Grid-Based Stochastic Model Predictive Control for Motion-Planning in Low-Friction Conditions

Addressing Perception Uncertainties in Challenging Road Conditions in Automated Driving

J.E. (Jelle) van Rijn

Delft University of Technology

TU Delft

Grid-Based Stochastic Model Predictive Control for Motion-Planning in Low-Friction Conditions

Addressing Perception Uncertainties in Challenging Road Conditions in Automated Driving

by

J.E. (Jelle) van Rijn

to obtain the degree of Master of Science Cognitive Robotics

at the Delft University of Technology,

to be defended publicly on Friday February 21, 2025 at 13:00.

Student number:	4933397	
Project duration:	March 14, 2024 – February 21, 2025	
Thesis committee:	Dr. ir. B. Shyrokau,	TU Delft, supervisor
	Dr. ir. G. Papaioannou,	TU Delft, external
	Ir. A. Bertipaglia,	TU Delft, daily supervisor

Cover:	Generated using ChatGPT
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

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

TU Delft

Preface

The first time I saw a car drive without any hands on the steering wheel, it was like magic to me. It made me feel like I was living in the future and it ultimately was the reason for me to pursue this masters degree in Cognitive Robotics. What better way to finish this degree than by taking a deep dive into the realm of automated vehicles during my final thesis and making my contribution to this field? I would like to use this opportunity to express gratitude to all the people that helped and supported me during the process of writing this thesis.

First and foremost, I would like to thank my TU Delft supervisors, Barys Shyrokau and Alberto Bertipaglia. I want to thank them for their guidance and valuable insights during the thesis project and thank them for their patience during our meetings. I could not have done it without them.

Furthermore, I would like to thank my friends and family for the support during the entirety of my education journey, for their support in tough times, and for all the fun times outside of studying.

Additionally, I thank my roommates Jan József, Nienke, and Diederick for putting up with me during the frustrating periods during this thesis. I want to thank them for their support and understanding in some of these stressful times.

Lastly I want to thank you, the reader, for taking the time to read this thesis. I hope you find it both insightful and engaging.

J.E. (Jelle) van Rijn
Delft, February 2025

Acknowledgment
I would like to acknowledge the use of OpenAI's ChatGPT to improve the readability and flow of parts of this thesis. All ideas are my own.

Abstract

Automated driving is poised to transform the transportation landscape of the future, but several challenges remain before full automation is achieved. One of these challenges lies in managing perception uncertainties, such as those arising from radar and sensor measurements, while maintaining control in low tire-road friction conditions. These challenges often occur simultaneously in adverse weather conditions, but are typically researched separately. Their combined effect on safety and performance remains underexplored, even though addressing them together is critical for robust and reliable automated driving systems.

Lower tire-road friction limits the available tire force required for obstacle avoidance. Properly modeling these tire friction limits is particularly important in dynamic and uncertain environments to adequately account for the changing environment responsively. Moreover, addressing perception uncertainties and modeling low-friction conditions individually can significantly increase computational demands which poses challenges to achieve real-time performance required for real-world implementation. Therefore, this thesis jointly considers perception uncertainties and low tire-road friction conditions, accounting for their interacting effects. This is accomplished by addressing the following research question: "How can perception uncertainties be effectively integrated into motion-planning models for obstacle avoidance to improve performance and safety of automated vehicles in various road conditions?"

To address this question, a grid-based stochastic model predictive control framework is extended, implementing a non-linear bicycle model and a Fiala tire model (brush model) to consider realistic vehicle capabilities in low-friction conditions. Grid-based stochastic model predictive control reduces the uncertain obstacle environment into a set of linear constraints, utilizing an occupancy probability grid to effectively consider perception uncertainties. While the developed framework is presented as a proof-of-concept with a focus on safety and feasibility over real-time implementation, computational efficiency is not overlooked. By reformulating the constraints, the uncertainties are effectively accounted for while also reducing the optimization time by simplifying the probabilistic obstacle space to a deterministic convex region. If the nominal reformulation fails, a novel back-up method generates a conservative back-up set of constraints improving the safety and feasibility of the method. Two different back-up strategies are proposed, providing a trade-off between accuracy and computational effort.

To evaluate the contributions, three simulations were conducted. The first comparing the non-linear bicycle model and Fiala tire model to more simplistic models at various tire-road friction coefficients, highlighting the improvement in control of the proposed model in low-friction conditions. The second simulation evaluated the performance of the proposed back-up methods and environment representation by simulating tight scenarios that would fail using only the nominal approach. Feasibility rates increased compared to the baseline back-up method (43.8%) with feasibility rates of 62.5% for the precomputed back-up method and 75.0% for the current-state back-up method. These simulation results demonstrate that the capability of the proposed framework to compute feasible solutions and that the framework is able to compute valid hulls that are safe when the nominal approach fails to reformulate the constraints. The final simulation evaluated the performance of the complete proposed method by simulating both tight scenarios requiring a back-up method as well as various friction levels. These simulations demonstrated that the proposed framework is effective at lower friction levels, achieving high feasibility rates of 87.5% for the current-state back-up method, and 91.6% for the precomputed back-up method. This high accuracy is particularly promising for real-time applications using the precomputed back-up strategy, since this method leverages parallel computation of the back-up constraints.

This thesis demonstrated the effectiveness of the proposed extended grid-based stochastic model predictive control framework in considering perception uncertainties in low-friction road conditions. While presented as a proof-of-concept with an emphasis on feasibility and safety rather than real-time implementation, the proposed framework lays the groundwork for real-time applications, marking a step in solving all edge cases required to reach fully automated vehicles.

Contents

Preface i

Abstract ii

Nomenclature viii

1 Introduction 1
- 1.1 Problem Statement & Scope . . . 3
- 1.2 Contributions . . . 4
  - 1.2.1 Prediction Model Complexity . . . 4
  - 1.2.2 Back-Up Method for Convex Hull Generation . . . 4
  - 1.2.3 Supporting Extensions for Safe Constraint Representation . . . 4
- 1.3 Thesis Structure . . . 5

2 Literature Review and Baseline Method 6
- 2.1 Automated Vehicles and Motion Planning . . . 6
  - 2.1.1 Graph-Based Techniques . . . 6
  - 2.1.2 Interpolation-Based Techniques . . . 7
  - 2.1.3 Sampling-Based Techniques . . . 8
  - 2.1.4 Optimization-Based Techniques . . . 9
- 2.2 Stochastic Model Predictive Control . . . 11
- 2.3 Grid-Based Stochastic Model Predictive Control . . . 12
  - 2.3.1 Limitations . . . 13

3 Methodology 16
- 3.1 Non-Linear Stochastic Controller . . . 16
  - 3.1.1 Coordinate System . . . 16
  - 3.1.2 Vehicle Model . . . 17
  - 3.1.3 Tire Model . . . 17
  - 3.1.4 Discretization . . . 18
  - 3.1.5 Target Vehicles . . . 18
  - 3.1.6 Ego Vehicle Dimensions . . . 19
  - 3.1.7 Cost Function . . . 19
  - 3.1.8 Reference Trajectory . . . 20
- 3.2 Constraint Formulation . . . 21
  - 3.2.1 Probability Grid . . . 21
  - 3.2.2 Binary Grid . . . 23
  - 3.2.3 Convex Hull Formulation . . . 24
  - 3.2.4 Soft Constraints . . . 27
- 3.3 Back-Up Method . . . 27
  - 3.3.1 Back-Up Necessity . . . 28
  - 3.3.2 Approach . . . 28

4 Simulation and Results 32
- 4.1 Model Complexity . . . 33
  - 4.1.1 Scenario . . . 33
  - 4.1.2 Results . . . 34
- 4.2 Back-Up Method . . . 39
  - 4.2.1 Scenario . . . 41
  - 4.2.2 Results . . . 42
- 4.3 Combined Method . . . 46

4.3.1 Scenario . . . 47
4.3.2 Results . . . 47

5 Discussion **50**
5.1 Simulation Analyses: Strengths and Limitations . . . 50
5.1.1 Simulation 1: Model Complexity . . . 50
5.1.2 Simulation 2: Back-Up Method . . . 52
5.1.3 Simulation 3: Complete Method . . . 54
5.2 Limitations . . . 55
5.2.1 Computational Overhead . . . 56
5.2.2 Simulation Environment Assumptions . . . 56

6 Conclusion **58**

References **61**

A Control Models **64**

List of Figures

1.1 Society of Automotive Engineers' six levels of driving automation [3] . . . 1
1.2 Main motion planning architecture approaches in automated driving [7] . . . 2
1.3 Driving scenario scope [15] . . . 3

2.1 State lattice [19] . . . 7
2.2 A* [20] . . . 7
2.3 Polynomial curves [21] . . . 7
2.4 Bezier curves [22] . . . 7
2.5 RRT* [28] . . . 9
2.6 Trajectory sampling [26] . . . 9
2.7 Model predictive control schematic [30] . . . 10
2.8 Expansion of probability density function uncertain target vehicle state [13] . . . 13
2.9 Baseline limitation: Unsafe back-up method in baseline framework . . . 14
2.10 Baseline limitation: ego vehicle dimensions . . . 14

3.1 Single track model coordinate system schematic . . . 16
3.2 Representation of the vehicle dimensions and convex free space . . . 19
3.3 Determination of a free lane . . . 20
3.4 Reference trajectory . . . 21
3.5 Computation of a deterministic obstacle free space from uncertain target vehicle states . . . 22
3.6 Influence of variance on probability density function shape . . . 23
3.7 Convex hull construction algorithm: explorations . . . 24
3.8 Convex hull construction algorithm: expansion . . . 25
3.9 Convex hull described by a set of linear constraints . . . 27
3.10 Constraint violation . . . 27
3.11 Flowchart of a back-up constraint computation . . . 29
3.12 Visualization of overlapping predictions across model predictive control iterations . . . 30
3.13 Iteration flowchart of the current-state back-up method . . . 31
3.14 Iteration flowchart of the precomputed back-up method . . . 31

4.1 Scenario unobstructed overtaking: Evaluating models in various friction coefficients . . . 33
4.2 Vehicle trajectories at friction coefficient 0.9 . . . 34
4.3 Control inputs at friction coefficient 0.9 . . . 34
4.4 x velocity (m/s) over time (s) at friction coefficient 0.9 . . . 35
4.5 y velocity (m/s) over time (s) at friction coefficient 0.9 . . . 35
4.6 Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.9 . . . 35
4.7 Tracking Error (m) over time (s) at friction coefficient 0.9 . . . 35
4.8 Vehicle trajectories at friction coefficient 0.45 . . . 36
4.9 Control inputs at friction coefficient 0.45 . . . 36
4.10 x velocity (m/s) over time (s) at friction coefficient 0.45 . . . 37
4.11 y velocity (m/s) over time (s) at friction coefficient 0.45 . . . 37
4.12 Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.45 . . . 37
4.13 Tracking Error (m) over time (s) at friction coefficient 0.45 . . . 37
4.14 Vehicle trajectories at friction coefficient 0.35 . . . 38
4.15 Control inputs at friction coefficient 0.35 . . . 38
4.16 x velocity (m/s) over time (s) at friction coefficient 0.35 . . . 39
4.17 y velocity (m/s) over time (s) at friction coefficient 0.35 . . . 39
4.18 Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.35 . . . 39
4.19 Tracking Error (m) over time (s) at friction coefficient 0.35 . . . 39

4.20 Narrow and shallow hull when target vehicles are close 40
4.21 Scenario varying gaps: evaluation back-up method 41
4.22 Baseline infeasible. Peeking behavior 43
4.23 Safe back-up hull proposed method 44
4.24 Peeking behavior in the proposed back-up methods 44
4.25 Infeasibility due to discrepancies between consecutive hulls 45
4.26 Improved continuity utilizing current-state back-up method 46
4.27 Scenario varying gaps: evaluating complete framework 47
4.28 Hulls alternating lanes 48
4.29 Growing uncertainty leading to tight scenario 49

5.1 Fiala tire model relating sideslip angle to the available tire force at various friction coefficients 51

List of Tables

3.1 Cost function weights and control parameters (based on [9]) 20

4.1 Vehicle model parameters (based on [9]) 32
4.2 Root Mean Square Error (m) at friction coefficient 0.9 35
4.3 Root Mean Square Error (m) at friction coefficient 0.45 37
4.4 Root Mean Square Error (m) at friction coefficient 0.35 39
4.5 Comparison of back-up hull method feasibility 42
4.6 Comparison of back-up method feasibility at friction coefficient 0.9 47
4.7 Comparison of back-up method feasibility at friction coefficient 0.45 47
4.8 Comparison of back-up method feasibility at friction coefficient 0.35 47

Nomenclature

Abbreviations

Abbreviation	Definition
CoG	Center of Gravity
EV	Ego Vehicle
erf.	Error function
GPS	Global Positioning System
inf.	Infeasible
LiDAR	Light Detection And Ranging
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
PDF	Probability Density Function
QRQP	Quadratic Regularized Quadratic Programming
Radar	Radio Detection And Ranging
ref.	Reference trajectory
RMPC	Robust Model Predictive Control
RMSE	Root Mean Square Error
RRT	Rapidly-exploring Random Trees
SAE	Society of Automotive Engineers
SMPC	Stochastic Model Predictive Control
TV	Target Vehicle

Symbols

Symbol	Definition	Unit
$\boldsymbol{A}$	Target vehicle state matrix	[-]
a_x	Longitudinal acceleration	[m/s^2]
$\dot{a}_x$	Longitudinal jerk	[m/s^3]
a_y	Lateral acceleration	[m/s^2]
$\boldsymbol{B}$	Target vehicle input matrix	[-]
B	Binary grid	[{0,1}]
$b_{i,j}$	Grid cell occupancy binary value	[{0,1}]
C	Cell column	[cell]
C_α	Axle cornering stiffness	[kN/rad]
C_{free}	Free section of column at exploration radius	[cell]
C_{range}	Column of grid cells at the exploration radius	[cell]
$\boldsymbol{c}_{i,j}$	Grid cell	[m]
c_{th}	Cantelli threshold	[-]
d	Inequality constraint scalar	[m]
e	Probabilistic part state	[various]
F_{x_f}	Longitudinal force on front tires	[N]
F_{y_f}	Lateral force on front tires	[N]
F_{y_r}	Lateral force on rear tires	[N]
$F_{y,max}$	Maximum lateral tire force	[N]
f	State equations of motion	[-]
$\boldsymbol{G}$	Target vehicle uncertainty matrix	[-]

Symbol	Definition	Unit
$\mathcal{G}$	Grid	[cell]
g	Gravitational constant	[m/s^2]
$\boldsymbol{g}$	Constraint coefficient vector	[-]
h	Time duration prediction step	[ms]
h_{CoG}	Height of the center of gravity	[m]
I	Identity matrix	[-]
I_{zz}	Moment of inertia about the z-axis	[kg·m^2]
i	Model predictive control iteration	[-]
J	Cost function	[-]
$\boldsymbol{K}$	Target vehicle feedback matrix	[-]
k	Prediction horizon step	[-]
L	Vehicle wheelbase	[m]
l_f	Distance from center of gravity to front axle	[m]
l_{lane}	Lane width	[m]
l_r	Distance from center of gravity to rear axle	[m]
l_y	Grid cell height	[m]
l_x	Grid cell width	[m]
m	Vehicle mass	[kg]
N	Number of prediction steps in prediction horizon	[-]
N_v	Number of target vehicles localized in the detection range	[-]
P	Probability grid	[0,1]
$p_{i,j}$	Grid cell occupancy probability value	[0,1]
p_{th}	Binary grid probability threshold	[0,1]
$\boldsymbol{Q}$	Kinematic model state cost matrix	[-]
Q_{ax}	Weight for longitudinal acceleration	[-]
$Q_{\dot{a}x}$	Weight for longitudinal jerk	[-]
$Q_{\boldsymbol{S}}$	Weight for constraint violation	[-]
Q_{vx}	Weight for longitudinal speed error	[-]
Q_y	Weight for lateral position error	[-]
Q_δ	Weight for steering angle	[-]
$Q_{\dot{\delta}}$	Weight for steering rate	[-]
q	Confidence boundary	[-]
$\boldsymbol{R}$	Kinematic model input cost matrix	[-]
R	Turning radius	[m]
r	Heading rate	[rad/s]
$\boldsymbol{S}$	Constraint slack vector	[m]
t	Time	[s]
$\boldsymbol{U}$	Input sequence	[various]
$\mathcal{U}$	Set of admissible inputs	[various]
$\boldsymbol{u}$	Input vector	[various]
v_x	Longitudinal velocity	[m/s]
v_y	Lateral velocity	[m/s]
$\boldsymbol{w}$	Uncertainty vector	[various]
W	Vehicle width	[m]
$\mathcal{X}$	Set of admissible states	[various]
$\boldsymbol{x}$	State vector	[various]
X	Example random variable	[-]
x	Longitudinal position	[m]
y	Lateral position	[m]
z	Deterministic part state	[various]
α	Body slip angle	[rad]
α_f	Tire slip angle for front tires	[rad]
α_r	Tire slip angle for rear tires	[rad]

Symbol	Definition	Unit
β	Risk parameter	[0,1]
β_{bu}	Risk parameter back-up method	[0,1]
δ	Steering angle	[rad]
$\dot{\delta}$	Steering rate	[rad/s]
γ	Tightening parameter	[-]
κ	Standard deviation scalar	[-]
μ	Friction coefficient	[-]
μ_X	Mean value	[-]
Ξ^{adm}	Admissible grid cells	[-]
ξ^{EV}	Ego vehicle state vector	[various]
ξ^{TV}	Target vehicle state vector	[various]
ψ	Heading angle	[rad]
Σ	Uncertainty covariance matrix	[various]
σ	Standard deviation	[-]

1

Introduction

Each year, 1.19 million deaths are caused by road traffic accidents [1]. 95% of the road traffic accidents are partially caused by human behavior and in three out of five cases, human behavior is the primary cause of the accident [2]. Therefore, a lot of research is being done to remove human behavior out of traffic entirely via fully automated vehicles. Automated vehicles represent a revolutionary advancement assured to reshape society's transportation landscape. From adaptive cruise control to experimental self-driving taxis, advanced driver-assistance systems promise to enhance transportation safety and comfort compared to current standards. However, several challenges must be overcome before achieving full automation.

The Society of Automotive Engineers (SAE) has delineated six levels of automation for vehicles [3], ranging from Level 0 to Level 5:

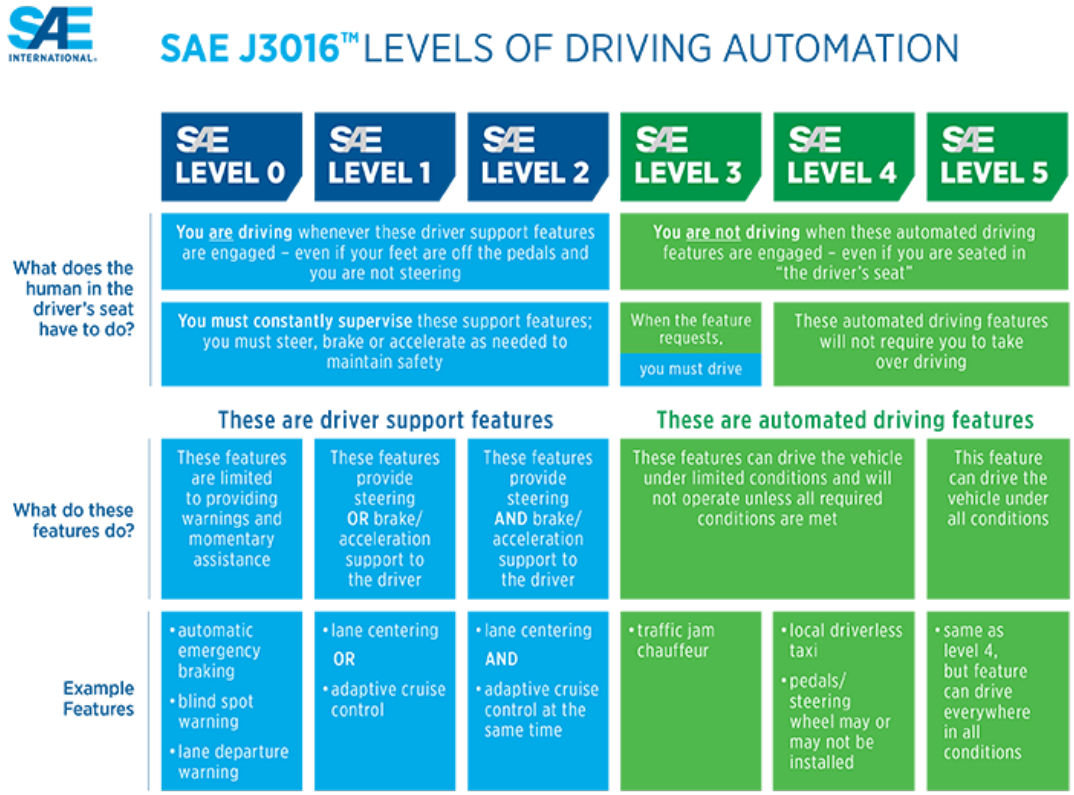
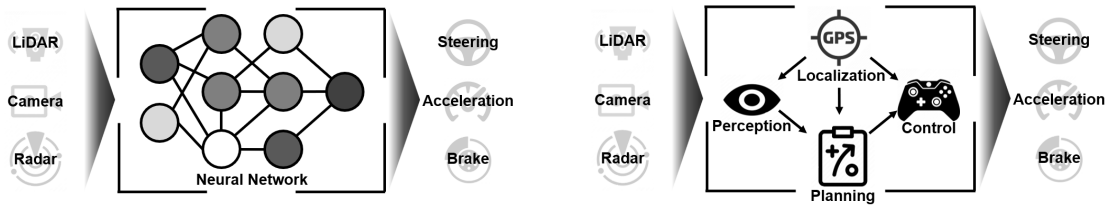


Figure 1.1: Society of Automotive Engineers' six levels of driving automation [3]

The levels described in Figure 1.1 represent a continuum of increasing automation, with higher levels reducing the need for human intervention. Examples such as the Waymo's fully self driving taxi could be considered level 4 automation, since these taxis are "designed to operate without a human driver" [4]. However, some challenges remain in achieving full automation in all scenarios, including technological limitations, regulatory considerations, and societal acceptance. The most important challenge to be tackled is safety, as it is essential to protect the lives of passengers, pedestrians, and other road users. Moreover, achieving societal and legislative acceptance of automated driving requires safety levels at least on par with human driving, if not higher [5]. This also means that automated vehicles should behave safely in non-trivial conditions, such as adverse weather, to achieve level 5 automation. Adverse weather and resulting reduced tire-road friction, limit the controllability and perception of the vehicle. The impact of adverse weather conditions is still an under-researched area [6] and by addressing this critical issue, automated driving could be brought a step closer to full automation.

Automated driving research can be separated into two main approaches. End-to-end planning methods and pipeline planning methods [7] (visualized in Figure 1.2). End-to-end methods aim to directly map raw sensor output to control outputs, often utilizing machine learning. This method has become popular with the current rise of powerful artificial intelligence. However, the opaqueness of these models is a crucial drawback. Deep learning algorithms are like a black box and are hard to interpret. This lack of interpretability makes pinpointing the occurrence of potential faults hard. This is a crucial flaw in a field where a fault could lead to fatal accidents. This black box property also makes it hard to deal with known critical aspects such as perception uncertainties. There is, therefore, still a lot of research being done in pipeline planning based methods.



(a) End-to-End approach

(b) Pipeline approach

Figure 1.2: Main motion planning architecture approaches in automated driving [7]

Pipeline methods tackle different parts of automated driving as subproblems, such as perception, motion planning and control. Considering these subproblems as individual modules creates more insight into what is happening in the system when unexpected behaviour occurs.

Perception
The perception module perceives the environment utilizing sensors such as cameras, LiDAR or radar or ultrasonic sensors. The raw sensor output is converted to a semantic representation of the current state of the ego vehicle and the environment by using various algorithms.

(Local) motion planning
During local motion planning a trajectory is computed for the coming period of time. The submodule generates a safe and efficient trajectory by taking obstacles, system dynamics and high-level goals into account. This is done based on the perceived state of the perception module.

Control
The control module submodule follows a given trajectory and gives the system inputs (e.g. acceleration or steering angle) while adhering to the given constraints and system dynamics.

Even though being more transparent, addressing subproblems individually introduces complexities, particularly regarding noise and uncertainty propagation across subsystems. Uncertainties originating from various modules, such as perception and control, can propagate throughout the system, impacting overall performance and safety. Conventional research often takes uncertain perception outputs as inputs in the motion planning module leading to uncertain or unsafe trajectory planning. Moreover, a trajectory could be planned by the motion planner which is infeasible due to the constraints of the controller. Thus, while approaching automated driving like a pipeline facilitates more transparency, it

is still in need of robust integration strategies to mitigate the effects of uncertainties across different subsystems. Particularly in complex scenarios, like low-friction conditions.

1.1. Problem Statement & Scope

Automated driving relies heavily on accurate motion planning. In low tire-road friction conditions, motion planning becomes significantly more challenging, since reduced tire-road friction affects vehicle dynamics and maneuverability. To ensure safety in low-friction conditions, a more complex model is required that accounts for these effects. Low-friction conditions are often caused by adverse weather conditions and thus likely go hand-in-hand with other weather effects. Snow and rain can seriously affect the performance of LiDAR and camera perception systems and could lead to high perception uncertainties. While radar is more robust to these conditions, it too is affected, particularly at longer detection distances [8]. Snow and rain will decrease signal strength and will result in less accurate perception, which will add to the perception uncertainty.

Existing approaches on motion planning under low-friction conditions often consider deterministic environments [9], [10]. Approaches considering (perception) uncertainties typically consider simple or no dynamics in their model [11], [12], [13], [14]. However, since the effects of lower tire-road friction and higher uncertainty are closely related, it is important to model them together, accounting for any interacting effects. Failing to considers both complex dynamics and perception uncertainties together could lead to unsafe or inefficient trajectories in real-world scenarios. Since modeling complex dynamics and uncertain environments both lead to high computational load, it is extra important to meticulously consider relevant aspect in order achieve real-time performance.

This gap highlights the need for obstacle avoidance models that can integrate perception uncertainties dynamically and account for vehicle limitations due to low-friction roads. Addressing this gap is critical for improving both the performance and safety of automated vehicles in real-world scenarios.

To address this gap, this thesis investigates the following research question:

How can perception uncertainties be effectively integrated into motion-planning models for obstacle avoidance to improve performance and safety of automated vehicles in various road conditions?

To answer the research question, this thesis focuses on developing an obstacle avoidance framework under specific conditions. The scope is limited to highway driving scenarios with varying tire-road friction coefficients, where the friction is assumed to be known. The work focuses on incorporating realistic vehicle dynamics and considering perception uncertainties, provided as input from perception submodules. This thesis will focus on a high-way scenario, utilizing common traffic behavior on these roads, such as no backwards driving and vehicles primarily staying within lanes. Urban scenarios are not considered in this work.

Figure 1.3: Driving scenario scope [15]

1.2. Contributions

This thesis builds upon a grid-based stochastic model predictive control framework [13] that handles perception uncertainties for trajectory planning in dynamic environments. This method, used as a baseline, introduces a probabilistic approach to obstacle avoidance. It aggregates the obstacle uncertainties probability density functions (PDFs) into an occupation likelihood grid, from which a simplified set of linear constraints is computed, describing the free space. This thesis extends its capabilities, proposing a method capable of driving in various road-conditions and proposing a safe back-up method, for when the nominal constraint formulation fails. The baseline method and its limitations are discussed in further detail in Section 2.3. This thesis proposes a novel approach to utilize the grid-based constraint formulation in a framework which is able to consider the effects of low friction in obstacle avoidance in dynamic and uncertain environments.

1.2.1. Prediction Model Complexity

The first contribution of this work is the implementation of a realistic dynamic bicycle model with the Fiala tire model (brush model) in the grid-based stochastic model predictive control framework to account for the effects of varying tire-road friction coefficients. The dynamic bicycle model provides an accurate representation of vehicle dynamics, capturing lateral and longitudinal forces. This model balances complexity and computational efficiency, making it suitable for real-time applications while ensuring an accurate representation of the vehicle's motion [9]. The Fiala model explicitly models non-linear tire forces at low friction coefficients, enabling the prediction of force limitations at reduced-friction conditions. These models allow for better prediction of the behavior of the ego vehicle under various friction conditions, improving trajectory tracking accuracy and control compared to more simplistic prediction models.

1.2.2. Back-Up Method for Convex Hull Generation

The second contribution comprises an extension of the convex hull computation of [13], proposing a novel back-up method that accurately computes a set of linear constraints to describe a convex hull within the free space, when the nominal approach fails. The proposed back-up method improves feasibility and ensures safety by generating a conservative realization of free space when a nominal hull cannot be found. This approach leverages the iterative prediction of states in the model predictive control framework and the principle that uncertainties grow over time. By incorporating this growing uncertainty into the back-up hull computation, the method prioritizes safety while maintaining computational efficiency. Two distinct strategies are introduced for the back-up hull generation, with trade-offs between computational efficiency and accuracy:

- **Current-State Back-Up Hull Method**. This approach computes a back-up hull by reconsidering the occupancy likelihood grids computed in the nominal constraint reformulation approach and considers a more conservative free space. A different, lower threshold is used to determine wether grid cells are classified as occupied. These back-up hulls are computed to fall back on when a hull cannot be found with the nominal approach, such that trajectory optimization is still possible.
- **Precomputed Back-Up Hull Method** This approach is similar to the current-state back-up hull method, however it utilizes the states and occupancy likelihood grids from the previous open-loop solution to generate the back-up hulls. Since uncertainties grow over the prediction horizon, these probability grids inherently represent a more conservative view of free space. The use of prior predictions allows the back-up hull computation to be performed in parallel with the optimization process, ensuring that the back-up hulls are precomputed and ready for immediate use if the nominal constraint reformulation fails. This parallel computation enhances computational efficiency, making this approach particularly well-suited for real-time applications.

The proposed back-up strategies provide a safe computation of constraints to fall back on when the nominal constraint formulation fails. This way the optimization problem can still be solved and vehicle control can continue, improving feasibility rates and safety in tight scenarios compared to the baseline.

1.2.3. Supporting Extensions for Safe Constraint Representation

To ensure that the computed back-up hulls are not only feasible but also safe, several extensions to the baseline method are introduced. Their implementation within the context of this framework are crucial

for ensuring safety and robustness in real-world settings.

Consideration of Ego Vehicle Dimensions

To further enhance safety and reliability, the convex hull generation method incorporates the physical dimensions of the ego vehicle, ensuring that the entire vehicle is contained within the computed hull. This consideration guarantees that if the vehicle is within the convex hull, it is also within free space. By accounting for the vehicle's width and length in the constraint formulation, the method provides a more realistic representation of free space. This improvement is applied to both the nominal and back-up methods, enhancing overall system performance.

Dynamic Uncertainty Threshold

To ensure a more conservative representation of free space compared to the nominal method, a proper uncertainty representation is required. This is achieved by employing a dynamic probability threshold that adapts to the aggregated uncertainties in the probability grids. These grids combine multiple probability density functions corresponding to different sources of uncertainty, which may have varying variances. PDFs with larger variances are flatter and have lower peak values, meaning that a static threshold would not consistently represent the same level of confidence across the uncertain environments. The dynamic threshold addresses this by computing the (user-defined) probability boundary for the highest variance among the aggregated uncertainties at each prediction step. This ensures that regions with greater uncertainty are more likely to be classified as occupied, providing a conservative representation of free space that aligns with the back-up method's role as a fallback mechanism. By dynamically adapting the threshold to reflect the uncertainty magnitudes at each step, the method enables tuning the level of conservativeness, both in the back-up method, as well as in in the nominal hull finding method.

1.3. Thesis Structure

The remainder of this thesis will be structured as follows. Chapter 2 discusses the relevant literature, considering various motion planning algorithms and discussing their strengths and weaknesses with respect to low-friction conditions, dealing with uncertainty and real-time applicability. Followed by a section discussing the promising framework stochastic model predictive control and its formulation. Finally the baseline method, grid-based stochastic model predictive control, is discussed upon which this work is based, together with its challenges to utilize the framework in the scope of this thesis. Chapter 3 discussed the proposed methodology, divided in 3 sections. The first section entails the stochastic model predictive controller, specifying the controller architecture and environment representation. The second section considers the constraint reformulation, dissecting the procedure to reformulate the uncertain obstacle environment to a set of linear constraints to describe the obstacle-free space. The final section discusses the proposed back-up method, formulating a conservative back-up set of constraints when the nominal constraint reformulation fails. Chapter 4 discusses the simulations and results conducted to validate the proposed methods. The first simulation comprises of an overtaking maneuver at various tire-road friction coefficients, to test the capabilities of the proposed prediction model. The second simulation contains an overtaking maneuver between two obstacle vehicles at various gaps, testing the capabilities of the proposed back-up constraint formulation method in tight scenarios. The final simulations combine the previous two simulations, evaluating the performance of the complete proposed model in tight scenarios and low-friction conditions. Chapter 5 discusses these results and general strengths and limitations of the proposed methods. Chapter 6 draws final conclusions and proposes interesting recommendations for future work.

2

Literature Review and Baseline Method

This chapter discusses the literature review conducted to identify a suitable approach to consider perception uncertainties in low tire-road friction conditions and discusses the baseline method upon which the proposed framework is build. The chapter starts with a high-level review of motion planning techniques in automated driving, followed by a discussion of stochastic model predictive control, explaining the framework and its capability to consider uncertainties. The last section discusses the baseline method, grid-based stochastic model predictive control, highlighting its potential to address perception uncertainties in low-friction conditions and the challenges that must be addressed to adapt the method to the problem at hand.

2.1. Automated Vehicles and Motion Planning

The motion planning problem for automated vehicles using pipeline approaches is categorized into four primary categories: graph-based techniques, interpolation-based techniques, sampling-based techniques, and optimization-based techniques. Each approach possesses distinct advantages and limitations, particularly when addressing perception uncertainties in dynamic environments.

2.1.1. Graph-Based Techniques

Graph-based or graph search-based techniques divide the state-space into a discrete graph, like an occupancy grid or a lattice (Figure 2.1). Nodes represent states and edges denote possible transitions, where the edges can have weights according to some cost function. Once this representation is established, a search algorithm, like Dijkstra's algorithm [16] or A* [17], is employed to find a path through the graph to reach the goal while minimizing the sum of weights of traversed edges. Graph search algorithms can be computationally efficient in static environments and are often used to find a global path. Dijkstra's algorithm is based on a min-priority queue, meaning the shortest path found is explored first, implying the first path found is the one with the lowest cost. The A* algorithm (Figure 2.2) extends this by using a heuristic, which is an (under)estimation of the cost-to-go, guiding the search direction. The optimality guarantee of the first found solution holds if the heuristic is smaller than or equal to the actual cost-to-go, as other optional paths are guaranteed to have equal or greater cost.

D* (short for dynamical A*) extends A* to dynamic environments by implementing an efficient replanning module, enhancing robustness in unknown or environments. D*Lite [18] is a simplified and more efficient variant of D*. These algorithms are similar to A* as they utilize a search heuristic. D*(lite) is designed to handle environmental changes and can recompute the optimal path efficiently by updating only the affected parts. Unlike A*, which executes the search from the start node to the goal node, D* performs the search from the goal node to the start node. This leverages the previous search efforts by propagating changes from the modification point back to the start.

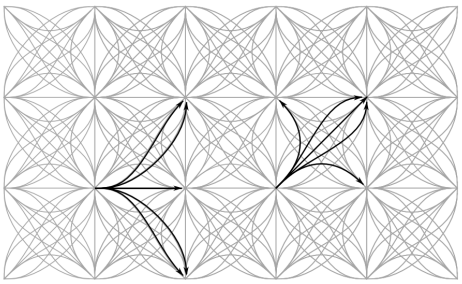


Figure 2.1: State lattice [19]

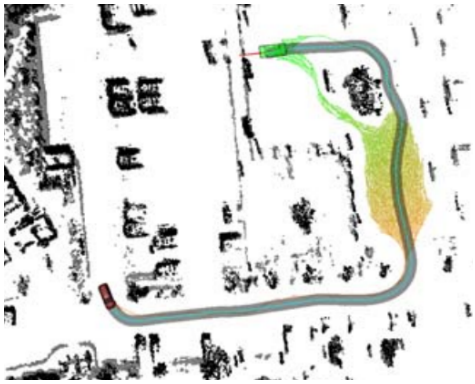


Figure 2.2: A* [20]

While graph-based techniques offer a relatively simple approach to motion planning in a static environment, they are not suitable for complex environments with dynamic obstacles or uncertain dynamics. Recomputing a graph is too expensive for this method to be a reliable motion planning technique in real-time applications, particularly in a high-dimensional state-space. Additionally, graph-based planners generally struggle to handle perception uncertainties effectively. The discrete nature of the graph representation can limit the planner's ability to adapt to uncertain and dynamic environments, where continuous updates and flexible adjustments are necessary. Despite their improvements, even advanced graph-based planners like D* and D* Lite have limitations when dealing with high levels of uncertainties and dynamic changes in the environment. The need for frequent replanning and the discrete nature of these algorithms make it challenging to achieve the responsiveness and adaptability required for safe and efficient motion planning under such conditions.

2.1.2. Interpolation-Based Techniques

Curve-based motion planners that employ interpolation techniques utilize predefined waypoints to construct smooth trajectories through these points. Interpolation-based methods are often used in combination with Global Positioning System (GPS), as GPS provides well-established means to acquire such waypoints. Curve-based motion planners, like polynomial curves (Figure 2.3) and Bezier curves (Figure 2.4), are often used when driving comfort is of importance since the generated trajectories are smooth.

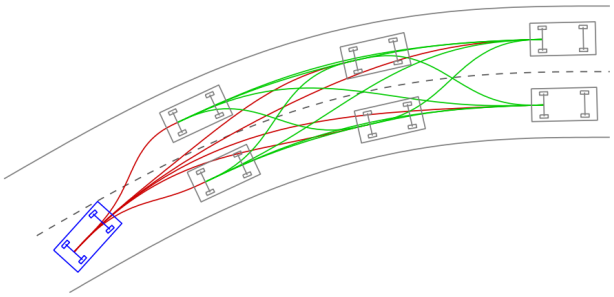


Figure 2.3: Polynomial curves [21]

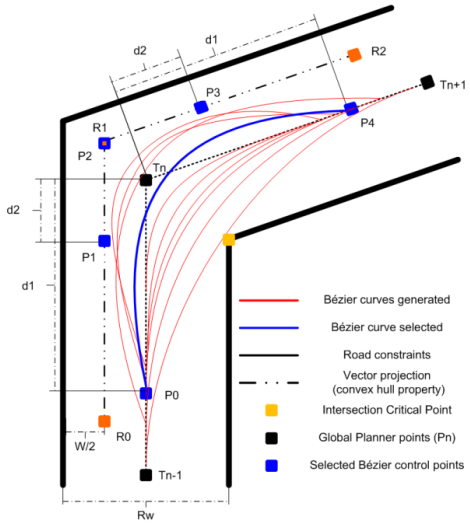


Figure 2.4: Bezier curves [22]

Interpolation-based techniques could deal with uncertainties by incorporating them into the path planning algorithm. In real world approaches however, obstacles and their uncertainty are rapidly changing continuously, which results in a constantly changing solving environment.

While interpolation-based techniques are efficient in static environments due to their ability to generate smooth and continuous paths, they face significant challenges in real-time settings [23]. The need for continuous recalculation in the presence of moving obstacle undermines their computational efficiency. Furthermore, interpolation-based methods rely heavily on predefined waypoints, making them less effective in complex and unstructured environments. Thus, while interpolation-based techniques offer comfort and efficiency in controlled setting, their applicability in dynamic and uncertain conditions is limited.

2.1.3. Sampling-Based Techniques

Sampling-based techniques randomly sample points in the environment and connect them to form a graph or tree structure. In automated driving, the most common sampling algorithm is Rapidly-exploring Random Trees (RRT) [23]. By iteratively sampling and connecting random points, RRT efficiently explores complex state-spaces and generates feasible paths. Sampling-based techniques are particularly effective in high-dimensional spaces and environments with complex obstacles. However, these techniques may struggle in environments with narrow passages or tight constraints, as random sampling may not adequately explore these regions. Additionally, while sampling-based techniques are well-suited for online planning and dynamic environments, the quality of generated paths can vary, and the computational complexity can be high.

Closed-loop RRT [24] incrementally grows a tree of probabilistic-feasible trajectories. To guarantee the kinodynamic feasibility of the generated trajectories, a closed-loop model containing the kinematic model and controllers is utilized to propagate the state. Perception uncertainty-aware rapidly-exploring random tree [12] extends the closed-loop RRT approach by incorporating chance constraints on the obstacles. The probability distribution of the perception uncertainties is assumed to be Gaussian. By utilizing the inverse error function, the rectangular obstacles are converted to ellipses, with the size of the ellipse in each dimension determined by the uncertainty in that dimension and the allowed risk parameter. While perception uncertainties are accounted for, motion of obstacles is not considered. Consequently, the found trajectories are only valid at the current state and need to be reconsidered at each time step, making the method unsuitable for real-time approaches. Similarly, distributionally robust RRT [25] implements chance constraints to deal with uncertainties. This approach does not only consider Gaussian distributions but implements a distributionally robust RRT instead, accounting for various distribution shapes and limiting the probability of collision to the worst-case distribution.

The Frenetix sampling-based trajectory planner [26] samples from a given range of end states, and quintic polynomial functions are generated to connect the start and end state smoothly while minimizing jerk for comfort (see Figure 2.6). The sampled trajectories are then evaluated to check feasibility and compare optimality. Initially, the trajectories are evaluated for kinematic feasibility, after which a cost is calculated for every feasible trajectory. This cost includes weighted factors considering comfort, efficiency, and safety, with an explicit term for collision probability based on models estimating other traffic participants' movement. Perception uncertainties could be incorporated into this sub-cost. The trajectories are then checked for collisions in order of the lowest found cost, allowing the process to stop once a collision-free trajectory is identified. This additional check is performed on top of the collision probability cost to ensure a collision-free path. This collision check is conducted for states between two time steps by checking for collisions with an oriented bounding box [27].

Sampling-based techniques excel in exploiting high-dimensional spaces and handling complex obstacles through random sampling. However, they can struggle with path smoothness and computational cost, particularly in environments requiring frequent rerouting. Methods such as RRT* (Figure 2.5) ensure that a solution, if it exists, will eventually be found, but the quality of the path can vary. Consequently, while sampling-based methods are effective for real-time applications, their performance can be inconsistent under highly dynamic and uncertain conditions.

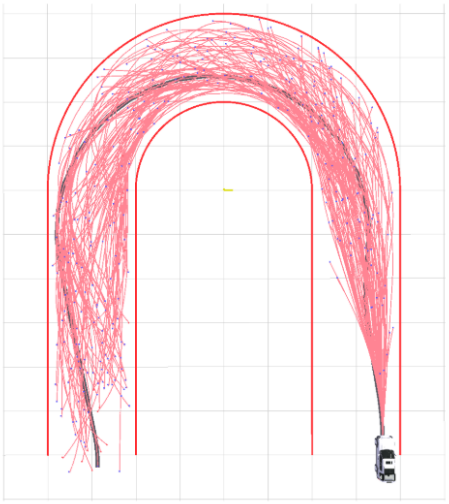


Figure 2.5: RRT* [28]

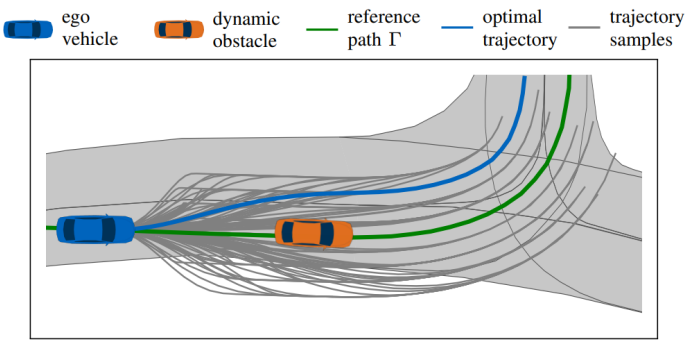


Figure 2.6: Trajectory sampling [26]

2.1.4. Optimization-Based Techniques

Optimization-based techniques formulate path planning as an optimization problem, aiming to find a path that minimizes or maximizes an objective function while satisfying constraints. These techniques have the advantage of finding optimal or near-optimal solutions systematically. They can handle complex cost functions and constraints, making them suitable for tasks requiring precise control. However, they can be computationally expensive, particularly in high-dimensional spaces or when dealing with non-linear, non-convex problems. Real-time application of these methods is challenging due to the time required to solve complex optimization problems, although approximation techniques could mitigate this issue. Due to computational complexity, optimization techniques often consider only a finite time horizon. Frequently used examples include Linear Quadratic Regulator (LQR) and Model Predictive Control.

The linear quadratic Gaussian framework [11] is an extension of the linear quadratic regulator that is able to deal with uncertainty. LQR is a feedback optimization technique that optimizes quadratic cost function by describing a dynamical system by some linear differential equations. A linear quadratic Gaussian extends the LQR by assuming the state variables follow a Gaussian distribution rather than being deterministic values. The state distribution is updated using a linear Kalman filter. This method is functional for simple systems but cannot handle non-linear systems.

Model Predictive Control (MPC) is a finite horizon optimization technique. This technique optimizes for the best control input for a limited amount of time or a certain number of time steps. After each optimization, the best control for the next time step is executed, and the process repeats. Limiting the time horizon makes MPC robust to changes and enables online optimization. Like other optimization techniques, MPC minimizes a cost function subject to constraints. The cost function describes an optimal path where weights are assigned to objectives such as path deviation, comfort, safety, and control input magnitude. Constraints describe the system's limitations, such as maximum acceleration or obstacles to avoid. The strength of MPC lies in its predictive model, which describes the (possibly non-linear) system dynamics. By incorporating the prediction of future states, MPC solves for the best inputs within the time horizon through closed-loop control. Obstacles are avoided using state constraints. Obstacle avoidance is a non-convex problem, which is often simplified by convexifying the obstacle constraints as a point, affine space, sphere, ellipsoid, or polytope [29]. Once the optimal sequence is determined, only the first control action is applied to the system. The process is repeated at the next time step, updating predictions and control inputs based on the system's new state. This iterative approach allows MPC to adapt to changes in the environment or system behavior in real time. A visualization can be seen in Figure 2.7.

Nominal model predictive control is not able to consider uncertainties. To address perception uncer-

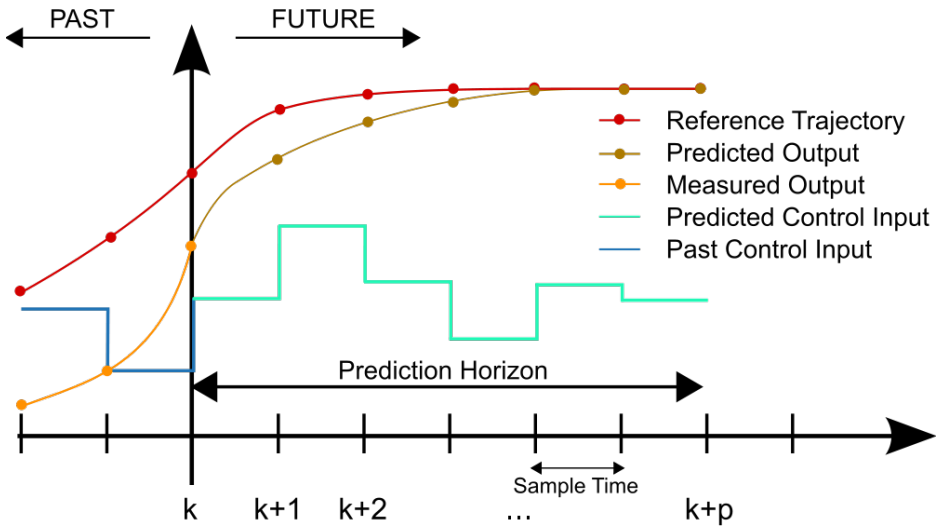


Figure 2.7: Model predictive control at time step k with a prediction horizon p [30]

tainties, it is necessary to describe them in the system. The two main approaches are robust MPC and stochastic MPC. Robust Model Predictive Control (RMPC) represents obstacles as hard constraints. These constraints consider the worst-case scenario, ensuring the solution, if found, will always be safe. A common RMPC approach is tube-based MPC [31]. This approach solves the nominal MPC problem and finds a tube around the found trajectory. All realizations of the uncertainty are present within this tube. Then, the constraints are tightened according to this tube, after which the optimization problem is solved with the tightened constraints. Min-max MPC [32] is another robust MPC approach. In min-max MPC, the uncertainty is incorparated in a cost function. The method first maximises the uncertainty loss, resulting in the worst case outcome of the uncertainty distribution. Then, under these worst case conditions, the loss function is minimized to find the optimal input sequence, similarly to nominal MPC. The drawback of the worst-case approach is that the solution is often overly conservative or could lead to infeasibility environments where there is a feasible trajectory. Another drawback of robust approaches is that finding a feasible trajectory under the worst-case uncertainty realization requires the uncertainty distribution to have bounded support, which is not always the case (e.g. a Gaussian distribution).

Stochastic Model Predictive Control (SMPC) allows, unlike robust optimization, for some acceptable level of risk in the obstacle constraints. This approach provides a balance between constraint violation and performance [33]. This is often achieved with chance constraints, which limit the probability of a state constraint violation at a certain threshold. Chance constraints accommodate uncertainties with unbounded support, which is not possible with robust optimization. The formulation of stochastic MPC remains similar to nominal MPC. However, given a state constraint $\boldsymbol{x}_k \in \mathcal{X}_k$ in nominal MPC, the constraint becomes

$$\mathbb{P}\left[\boldsymbol{x}_k \in \mathcal{X}_k\right] \geq \beta \tag{2.1}$$

Where β is a risk parameter. A higher parameter relates to a lower allowed risk.

Chance constraints are often made deterministic for better online tractability. The uncertainty is often assumed to be Gaussian. Converting the probabilistic constraints to deterministic constraints can be done using the inverse error function. Other approaches can be employed if the uncertainty distribution is not assumed to be Gaussian. More details on this will be discussed in the next section.

Optimization-based techniques are promising due to their ability to generate globally optimal solutions while incorporating complex constraints. Particularly, stochastic model predictive control stands out as an effective method for handling dynamic and uncertain environments. SMPC's capability to model uncertainties and balance risk with conservativeness makes it especially suitable for challenging scenarios with uncertainties. By leveraging the strengths of optimization-based approaches, robust and reliable motion planning can be achieved, improving safety and performance.

2.2. Stochastic Model Predictive Control

Stochastic model predictive control is an extension of nominal model predictive control. Model predictive control is a widely used trajectory planning technique, but it cannot consider uncertainty in its nominal form. Stochastic model predictive control is able to account for uncertainties while remaining efficient in contrast to other MPC approaches like robust model predictive control. By allowing a low probability of constraint violation, SMPC can reduce conservativeness compared to other approaches like robust model predictive control.

To understand stochastic model predictive control, it is essential to revisit nominal model predictive control before extending it to SMPC. The approach of nominal model predictive control is as follows:

$$\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} J\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{2.2a}$$

$$\text{s.t.:} \quad \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{2.2b}$$

$$\boldsymbol{u}_k \in \mathcal{U}_k, \quad k \in \mathbb{I}_{0,N-1} \tag{2.2c}$$

$$\boldsymbol{x}_k \in \mathcal{X}_k, \quad k \in \mathbb{I}_{1,N} \tag{2.2d}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}} \tag{2.2e}$$

This formulation tries to find the control input sequence $\boldsymbol{U} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1})$ by minimizing cost function $J\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$ over the time horizon $N \in \mathbb{Z}_+$, where $\boldsymbol{x}_k$ is the system's state at time step k. Equation (2.2b) describes the dynamics of the system. Equation (2.2c) describes the input constraints of the system, where $\mathcal{U}$ describes the set of admissible inputs, and (2.2d) describes the state constraints, including the obstacle constraints, where $\mathcal{X}$ describes the set of admissible states. Equation (2.2e) describes the initial state of the system.

Now let us consider a system with uncertainties described by $\boldsymbol{w}_k$. Equation (2.2b) becomes

$$x_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \tag{2.3}$$

The nominal model predictive controller cannot handle these uncertainties. Stochastic model predictive control addresses these uncertainties by using chance constraints. The state constraint (2.2d) is replaced by probabilistic constraint (2.4):

$$\mathbb{P}\left[\boldsymbol{x}_k \in \mathcal{X}_k\right] \geq \beta \tag{2.4}$$

where β is a tunable risk parameter. A higher parameter relates to a lower allowed risk. By tuning this parameter, it is possible to find a balance between constraint violation (collision risk) and trajectory efficiency and feasibility. A common used value of β is 0.95

Since these chance constraints are probabilistic, they are non-linear and not directly tractable. The probabilistic constraints are made deterministic to solve the optimization problem.

If the distribution is Gaussian, the well-known properties of a Gaussian distribution can be used to find a deterministic threshold using the cumulative distribution function or error function. Let us consider the j-th polytopic constraint from a set of constraints at time step k:

$$\boldsymbol{x}_k \in \mathcal{X}_k = \left\{\boldsymbol{x} \mid \boldsymbol{g}_j^\top \boldsymbol{x} \leq d_j\right\} \tag{2.5}$$

Where $\boldsymbol{g}_j$ is a vector of coefficients associated with the j-th inequality constraint and d_j is a scalar representing the threshold for the j-th inequality constraint. The state variable can be split into deterministic and probabilistic parts

$$\boldsymbol{x}_k = \boldsymbol{z}_k + \boldsymbol{e}_k \tag{2.6}$$

Given uncertainty covariance matrix $\mathbf{\Sigma}_k$, the probabilistic constraint can be reformulated as a deterministic constraint:

$$\boldsymbol{g}_k^\top \boldsymbol{z}_k \leq d_k - \gamma_k \tag{2.7}$$

where γ_k is the tightening parameter given by

$$\gamma_k = \sqrt{2\boldsymbol{g}_k^\top \mathbf{\Sigma}_k \boldsymbol{g}_k}\, \mathrm{erf}^{-1}(2\beta - 1) \tag{2.8}$$

With $\mathrm{erf}^{-1}(.)$ denoting the inverse error function. Making the constraint deterministic makes it tractable. However, perception uncertainties are not necessarily Gaussian. Noise due to multi-path effects, for example, can lead to different, non-Gaussian noise (e.g., Rayleigh-distributed noise [34]). When the uncertainties are non-Gaussian, another approach is needed.

When the uncertainty distribution is not known, but at least the fist two moments of the distribution are known, it is still possible to convert the probabilistic constraint into a deterministic bound using Chebyshev's inequality. Let us consider a random variable X with mean μ_X and variance σ^2. Chebyshev's inequality states that

$$\Pr(|X - \mu_X| < \kappa\sigma) \geq 1 - \frac{1}{\kappa^2} \tag{2.9}$$

for any number $\kappa > 0$. If we consider a threshold $c_{th} = \kappa\sigma$ it follows that

$$\Pr(|X - \mu_X| < c_{th}) \geq 1 - \frac{\sigma^2}{c_{th}^2} \tag{2.10}$$

The one-sided Chebyshev's inequality, known as Cantelli's inequality, can be derived in a similar fashion

$$\Pr(|X - \mu_X| < c_{th}) \geq 1 - \frac{\sigma^2}{\sigma^2 + c_{th}^2} \tag{2.11}$$

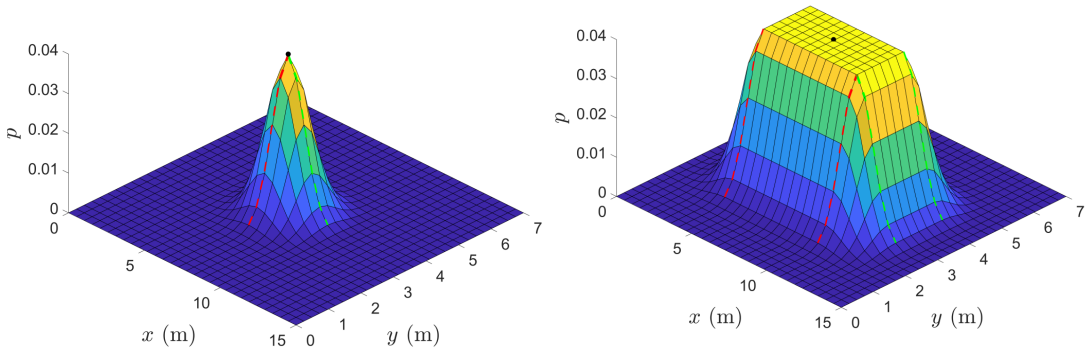
Chebyshev's inequality and Cantelli's inequality do not assume a specific distribution shape and therefore overestimate the risk bound. However, since the risk parameter β controls a trade-off between efficiency and risk, it is possible to increase the risk if the model becomes too conservative. For more details on stochastic model predictive control, see [35].

2.3. Grid-Based Stochastic Model Predictive Control

Grid-based stochastic model predictive control [13] is a promising approach for efficient optimization by reformulation constraints in uncertain environments as tractable constraints. The method deals with perception uncertainties by generating a probabilistic grid of the environment's obstacles and transforming it into a simplified deterministic set of linear state constraints. The aim of grid-based stochastic model predictive control is to balance computational efficiency with effective obstacle avoidance in dynamic and uncertain environments.

In grid-based stochastic model predictive control, obstacles are represented by a probability grid, where each grid cell holds a value related to the occupancy likelihood of surrounding obstacles, much like the values of a probability density function. Occupancy grids are a well-established approach for representing uncertain environments, particularly in automated driving applications. They effectively manage perception uncertainties and dynamic state estimation [36]. More specifically, approaches such as dynamic occupancy grids [37], incorporate shape estimation and motion dynamics to address the uncertain state and geometry of obstacles. In grid-based SMPC, the obstacle shape and state uncertainty is assumed to be known and the center of gravity location uncertainty is expanded along the vehicle dimensions to obtain a probabilistic grid representation of the free space. This probability grid is

transformed into a binary grid via a user-defined probability threshold by designating individual cells as "occupied" or "free", effectively creating a binary grid that represents safe and unsafe regions.



(a) Probability occupancy grid given TV center of gravity

(b) Probability occupancy grid expanded along the target vehicle dimensions

Figure 2.8: Expansion of probability density function uncertain target vehicle state [13]

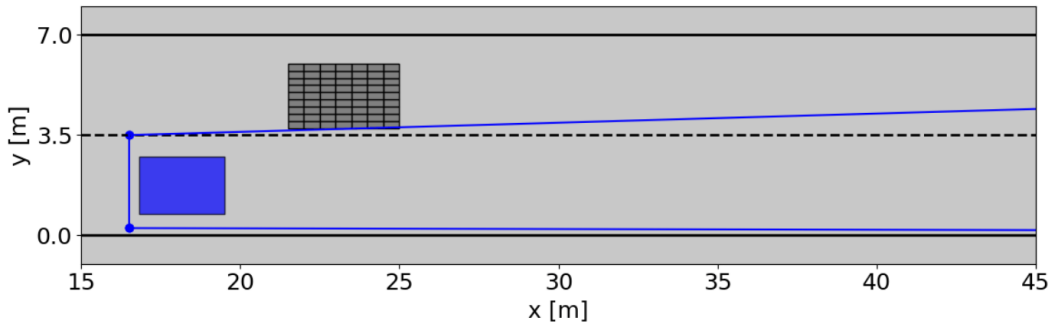
Within the binary grid, Bresenham's line algorithm is applied to check whether lines connecting cells are free from obstacles. Based on the identified free lines, a convex hull of free space is generated, providing an approximation of the safe area that accounts for obstacle uncertainty. This convex free space allows the algorithm to simplify and speed up the optimization process, as it restricts the search to a defined, convex area [38].

2.3.1. Limitations

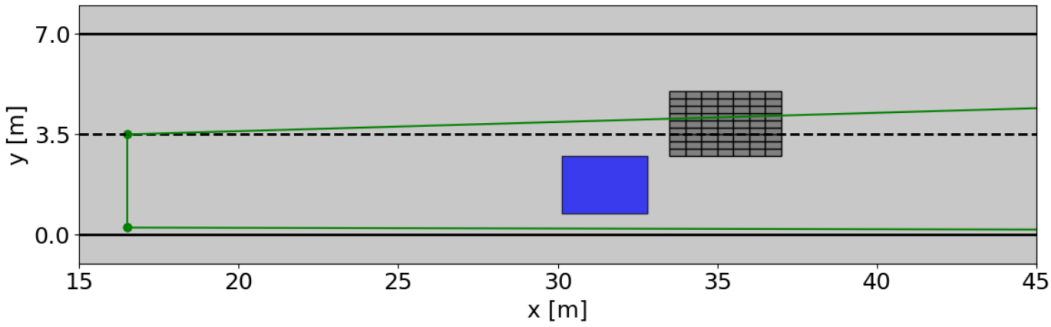
Grid-based stochastic model predictive control is a promising approach for the problem at hand. The method allows for simplification of the state constraints in an efficient manner, that scales well with the number of obstacles. This reduced complexity allows for a more efficient optimization by reducing the free space to a convex region. The method however, is subject to some limitations, presenting challenges for implementation in real-world low-friction conditions.

One such limitation is the ego vehicle prediction model complexity. The method employs a kinematic bicycle model. While the kinematic model allows for the demonstration of the constraint reformulation, it is limited when applied to scenarios in low-friction conditions. A kinematic model assumes the vehicle motion to be determined by geometric and kinematic relationships. In low-friction conditions, the maximum tire forces are influenced by the wheel-road friction coefficient. Since the kinematic model does not account for tire forces, it fails to capture possible wheel friction limitations, which is essential for accurate trajectory predictions. As a result, predictions may deviate from the vehicle's actual behavior, leading to unsafe or infeasible planned trajectories. Properly considering these friction limitations would require integrating a more complex dynamic model and a tire model that considers the friction coefficient.

Another limitation lies in the way the method handles situations where no convex hull can be found for a given prediction step k. In such cases, the method falls back on the convex hull computed for the previous prediction step $(k-1)$. While this approach ensures the optimization process can continue without interruption, it introduces a significant risk, since the convex hull of a preceding prediction step represents free space based on the state of the environment at an earlier moment in time. Reusing this hull assumes that the same space remains unoccupied at the current time step. However, since the environment is dynamic, this assumption is likely not true. As a result, the method might consider space as free, while it is occupied. Figure 2.9 shows an example of how this approach could lead to unsafe scenarios. To ensure safe trajectories, a new back-up method is required that considers the current occupied space. Such a method should reflect real-time environmental changes, ensuring that the optimization problem aligns with the actual state of the surroundings.



(a) Convex hull at the beginning of the prediction horizon, beginning of the lane change of the target vehicle.



(b) Convex hull later in the prediction horizon. Target vehicle moves in front of ego vehicle and nominal hull method fails. Falling back on the hull of fig. 2.9a. Occupied cells are inside the convex hull, thus occupied cells are deemed free space.

Figure 2.9: Unsafe back-up method in baseline framework. Both vehicles are moving in the positive x direction and the target vehicle is changing lanes. When the nominal hull finding algorithm fails (fig. 2.9b), the method falls back on the hull computed at a previous prediction step (fig. 2.9a), while the environment has changed, leading to an unsafe hull deemed safe. The ego vehicle is represented by a blue rectangle, occupied grid cells are dark gray. The nominal convex hull (in fig. 2.9a) is given in blue. The back-up convex hull (in fig. 2.9b) is given in green. These hulls represent the reformulated free space.

A further limitation of the method is that it does not consider the physical dimensions of the ego vehicle. The approach assumes that ensuring the center of gravity of the ego vehicle remains inside the computed convex hull is sufficient for obstacle avoidance. Not considering the physical dimensions of the ego vehicle can lead to unsafe scenarios where the center of gravity might be within the convex hull, but corners and edges extend beyond the hull (see Figure 2.10), resulting in collisions with obstacles. In real-world applications, considering the dimensions of the ego vehicle are critical for ensuring safe and feasible maneuvers, particularly in scenarios where obstacles are close to the vehicle's trajectory. Not considering the physical dimensions makes the method less reliable, as it is not guaranteed that the entire vehicle remains within safe boundaries.

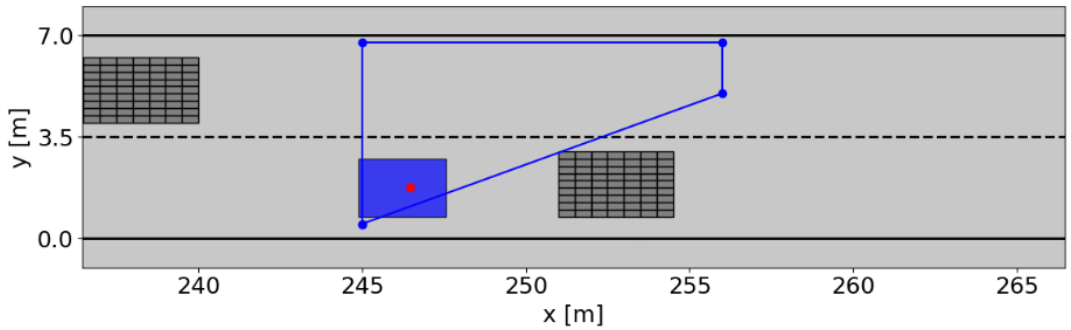


Figure 2.10: Composed hull encapsulates ego vehicle center of gravity, but not entire vehicle

The final limitation of the method lies in its use of a static threshold to determine the binary occupancy grid from the probability grid. PDFs with higher variances are flatter and distribute likelihood values across a larger area, whereas PDFs with lower variances are more peaked and localized, since probabilities sum to one. A static threshold applied uniformly across all grid cells fails to adapt to these differences in variance. This can lead to inconsistencies in the representation of occupied and free space. This problem is even more pronounced when the uncertainties grow over time, leading to flatter PDFs. To address this limitation, an adaptive threshold could be employed. This threshold would be dynamically adjusted based on current uncertainty characteristics, ensuring that the binary grid more accurately reflects the underlying PDFs.

3

Methodology

To effectively account for perception uncertainties in motion planning in low tire-road friction conditions, this chapter presents the proposed framework. The methodology builds upon the grid-based stochastic model predictive control framework [13] discussed in the previous chapter and incorporates a non-linear vehicle model (based on [9]) to account for low-friction conditions. Additionally, a novel back-up method is proposed to improve safety and feasibility in challenging scenarios. This chapter is structured as follows: The architecture of the stochastic controller is described first, followed by a detailed explanation of constraint formulation. The final section discusses the novel back-up method.

3.1. Non-Linear Stochastic Controller

An accurate predictive model is essential for performance in dynamic and uncertain environments. This section will discuss the architecture of the proposed prediction model and how it will solve the optimal control problem.

3.1.1. Coordinate System

The ego vehicle's state is described by state vector $\xi^{\mathrm{EV}} = \left[x, y, \psi, v_x, v_y, r, F_{xf}, \delta\right]$, where x and y represent the longitudinal and lateral position of the ego vehicle center of gravity (CoG) in the world frame, ψ describes the angle between the vehicle reference frame and the world reference frame. Longitudinal and lateral velocity of the ego vehicle CoG in the vehicle frame are given by v_x and v_y respectively, and heading rate is given by r. F_{xf} is the longitudinal force on the front tires and δ is the steering angle. A schematic of the single-track model is given in Figure 3.1.

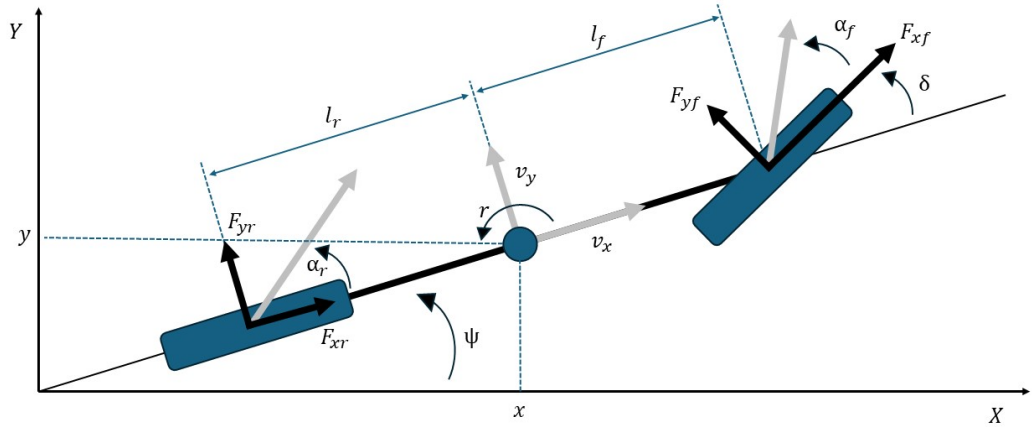


Figure 3.1: Single track model coordinate system schematic

3.1.2. Vehicle Model

A non-linear single-track dynamics prediction model is used to describe the motion of the controlled ego vehicle. The equations of motion are given by:

$$\begin{aligned}
\dot{x} &= v_x \cos(\psi) - v_y \sin(\psi) \\
\dot{y} &= v_x \sin(\psi) + v_y \cos(\psi) \\
\dot{\psi} &= r \\
\dot{v}_x &= \frac{-F_{yf} \sin(\delta) + F_{xf} \cos(\delta)}{m} + r \cdot v_y \\
\dot{v}_y &= \frac{F_{yf} \cos(\delta) + F_{xf} \sin(\delta) + F_{yr}}{m} - r \cdot v_x \\
\dot{r} &= \frac{l_f F_{yf} \cos(\delta) + l_f F_{xf} \sin(\delta) - l_r F_{yr}}{I_{zz}} \\
\dot{F}_{xf} &= m\dot{a}_x \\
\dot{\delta} &= \dot{\delta}
\end{aligned} \tag{3.1}$$

l_f and l_r are the distances from the center of gravity to the wheel axes of the front and rear wheels respectively. F_{yf} and F_{yr} are the lateral forces and F_{xf} and F_{xr} are the longitudinal forces for the front and rear wheel respectively. m correspond to the vehicle mass and I_{zz} is the vehicle moment of inertia around the z-axis. The vehicle is modeled as front-wheel driven where acceleration and braking is described by F_{xf}, thus the back wheel is assumed to be free-rolling. $\dot{a}_x$ and $\dot{\delta}_x$ are longitudinal jerk and steering rate respectively, which are given to the system as inputs.

3.1.3. Tire Model

The maximum tire forces are modeled, to account for reduced tire-road friction. Tire forces are modeled using a Fiala tire model (often referred to as brush tire model) similarly to [9] to consider different tire-road friction coefficients. A brush model is chosen for its ability to model the effects of lower friction conditions, while being relatively simple, restraining the computational complexity [39]. The Fiala tire model relates lateral tire force to tire slip angles α:

$$\alpha_f = \tan^{-1}\left(\frac{v_y + l_f r}{v_x}\right) - \delta \tag{3.2}$$

$$\alpha_r = \tan^{-1}\left(\frac{v_y - l_r r}{v_x}\right) \tag{3.3}$$

Where subscripts f and r denote the combined front and rear wheels in the bicycle model (see Figure 3.1). The maximum lateral tire force $F_{y,max}$ is modeled using the friction circle. Lateral force can then be described as a function of lateral slip angle α, longitudinal force F_{xf} and normal load F_z, friction coefficient μ and axle cornering stiffness C_α

$$F_{y,\max} = \sqrt{\left(\mu F_z\right)^2 - F_x^2} \tag{3.4}$$

$$F_y = \begin{cases} -C_\alpha \tan\alpha + \frac{C_\alpha^2}{3F_{y,\max}} |\tan\alpha| \tan\alpha - \frac{C_\alpha^3}{27\left(F_{y,\max}\right)^2} \tan^3\alpha, & |\alpha| < \tan^{-1}\left(\frac{3F_{y,\max}}{C_\alpha}\right) \\ -F_{y,\max} \operatorname{sgn}\alpha, & \text{otherwise} \end{cases} \tag{3.5}$$

Normal load on the front and rear tires is modeled

$$\begin{aligned}
F_{zf} &= \frac{1}{L}\left(mgl_r - h_{CoG} F_{xf}\right) \\
F_{zr} &= \frac{1}{L}\left(mgl_f + h_{CoG} F_{xf}\right)
\end{aligned} \tag{3.6}$$

where h_{CoG} is the height of the center of gravity above the ground and g is the gravitational constant. This weight transfer function captures the effect increased force of the front wheel when braking, without adding any additional states. Lateral weight transfer is ignored.

3.1.4. Discretization

To solve the optimal control problem, the proposed method relies on the discretization of continuous-time dynamics. The prediction model is discretized using Runge-Kutta 4, known for its accuracy. While Runge-Kutta 2 could be a more efficient integrator, balancing performance and efficiency [9], a built-in Runge-Kutta 2 integrator was not readily available in the utilized CasADi environment. Given the time constraints of this project and the focus on model development rather than real-time performance, implementing a Runge-Kutta 2 integrator is beyond the scope of this thesis and the Runge-Kutta 4 method was utilized:

$$\begin{aligned} x(t+h) &= x(t) + \frac{1}{6} f(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 &= h \cdot f\left(x(t), u(t), t\right), \\ k_2 &= h \cdot f\left(x(t) + \frac{k_1}{2}, u(t), t + \frac{h}{2}\right), \\ k_3 &= h \cdot f\left(x(t) + \frac{k_2}{2}, u(t), t + \frac{h}{2}\right), \\ k_4 &= h \cdot f\left(x(t) + k_3, u(t+h), t + h\right). \end{aligned} \tag{3.7}$$

While Runge-Kutta 4 is generally more computationally demanding than Runge-Kutta 2, it provides similar levels of accuracy. Since this implementation is focused on safety and feasibility, where real-time execution is not a primary concern, the additional computational cost of Runge-Kutta 4 is neglected and this possible improvement in computational performance is left to possible future work.

3.1.5. Target Vehicles

A clear representation of other road participants (Target vehicles) and the ego vehicle (EV) is required for proper obstacle avoidance. Target vehicles (TVs) are defined as obstacles with uncertain state. This uncertainty is caused by both the uncertainty in perception as well as uncertain behavior of the traffic participants. The target vehicle state is predicted using a linear state-space model with an additional uncertainty term $\boldsymbol{G}\boldsymbol{w}_k^{\mathrm{TV}}$, where $\boldsymbol{w}_k^{\mathrm{TV}}$ is a continuous random variable and $\boldsymbol{G} = \mathrm{diag}(g_1, g_2, g_3, g_4)$ is a diagonal matrix describing different uncertainties in the target vehicle state. The target vehicle state is modeled by the system

$$\boldsymbol{\xi}_{k+1}^{\mathrm{TV}} = \boldsymbol{A}\boldsymbol{\xi}_k^{\mathrm{TV}} + \boldsymbol{B}\boldsymbol{u}_k^{\mathrm{TV}} + \boldsymbol{G}\boldsymbol{w}_k^{\mathrm{TV}} \tag{3.8}$$

Where $\boldsymbol{\xi}_k^{\mathrm{TV}} = \left[x_k^{\mathrm{TV}}, v_{x,k}^{\mathrm{TV}}, y_k^{\mathrm{TV}}, v_{y,k}^{\mathrm{TV}}\right]^\top$ is the target vehicle state at time step k, existing of longitudinal position and velocity and lateral position and velocity respectively. $\boldsymbol{u}_k^{\mathrm{TV}} = \left[a_{x,k}^{\mathrm{TV}}, a_{y,k}^{\mathrm{TV}}\right]$ is the control input of the target vehicle, given by the longitudinal and lateral acceleration respectively, with the reference trajectory $\boldsymbol{\xi}_{k,\mathrm{ref}}^{\mathrm{TV}}$ and feedback law $\boldsymbol{u}_k^{\mathrm{TV}} = \boldsymbol{K}\left(\boldsymbol{\xi}_k^{\mathrm{TV}} - \boldsymbol{\xi}_{k,\mathrm{ref}}^{\mathrm{TV}}\right)$, where $\boldsymbol{K} = \begin{bmatrix} 0 & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}$

The uncertainty at prediction step k is given by $\boldsymbol{\Sigma}_k$. It assumed that the reference trajectory $\boldsymbol{\xi}_{\mathrm{ref}}^{\mathrm{TV}}$ of the target vehicle is known and that a representative uncertainty of the states is known. The uncertainty is assumed to be Gaussian. The uncertainty covariance matrix is propagated through the prediction horizon using linear Gaussian propagation to obtain the uncertain vehicle states at every time step in the prediction horizon.

$$\boldsymbol{\Sigma}_{k+1} = (\boldsymbol{A} + \boldsymbol{BK})\boldsymbol{\Sigma}_k(\boldsymbol{A} + \boldsymbol{BK})^\top + \boldsymbol{G}\boldsymbol{\Sigma}_w\boldsymbol{G}^\top \tag{3.9}$$

Where $\boldsymbol{\Sigma}_0$ is the initial perception uncertainty, and $\boldsymbol{G}\boldsymbol{\Sigma}_w\boldsymbol{G}^\top$ describes the prediction uncertainty.

Using these uncertain vehicle states as chance constraints directly is very computationally expensive [40]. Therefore, they are converted to tractable constraints. This is done by converting the probabilities into a hull describing a convex free space with a set of linear constraints (see Figure 3.2). A detailed description of this constraint formulation is discussed in Section 3.2.

3.1.6. Ego Vehicle Dimensions

The physical dimensions of the ego vehicle are modeled as a circle around the center of gravity of the ego vehicle with a radius l_f (see Figure 3.2). A circle representation allows for simple constraint formulation to ensure the circle is inside the determined free space, by constraining the distance of the ego vehicle CoG to the occupied space boundary to be at most radius l_f. It is assumed that rotation are small, and $v_x \geq 0$. Since the rear of the vehicle follows the trajectory of the front, its impact on collision checks and free-space constraints is minimal. Representing the ego vehicle as a singular circle allows for an approximate representation of the ego vehicle dimensions without adding significant computational load.

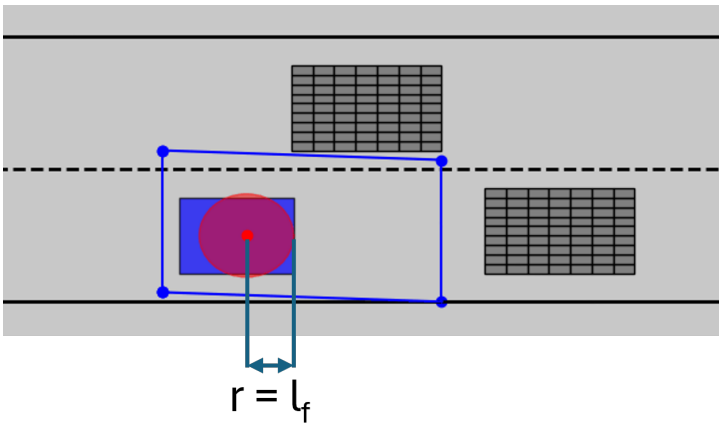


Figure 3.2: Representation of the vehicle dimensions (red circle) and convex free space (blue lines)

3.1.7. Cost Function

As discussed in Section 2.2 model predictive control minimizes a cost function in order to find an optimal control input. The cost function includes multiple terms relating to states, inputs and constraint violation. The cost function is given by

$$J(\boldsymbol{x}_k, \boldsymbol{u}_k) = \Bigg(Q_y \left(y^k - y^k_{\text{ref}}\right)^2 + Q_{v_x}\left(v^k_x - v^k_{x,\text{ref}}\right)^2 + Q_{a_x}\left(a^k_x\right)^2 + Q_\delta\left(\delta^k\right)^2 + Q_{\dot{a}_x}\left(\dot{a}^k_x\right)^2 + Q_{\dot{\delta}}\left(\dot{\delta}^k\right)^2 + Q_S(\|\boldsymbol{S}^k\|_2)^2 \Bigg) \tag{3.10}$$

Where Q_y and Q_{v_x} are quadratic weights on the lateral position error and longitudinal velocity error, Q_{a_x} and $Q_{\dot{a}_x}$ are quadratic weights on longitudinal acceleration and its derivative and Q_δ and $Q_{\dot{\delta}}$ are quadratic weights on steering angle and steering rate. Q_S is a large quadratic weight on the magnitude of slack vector $\boldsymbol{S}$. These cost parameters are similar to [9] such that reasonable values for each term are scaled to unity, except for the constraint violation term. This term is higher by orders of magnitude to discourage any constraint violation. More on the role of this cost term will be discussed in Section 3.2.4. The cost function weights and control parameters for the proposed controller are given in Table 3.1

Parameter	Description	Value
N	Number of stages in prediction horizon	50
h	Number of ms between stages	50
Q_y	Weight for lateral position error	$\frac{1}{(0.50)^2}$
Q_{vx}	Weight for longitudinal speed error	$\frac{1}{(5)^2}$
Q_{ax}	Weight for longitudinal acceleration	0.1
Q_δ	Weight for steering angle	10
$Q_{\dot{a}x}$	Weight for longitudinal jerk	$\frac{1}{(5)^2}$
$Q_{\dot{\delta}}$	Weight for steering rate	$\frac{1}{(10)^2}$
Q_S	Weight for constraint violation	1000

Table 3.1: Cost function weights and control parameters (based on [9])

3.1.8. Reference Trajectory

The proposed cost function minimizes the state difference with a reference trajectory. This subsection will discuss how the reference trajectory is determined. The cost function considers a cost on velocity in longitudinal direction and in lateral position. Since there is no cost in longitudinal position, only the lateral position reference will be considered.

A check is performed at the first prediction step of every control iteration to check which lanes are free in the proximity of the ego vehicle. This check is similar to the method to determine the free space described in further detail in Section 3.2.3. At 20 meter in front of the ego vehicle, the entire width of the road is analyzed to determine which lanes are free at this exploration radius. If a lane is free at this radius, a check is performed to verify wether the lane segment between the ego vehicle and the free lane at this radius is free (green area in Figure 3.3). If both checks are successful, the lane is considered free. If the current lane is free, the reference lateral position will remain the middle of the current lane. If not, the closest free lane is considered. If a free path exists to any of the lanes, this lane will be considered the new reference lane. If no free lane is found at the current exploration radius, the process is repeated at a slightly shorter exploration radius, until a free lane is found.

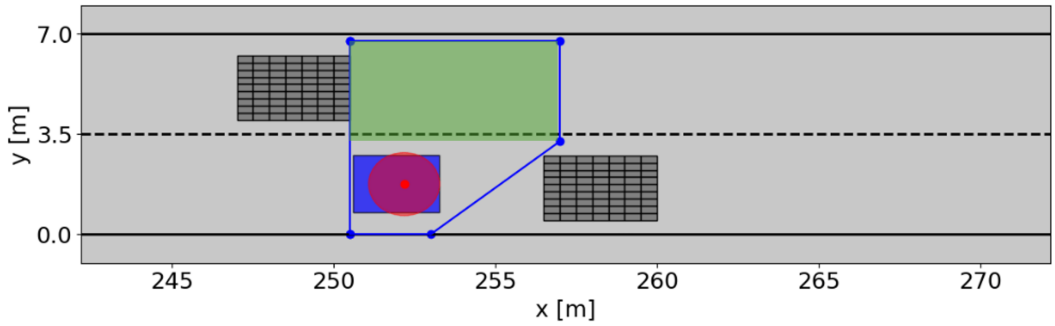


Figure 3.3: Determination of a free lane. Lane considered free when entire lane segment from the back of the vehicle to the exploration radius is free. Adjacent lane is free, so reference lane is updated to the adjacent lane.

The reference lateral position is the middle of the current lane if the current lane is free. For good optimization performance, it is important that the reference trajectory of the lane transition is smooth. The reference lateral position is generated using the following function:

$$N_{\text{change}} = \left\lfloor \frac{v_x \cdot |y_{\text{new}} - y_{\text{old}}|}{25 \cdot \Delta t} \right\rfloor \tag{3.11}$$

$$y_{\text{ref}}(k) = \begin{cases} \frac{y_{\text{old}}+y_{\text{new}}}{2} + \frac{y_{\text{old}}-y_{\text{new}}}{2} \cos\left(\pi \frac{k-k_{\text{current}}}{N_{\text{change}}}\right), & k \in [k_{\text{current}}, k_{\text{current}} + N_{\text{change}}] \\ y_{\text{new}}, & k > k_{\text{current}} + N_{\text{change}} \end{cases} \tag{3.12}$$

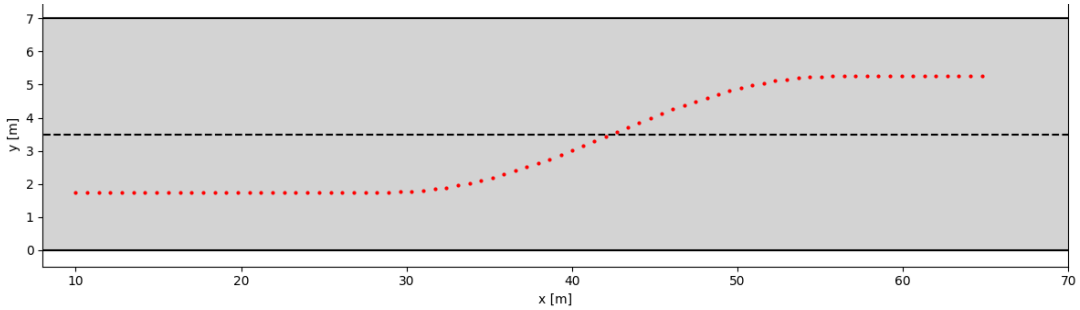


Figure 3.4: Reference trajectory

This function describes half a wavelength of a scaled and transposed cosine function such that the the start of the function lies on the original lane with zero gradient and the end lies on the target lane, with zero gradient (see Figure 3.4). This allows for a smooth transition between the two lanes. At lower tire-road friction, available lateral tire force is limited. To account for lateral forces, the term N_{change} considers the transition duration. This parameter is determined by considering the lateral acceleration relationship

$$a_y = v_x^2/R \tag{3.13}$$

When considering the traveled lateral distance and the amount of time steps to do this we can consider this relationship as an effective turning radius by

$$R \approx \frac{x_{end} - x_{begin}}{y_{new} - y_{old}} = \frac{N_{change} v_x \Delta t}{y_{new} - y_{old}} \tag{3.14}$$

Where $x_{end} - x_{begin}$ is the difference in longitudinal position between the end and beginning of the lane change. y_{old} is lateral position of the middle of the initial lane and y_{new} is the lateral position of the middle of the new reference lane. Combining (3.13) and (3.14) gives the proportionality:

$$a_y \propto \frac{v_x(y_{new} - y_{old})}{N_{change}\Delta t} \tag{3.15}$$

Using this proportionality, N_{change} can be described as function of the vehicle velocity and time step resulting in similar lateral accelerations and tire forces at various longitudinal velocity values.

3.2. Constraint Formulation

To achieve real time performance it is essential to define state constraints which are representative of the real environment while not significantly increasing the computational load. This section discusses how representative and tractable constraints are formulated, while considering the uncertain states of obstacles. This is done by dividing the space into a grid and the considering the uncertainty as probabilities of occupation of certain grid cells. This probability grid is converted to a binary grid by defining a probability threshold above which cells are considered occupied space. This threshold is dynamically set considering uncertainty magnitudes. From this binary grid, a convex hull is derived, constraining the optimization problem to a convex region, reducing computational complexity. Furthermore, the method remains efficient with increasing obstacles since the method requires a single iteration independent of the number of obstacles.

3.2.1. Probability Grid

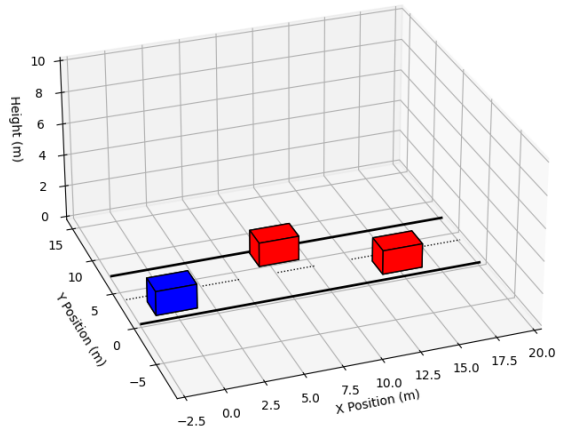
For each time step in the SMPC time horizon, a probabilistic occupancy grid is formulated where grid cell values are related to occupancy likelihood in the open-loop solution in (3.8). The environment is represented by a grid $\mathcal{G}$, i.e., a uniformly spaced field of cells $\boldsymbol{c}_{i,j} \in \mathcal{G}$ with $\boldsymbol{c}_{i,j} = (c_i, c_j)$. Cells

have dimensions l_x and l_y accounting for length and width respectively, with 2D indices i and j. For every time step of the optimization horizon, a probabilistic grid is generated. The probabilistic grid is represented by a matrix P consisting of elements $p_{i,j}$ which describe the occupancy probability of cell $\boldsymbol{c}_{i,j}$ being occupied by a target vehicle. Considering Gaussian probability distributions of the location of the center of gravity of the target vehicles, the probability density function assigning probability values to matrix P is given by

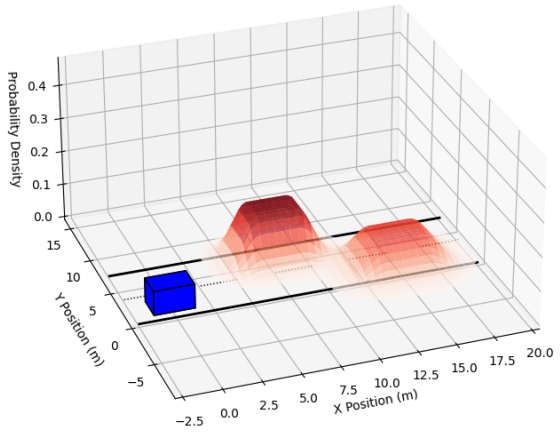
$$f_{c^{\mathrm{TV}}}(\boldsymbol{c}) = \frac{\exp\left(-\frac{1}{2}\left(\boldsymbol{c}-\boldsymbol{c}_k^{\mathrm{TV}}\right)^{\top}\boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{c}-\boldsymbol{c}_k^{\mathrm{TV}}\right)\right)}{\sqrt{(2\pi)^2\,|\boldsymbol{\Sigma}_k|}} \tag{3.16}$$

here $\boldsymbol{c}_k^{\mathrm{TV}}$ denotes the grid cell associated with the center of gravity of the TV at prediction step k and $\boldsymbol{\Sigma}_k$ is the covariance matrix of the target vehicle state.

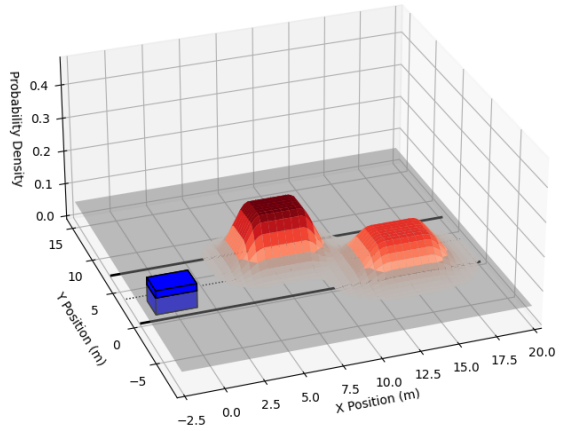
The peak value in this grid representation related to the mean value of the uncertain CoG location is converted to a spatial representation of the occupancy likelihood by considering the target vehicle dimensions and expanding the maximum value of the probability density function along the axes of the target vehicle (visualized in Figure 2.8). Note that Figure 2.8b does not show a probability distribution anymore since the distribution is artificially expanded.



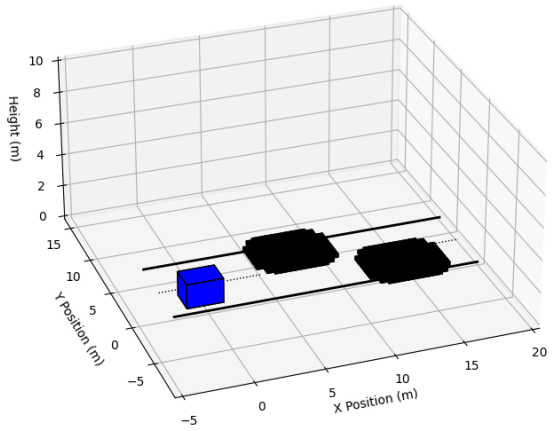
(a) Environment, ego vehicle in blue, target vehicles in red



(b) Occupancy probabilities expanded to consider vehicle dimensions



(c) Probability threshold converting probability grid to a binary grid



(d) Binary grid

Figure 3.5: Computation of a deterministic obstacle free space from uncertain target vehicle states

This approach is extended to multiple target vehicles by computing probability grids for every target vehicle and summing the values for each cell p_{i,j,n_v} to obtain a value for each cell $p_{i,j}$

$$p_{i,j} = \sum_{n_v=1}^{N_v} p_{i,j,n_v} \tag{3.17}$$

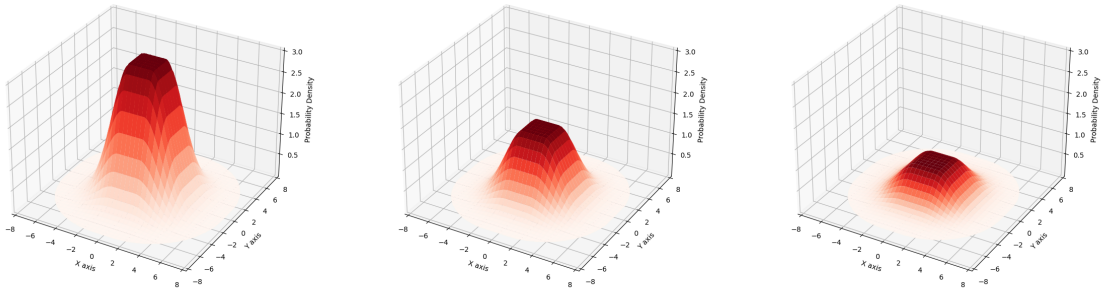
where N_v is the number of target vehicles localized in the detection range of the ego vehicle. The aggregated probability grid is visualized in Figure 3.5b.

3.2.2. Binary Grid

The obtained probability grid represents the likelihood of the occupancy of the environment. This probability grid is converted to a binary grid to obtain a deterministic constraint formulation. This is achieved by transforming the probabilistic grid P into a binary grid B where elements $b_{i,j}$ are determined with probability threshold p_{th} (see Figure 3.5c),

$$b_{i,j} = \begin{cases} 1, & \text{if } p_{i,j} \geq p_{\text{th}} \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

Here, a $b_{i,j}$ value of 1 indicates that a grid cell is considered occupied, a value of 0 indicates free space (see Figure 3.5d). The parameter p_{th} represents a trade-off between risk and conservatism.



(a) Probability grid $\sigma = 0.05m$ **(b)** Probability grid $\sigma = 0.1m$ **(c)** Probability grid $\sigma = 0.2m$

Figure 3.6: Influence of variance on probability density function shape. Higher variance corresponds with probability density functions that are wider with lower peaks. Similarly, probability grids derived from extended probability density grids exhibit broader distributions with reduced peak values.

Aggregating different probability density functions into a single probability grid allows for efficient computation of a binary grid. However, to accurately consider the uncertainty of the obstacle states, it is important to consider how p_{th} is determined and what this threshold represents in the probability grid. Probability values are generated according to (3.16), where the peak value of this distribution is extended along the vehicle dimensions. Since this is a probability density function, the sum of the function is equal to 1. It follows that a higher variance and thus a wider distribution have a lower peak value than a distribution with a lower variance (See Figure 3.6). It is therefore important that the probability threshold is representative of the uncertainties at the current state in order to accurately represent the admissible space. This is particularly important when considering consecutive time steps in the prediction horizon, because the uncertainty and thus distribution variance grows significantly over time (Equation 3.9).

A dynamic threshold is therefore proposed, which is dynamically determined, based on the uncertainties present at the predicted time step. The variances of all uncertain target vehicles are sorted by variance. The distribution with highest variance is considered and a used-defined confidence interval boundary (β%) is determined using the quantile function of the chi-squared distribution

$$q = \chi_2^2(\beta) \tag{3.19}$$

where $\beta\%$ of the probability density lies within this boundary. The value of the PDF at this boundary of the corresponding probability density function is used as threshold.

$$p_{th} = \frac{e^{-\frac{q}{2}}}{2\pi\sqrt{\sigma_x^2\sigma_y^2}} \tag{3.20}$$

The threshold is based on the uncertainty with the highest variance since it will lead to a conservative approximation of the free space where at least $\beta\%$ of all other probability density functions are considered occupied space.

Inserting this threshold in (3.18), the chance constraint can now be expressed as a deterministic constraint

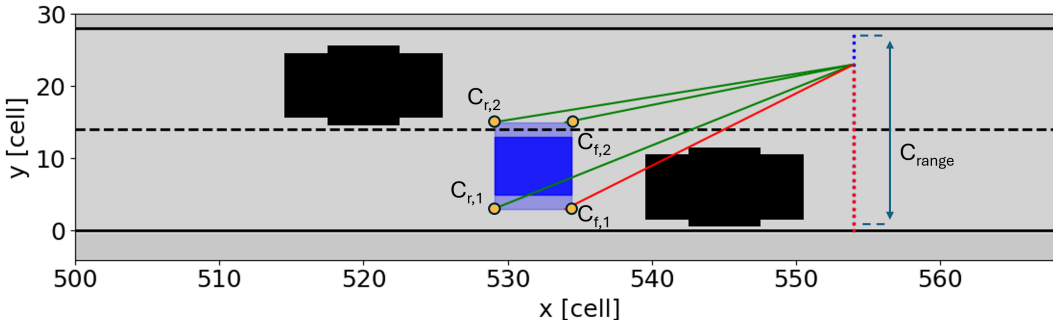
$$\xi^{\mathrm{EV}} \in \Xi^{\mathrm{adm}} \tag{3.21}$$

where Ξ^{adm} includes all admissible cells in the binary grid.

3.2.3. Convex Hull Formulation

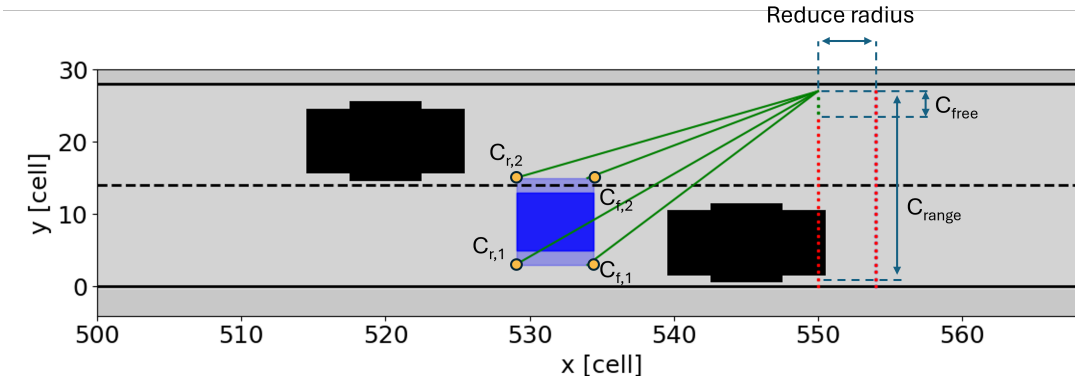
Once the binary grid is obtained, a convex, safe state space is determined. The state constraint is transformed into a set of linear inequality constraints to enable rapid optimization, similar to [13], but extending it to consider the physical dimension of the ego vehicle. Bresenham's line algorithm [41] is used to check lines between grid cells from the corners of the ego vehicle into the driving direction. This algorithm yields all cells connecting two input cell by a straight line in a grid. Checking these cells yields admissible cells between the two input points, effectively checking wether the straight line between the points is free. This check is performed between all corners of the ego vehicle and a point at a exploration radius, where the ego vehicle state is the predicted state given by the open-loop solution of (2.3) corresponding to the time step of the considered binary grid. If all four corners yield in a clear view to the point, it is considered a free line and the point at the exploration radius is added to the free range (See Figure 3.7b).

This check is performed across the road width at the exploration radius (Figure 3.7a). If a sufficiently wide free range (at least three meters) is found for the vehicle to pass through, a convex hull is composed around the outer cells of the free range and the corners of the ego vehicle (see Figure 3.8a). If no valid hull is found, the same algorithm is performed again at a reduced exploration radius (see Figure 3.7b), until a valid hull is found or the minimum exploration radius (one vehicle length in front of the vehicles) is reached without finding a valid hull.



(a) Line connecting the corners of the vehicle to cell in the exploration radius is checked for obstacles. No free line found for one of the corners, cells is not added to C_{free}

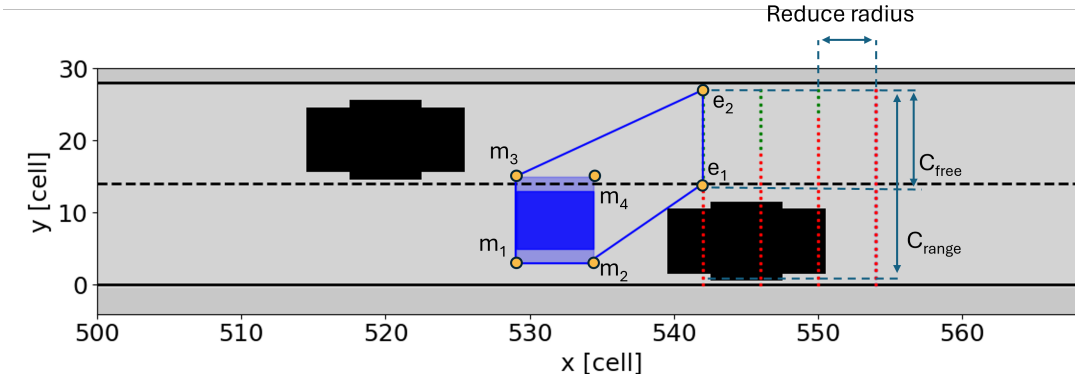
Figure 3.7: Convex hull construction algorithm. Exploration of free lines in the admissible space to construct a convex hull. Column of the current exploration radius is indicated with C_{range}, the largest free section of this column is indicated with C_{free}. Note that the checks are conducted slightly outside the corners of the ego vehicle, to allow for some maneuverability.



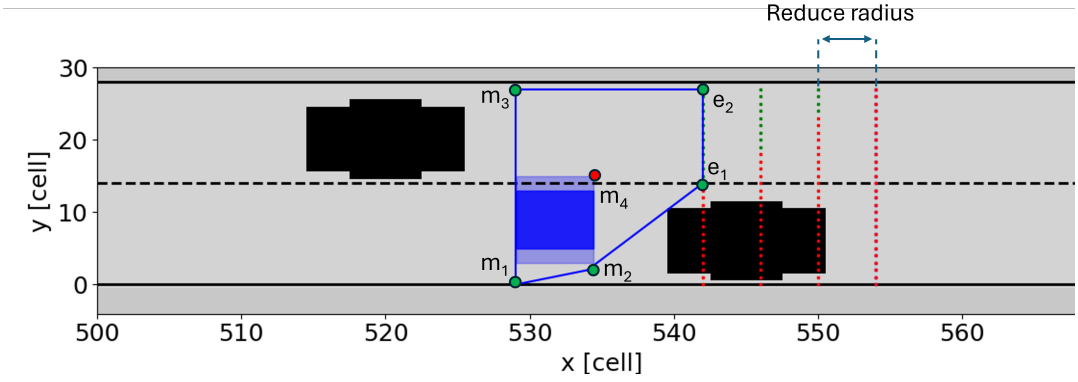
(b) No free range of at least $3m$ wide is found, the exploration radius is reduced. Free line for all four corners, cell is added to C_{free}

Figure 3.7: Convex hull construction algorithm. Exploration of free lines in the admissible space to construct a convex hull. Column of the current exploration radius is indicated with C_{range}, the largest free section of this column is indicated with C_{free}. Note that the checks are conducted slightly outside the corners of the ego vehicle, to allow for some maneuverability. (cont.)

When a valid hull is found it is expanded maximize the free space around the ego vehicle, allowing for more maneuverability. Dependent on the shape and orientation of the hull, either the front or the rear corners are expanded outwards, checking for free space using the Bresenham algorithm (see Figure 3.8b). This done while ensuring the hull remains within the road boundaries and remains convex. Creating a convex hull this way prioritizes forward exploration, while considering the required space due to the dimensions of the ego vehicle. The approach is described in more detail in Algorithm 1



(a) A free range (C_{range}) of at least $3m$ is found, hull is composed around the outer vertices



(b) Hull is expanded to maximize the convex area

Figure 3.8: Construction and expansion of a convex hull

Algorithm 1 Admissible Safe State Space search

Require: ξ_k^{EV}, B_k
Identify ego vehicle center c^{EV} and corners $c_{r,1}$, $c_{r,2}$, $c_{f,1}$ and $c_{f,2}$ (as in Figure 3.7a)
Select admissible cells in the detection range of the ego vehicle and store them in a matrix C_{range} (dotted line in Figure 3.7a)
for each cell c in C_{range} **do**
    Verify free path from c to $c_{r,1}$, c to $c_{r,2}$, c to $c_{f,1}$ and from c to $c_{f,2}$
    if free paths are confirmed **then**
        Store c in matrix C_{free}
    end if
end for
if width of $C_{\text{free}} \geq 3m$ **then**
    Select edge cells e_1 and e_2 of C_{free} (as in Figure 3.8a)
    Define m_1 as $c_{r,1}$, m_2 as $c_{f,1}$, m_3 as $c_{r,2}$ and m_4 as $c_{f,2}$
    if $e_{1,y} > m_{2,y}$ **then**
        Move down m_2 as long as free path exists between m_2, e_1 and m_2, m_1
        while slope$(m_1, m_2) \leq$ slope(m_2, e_1) **do**
            if free path exists between m_1 and m_2 **then**
                Move down m_1
            else
                Break
            end if
        end while
    else
        Move down m_1 as long as free path exists between m_1, e_1
    end if
    if $e_{2,y} < m_{4,y}$ **then**
        Move up m_4 as long as free path exists between m_4, e_2 and m_4, m_3
        while slope$(m_3, m_4) \geq$ slope(m_4, e_2) **do**
            if free path exists between m_3 and m_4 **then**
                Move up m_3
            else
                Break
            end if
        end while
    else
        Move up m_3 as long as free path exists between m_3, e_2
    end if
    Select the outer vertices of the convex hull
else
    Reduce exploration radius
end if
Ensure: Convert vertex-representation into linear inequality representation (3.22)

This algorithm results in the set of linear constraints (visualized in Figure 3.9)

$$\boldsymbol{A}_k^{\text{safe}} \boldsymbol{\xi}_k^{\text{EV}} \leq \boldsymbol{b}_k^{\text{safe}} \tag{3.22}$$

which are used the SMPC optimal control problem. A backup method is used in the event that no admissible convex hull is found. A detailed description on the implementation of this back-up method is given in Section 3.3.

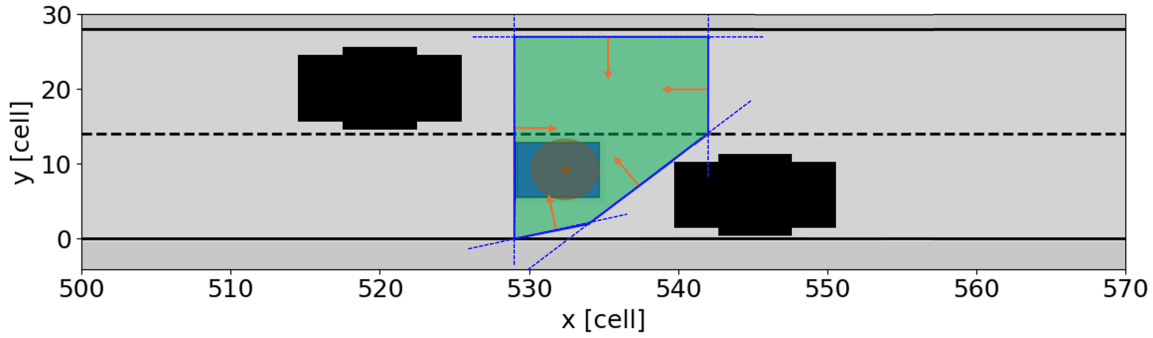


Figure 3.9: Convex hull described by a set of linear constraints

3.2.4. Soft Constraints

Dynamic and uncertain environments constantly evolve, and the perception of such environments is inherently noisy. This means that the perceived state of the environment changes frequently based on uncertain behavior or sensor updates. Consequently, the constraints derived from the perceived environment are also subject to frequent change over time. These changes can result in short inconsistencies in the sequence of convex hulls. Strict enforcement of these hull-based constraints under such conditions may lead to infeasibility in the optimization problem. To account for this, soft constraints are introduced. Rather than treating the composed constraints as hard boundaries, soft constraints allow for some constraint violation when necessary. These violation are penalized in the cost function with a high weight. This high weight discourages the constraint violation, but feasibility is maintained if it occurs.

By including soft constraints (3.22) becomes

$$\boldsymbol{A}_k^{\text{safe}}\, \boldsymbol{\xi}_k^{\text{EV}} \leq \boldsymbol{b}_k^{\text{safe}} + \boldsymbol{S}_k \tag{3.23}$$

Where $\boldsymbol{S}_k \geq 0$ describes the set of slack vectors at time step k and $||\boldsymbol{S}_{k,i}||_2$ is the magnitude of the slack vector $\boldsymbol{S}_{k,i}$ to constraint boundary i as shown in figure 3.10

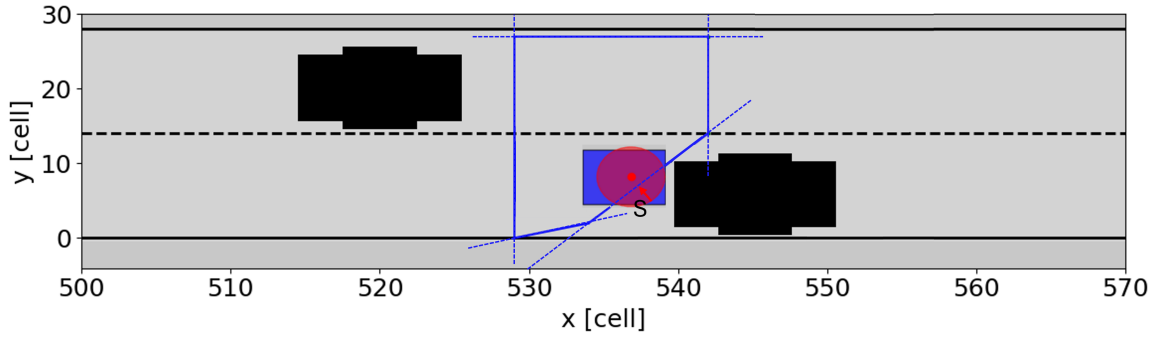


Figure 3.10: Constraint violation occurs when the ego vehicle is outside of the convex hull, S denotes the slack vector

3.3. Back-Up Method

The previous section described the constraint formulation, simplifying uncertain states into a set of linear inequality constraints, enabling the identification of a convex hull representing free space. However, the

method is not always capable of finding a viable hull. This section examines the circumstances under which no hull can be found and proposes an improved back-up strategy to address such scenarios.

3.3.1. Back-Up Necessity

To understand when the nominal approach fails, it is useful to revisit the process by which the hull is determined. As described in Section 3.2.3, the hull explores free space in the driving direction, aiming to maximize a convex admissible space with a bias toward the ego vehicle's intended trajectory. A hull is considered valid if it is at least 3 meters wide and spans at least the length of a vehicle in the driving direction.

Intuitively, these conditions imply that a hull will fail to be found if there is no sufficiently large region of free space around the ego vehicle in its intended trajectory. Such failures are observed in the following scenarios:

- Nearing towards obstacles: The ego vehicle drives toward the edge of the previously defined hull, and further expansion is blocked by obstacles, such as slower-moving vehicles ahead.
- Prediction variability: Due to environmental uncertainty or sensor noise, consecutive prediction horizons may differ, causing disruptions in hull continuity.
- Uncertainty growth: Over time, growing uncertainty in the vehicle's trajectory or obstacle positions may interfere with otherwise clear paths, leading to invalid hulls.

When no nominal hull is found, it typically indicates one of three situations:

- The ego vehicle is approaching or is in close proximity to other vehicles or obstacles.
- A realization of uncertainty has occurred that was not adequately accounted for in the preceding control iteration.
- The ego vehicle is violating the previously computed constraints.

In these scenarios, a robust back-up method is essential for maintaining safety and feasibility.

3.3.2. Approach

Since the situations necessitating a back-up hull describe tight scenarios, safety should be prioritized over trajectory efficiency. The proposed methods utilize the flexibility to tune the level of conservativeness of stochastic model predictive control, as detailed in Section 3.2.2. Considering that the situations where a back-up is required often involve tight scenarios, adopting a more conservative approach in these situations allows for additional safety. In this section, two novel back-up methods are proposed, both of which emphasize safety by considering a conservative realization of free space. The proposed methods differ in their underlying strategies and pose a trade-off between computational efficiency and accuracy.

The first approach computes nominal hulls for the steps in the prediction horizon, until such a hull cannot be found. If no nominal hull is found, the back-up method computes a more conservative hull for that particular prediction step. This hull is based on the computed probability grid during the nominal method (as described in Section 3.2.1). This grid is converted to a binary grid as described in Section 3.2.2. However, instead of the nominal computed threshold, a new, more conservative threshold is computed with a higher value for β in (3.18). As a result, the obtained binary grid represents a more conservative free space. Within this binary grid, the hull finding algorithm described in Section 3.2.3 is utilized. A flowchart displaying the architecture of a single optimization iteration utilizing this back-up method is given in Figure 3.13.

Since the free space is more conservative than the nominal binary grid, it is unlikely that a valid hull will be found in this space. The algorithm computing a convex hull from a binary grid however, is not limited to using the current ego vehicle position as an exploration starting point. Since the binary grid derived for the back-up is a more conservative representation of the free space, the approach is less likely to identify a viable convex hull in scenarios where a nominal hull was not found. However, the state of the obstacles and their associated uncertainties are assumed to be independent of the ego vehicle state, which allows for a key advantage: the hull exploration can be executed from different starting exploring states in the environment (c^{EV} in Algorithm 1). Specifically, the previous state of the open-loop solution is utilized as a reference state, since it is close to the current ego vehicle state,

maximizing continuity between successive prediction horizons. This process is repeated recursively until a back-up hull is found or no valid hull could be found for any of states in the prediction horizon, which would deem the problem infeasible. A flowchart of this process is given in Figure 3.11. The resulting back-up hull is a safe and more conservative representation of the free space that provide the necessary safety constraints to enable efficient optimization, even when no convex hull could be found utilizing the nominal approach.

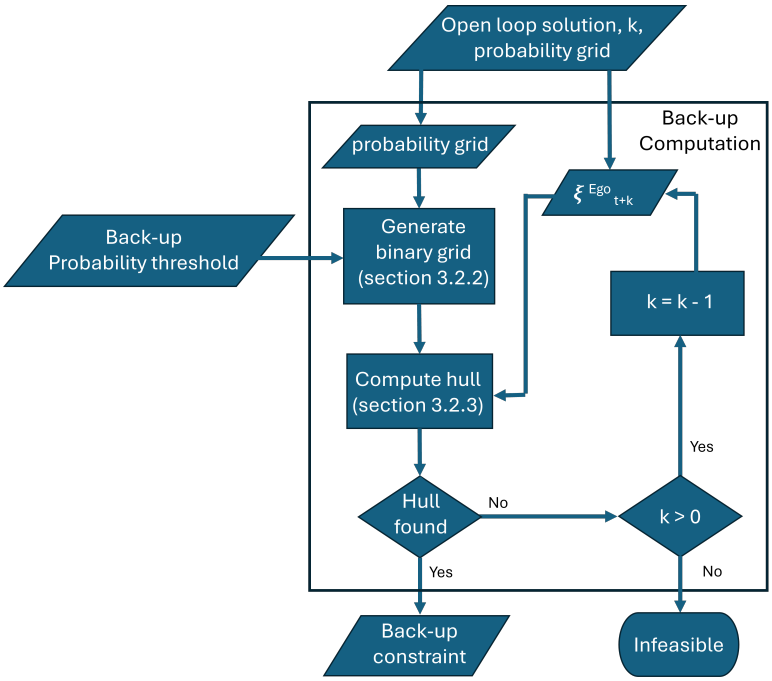


Figure 3.11: Flowchart of a back-up constraint computation. Converts the probability grid to a set of linear constraints describing a safe scape as back-up when the nominal approach fails. Back-up threshold allows for conservativeness tuning of the back-up method. Exploration for the free space is conducted initially from the input state ξ_k and recursively retraced along the prediction horizon until a back-up hull is found or the beginning of the prediction horizon is reached.

The second approach is similar to the first approach, but differs in a few critical aspects. This approach is based on the iterative prediction horizon of the model predictive control framework and the modeled prediction uncertainty that grows along the time dimension in the prediction horizon, since predictions further into the future are less certain than more recent predictions (see Equation 3.8). The iterative behavior of model predictive control is exploited to improve computation times by computing the back-up constraints in parallel.

In the grid-based model predictive control framework, probability grids are computed for all time steps in the prediction horizon. This process is repeated every iteration, leading to many predictions for the same moment in time among the iterations. This concept is visualized in Figure 3.12. The proposed back-up approach utilizes these perviously made predictions by computing back-up hulls using the states and the probability grids of the previous open-loop solution, generated during the preceding prediction iteration. A schematic flowchart for this approach is given in 3.14. Since the predicted hulls along the prediction horizon model an increasing prediction uncertainty along the prediction horizon, the previously made prediction predictions will represent a more conservative representation of the environment, since the considered time step is further in the future of the respective prediction horizon. With these probability grids, the back-up hull is computed similarly to the back-up method using the current-state probability grids.

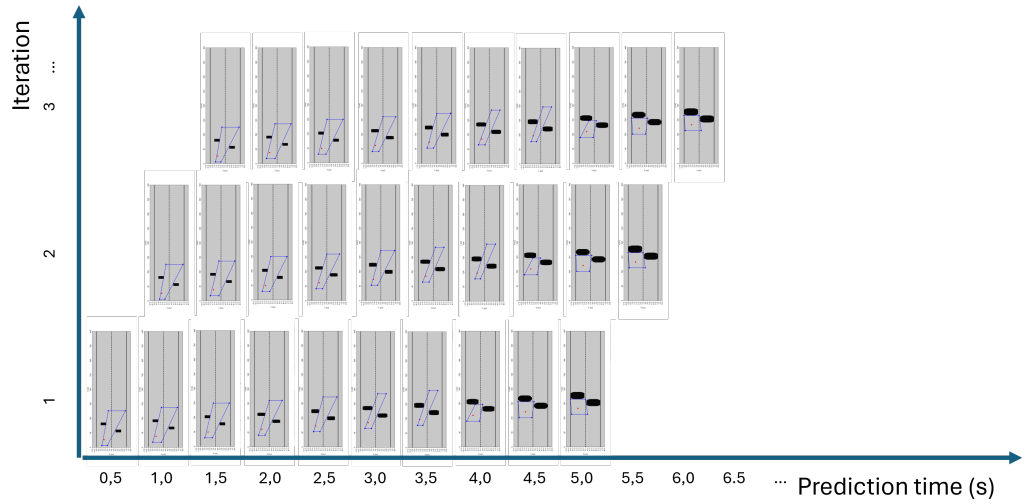


Figure 3.12: Visualization of overlapping predictions across model predictive control iterations. Prediction steps (displayed horizontally) over the prediction horizons at different iterations (displayed vertically). Since uncertainties grow over time, the previously made predictions for a certain point in time are generally more conservative.

A major advantage of using the prior open-loop solutions is that this allows the back-up hull computation to be executed in parallel to the optimization step. This ensures that the back-up hull is precomputed and ready for immediate use when needed. This parallel computation not only enhances computational efficiency but also allows for a seamless transition to the back-up trajectory in the event that the nominal hull fails, making this method particularly well-suited for real-time applications. To distinguish between the two proposed methods, this method will be referred to as the precomputed back-up method in the remainder of this thesis. The previously proposed method will be referred to as the current-state back-up method.

The two distinctive methods present a trade-off between uncertainty representation accuracy and real-time performance. The precomputed back-up method allows the back-up to be computed in parallel, which can improve computation times, critical for real-time performance. However, the method does not consider the most recent environment information. The current-state back-up method does consider the most recent environment information, which could be beneficial in highly dynamic scenarios. On the other hand, the method relies on the the sequential computation of the back-up after the nominal approach fails. This could be critical in time constraint scenarios. The chosen back-up strategy should depend on the scenario at hand, considering the priority for faster computation times or a more accurate representation of the environment.

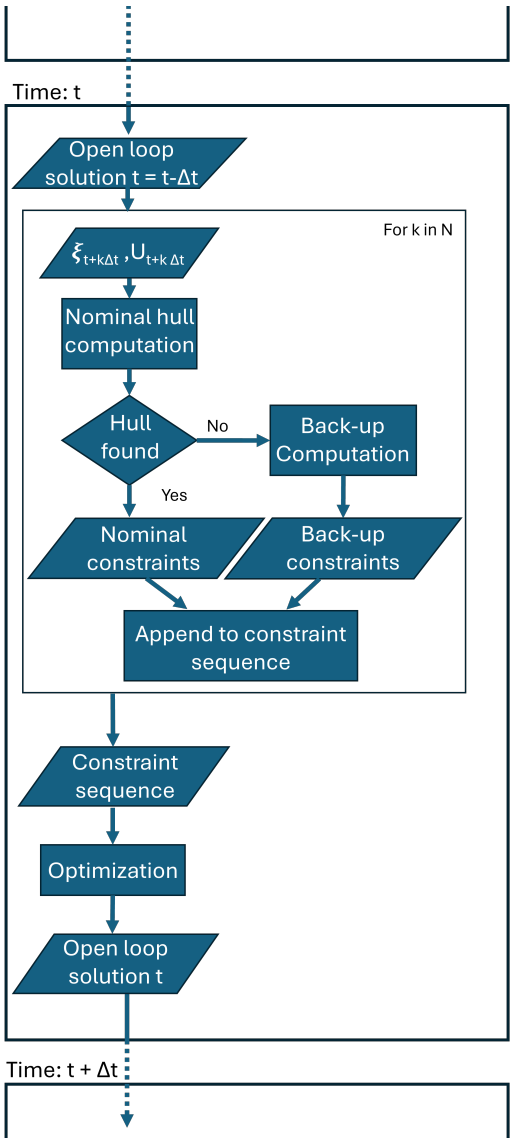


Figure 3.13:
Iteration flowchart of the current-state back-up method. Using the current open-loop solution, back-up hull constraints are computed (Figure 3.11) if no nominal hull is found. Either the nominal hull or the back-up hull is appended to the sequence of hull constraints, which is used to solve the optimization problem. After the first control input is applied to the system, this process is repeated. The open-loop state and input sequence are denoted by ($\xi_{t\rightarrow t+N\Delta t} = \{\xi_t, \xi_{t+\Delta t} ... \xi_{t+(N-1)\Delta t}, \xi_{t+N\Delta t}\}$, $\boldsymbol{u}_{t\rightarrow t+N\Delta t} = \{\boldsymbol{u}_t, \boldsymbol{u}_{t+\Delta t} ... \boldsymbol{u}_{t+(N-1)\Delta t}, \boldsymbol{u}_{t+N\Delta t}\}$)

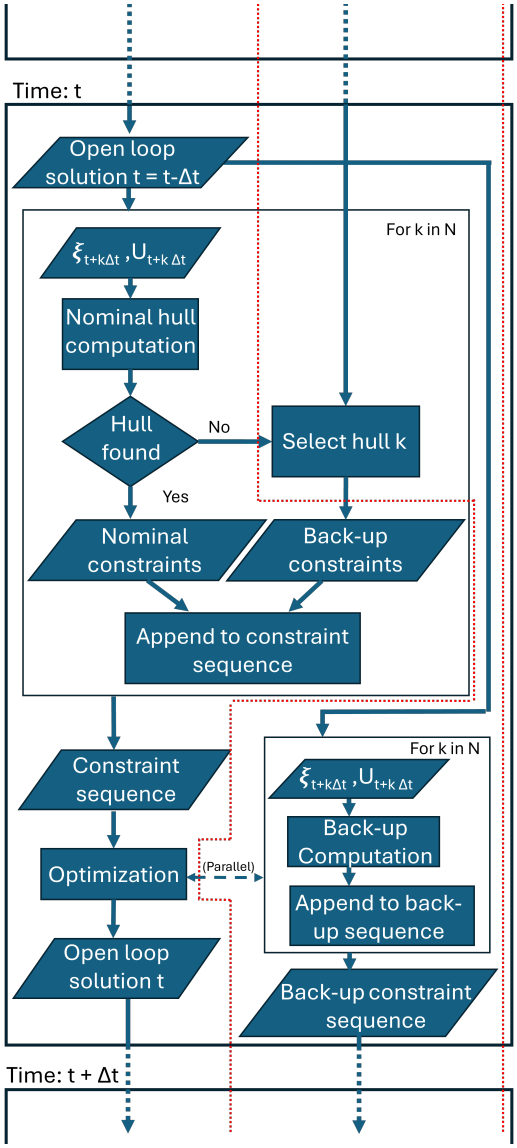


Figure 3.14:
Iteration flowchart of the precomputed back-up method. Using the open-loop solution of the previous iteration, nominal hull constraints are computed for every step k in the prediction horizon N. If no hull is found, a back-up hull for this particular time step is chosen from the precomputed back-up sequence, computed during the previous iteration. Either the nominal hull or the back-up hull is appended to the sequence of hull constraints, which is used to solve the optimization problem. In parallel to this optimization, the back-up hull constraint sequence for the full prediction horizon N of time step $t + \Delta t$ is computed (Figure 3.11). After this, the first control input is applied to the system. This process is repeated for the next time step utilizing this open-loop solution. The open-loop state and input sequence are denoted by ($\xi_{t\rightarrow t+N\Delta t} = \{\xi_t, \xi_{t+\Delta t} ... \xi_{t+(N-1)\Delta t}, \xi_{t+N\Delta t}\}$, $\boldsymbol{u}_{t\rightarrow t+N\Delta t} = \{\boldsymbol{u}_t, \boldsymbol{u}_{t+\Delta t} ... \boldsymbol{u}_{t+(N-1)\Delta t}, \boldsymbol{u}_{t+N\Delta t}\}$)

4

Simulation and Results

This chapter presents the simulations conducted to evaluate the performance of the proposed grid-based stochastic model predictive control framework. The first simulation evaluates the feasibility and control performance of the prediction model in a controlled environment and compares its performance against baseline models at various friction conditions. The second simulation evaluates the back-up model in a high tire-road friction environment, evaluating the effectiveness of a back-up method in tight scenarios. The final simulation is a combination of the prior simulations, testing the complete proposed method in tight scenarios at various friction levels.

The framework proposed in Chapter 3 is implemented in Python utilizing a sequential quadratic programming method with a QRQP solver from the CasADi package. All simulations have been conducted in a two-lane scenario with a lane width $l_{lane} = 3.5m$. Grid cell dimensions are $l_x = 0.5m$ and $l_y = 0.25m$. The vehicle model parameters are given in Table 4.1.

Parameter	Description	Unit	Value
l_f	Distance between CoG and front axle	m	1.1
l_r	Distance between CoG and rear axle	m	1.57
m	Mass	kg	1600
I_{zz}	Moment of inertia in vertical direction	$\mathrm{kg} \cdot \mathrm{m}^2$	2100
h_{CoG}	Height of center of gravity above ground	m	0.3
$C_{\alpha f}$	Front axle cornering stiffness	kN/rad	114
$C_{\alpha r}$	Rear axle cornering stiffness	kN/rad	94
g	Gravitational constant	$\mathrm{m/s}^2$	9.81
$a_{x,\mathrm{UB}/LB}$	Upper and lower bounds on longitudinal acceleration	$\mathrm{m/s}^2$	± 3
$\delta_{\mathrm{UB}/LB}$	Upper and lower bounds on steering angle	rad	$\pm\frac{\pi}{10}$
$\dot{a}_{x,\mathrm{UB}/LB}$	Upper and lower bounds on longitudinal jerk	$\mathrm{m/s}^3$	± 0.5
$\dot{\delta}_{\mathrm{UB/LB}}$	Upper and lower bounds on steering rate	rad/s	$\pm\frac{\pi}{4}$

Table 4.1: Vehicle model parameters (based on [9])

Target vehicles are modeled as described in 3.1.5, where

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0.5(\Delta t)^2 & 0 \\ \Delta t & 0 \\ 0 & 0.5(\Delta t)^2 \\ 0 & \Delta t \end{bmatrix} \tag{4.1}$$

$[k_{12}, k_{21}, k_{22}] = [-1, -0.8, -2.2]$. We assume Gaussian noise $\boldsymbol{w}_k^{\mathrm{TV}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$ in (2.3)) with covariance matrix $\boldsymbol{\Sigma}_w = \mathrm{diag}(1, 1, 1, 1)$. These target vehicle parameters are similar to [13] and [42]. The

vehicles have a length of $L = l_f + l_r = 2.67m$ and a width of $W = 2m$. The physical dimensions of all simulated vehicles are equal.

4.1. Model Complexity

The performance of a model predictive controller is highly dependent on the accuracy of its prediction models. This section presents a comparative analysis of four prediction models evaluated under varying friction coefficients, reflecting various road conditions. The aim of this comparison is to identify the strengths and limitations of the various model complexities. The equations of motion describing the compared models are given in Appendix A.

Utilizing a non-linear prediction model, enables the modeling of more complex scenarios. This is particularly useful in low-friction conditions, where friction coefficients need to be modeled. Four models were compared in a simulation to demonstrate the effectiveness of the proposed method, namely a point-mass model, a linear-tire bicycle model (both used by [13]) and the proposed bicycle model using a non-linear Fiala tire model (see Section 3.1). The linear tire model is included in this comparison to understand wether differences in results are stemming from the dynamic bicycle model or the tire model.

The simulation is modeled with a plant model utilizing a dynamical bicycle model with a Fiala tire model (described in Section 3.1.2 and Section 3.1.3 respectively) at a sampling time of 1ms. Since this model requires longitudinal acceleration and steering angle as inputs, and the point-mass model computes longitudinal acceleration and lateral acceleration, the desired lateral acceleration is converted to a steering angle according to the following relationships

$$a_y = \frac{{v_x}^2}{R} \tag{4.2}$$

$$\delta = \tan\left(\frac{l_r + l_f}{R}\right) \approx \frac{l_r + l_f}{R} = \frac{a_y(l_r + l_f)}{{v_x}^2} \tag{4.3}$$

4.1.1. Scenario

To compare the performance of the non-linear model the following scenario is considered. The ego vehicle is driving at a velocity of $14m/s$ in lane 1. A target vehicle is driving a velocity of $12m/s$, 15 meter in front of the ego vehicle in lane 1. Both vehicles have a reference velocity equal to their initial velocity. The ego vehicle will approach the target vehicle until the vehicle is close and the ego vehicle reference trajectory changes to the second lane following the reference trajectory generated as described in Section 3.1.8. The obstacle states are assumed to be deterministic ($\boldsymbol{G} = 0$ and $\Sigma_0 = 0$ in (3.8) and (3.9)), since only the adaptability of the dynamic model is of interest in this simulation. A simulation including both low-friction conditions and uncertainty is discussed later in Section 4.3.

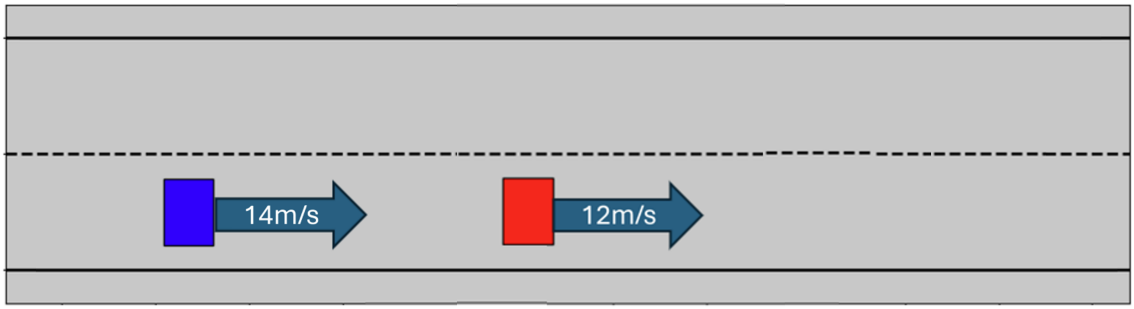


Figure 4.1: Scenario unobstructed overtaking

4.1.2. Results

In this section, the results of the conducted simulations are presented and the traversed trajectories and the states during the simulation are compared. Note that the point-mass model and the kinematic bicycle model exhibit staggered transitions between time steps, in contrast to the dynamic bicycle models. This effect, known as zero-order hold effect, is caused by the difference in control frequency and frequency of the the plant dynamics. Inputs are assumed constant between control time steps, causing step-like updates every control iteration. This effect is not present in the dynamic bicycle models, because the computed inputs are derivatives of the longitudinal acceleration and steering input, resulting in a smoother transition between states.

Friction coefficient 0.9

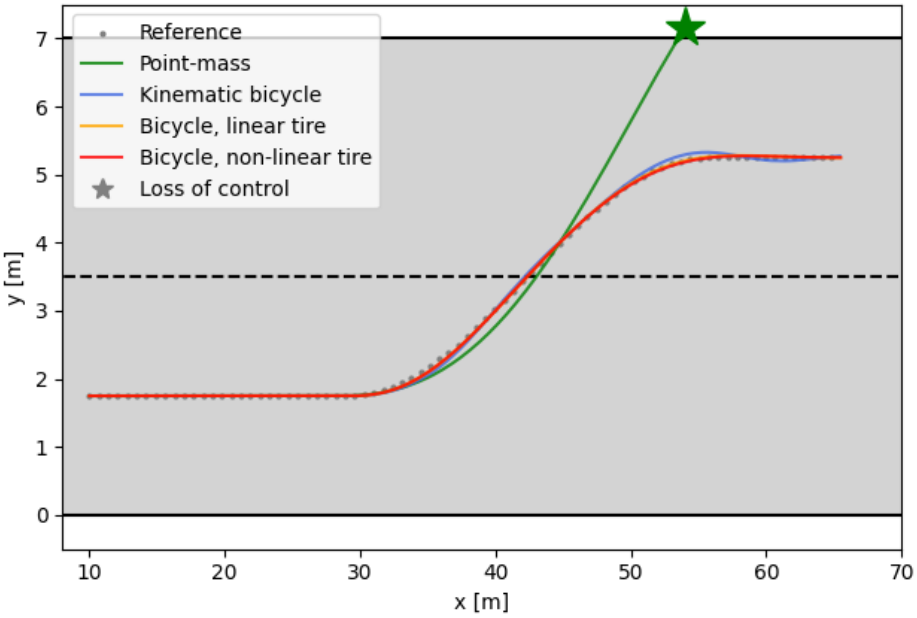
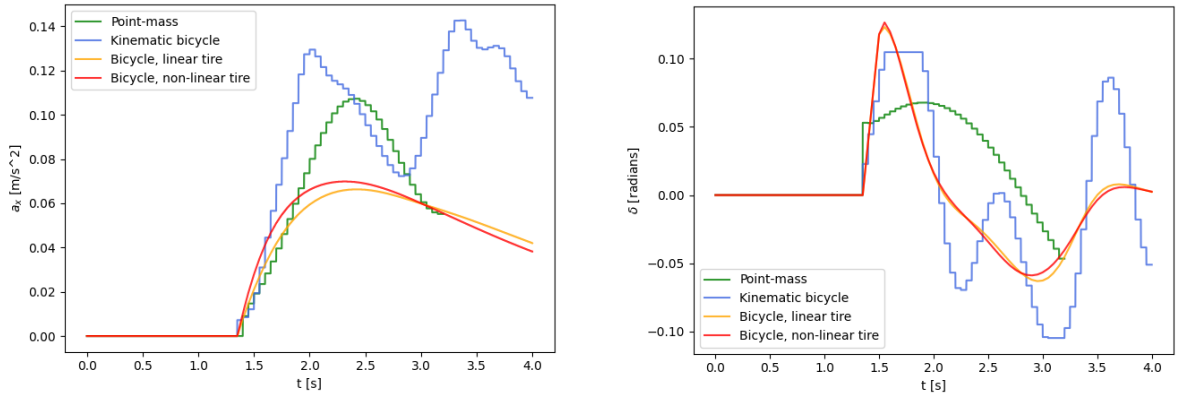


Figure 4.2: Vehicle trajectories at friction coefficient 0.9

At a friction coefficient of 0.9 it is observed that the new models, both with a linear tire- and a non-linear tire-model, are able to follow the reference trajectory without any difficulties (see Figure 4.2 and Table 4.2). The kinematic bicycle model is able to follow the reference trajectory with some overshoot and the point-mass model loses control when steering back (see Figure 4.7).



(a) Longitudinal acceleration (m/s^2) over time (s)

(b) Steering angle (radians) over time (s)

Figure 4.3: Control inputs at friction coefficient 0.9

Figure 4.3 shows the control inputs. The results demonstrate small positive longitudinal accelerations during the lane-change maneuver, which coincide with the reduction in x-velocity (Figure 4.4) as the vehicle begins to turn. Significant differences in trajectory are mainly due to different steering angle inputs. The point-mass model results show a lower, longer peak in steering angle, before steering back compared to the other models. The steering inputs of the kinematic bicycle demonstrate oscillating behavior in line with the observed oscillating trajectory around the reference. The dynamic bicycle model, both with a linear and non-linear tire model, exhibit a smooth sequence of steering inputs without large deviations or oscillations, resulting in a smooth trajectory and low lateral velocities (Figure 4.5).

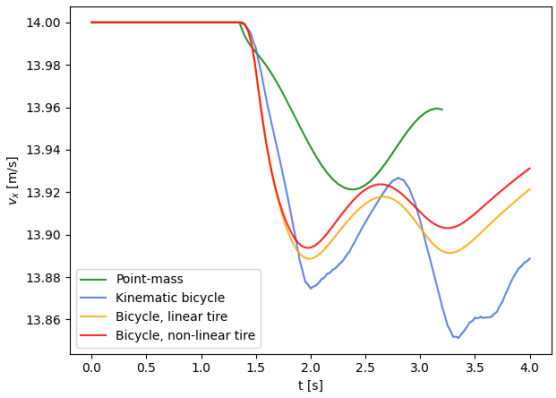


Figure 4.4: x velocity (m/s) over time (s) at friction coefficient 0.9

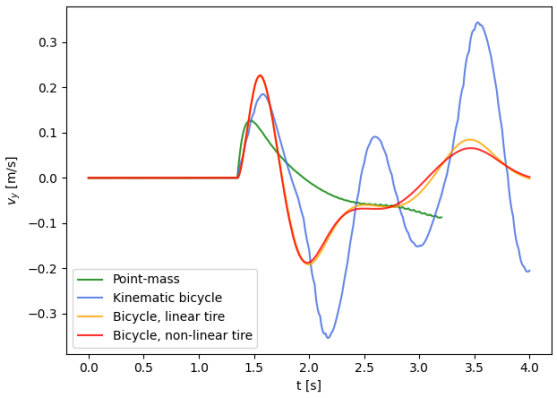


Figure 4.5: y velocity (m/s) over time (s) at friction coefficient 0.9

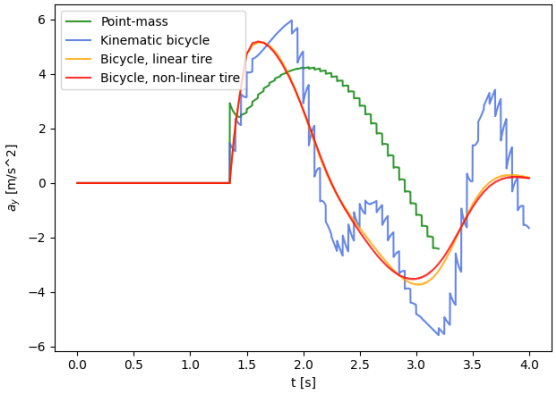


Figure 4.6: Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.9

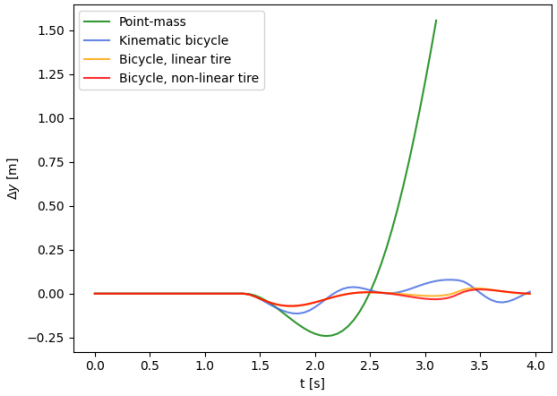


Figure 4.7: Tracking Error (m) over time (s) at friction coefficient 0.9

Model	**RMSE (m)**
Point-Mass	Loss of control
Kinematic Bicycle	0.0441
Dynamic Bicycle, Linear Tire	0.0252
Dynamic Bicycle, Non-Linear Tire	0.0245

Table 4.2: Root Mean Square Error (m) at friction coefficient 0.9

Friction coefficient 0.45

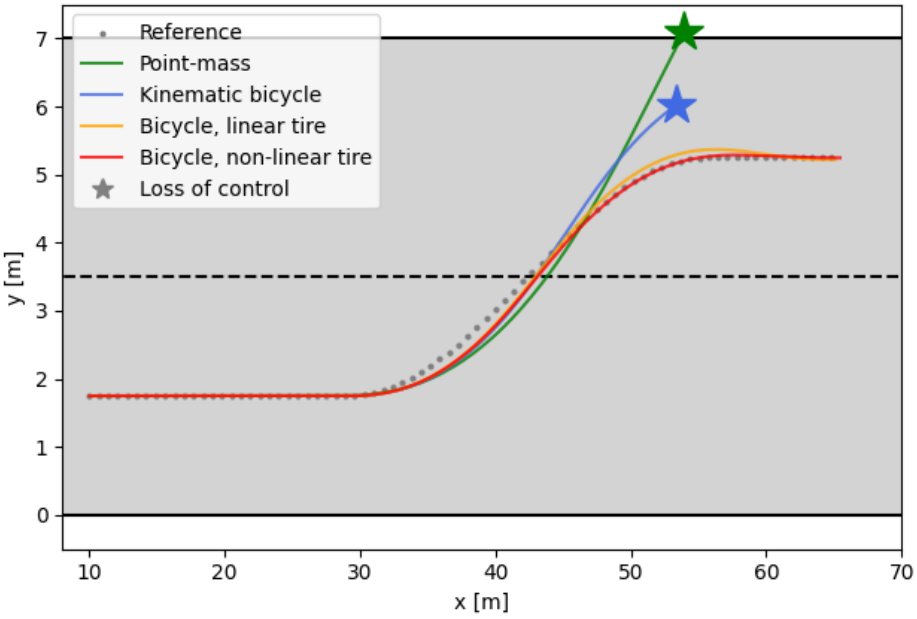
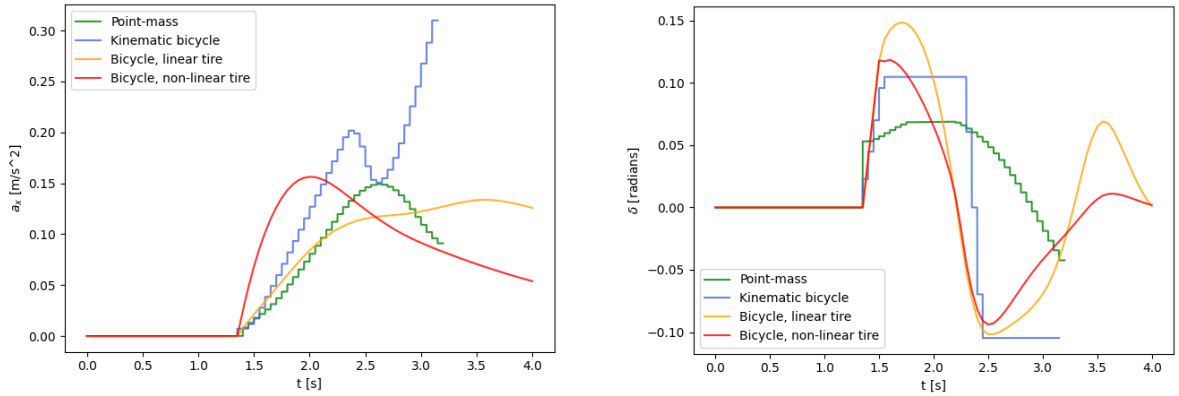


Figure 4.8: Vehicle trajectories at friction coefficient 0.45

Considering Figure 4.8 it is observed that at this reduced friction coefficient, both the point-mass model and the kinematic bicycle model experience loss of control. It is also observed that slight differences in the successfully controlled trajectories by the dynamic bicycle models occur at this friction coefficient, where the linear tire model slightly overshoots the reference trajectory and the non-linear tire model is better able to follow the reference trajectory at the final stages of the lane change maneuver (see Figure 4.13 and Table 4.3).



(a) Longitudinal acceleration (m/s^2) over time (s)

(b) Steering angle (radians) over time (s)

Figure 4.9: Control inputs at friction coefficient 0.45

The control input at a friction coefficient of 0.35 (Figure 4.9) display similar behavior for the point-mass model before loss of control. The kinematic model shows a longer peak in steering angle once more, followed by steep counter steering coinciding with a peak in lateral acceleration magnitude (Figure 4.12) in the second half of the lane change maneuver before loss of control. An increase in longitudinal acceleration is also observed (Figure 4.9a) , coinciding with the reduction in longitudinal velocity (Figure 4.10). It can be seen that with the long peak in steering angle also comes a larger peak in (negative) lateral velocity (Figure 4.11).

It is observed that the dynamic bicycle models are able to follow the reference trajectory. However, the dynamic bicycle model with a linear tire model slightly overshoots the reference trajectory in the second half of the lane change, whereas the non-linear tire model more accurately follows the desired trajectory. In the control inputs, the non-linear tire model exhibits early longitudinal acceleration that declines over time, whereas the linear tire model has a more constant longitudinal acceleration (see Figure 4.9a). For the steering angle input, both models show similar overall shapes, but the non-linear tire model produces less pronounced peaks (see Figure 4.9b), which is also observed in the lateral acceleration profiles (see Figure 4.12).

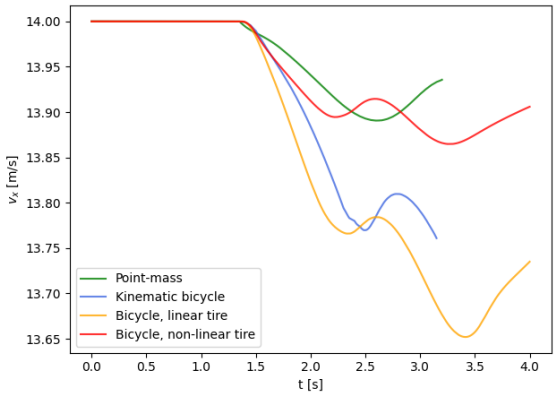


Figure 4.10: x velocity (m/s) over time (s) at friction coefficient 0.45

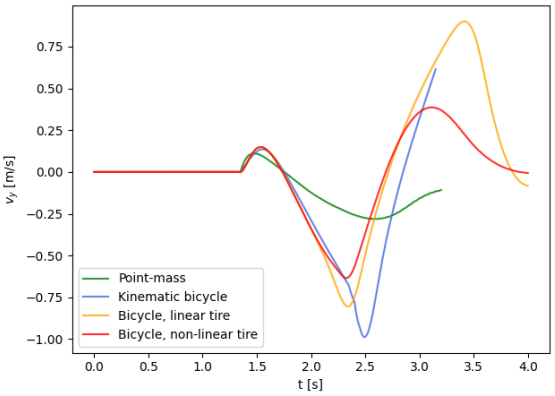


Figure 4.11: y velocity (m/s) over time (s) at friction coefficient 0.45

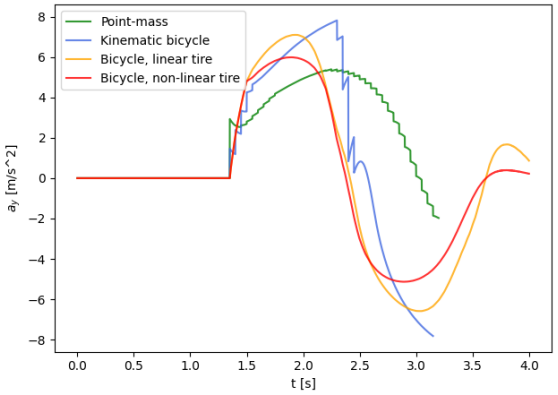


Figure 4.12: Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.45

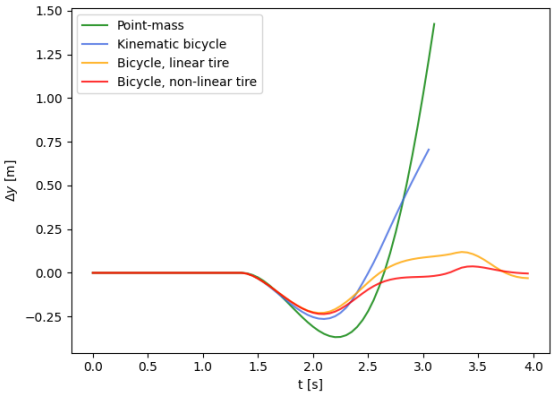


Figure 4.13: Tracking Error (m) over time (s) at friction coefficient 0.45

Model	**RMSE (m)**
Point-Mass	Loss of control
Kinematic Bicycle	Loss of control
Dynamic Bicycle, Linear Tire	0.0945
Dynamic Bicycle, Non-Linear Tire	0.086

Table 4.3: Root Mean Square Error (m) at friction coefficient 0.45

Friction coefficient 0.35

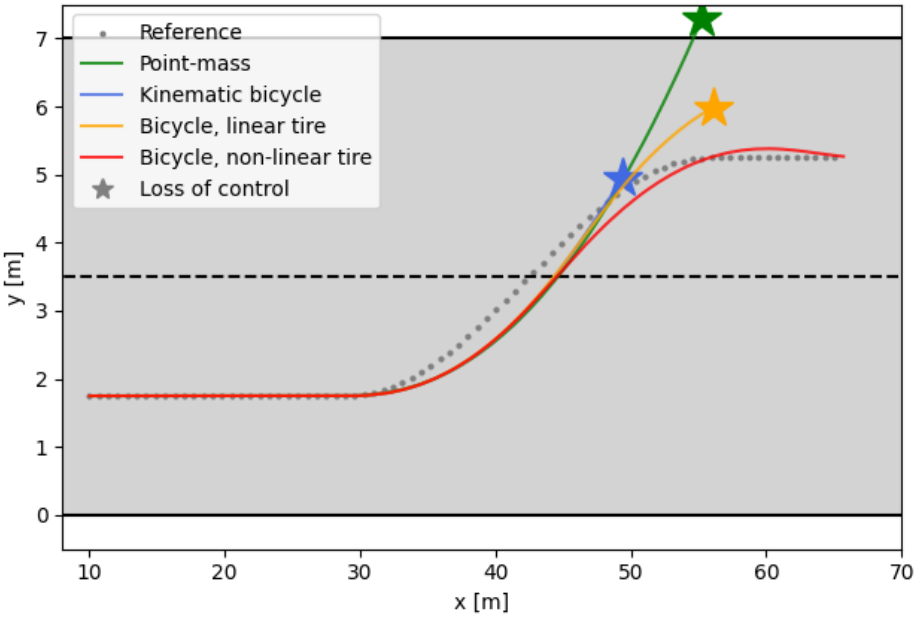
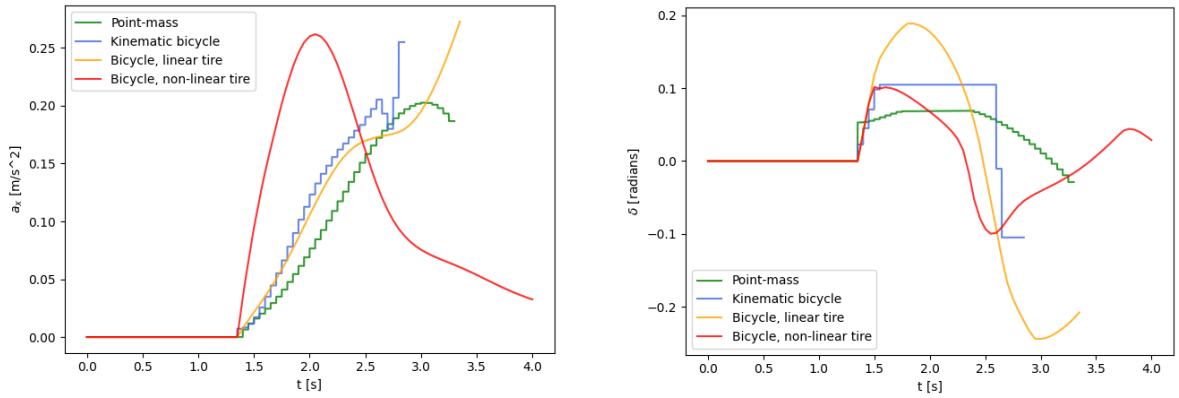


Figure 4.14: Vehicle trajectories at friction coefficient 0.35

At a friction of 0.35, it is observed that the dynamic bicycle model with a Fiala tire model is the only model still able to stay in control (see Figure 4.14). Similarly to the other models, the followed trajectory is experiencing some tracking lag, however, the bicycle model with a Fiala tire model is able to stabilize the followed trajectory with a slight overshoot in the second half of the lane change maneuver, where the other methods lose control.



(a) Longitudinal acceleration (m/s^2) over time (s)

(b) Steering angle (radians) over time (s)

Figure 4.15: Control inputs at friction coefficient 0.35

Comparing the control inputs (Figure 4.15), we observe that the point-mass model and the kinematic bicycle model display similar behavior of that observed at a friction coefficient of 0.45 before losing control. Comparing the dynamic bicycle models, we observe significantly bigger steering angle magnitudes for the linear tire model, compared to the Fiala model. Furthermore, the Fiala tire model steers back earlier than the linear tire model (see Figure 4.15b and Figure 4.17). Similarly to the results of friction coefficient 0.45, the Fiala tire model exhibits an early longitudinal acceleration before a decline, whereas the linear model increases the longitudinal acceleration (Figure 4.15a), coinciding with the

longitudinal velocity drop (Figure 4.16). In lateral acceleration the linear tire model reaches high values, whereas the Fiala model resulted in lower maximum lateral accelerations.

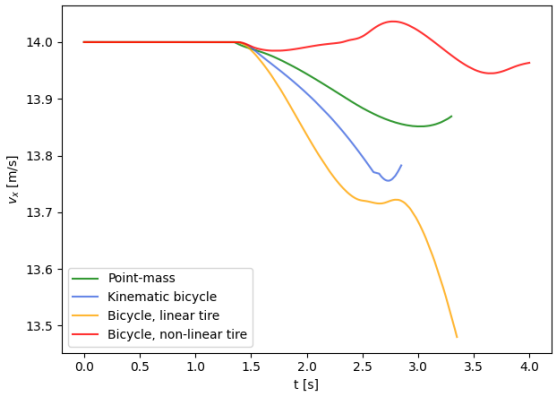


Figure 4.16: x velocity (m/s) over time (s) at friction coefficient 0.35

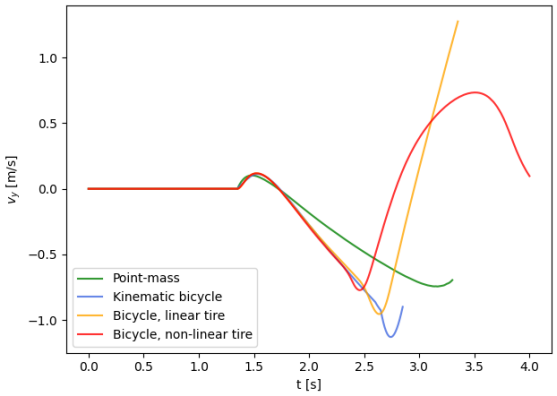


Figure 4.17: y velocity (m/s) over time (s) at friction coefficient 0.35

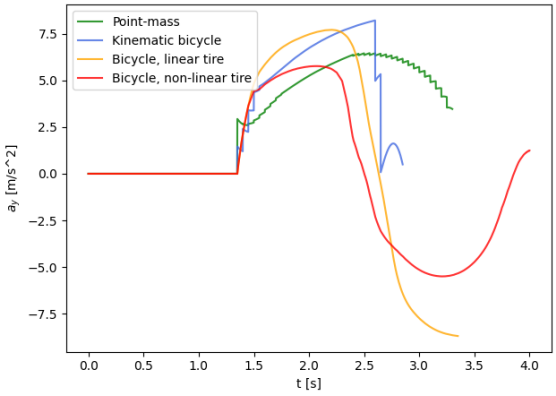


Figure 4.18: Lateral acceleration (m/s^2) over time (s) at friction coefficient 0.35

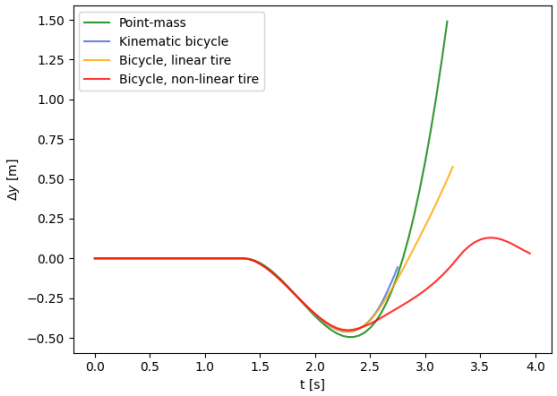


Figure 4.19: Tracking Error (m) over time (s) at friction coefficient 0.35

Model	RMSE (m)
Point-Mass	Loss of control
Kinematic Bicycle	Loss of control
Dynamic Bicycle, Linear Tire	Loss of control
Dynamic Bicycle, Non-Linear Tire	0.207

Table 4.4: Root Mean Square Error (m) at friction coefficient 0.35

4.2. Back-Up Method

A critical limitation in the baseline grid-based stochastic model predictive control framework is the unsafe handling of failed constraint formulation (as discussed in Section 2.3.1). To address this issue, a novel back-up method is developed. This simulation is set-up to result in many failed nominal constraint reformulations, to evaluate the performance of the proposed back-up strategies against the baseline method proposed in [13], comparing their safety and feasibility rates.

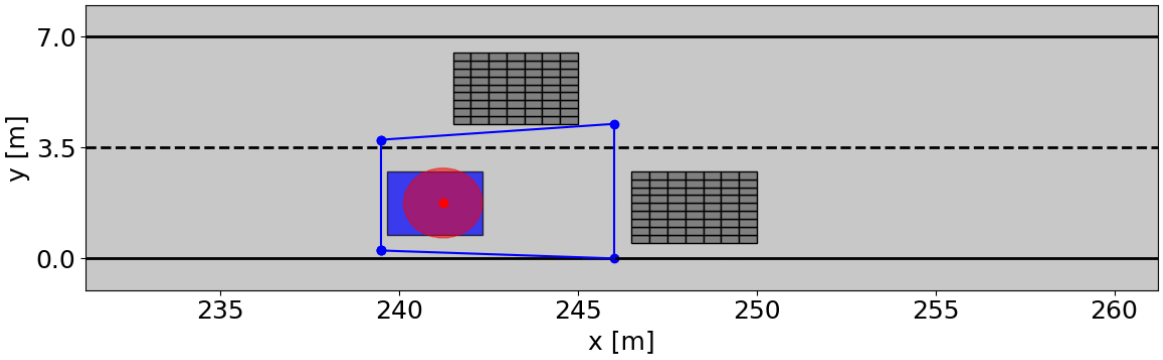


Figure 4.20: Narrow and shallow hull when target vehicles are close

To make a valid comparison between the proposed method and the baseline, it is important to conduct the simulation in a valid simulation environment. Considering Section 4.1, it can be concluded that the point-mass model does not work properly when considering more realistic dynamics including forces and possible friction limitations. It is also not correct to compare the approaches in the newly proposed model with a non-linear tire model, since the baseline back-up method was not designed for these dynamics. Therefore, the comparison will be made between the new methods and the baseline method using the plant dynamics and control dynamics of the baseline ([13]) approach. This allows for a fair comparison and allows for a proper discussion. The complete proposed method including the proposed control model is evaluated in Section 4.3 and interpreted in its corresponding discussion in Section 5.1.3.

Simulations are run at sampling time 0.2s, The SMPC minimizes the cost function

$$\min_{\boldsymbol{u}} \sum_{k=0}^{N-1} \left(\left\| \Delta \xi_k^{\mathrm{EV}} \right\|_Q^2 + \left\| \boldsymbol{u}_k^{\mathrm{EV}} \right\|_R^2 \right) \tag{4.4}$$

where $\xi_{k+1}^{\mathrm{EV}} = f\left(\xi_k^{\mathrm{EV}}, \boldsymbol{u}_k^{\mathrm{EV}}\right)$ is described by the kinematic bicycle ego vehicle model

$$\begin{aligned} \dot{x} &= v\cos(\psi + \alpha) \\ \dot{y} &= v\sin(\psi + \alpha) \\ \dot{\psi} &= \frac{v}{l_{\mathrm{r}}}\sin(\alpha) \\ \dot{v} &= a \\ \alpha &= \arctan\left(\frac{l_{\mathrm{r}}}{l_{\mathrm{r}} + l_{\mathrm{f}}}\tan(\delta)\right) \end{aligned} \tag{4.5}$$

Where α is the body slip angle. Prediction horizon with $N = 20$, weighting matrices $\boldsymbol{Q} = \mathrm{diag}(0, 2, 0.5, 0.1)$ and $\boldsymbol{R} = \mathrm{diag}(0.1, 1)$, $\Delta\xi_k^{\mathrm{EV}} = \xi_k^{\mathrm{EV}} - \xi_{k,\mathrm{ref}}^{\mathrm{EV}}$ where reference trajectory $\xi_{k,\mathrm{ref}}^{\mathrm{EV}}$ is determined similar to Section 3.1.8, however, unlike in Section 3.1.8, the y reference is updated to the new lane when a new reference lane is determined, in contrast to the smooth transition described in (3.12). This reference trajectory generation is similar to [13]. Disturbance matrix $G = \mathrm{diag}(0.05, 0.067, 0.013, 0.03)$ describes the prediction uncertainty, and initial perception uncertainty $\Sigma_0 = \boldsymbol{G}I\boldsymbol{G}^\top$, where I is the identity matrix, similar to [13] and [42]. Probability threshold $\beta = 0.98$ and the back-up threshold $\beta_{bu} = 0.99$ for the proposed back-up method.

As discussed before (see Section 3.3.2), there are two proposed back-up methodologies. The first being a back-up hull precomputed with the open-loop of the previous control iteration, such that it could be computed in parallel during the optimization step, allowing for rapid back-up hull computations. The second method utilizes the current uncertainty predictions and ego vehicle states, allowing for more up-to-date predictions, but not allowing for parallel computation during the optimization step. These

two methods are compared against the baseline method, where the differences between the new models and the baseline model are as follows:

- The new models consider the ego vehicle dimensions by defining a minimum distance around the vehicle's CoG as described in Section 3.1.6, whereas the baseline considers a single point at the CoG of the vehicle.
- The new models consider a hull to be valid if the width of the hull is at least $3m$ wide, whereas the the baseline model considers hulls to be valid if the lane width is at least $2m$. This difference is the result of the ego vehicle dimensions being accounted for in the constraint formulation (see Section 3.2.3), requiring more space inside the hull than a single point.
- The new models compute a convex hull described in Algorithm 1 which considers all four corners as a basis for the hull generation, guarantying the entire vehicle to be inside the hull,whereas the baseline explores free lines only from the back two corners of the ego vehicle.
- The new models are based on the back-up methods and soft constraints described in Section 3.3 and Section 3.2.4 respectively, whereas the baseline method considers the hull of the previous prediction time step $(k-1)$ when no convex hull can be found.

4.2.1. Scenario

In the simulation, the ego vehicle is approaching two target vehicles from behind (see Figure 4.21). The ego vehicle has a reference velocity of $30m/s$ and the target vehicles have a reference velocity of $26m/s$. The ego vehicle is positioned on the bottom lane. The first target vehicle is positioned on the top lane 30 meters in front of the ego vehicle. The second target vehicle is positioned on the bottom lane at a varying distance in front of the first target vehicle as a simulation variable, this gap will remain constant during the simulation, since the target vehicles have the same velocity. By varying the distance between the target vehicle across simulations, the behavior can be evaluated of scenarios where there is not enough space to overtake; where there is enough space to overtake; and tight scenarios in between.

This scenario is chosen since it will lead to shallow and narrow hulls, because the gaps between target vehicles are small and the space between the two target vehicles is not wide enough for the ego vehicle to pass through (see Figure 4.20). As a result the vehicle is limited to a small convex area, which hinders finding a feasible trajectory in the positive driving direction. Since the reference velocity of the ego vehicle is higher than the velocity of the target vehicles, this results in the ego vehicle driving towards the front of the hull, striving to speed up, resulting in the vehicle driving close to the hull boundaries. As discussed in Section 3.3.1, this results in failure of the nominal hull finding method. When the gap size is increased and the gap is wide enough to change reference lane, the vehicle can get into tight scenarios, which is also interesting to analyze. It is crucial to have well functioning back-up method for both of these scenarios. The simulation is conducted with gaps between 1 and 16 meters, since is has been observed that all models are able find a valid hull sequence with enough space to overtake at gaps greater than 16 meters.

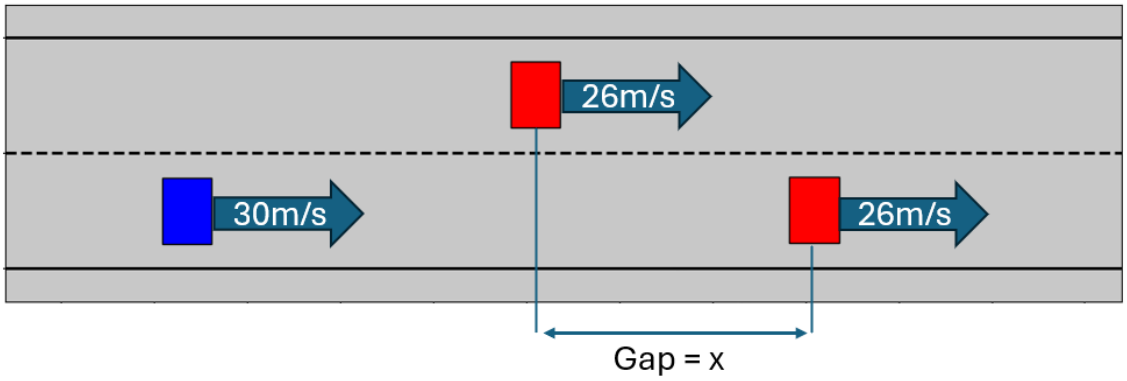


Figure 4.21: Scenario varying gaps

4.2.2. Results

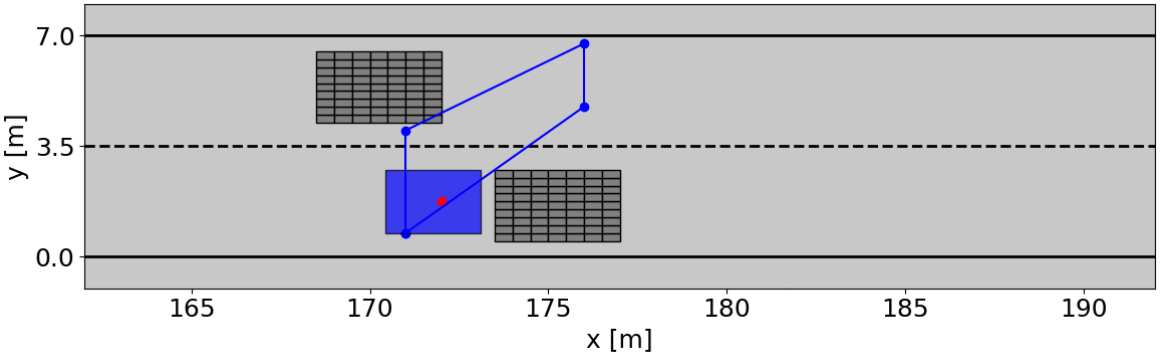
The feasibility rate for the baseline method is 43.8%. The new current-state back-up method and the new precomputed back-up method achieve 75.0% and 62.5% respectively (see Table 4.5).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Feasible
Baseline	✓	✓	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓	✓	43.8%
Current-state	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	✓	75.0%
Precomputed	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×	✓	✓	✓	✓	62.5%

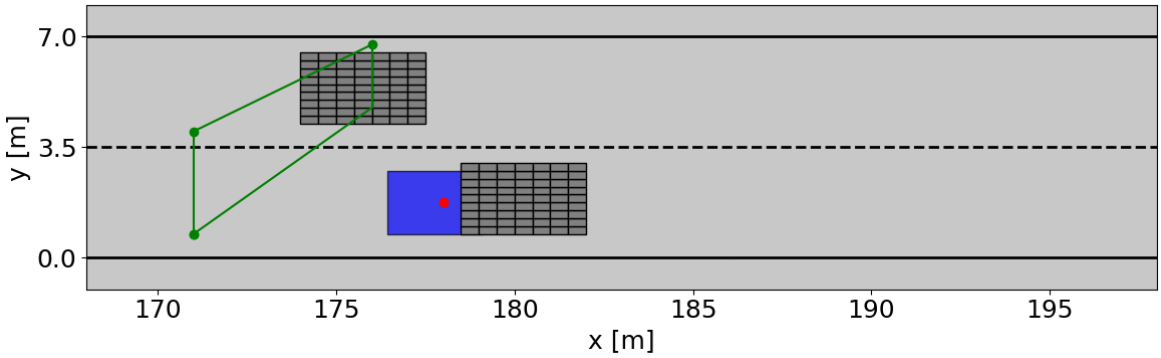
Table 4.5: Comparison of back-up hull method feasibility

Note: Before discussing the simulation results, it is important to clarify how to interpret the provided visualizations. As explained in Section 3.2, hulls describing the constraints are computed at every time step in the prediction horizon based on the open-loop states. These hulls then serve as constraints for the subsequent optimization step. Consequently, the hulls computed at one iteration directly influence the open-loop solution of the next MPC iteration.
In the figure captions, “time step” refers to a specific moment in time, while “iteration” indicates a control iteration. Each subfigure illustrates an open-loop state and the hull computed at that state for the next optimization step. For instance, in Figure 4.22, subfigure 4.22a shows the predicted hull for time step t at iteration i. Subfigure 4.22b shows the following hull for time step $t + \Delta t$ in the same prediction horizon (iteration i). Because no nominal hull is found at this new state, the baseline method falls back on the hull computed for t (i.e. subfigure 4.22a). Subfigure 4.22c then shows the open-loop solution at the next MPC iteration ($i + 1$), corresponding to the same time step ($t + \Delta t$) shown in subfigure 4.22b. This open-loop solution uses the hulls from iteration i (subfigures 4.22a an 4.22b). As a result, the ego vehicle in subfigure 4.22c remains within the hull computed in subfigure 4.22b.

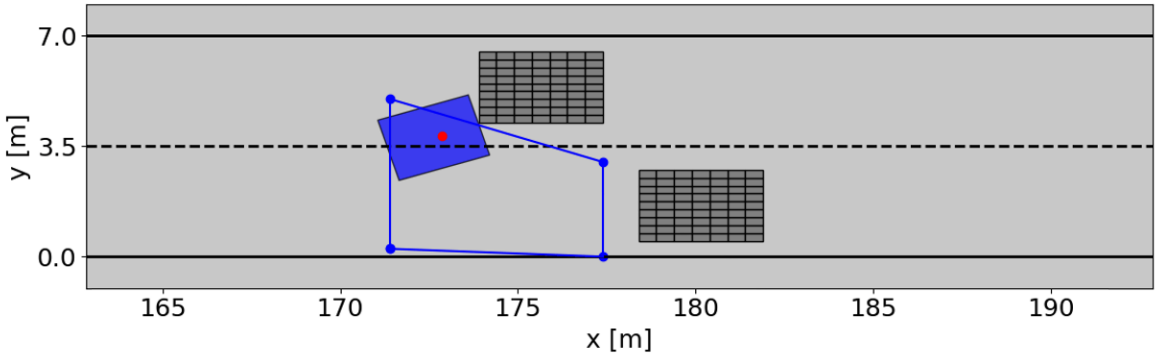
Evaluating the results, it is observed that at gaps narrower than 3 meters, the target vehicles are effectively driving side by side. The ego vehicle reduces its velocity and stays behind the target vehicles for all methods in this scenario. At gaps from 3 meter and wider, the hulls generated by the baseline method have a tendency to "peek" into the other lane (see Figure 4.22a). Figure 4.22 shows a sequence of prediction steps, shortly before infeasibility, where Figure 4.22a and Figure 4.22b represent two consecutive prediction steps. In Figure 4.22b, the ego vehicle gets too close to the target vehicle in front of the vehicle to generate a hull, so the method falls back on the hull of the prediction step before (Figure 4.22b). Figure 4.22b shows the resulting open loop solution of the prediction of the next iteration. The vehicle moves into the target vehicle in order to satisfy the back-up hull of Figure 4.22b as a result of the back-up hull computed at that iteration.



(a) Prediction of time step t at iteration i. Ego vehicle is approaching the target vehicles, hull exploration in other lane.



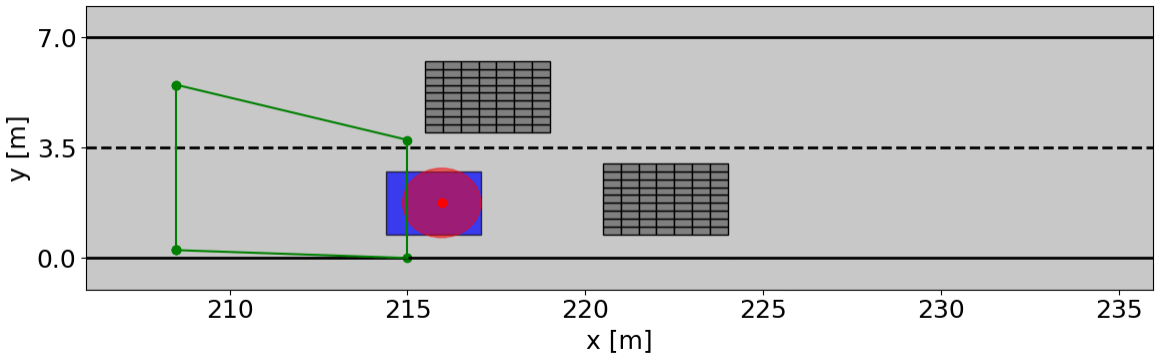
(b) Prediction of time step $t + \Delta t$ at iteration i. Ego vehicle is too close to target vehicle and no valid hull is found. As a back-up the previous prediction hull (of time step t) is used as hull, this hull overlaps with occupied cells.



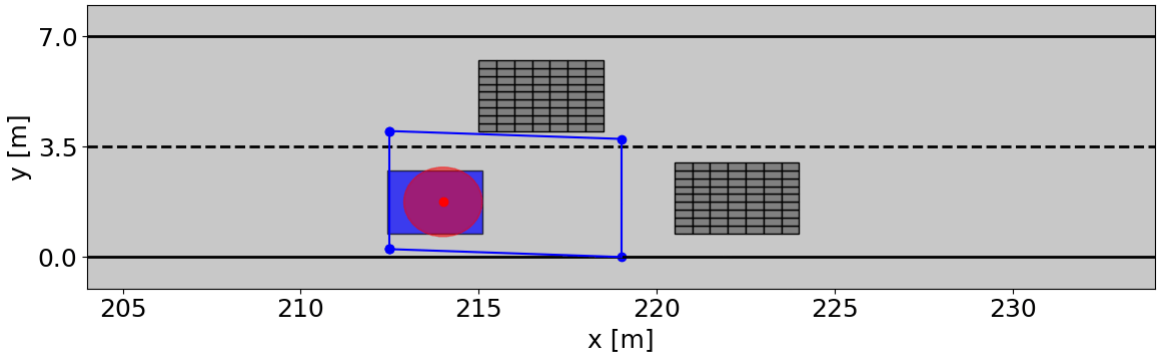
(c) Prediction of time step $t + \Delta t$ at iteration $i + 1$. Prediction state sequence based on the result of the optimization done with hulls found at iteration i. Ego vehicle moves over into the the target vehicle to adhere to previously determined back-up hull of Figure 4.22b

Figure 4.22: Sequence of open-loop solutions in the baseline method shortly before infeasibility. Baseline method computes unsafe back-up hulls that include occupied space.

For gaps smaller than 7 meter, the hull computed by the new methods remain in their own lane (see Figure 4.23). The back-up method is only necessary when the ego vehicle gets too close to the target vehicle in front and no nominal hull can be found. The back-up method generates a hull close to the actual ego vehicle states, which leads to a slight reduction in longitudinal acceleration, keeping the vehicle stable behind the target vehicles.



(a) Prediction of time step t at iteration i



(b) Prediction of time step t at iteration $i + 1$

Figure 4.23: Open-loop prediction of two consecutive iterations of the same time step. When the vehicle gets close to the obstacles, a safe back-up hull is computed (a), in the following optimization the longitudinal input is reduced and such that the vehicle is at a safe distance in the next iteration (b), adhering to the hull computed in (a).

When gaps are wider than 7 meters, the new methods display similar behavior to that of the baseline method at more narrow gaps, where the computed hulls reach into the adjacent lane (see Figure 4.24). Similarly to the baseline at narrower gaps, the trajectory computed by the new methods tend to move over to the other lane in the prediction horizon. When the new methods get into this tight space, they fall back on their respective back-up methods. These back-up hulls are in the original lane. The optimization becomes infeasible.

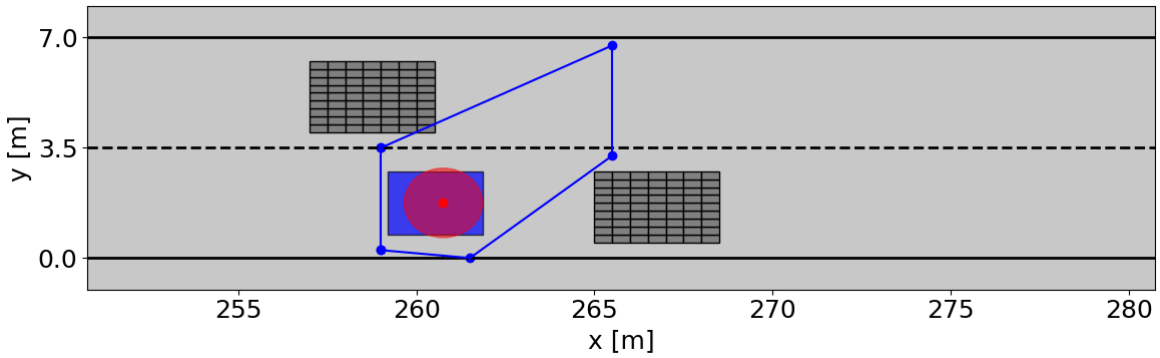
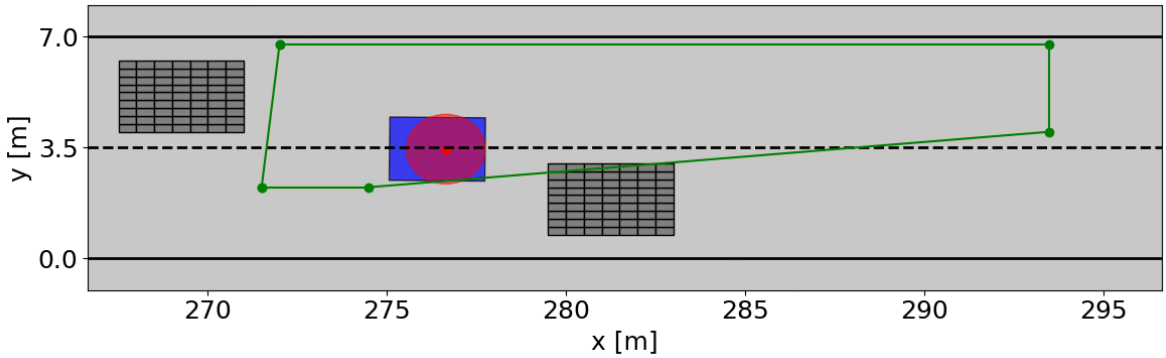


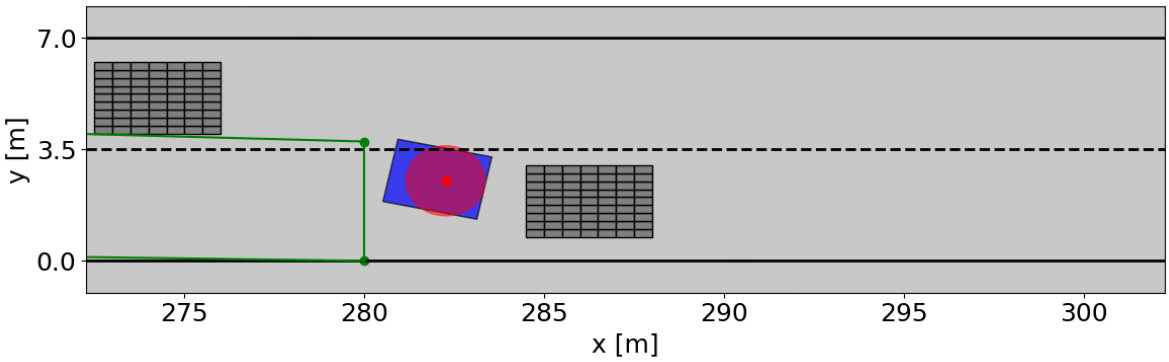
Figure 4.24: Peeking behavior in the proposed back-up methods

At gaps of 11 meters and wider the ego vehicle is able to overtake in some scenarios. However, the spaces between the target vehicles remain tight. At gaps of 11 and 12 meters a difference is observed

between the precomputed method and the current-state method. The precomputed method displays oscillating behavior and erratic steering, increasing over time, ultimately leading to infeasibility. It is observed that there are relatively large distances between consecutive hulls (See Figure 4.25).



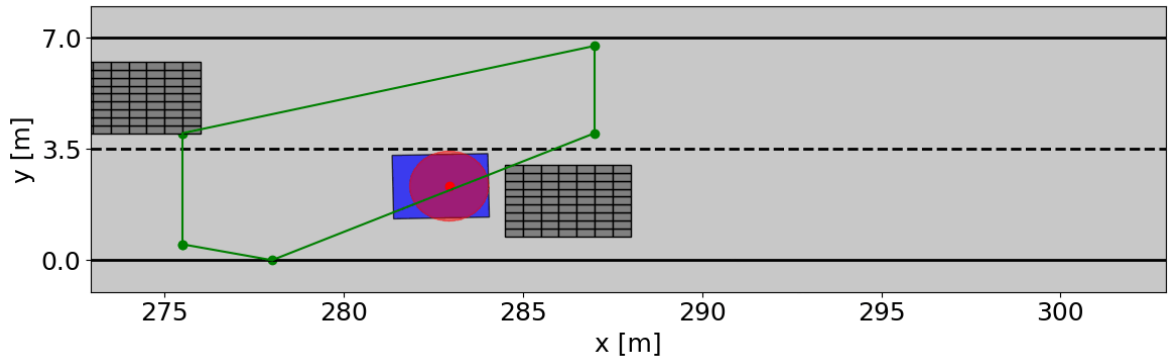
(a) Prediction hull of time step t at iteration i



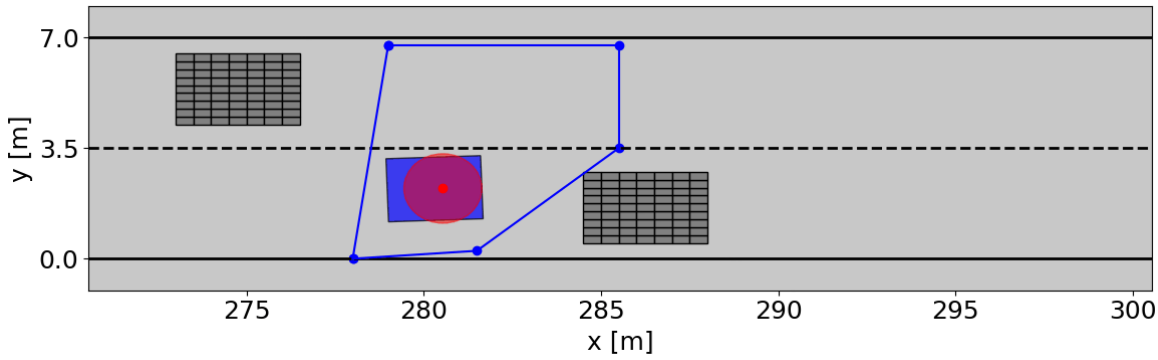
(b) Consecutive prediction hull of time step $t + \Delta t$ at iteration i. Distance between consecutive hulls is large.

Figure 4.25: Two consecutive hulls in the same prediction horizon. Large distance between consecutive hulls.

The current-state back-up method shows less oscillation and consecutive hulls are closer to each other (See Figure 4.26). Due to the improved continuity between states, the method is able to compute a feasible trajectory.



(a) Prediction hull of time step t at iteration i, computed with the current-state back-up method



(b) Prediction of time step t at iteration $i + 1$

Figure 4.26: Prediction of the same time step in two consecutive prediction iterations, the second iteration corrected the open loop solution, such that the vehicle is inside the computed back-up hull and stabilizes.

4.3. Combined Method

The preceding simulations displayed the promising results of implementing the Fiala tire model and a non-linear bicycle prediction model and the performance of the proposed back-up methods in their respective controlled environments. This final simulation is conducted to evaluate the performance of the complete proposed framework. The simulation is conducted at various friction coefficients at varying gaps, such that the back-up method is tested at lower friction levels, validating that the proposed method performs well in scenarios where both tight gaps and low friction are critical. The aim of this simulation is to verify that the proposed method performs when the previously individually considered critical aspects are combined.

The simulation is conducted with the complete control framework as described in Chapter 3, including the proposed back-up methods. The probability threshold is set at $\beta = 0.98$ and the back-up threshold $\beta_{bu} = 0.99$ for both back-up strategies similar to Section 4.2. Disturbance matrix $G = \text{diag}(0.05, 0.067, 0.013, 0.03)$ describes the prediction uncertainty, and initial perception uncertainty $\Sigma_0 = \boldsymbol{GIG}^\top$ similar to Section 4.2, and [13] and [42]. The simulation is modeled with a plant model utilizing a dynamical bicycle model with a Fiala tire model (described in Section 3.1.2 and Section 3.1.3 respectively) at a sampling time of 1ms, similar to Section 4.1. The simulation can be seen as a variant of the scenario of Section 4.2, in a more realistic simulation environment including lower tire-road friction coefficients.

As discussed in section 4.1, the baseline method is not designed for a simulation environment modeling low tire-road friction coefficients. Implementing this baseline method in this simulation environment would not lead to a valid comparison and is therefore omitted in these results. This simulation focuses on the performance of the proposed method, without a comparison to a baseline method.

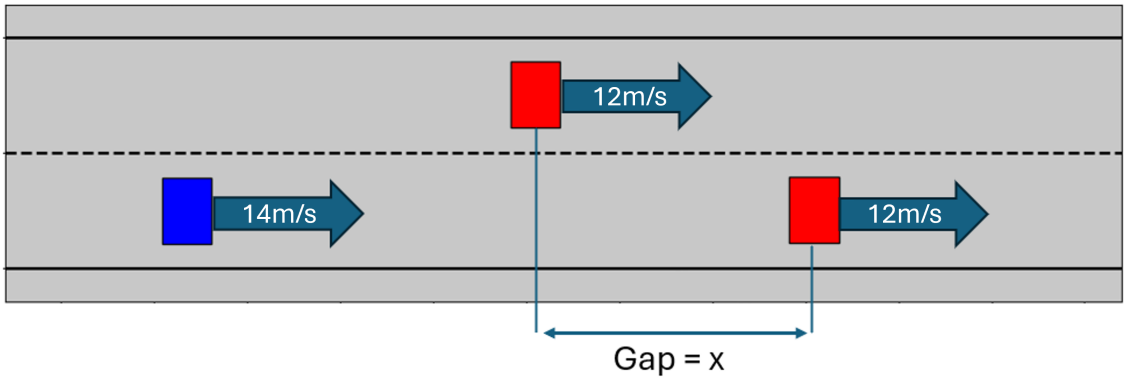


Figure 4.27: Scenario varying gaps

4.3.1. Scenario

In the simulation, the ego vehicle is approaching two target vehicles from behind (see Figure 4.27). The ego vehicle has a reference velocity of $14m/s$ and the target vehicles have a reference velocity of $12m/s$ (like in simulations of Section 4.1). The ego vehicle is positioned on the bottom lane. The first target vehicle is positioned on the top lane, 20 meters in front of the ego vehicle. The second target vehicle is positioned on the bottom lane at a various distance in front of the ego vehicle. The gap between the target vehicles is varied between the target vehicles between 1 and 16 meters. The simulation is conducted at the different tire-road friction coefficients deemed interesting in Section 4.2, namely 0.9, 0.45, and 0.35. The presented scenario is chosen to be able to make a comparison with the previous two simulations. The varying gaps and uncertainty levels are similar to the simulation scenario of Section 4.2, while the ego vehicle velocity will result in similar tire forces to the simulation scenario of Section 4.1.

4.3.2. Results

The average feasibility rate for the current-state back-up method is 87.5% and the average feasibility rate for the precomputed back-up method is 91.6% (see Tables 4.6 - 4.8).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Feasible
Current-state	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	87.5%
Precomputed	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	93.8%

Table 4.6: Comparison of back-up method feasibility at friction coefficient 0.9

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Feasible
Current-state	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	87.5%
Precomputed	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	93.8%

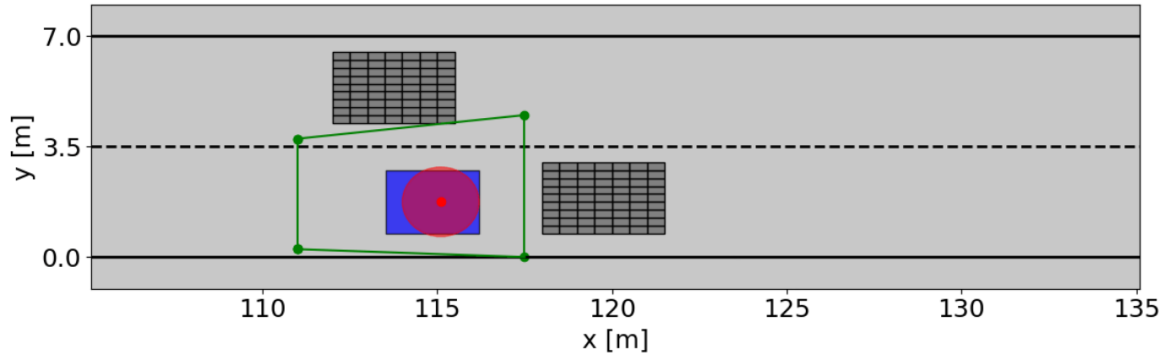
Table 4.7: Comparison of back-up method feasibility at friction coefficient 0.45

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Feasible
Current-state	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	87.5%
Precomputed	✓	✓	✓	✓	✓	×	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	87.5%

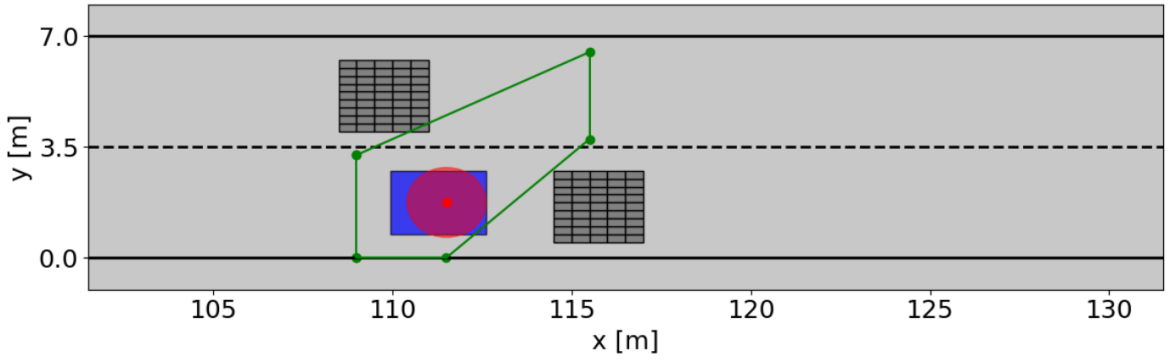
Table 4.8: Comparison of back-up method feasibility at friction coefficient 0.35

It can be seen in Tables 4.6 - 4.8 that both back-up methods lead to high feasibility rates, outperforming the simulation results of Section 4.2, but still resulting in infeasibility at certain gap widths. At narrow gaps the ego vehicle stays behind the target vehicles, and at wider gaps, the ego vehicle is able to overtake the target vehicles without any problems. Intermediate gaps, where there is a tight passage to overtake, are scenarios where infeasibility occurs, similarly to the preceding simulations. In Section 4.2, there was a significant difference in performance between the current-state and the precomputed back-up method. In contrast, the feasibility rates in this simulation are more similar. In contrast to Section 4.2, the two proposed method behave similarly across the various gap widths, in terms of generated back-up hulls. Furthermore it is observed that the feasibility rates are similar across the different levels of tire-road friction.

At the most narrow gaps, the ego vehicle reduces its velocity and reaches stable driving behind the target vehicles. When the simulated gaps are wider than 5 meters, some composed hulls in the prediction horizon have the tendency to "peek" into the new lane see Figure 4.28b, similarly to the behavior observed in Section 4.2. However, contrary to these simulations, it is observed that open-loop solutions remain more in the original lane and maneuvers in the open-loop solutions are less erratic. In contrast to Section 4.2, the method is able to deal with hull reaching over into the adjacent lane in this simulation, if the number of these hulls in the prediction horizon is limited.



(a) Computed (back-up) hull within initial lane at iteration i



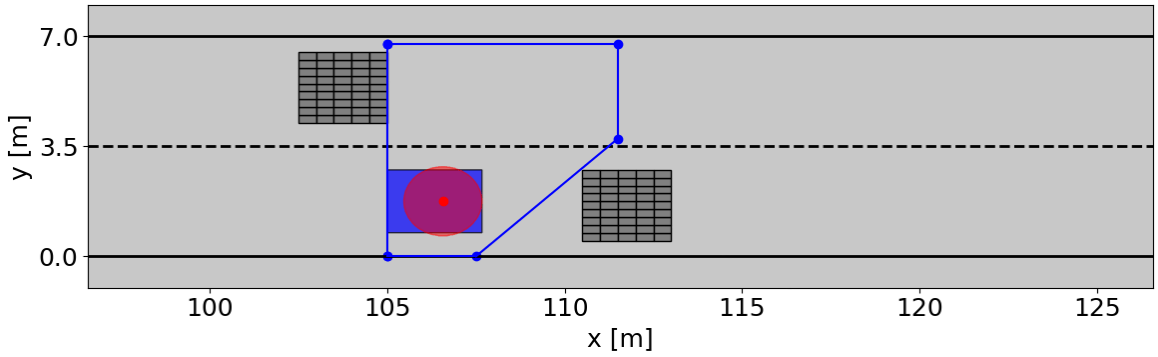
(b) Computed (back-up) hull reaching into other lane at iteration i

Figure 4.28: Examples of two back-up hulls computed within the same prediction horizon. Some computed hulls in the prediction horizons reach into the second lane, others remain fully in the initial lane.

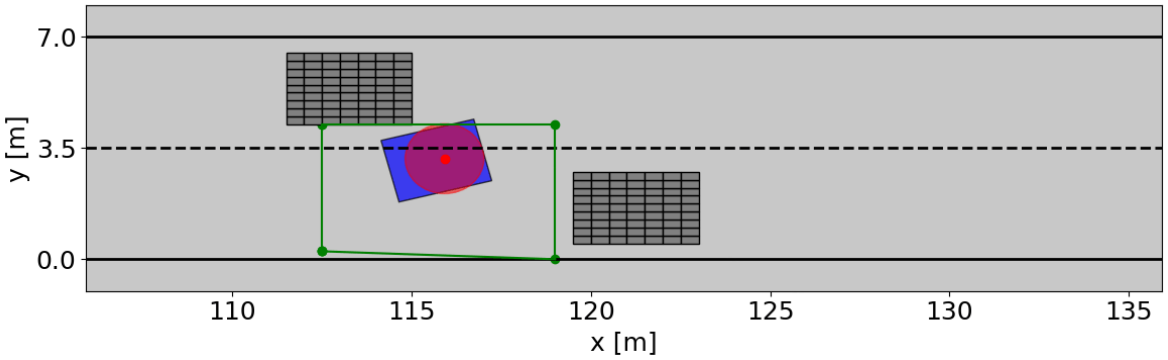
When gaps are wider, there are also more hulls reaching into the adjacent lane. Particularly, at gaps of 6 and 7 meters wide, this peeking behavior occurs more and leads to prediction horizons containing both hulls reaching into the other lane, as well as hulls remaining to the initial lane. In simulations where the problem is deemed infeasible, a lot of alternation occurs between the two types of hulls.

Gaps with a width of 8 meter and wider display different behavior. At these widths there is enough space for the ego vehicle to get past the first target vehicle. As discussed in 3.1.8, this results in the change of reference lane. At a gap of exactly 8 meter however, it is observed that the method fails. As

seen in Figure 4.29a, the change of reference trajectory happens when the first prediction step is in front of the first target vehicle. However, later in the prediction horizon, it can be seen that the nominal hull computation fails when the uncertainty has grown and the binary grid represents a greater occupied space (see Figure 4.29b). The back-up hulls are computed in the initial lane. As a result, the consecutive nominal and back-up hulls are far apart, leading to infeasibility.



(a) First prediction step time step t at iteration i. Reference trajectory is changed to second lane, since the lane is free



(b) Back-up hull at time step t + 15Δt at iteration i. Uncertainty grows, leading to infeasible nominal hulls. Back-up hull is computed in the initial lane

Figure 4.29

At wider gaps, the ego vehicle overtakes the target vehicles smoothly. At these gaps, the ego vehicle is able to overtake the target vehicles smoothly and no back-up hulls in the initial lane are observed.

5 Discussion

This chapter discusses and evaluates the performance of the proposed framework. First, the simulations conducted in the previous chapter and the corresponding results will be meticulously analyzed and interpreted, discussing the framework's strengths and limitations illuminated by the simulation. Next, certain limitations will be discussed, including general system limitations and assumptions that might not align with reality, highlighting interesting areas for future research.

5.1. Simulation Analyses: Strengths and Limitations

The results presented in the preceding chapter provide valuable insights in the performance of the proposed framework. This discussion aims to contextualize those findings and their implications and discuss the strengths and limitations of the proposed methods. The structure of this section is similar to the structure of the preceding chapter, discussing the implications of the proposed prediction model first, then discussing the proposed back-up strategies and finally discussing the results of the total proposed method. It is encouraged to keep the results from the previous chapter at to hand for better understanding of the interpretations presented in this section.

5.1.1. Simulation 1: Model Complexity

From the results presented in Section 4.1, it is evident that the proposed dynamic bicycle model with a non-linear Fiala tire model outperforms the other methods at various tire-road friction coefficients. In this section, the different simulations will be analyzed, and the results at different friction coefficients will be interpreted.

Friction Coefficient 0.9

At a high friction coefficient of 0.9, all methods, except for the point-mass model, successfully follow the reference trajectory. Despite sufficient available tire force provided by the high friction coefficient, the point-mass model fails to maintain control. This failure can be attributed to the mismatch between the inputs required by the plant and the inputs generated by the point-mass model. While the plant requires longitudinal acceleration and steering angle as inputs, the point-mass model computes a desired lateral acceleration, which must then be converted to the appropriate inputs using an approximation (Equation 4.3). However, this conversion is an oversimplification of the vehicle's dynamics, neglecting critical factors such as steering limitations and sideslip angles. Consequently, the intended lateral acceleration cannot always be accurately translated into the required system inputs, resulting in control loss.

The kinematic bicycle model demonstrates the ability to follow the reference trajectory, albeit with some overshoot. This behavior is clearly observed in the steering angles (Figure 4.3b), where the model exhibits a longer peak between 1.2s and 2.0s. This overshoot is due to the imposed steering angle limitation of $-\frac{\pi}{30} < \delta < \frac{\pi}{30}$, which constrains the vehicle's ability to make rapid steering adjustments. The steering constraint appears to have been introduced in the baseline method as an indirect way to limit lateral acceleration. However, since the method is limited to kinematic constraints and does not

account for the tire-road dynamics, it fails to adapt to changing conditions, leading to non-optimal steering inputs and the subsequent overshoot.

In contrast, the proposed dynamic bicycle models are able to follow the reference trajectory smoothly and without significant deviations. These models provide more accurate predictions of the vehicle's dynamics and generate a sequence of feasible control inputs, resulting in stable and precise trajectory tracking. At this friction coefficient, the results of the two dynamic bicycle models are nearly identical. The Fiala model describes a continuous transition from the linear tire dynamics at small sideslip angles to the fully saturated tire forces at large sideslip angles (see Figure 5.1). At high friction coefficients μ, the maximum lateral force F_{max} is relatively high (as described Equation 3.4). This means that the range of slip angles over which the tire force response is approximately linear is larger. In the gradual lane change maneuver, sideslip angles are relatively small, which allows for the approximation $\tan(\alpha) \approx \alpha$. Since the angles are relatively small, higher order terms in (3.5) including $|\tan(\alpha)|\tan(\alpha)$ and $\tan^3(\alpha)$ become negligible. As a result, the Fiala model simplifies to the linear relation $F_y \approx -C_\alpha \alpha$, which is why the results of the Fiala tire model closely resemble the results of the linear tire model at high friction coefficients.

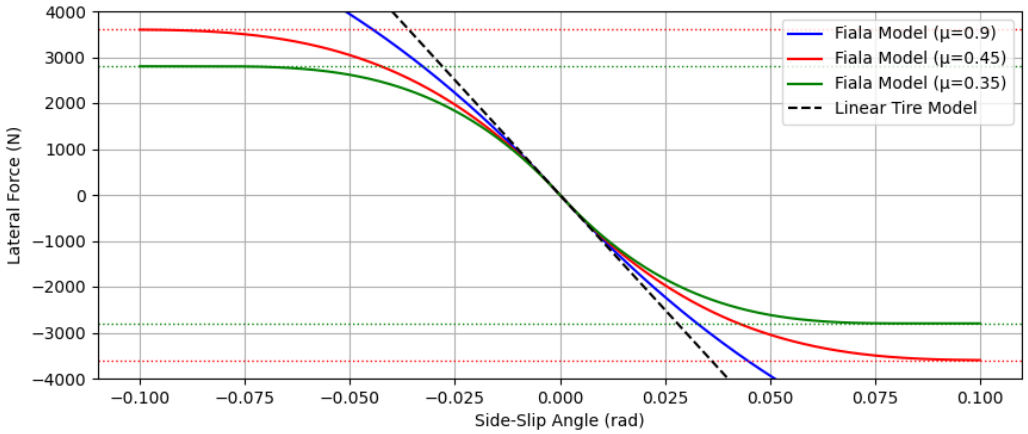


Figure 5.1: Fiala tire model relating sideslip angle to the available tire force at various friction coefficients

Friction Coefficient 0.45

At a reduced friction coefficient of 0.45, the kinematic bicycle model loses control. Similar to the high-friction simulation, the imposed steering input limitation results in an extended first peak in the steering angle, as shown in Figure 4.9b. This longer peak causes the need for compensation, which requires lateral acceleration (Figure 4.12) beyond what the tires can generate in reduced friction conditions. As a result, the vehicle cannot maintain stability, leading to a loss of control.

In contrast, both dynamic bicycle models successfully follow the trajectory at this lower friction level. At this friction level however, there are notable differences in the performance of the different models. The dynamic bicycle model with the linear tire model shows a slight overshoot of the reference trajectory (RMSE of $0.095m$), whereas the model with the non-linear Fiala tire model tracks the trajectory more accurately (RMSE of $0.085m$). This discrepancy can be attributed to the ability of the Fiala tire model to account for tire saturation effects. By Equation (3.4), at lower friction coefficients, F_{max} will be reduced and tire saturation effects occur at smaller sideslip angles, shrinking the linear domain of the function. In (3.5), the higher order terms $\frac{C_\alpha^2}{3F_{y,\max}}|\tan\alpha|\tan\alpha$ and $\frac{C_\alpha^3}{27(F_{y,\max})^2}\tan^3\alpha$ become more significant due to their relatively larger coefficients and as a result the model deviates more from the linear model (see Figure 5.1).

The Fiala tire model incorporates the maximum lateral tire forces, preventing the steering input from increasing once the tires are saturated. This ensures that additional steering input is not applied unnecessarily, as the model recognizes that further increases would not produce additional lateral acceleration due to tire saturation. The linear tire model, on the other hand, assumes a linear relationship between slip angle and lateral force, leading to a continued increase in steering input even when the tires

are saturated. This overestimation of the available tire force causes reduced responsiveness, resulting in overshoot. By accurately capturing the effects of tire saturation, the non-linear tire model achieves better trajectory tracking in the second turn of the lane-change maneuver.

Friction Coefficient 0.35
At a friction coefficient of 0.35, only the dynamic bicycle model with the Fiala tire model maintains control throughout the lane-change maneuver and demonstrates its differences with the dynamic bicycle model with the linear tire model, which loses control under these conditions.

The key difference lies in the ability of the non-linear tire model to account for tire saturation. The linear tire model assumes that the sideslip angle is proportional to the lateral acceleration, which holds true only in the linear operating range of the tire. This assumption breaks down at low friction coefficients, where the lateral forces saturate at moderate sideslip angles. As a result, during the lane change maneuver, the controller using the linear tire model continues to increase the steering input in an attempt to generate the required lateral acceleration. However, due to tire saturation, the lateral forces plateau, and further increasing the steering input does not lead to higher lateral forces. This discrepancy becomes critical in the second corner, where the vehicle requires significant lateral acceleration to stabilize. The linear model predicts forces that exceed the physical limits, which leads to a loss of control.

In contrast, the non-linear tire model captures the saturation behavior of the tires, recognizing that increasing the sideslip angle beyond the saturation point does not lead to additional lateral force. This results in more realistic steering inputs that align with the achievable forces, allowing for the vehicle to remain stable even under low-friction conditions.

General Observations and Conclusion
The results of the simulations demonstrate that the proposed controller utilizing a dynamic bicycle model and a non-linear Fiala tire model outperforms the baseline models at lower tire-road friction. The ability of the non-linear tire model to account for tire saturation allows for maintained control and tracking of the reference trajectory, even at reduced friction coefficients where the linear models fail to maintain control.

The simulations further reveal the limitations of the baseline models, particularly in their inability to account for the physical limits of lateral acceleration, which results in loss of control under demanding conditions. The kinematic bicycle model and point-mass model, while less computationally demanding, lack the necessary prediction model complexity to account for these physical limits effectively, making them unsuitable for lower friction conditions.

These results highlight the necessity of a higher complexity prediction model in order to maintain control at lower friction conditions. The requirement of the increased complexity in the prediction model underline the importance of an efficient constraint reformulation algorithm, to keep computational load manageable for real-time performance.

5.1.2. Simulation 2: Back-Up Method

The back-up methods increased the feasibility rates from 48.3% to 62.5% for the precomputed back-up hull method and 75.0% for the current-state back-up method across the tested scenarios. The improved feasibility highlights the adaptability of the proposed back-up methods.

The improvement in feasibility is mainly attributed to the improved performance in the tested scenarios with narrower gaps. Unlike the baseline method, which has a tendency to "peek" into the adjacent lane, the hulls computed with the new methods remain limited to the current lane at these narrow gaps. This improvement is a result of incorporating the physical dimensions of the ego vehicle into the hull-finding process. The proposed methods require a hull to be at least three meters wide to be considered valid, refraining the controller to explore tight gaps. In contrast, the baseline method does explore these overly tight gaps, leading to feasibility problems, when a back-up hull is required. This is shown in Figure 4.22, where Figure 4.22a and Figure 4.22b represent two consecutive prediction steps. In Figure 4.22b, the ego vehicle gets too close to the target vehicle in front of the vehicle to generate a hull, so the method falls back on the hull of the prediction step before (Figure 4.22a). Figure 4.22c shows the resulting open loop solution of this time step in the next iteration. The vehicle moves into the target vehicle in order to satisfy the back-up hull of Figure 4.22b as a result of the back-up hull computed at the previous iteration.

The new methods, on the other hand, require a wider space, since the physical dimensions of the vehicle are considered. As a result, the method will fall back on a safe back-up hull when the vehicle gets too close to the hull boundary and simply reduces longitudinal acceleration slightly so the ego vehicle is inside of the found prediction hull at the next prediction iteration (see Figure 4.23). This results in stable driving behind both target vehicles, without leading to infeasibility or risky maneuvers.

At wider gaps, the proposed methods also exhibit open-loop trajectories that extend into the adjacent lane (e.g. Figure 4.24), similar to the baseline method at more narrow gaps. This leads to tight situations. This behavior arises only at larger gaps, because the proposed methods require wider hulls to be considered valid (see Section 3.3.1). In these tight scenarios, the nominal hull finding algorithm often fails and the framework falls back on the back-up method. Unlike the baseline method (see Section 2.3.1), the back-up hulls generated by the proposed approach accurately represent the free space at the current time step. However, because these back-up hulls use a more conservative representation of free space, the already tight spaces become even tighter. As a result, the back-up method struggles to compute hulls that are close to the nominal hulls. This could lead to consecutive hulls that are far apart in the environment (see Figure 4.25), making it impossible for the vehicle to satisfy the hull constraints for consecutive time steps, due to the vehicle's physical limitations. As a result, the optimization problem becomes infeasible in these scenarios.

At gaps of 11 and 12 meters another difference is observed between the precomputed back-up method and the current-state back-up method. The current-state back-up method is able to compute a safe and feasible trajectory. In contrast, the precomputed back-up method, shows oscillating behavior before ultimately becoming unstable and infeasible. This difference can be explained by analyzing the architecture of both methods. The current-state method, computes back-up hulls that are closer to the nominal hulls, since the free space representation is closer to that of the nominal approach. Since nominal hulls and back-up hulls are close, this results in minor corrections, keeping the trajectory stable. The precomputed method on the other hand, computes back-up hulls based on the open-loop solution of the previous optimization step. This results in back-up hulls that differ more from the nominal computed hulls, since the back-up hulls are computed based on a slightly different environment representation. Since the back-up hulls and nominal computed hulls are combined in the prediction sequence, this results in larger differences between consecutive prediction steps. To account for these larger differences in consecutive prediction steps, more severe corrections are required. These harsher corrections in turn magnify the discrepancy between the current open-loop solution and the open-loop solution computed at the previous iteration (which are used for this iteration's back-up computation), causing the discrepancies to increase causing the system to become unstable and lose control.

General Observations and Conclusion

The simulation results demonstrate that the proposed methods outperform the baseline approach, achieving a significantly higher feasibility rate, particularly in scenarios with narrow gaps. This improvement can be attributed to the consideration of the physical dimensions of the vehicle in the constraint formulation. By requiring the convex hulls to account for the dimensions of the ego vehicle, the proposed methods ensure that only spaces are considered that are truly wide enough.

A critical advantage of the proposed methods is their inherent safety-by-design framework. Unlike the baseline method, which may compute back-up hulls that include occupied space, the proposed methods guarantee that the back-up hulls represent actual free space. This ensures that all computed feasible trajectories are safe, reducing the risk of collisions in scenarios where the baseline method might generate unsafe trajectories despite appearing feasible.

Furthermore, the two proposed back-up methods illustrate a trade-off between computational efficiency and performance. The precomputed back-up hull method based on the open-loop solution from the previous time step, allows for ready-to-go back-up solutions when required. However, since these back-up hulls are based on earlier predictions, they can display larger discrepancies from the current nominal hulls due to changes in the environment. This increases the likelihood of infeasible consecutive hulls in tight scenarios, which is one of the main observed causes for infeasibility. The current-state back-up method, which computes back-up hulls only when needed using the most up-to-date open-loop solution, displays less of these discrepancies, resulting in more consistent and feasible sequences of hulls. However, this approach requires sequential computation, which limits its ability to compute the

back-up hulls beforehand in parallel.

Overall, the proposed methods offer a significant advancement in terms of feasibility and safety compared to the baseline approach. The observed trade-offs between the proposed methods provide opportunities to balance computational efficiency and robustness in tight scenarios.

5.1.3. Simulation 3: Complete Method

The combined method showed feasibility rates of 87.5% for the current-state back-up method and 91.6% for the precomputed back-up method across the different friction levels. These results show an improvement to the results of the simulation of section 4.2.2, while remaining in control. This indicates that the back-up method produces promising results under low-friction conditions. To interpret the results, these simulations are compared to the results of the simulation of Section 4.2 and its corresponding interpretation from Section 5.1.2. The significant improvement in feasibility rate observed in the simulation results of the combined method can be attributed primarily to two factors: the increased control frequency and a more realistic consideration of vehicle limitations. These modifications collectively resolve critical challenges identified in the previous simulation, leading to enhanced performance.

The higher control frequency reduces the temporal gap between consecutive simulation steps, resulting in greater continuity between consecutive hulls in the prediction horizons. This continuity ensures that predicted vehicle states are less erratic, as the environment undergoes smaller changes between steps. Moreover, the incorporation of realistic lateral acceleration limits, prevents the vehicle from predicting abrupt maneuvers to satisfy the predicted constraints, which was a prominent issue in the previous simulations. The combination of these two factors allows smoother open-loop solutions and more consistent nominal and back-up hulls.

The reduction in erratic behavior not only increased the continuity of consecutive prediction steps within the prediction horizon, but also the continuity between consecutive open-loop solutions, which is critical for the precomputed back-up method, since this method relies on the preceding open-loop solution to compute back-up hulls. Smoother transitions between iterations ensure less discrepancy between nominal and back-up hulls, further improving feasibility. The shortened prediction horizon in this simulation also contributes to this improvement by reducing the accumulation of compounding differences over time. Because the difference of open-loop solutions is significantly reduced, the precomputed hulls are more similar to the back-up hulls computed by the current-state method, addressing one of the main issues with the precomputed back-up method in simulation 4.2, where large differences between nominal and back-up hulls led to infeasibility. Where in the previous simulation, there were significant and interpretable differences in accuracy between the precomputed back-up method and the current-state back-up method, here the methods' accuracy and behavior are similar.

Even though the feasibility rates have increased, there are still some scenarios where the proposed method encounters difficulties. Similarly to the results observed in the second simulation, the combined method displayed instances where the computed hulls extended into the adjacent lane (e.g. Figure 4.28b), particularly in scenarios with wider gaps that were insufficient for a complete overtaking maneuver. In the second simulation, this peeking behavior often resulted in infeasibility due to the generation of erratic open-loop solutions. These erratic solutions caused large discrepancies between consecutive prediction steps, ultimately destabilizing the system. In contrast, this simulation demonstrates improved performance under similar conditions. The proposed vehicle model, incorporating more realistic steering dynamics, reduces the occurrence of these erratic behaviors. This leads to smoother and more stable open-loop solutions, which in turn minimize the discrepancies between consecutive prediction steps. For instance, at a gap width of 6 meters, the optimization remained feasible despite hulls partially extending into the adjacent lane for most simulations (see Figure 4.28). This improvement can be primarily attributed to smoother steering dynamics and an increased control frequency, which enhanced the system's ability to maintain feasibility despite minor inconsistencies in hull placement.

However, the more constricted steering maneuverability introduced by the proposed model imposes its own challenges. When hulls extend into the adjacent lane (as shown in Figure 4.28b), the effective size of the feasible hull in the driving lane is reduced. This effect occurs because the vehicle cannot effectively utilize the hulls extending into the adjacent lane without fully committing to a lane change.

In scenarios where both types of hulls (those constrained to the original lane (Figure 4.28a) and those partially extending into the adjacent lane (Figure 4.28b)), appear in the prediction horizon, the vehicle encounters difficulty. Specifically, it struggles to adhere to the hulls in the adjacent lane while maintaining compliance with the hulls in the original lane. As a result, the open-loop solutions remain confined to the original lane, effectively rendering the free space in the adjacent lane unusable and reducing the effective feasible space for the ego vehicle, leading to infeasibility in some scenarios.

In scenarios where the gap between target vehicles is 8 meters or wider, the observed behavior shifts. At these gap widths, the reference trajectory transitions to the adjacent lane. This reference trajectory change is determined based on the first prediction time step as described in more detail in Section 3.1.8, under the assumption that if the lane is free at the first prediction step, it will be free for the rest of the prediction horizon. This is often a valid assumption, since the ego vehicle is likely to have a higher velocity than the vehicle it is overtaking. If the ego vehicle is in front of the vehicle to be overtaken at the first time step, it will likely be in front of the vehicle for the remainder of the the prediction horizon as well. However, when the gap is tight, the ego vehicle reduces its velocity before reaching the point where this trajectory shift occurs. As a result, the velocity of the ego and target vehicle become nearly identical. This, coupled with growing prediction uncertainties, can lead to situations where the adjacent lane is predicted to be partially occupied later in the prediction horizon (shown in Figure 4.29b), because the occupied grids cells grow faster than the relative velocity of the vehicles. In such cases, the reference trajectory change to the adjacent lane leads to a sequence of inputs where the ego vehicle begins switching lanes, only to fall back on back-up hulls in the original lane when the hull-finding method fails to identify a valid hull in the adjacent lane. These discrepancies lead to infeasibility. For gaps wider than 8 meters, this issue is mitigated. The wider gaps allow the ego vehicle to maintain a higher velocity relative to the target vehicles, ensuring that the trajectory change to the adjacent lane occurs smoothly. This results in stable and feasible overtaking maneuvers, even under low-friction conditions.

General Observations and Conclusion

The simulation results demonstrate that the proposed methods, which model variable friction coefficients and utilize the proposed back-up methods, still show their respective improvements in performance when both back-up hulls are required and when the tire-road friction is limited. Feasibility rates of 87.5% for the current-state back-up method and 91.6% for the precomputed back-up method illustrate that the combined method performs well when both critical aspects are integrated.

A particularly promising outcome is the similar feasibility rate of the precomputed back-up method compared to the current-state method. This result is significant because the precomputed method allows for parallel computation, offering computational efficiency while maintaining similar accuracy. The reduced discrepancies between nominal and back-up hulls observed in this simulation improve feasibility, reducing the challenges observed in previous simulations.

While some challenges remain regarding the continuity between consecutive hulls in the prediction sequence, this simulation demonstrates that the increased frequency and the incorporation of realistic vehicle limitations have resolved most of the issues observed in the previous simulation.

In conclusion, this simulation validates the effectiveness of the combined method, confirming that it reliably integrates low-friction modeling with robust back-up methods. The approach ensures safe and feasible trajectories even in challenging scenarios.

5.2. Limitations

The proposed framework demonstrates significant strengths in addressing perception uncertainties and obstacle avoidance in various friction conditions. Specific strengths and limitations regarding the conducted simulations have been discussed. However certain general system limitations remain that must be acknowledged. Furthermore there are limitations in the simulation environment assumptions that might not fully capture reality. These limitations highlight areas for future research to further improve the method's performance and applicability.

5.2.1. Computational Overhead

One limitation of the proposed framework is its computational overhead, which affects the real-time applicability. While the method successfully generates safe and feasible trajectories and back-up trajectories, the current implementation is unable to solve the optimization problem within the time constraints required for real-time performance. This research was conducted as a proof-of-concept to demonstrate the feasibility of applying grid-based stochastic model predictive control across various road conditions. However, the implementation was not optimized for computational efficiency. Specifically, the framework was developed in Python, a high-level language chosen for its flexibility during research and prototyping. Although Python enables rapid development, it is not well-suited for computationally intensive tasks, resulting in performance limitations.

While the framework was implemented in Python, which is not known for its computational efficiency, we are able to make some comparisons with the state of the art, considering potential computation times. The proposed ego vehicle prediction model is very similar to [9], which claims real-time performance. The main difference being that the proposed method in this thesis considers the reformulation of the uncertain obstacle states to a convex hull, as opposed to the the signed distance to obstacles (of which the state is assumed to be deterministic) in the state constraints. Reformulating the uncertain states to convex hull to account for uncertainties requires computation, however, it results in a set of linear constraints describing a convex free space, independent of the number of obstacles. This could significantly improve optimization time with respect to the state of the art, which requires a non-convex constraint for every obstacle in the environment. It would be interesting to investigate the computational complexity trade-off between calculating the convex hulls and optimization time gains.

A notable observation is that optimization times increased significantly in scenarios where constraints were tighter and friction coefficients were lower. This behavior can be attributed to the limited feasible inputs and reduced search space inherent in such conditions. These limitations affect the optimization process by increasing complexity and requiring more computational effort to resolve. Recognizing this relationship between scenario tightness, friction coefficients, and computational requirements is critical for guiding future enhancements and adapting the framework to ensure real-time feasibility.

Another area of improvement in the computational overhead is associated with the hull-finding method. While relatively efficient at lower frequencies, the current implementation computes a convex hull from scratch at every prediction time step. This process involves extensive exploration of the grid to check for free space, with many grid cells being redundantly validated during the exploration phase (see Section 3.2.3). Moreover, the current method does not leverage the fact that consecutive convex hulls are closely related, as their temporal and spatial distance is small and their shapes are often similar. While the current approach reliably generates valid convex hulls, it does introduce unnecessary computational load.

5.2.2. Simulation Environment Assumptions

It must be noted that these simulations are performed using the Fiala tire model, which is utilizes as both the control model (used in the proposed predictive control framework) and the plant model (used to simulate the vehicle's true dynamics). While the Fiala tire model is capable of capturing tire saturation effects, it does not represent the full complexity of real tire behavior. As described in Section 3.1.3, the Fiala model accounts for a maximum lateral acceleration. However, the model does not account for phenomena such as a reduction in lateral force at larger sideslip angles. These effects can be critical to accurately model real tire dynamics and could influence the behavior of the proposed framework in more complex scenarios. The use of the same Fiala model for both control and plant introduces limitations in assessing the robustness of the method. Therefore, further testing in a simulation environment that incorporates more detailed tire dynamics is necessary to validate and refine the approach. Additionally, it must be noted that the method relies on the assumption of a known tire-road friction coefficient, which may not always be available in practice. Sudden changes in tire-road friction, such as encountering icy or wet patches, could challenge the model's performance. Further extensions to dynamically estimate or adapt to changing friction conditions could enhance the model's applicability.

In the simulations, relatively low uncertainty levels are considered. These levels are similar to those of the baseline method [13], selected to ensure a fair comparison. However, these low uncertainty values

might not fully capture the challenges of real-world scenarios, where uncertainty levels are often higher and more variable. A key simplification in the current framework is the assumption that the reference trajectory of target vehicles is known. This assumption deviates from real-world conditions, where the behavior of other road users is uncertain without communication between vehicles. Additionally, the additive uncertainty model assumes a low initial uncertainty that grows over time due to prediction inaccuracies. While this approach captures temporal uncertainty growth, it does not fully account for scenarios with higher initial uncertainty, such as poorly detected obstacles or rapidly changing conditions. An example of the importance of addressing representative uncertainties was observed in the final simulation where tightly constrained gaps and growing prediction uncertainties led to infeasibility. In these scenarios, the relative velocities of the ego and target vehicles became nearly identical, and the uncertainty growth rate outpaced the relative motion, causing the adjacent lane to be incorrectly predicted as partially occupied. Similar issues could become even more prominent in environments with larger initial uncertainties, or faster-growing uncertainties.

Another limitation is the reliance on Gaussian distributions to represent uncertainty. While Gaussian models are computationally convenient, they may not accurately reflect the true nature of uncertainties in real-world environments. For example, the behavior of other traffic participants often follows distinct patterns, such as a lower likelihood of lane changes into already-occupied lanes, which Gaussian models may not capture effectively. Furthermore, perception uncertainties might not follow Gaussian distributions [34]. Exploring alternative distributions that better reflect real-world uncertainties could enhance the method's robustness and applicability.

Addressing these limitations would allow for a more comprehensive and realistic integration of uncertainties, further improving the reliability and performance of the proposed framework in diverse and dynamic environments.

6

Conclusion

This thesis aimed to address the challenge of dealing with perception uncertainties in various road conditions, with a focus on improving safety and feasibility in automated driving. Low tire-road friction conditions introduce challenges for motion planning due to reduced available tire-road friction forces, which affects vehicle dynamics and maneuverability. These conditions often coincide with other adverse weather effects that increase perception uncertainties, compounding the difficulty of trajectory planning. Existing approaches frequently treat these aspects in isolation, either ignoring complex dynamics or neglecting uncertainty, leading to limitations in real-world applicability. To bridge this gap, this research tackled the problem of integrating perception uncertainties while also accounting for non-linear vehicle dynamics at lower friction. The proposed framework addressed the research question: *How can perception uncertainties be effectively integrated into motion-planning models for obstacle avoidance to improve performance and safety of automated vehicles in various road conditions?*

This thesis demonstrates that perception uncertainties can be effectively integrated into obstacle avoidance models under various tire-road friction levels by incorporating a non-linear dynamic bicycle model combined with a Fiala tire model into a grid-based stochastic model predictive control framework. The proposed framework mitigates the compounding computational complexity typically introduced when dealing with uncertainties and incorporating complex models that can account for lower friction conditions. The grid-based approach reformulates the non-convex obstacle avoidance problem to a set of linear constraints, effectively dealing with the perception uncertainties while reducing the optimization complexity.

To effectively consider uncertainties in lower friction conditions, this thesis has produced two significant contributions. First of all, this thesis implemented a dynamic prediction bicycle model and a Fiala tire model into the grid-based stochastic model predictive control framework for the first time, allowing the consideration of the dynamic limitations related to reduced tire-road frictions in adverse weather conditions, showing that the framework can be used in combination with a realistic prediction model representing real-world dynamics. Furthermore this thesis has improved the safety and feasibility of the trajectories computed using the grid-based constraint reformulation, by developing a novel back-up method to safely fall back on when the nominal constraint formulation fails. The method offers flexibility in application, as it can be employed with two distinct strategies: one prioritizing accuracy, and the other prioritizing computational efficiency for time-critical situations. The method ensures that the found hull reflects an accurate and conservative representation of the free space, ensuring a safe set of constraints. To ensure safe constraint formulation some supporting extensions to the framework were implemented. These include the consideration of the physical ego vehicle dimensions in the hull finding algorithm and the implementation of a dynamic probability threshold, allowing for a better representation and comparison of uncertainty.

The simulation at various friction coefficients demonstrated the effectiveness of the proposed dynamic bicycle model with a non-linear Fiala tire model in handling low-friction conditions, significantly outperforming the other compared linear models. At higher friction levels, all models except the

point-mass model were able to follow the reference trajectory, with the dynamic models providing smoother and more accurate trajectory tracking. As friction coefficients decrease, the non-linear tire model's ability to account for tire saturation becomes crucial, being capable to maintain control where other methods fail. This strength highlights the importance of incorporating realistic vehicle dynamics into motion planning in lower friction conditions.

The simulations considering an overtaking maneuver at various gaps highlighted the significant improvement in feasibility rates achieved by the proposed back-up framework (75.0% for the current-state method and 62.5% for the precomputed method) compared to the baseline approach (43.8%), particularly in scenarios with narrow gaps. Due to the consideration of the physical dimensions of the vehicle, the proposed methods ensure that only spaces wide enough to accommodate the ego vehicle are considered valid for trajectory planning, improving feasibility rates with respect to the baseline method. Furthermore, the inherent safety-by-design framework of the proposed methods showed its capability to compute back-up hulls that, contrary to the baseline method, are guaranteed to represent the free space determined in the binary grid. Thereby mitigating the risks of trajectories that are deemed feasible, but are unsafe. A comparison between the two proposed back-up strategies revealed a trade-off between computational efficiency and feasibility rates. The precomputed back-up method benefits from parallel computation, offering ready-to-use back-up hulls. However, the strategy introduces larger discrepancies between consecutive nominal and back-up hulls due to reliance on older environment representations, which can lead to infeasibility in tight scenarios. In contrast, the back-up method utilizing up-to-date environment information computes back-up hulls closer to the nominal ones, resulting in more stable and feasible trajectory sequences more often. While this sequential approach improves robustness in this simulation, it requires computation of the back-up hull in real time, limiting its precomputation capabilities. Ultimately, the choice between these methods is a trade-off between computational efficiency and feasibility performance, and should be determined by the specific requirements of the system.

The third and final simulation integrated the critical aspects of the preceding simulation scenarios, combining the challenges of varying tire-road friction conditions with narrow overtaking gaps to test the robustness of the proposed back-up methods under more challenging friction conditions. This simulation was conducted with the complete proposed framework, incorporating a higher planning frequency and a shorter prediction horizon than in the second simulation, and utilizes a non-linear bicycle prediction model and a Fiala tire model. The results demonstrated that both the precomputed and current-state back-up method remained effective and that the ego vehicle maintained control, even in scenarios with reduced tire-road friction and tighter gaps. The precomputed method performed comparably to the current-state approach in terms of feasibility (rates of 91.6% and 87.5% respectively), suggesting that the higher planning frequency and shorter prediction horizon reduces the trade-off in accuracy observed with the precomputed strategy in the second simulation. This highlights the potential of the precomputed method to offer computational advantages without compromising performance in higher-frequency planners.

While the simulation results underscore the strengths of the proposed framework, they also highlight several limitations that must be addressed to improve the applicability of the framework in real-world settings. A key limitation lies in the observed discrepancies between consecutive nominal and back-up hulls, which are the primary cause of infeasibility in tight scenarios. Furthermore, the computational overhead presents a challenge, with the current Python-based implementation not being able to achieve real-time performance. The hull-finding algorithm, while effective, computes hulls from scratch at every step and often redundantly validates grid cells, contributing to the computational load. Finally, this work is limited in some assumptions made, which might not fully capture real-world driving scenarios. This includes the assumption of relatively low Gaussian state uncertainties, based on the known target vehicle reference trajectory; the assumption of a known tire-road friction coefficient; and neglected dynamic effects of real-world driving in the simulation environment. Addressing these limitations offers opportunities for improvements. Therefore, the following recommendations for future work are proposed:

Enhanced Uncertainty Representation Future research should focus on improving the representation of uncertainties to better reflect real-world scenarios. This includes testing with higher uncertainty magnitudes and exploring non-Gaussian distributions that better capture sensor characteristics and

traffic participant behavior. Incorporating models that describe uncertainty in obstacle states based on common traffic behavior, like staying in the lane or smoothly changing lanes, could provide a more realistic representation of the environment uncertainty. These refinements would enhance the robustness of the framework in dynamic and uncertain settings and improve its applicability to real-world conditions.

Improving Computational Efficiency The computational performance of the proposed framework is another key area for future improvement. Implementation in a real-time optimized programming language, such as C++ or Rust, could significantly reduce computation time. Furthermore, optimizing the hull-finding method could reduce redundant grid cell checks and streamline the process by leveraging previously computed hulls as initial guesses. These improvements would not only reduce computational overhead but also improve the continuity between consecutive hulls, addressing one of the observed causes of infeasibility in the current framework. Together, these optimizations could bring the framework closer to real-time applicability.

Integration into a Full Automated Driving Framework The proposed framework could benefit from integration into a larger system with a higher-level behavioral planner. Such a planner could leverage the established rules and behavior that are common on highways, to guide the hull-finding process. For example, the planner could prioritize exploration within the ego vehicle's current lane when adjacent lanes are clearly occupied, mimicking human decision-making. Additionally, the planner could incorporate friction coefficient estimation to adapt the reference trajectory generation and hull exploration strategy to the vehicle's current limitations. This approach would allow the system to operate closer to its physical limits while reducing unnecessary infeasibility caused by exploring unsafe areas.

Urban Driving Scenarios As a final recommendation, a promising direction for future research is adapting the proposed framework to urban driving scenarios. While friction limits are generally less critical due to lower velocities, scenarios involving extremely low friction coefficients, such as icy or snow-covered roads, could still pose a challenge for which this framework could be promising. Urban environments also introduce different challenges, such as less predictable traffic behavior and a higher density of dynamic obstacles, for which the efficient constraint reformulation could be promising.

The advancements in automated driving continue to push the boundaries of what is possible. While several challenges remain, full automation is gradually approaching reality. This thesis has addressed one of these challenges by developing a framework that integrates perception uncertainties into motion planning under varying tire-road friction conditions. While further improvements are necessary, the proposed framework enhances safety and feasibility in adverse weather conditions. By considering both the effects of reduced tire-road friction and the uncertainties inherent in perception systems, this work contributes to addressing critical edge cases, bringing automated driving a step closer to full automation.

References

[1] World Health Organization. *Global Status Report on Road Safety 2023*. World Health Organization, Dec. 2023.

[2] Eleni Petridou and Maria Moustaki. "Human factors in the causation of road traffic crashes." In: *European journal of epidemiology* 16.9 (Jan. 2000), pp. 819–826. DOI: 10.1023/a:1007649804201. URL: https://doi.org/10.1023/a:1007649804201.

[3] *SAE International releases updated Visual Chart for its "Levels of Driving Automation" standard for Self-Driving Vehicles*. Dec. 2018. URL: https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles.

[4] Waymo. *Waymo Safety Report on the road to fully Self-Driving*. Tech. rep. URL: https://storage.googleapis.com/sdc-prod/v1/safety-report/Safety%20Report%202018.pdf.

[5] Nidhi Kalra and David G. Groves. *The enemy of good*. Jan. 2017.

[6] Tim Brophy et al. "A Review of the Impact of Rain on Camera-Based Perception in Automated Driving Systems". In: *IEEE Access* 11 (2023), pp. 67040–67057. ISSN: 2169-3536. DOI: 10.1109/access.2023.3290143. URL: https://dx.doi.org/10.1109/access.2023.3290143.

[7] Siyu Teng et al. "Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives". In: *IEEE Transactions on Intelligent Vehicles* 8.6 (2023), pp. 3692–3711. ISSN: 2379-8904. DOI: 10.1109/tiv.2023.3274536. URL: https://dx.doi.org/10.1109/tiv.2023.3274536.

[8] Yuxiao Zhang et al. *Perception and sensing for autonomous vehicles under adverse weather conditions: A survey*. Tech. rep. Jan. 2023, pp. 146–177. URL: https://doi.org/10.1016/j.isprsjprs.2022.12.021.

[9] Matthew Brown and J. Christian Gerdes. "Coordinating Tire Forces to Avoid Obstacles Using Nonlinear Model Predictive Control". In: *IEEE Transactions on Intelligent Vehicles* 5.1 (Nov. 2019), pp. 21–31. DOI: 10.1109/tiv.2019.2955362. URL: https://doi.org/10.1109/tiv.2019.2955362.

[10] Janick V. Frasch et al. "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles". In: *2013 European Control Conference (ECC)*. 2013, pp. 4136–4141. DOI: 10.23919/ECC.2013.6669836.

[11] Wenda Xu et al. "Motion planning under uncertainty for on-road autonomous driving". In: IEEE. DOI: 10.1109/icra.2014.6907209. URL: https://dx.doi.org/10.1109/icra.2014.6907209.

[12] Dachuan Li et al. "Safe Motion Planning for Autonomous Vehicles by Quantifying Uncertainties of Deep Learning-Enabled Environment Perception". In: *IEEE Transactions on Intelligent Vehicles* 9.1 (2024), pp. 2318–2332. ISSN: 2379-8904 2379-8858. DOI: 10.1109/tiv.2023.3297735.

[13] Tim Brüdigam et al. "Grid-Based Stochastic Model Predictive Control for Trajectory Planning in Uncertain Environments". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–8. DOI: 10.1109/ITSC45102.2020.9294388.

[14] Ashwin Carvalho et al. "Stochastic predictive control of autonomous vehicles in uncertain environments". In: 2014.

[15] Bonnie Kittle. *vehicle on black paved road towards green hill*. Jan. 2018. URL: https://unsplash.com/photos/vehicle-on-black-paved-road-towards-green-hill-aQnyyf-4uZQ.

[16] Rahul Kala and Kevin Warwick. "Multi-Level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization". In: *Journal of intelligent & robotic systems* 72.3-4 (Mar. 2013), pp. 559–590. DOI: 10.1007/s10846-013-9817-7. URL: https://doi.org/10.1007/s10846-013-9817-7.

[17] Michael Montemerlo et al. “Junior: The Stanford entry in the Urban Challenge”. In: *Journal of field robotics* 25.9 (Aug. 2008), pp. 569–597. DOI: 10.1002/rob.20258. URL: https://doi.org/10.1002/rob.20258.

[18] S Koenig and M Likhachev. “D* lite”. In: *Eighteenth national conference on Artificial intelligence*. 2002, pp. 476–483.

[19] Julius Ziegler and Christoph Stiller. “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”. In: Oct. 2009. DOI: 10.1109/iros.2009.5354448. URL: https://doi.org/10.1109/iros.2009.5354448.

[20] Michael Montemerlo et al. “Junior: The Stanford entry in the Urban Challenge”. In: *Journal of field robotics* 25.9 (Aug. 2008), pp. 569–597. DOI: 10.1002/rob.20258. URL: https://doi.org/10.1002/rob.20258.

[21] Wenda Xu et al. “A real-time motion planner with trajectory optimization for autonomous vehicles”. In: May 2012. DOI: 10.1109/icra.2012.6225063. URL: https://doi.org/10.1109/icra.2012.6225063.

[22] *Continuous curvature planning with obstacle avoidance capabilities in urban scenarios*. Oct. 2014. URL: https://ieeexplore.ieee.org/abstract/document/6957887.

[23] David Gonzalez et al. “A review of motion planning Techniques for Automated vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.4 (Nov. 2015), pp. 1135–1145. DOI: 10.1109/tits.2015.2498841. URL: https://doi.org/10.1109/tits.2015.2498841.

[24] Y. Kuwata et al. “Real-Time motion planning with applications to autonomous urban driving”. In: *IEEE transactions on control systems technology* 17.5 (Sept. 2009), pp. 1105–1118. DOI: 10.1109/tcst.2008.2012116. URL: https://doi.org/10.1109/tcst.2008.2012116.

[25] Tyler Summers. “Distributionally Robust Sampling-Based Motion Planning Under Uncertainty”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 6518–6523. DOI: 10.1109/IROS.2018.8593893.

[26] Korbinian Moller et al. *Frenetix Motion Planner: High-Performance and Modular Trajectory Planning Algorithm for Complex Autonomous Driving Scenarios*. 2024. arXiv: 2402.01443 [cs.RO].

[27] Christian Pek et al. “CommonRoad Drivability Checker: Simplifying the Development and Validation of Motion Planning Algorithms”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 1013–1020. DOI: 10.1109/IV47402.2020.9304544.

[28] None Jeong Hwan Jeon et al. “Optimal motion planning with the half-car dynamical model for autonomous high-speed driving”. In: June 2013. DOI: 10.1109/acc.2013.6579835. URL: https://doi.org/10.1109/acc.2013.6579835.

[29] *Collision Avoidance for Dynamic Obstacles with Uncertain Predictions using Model Predictive Control*. Dec. 2022. URL: https://ieeexplore.ieee.org/document/9993319.

[30] Pallav Rawat. “Model predictive Control 101 - Pallav Rawat - Medium”. In: (Feb. 2023). URL: https://pallavrawat.medium.com/model-predictive-control-101-28de17208c39.

[31] Tianqi Qie et al. “An Improved Model Predictive Control-Based Trajectory Planning Method for Automated Driving Vehicles Under Uncertainty Environments”. In: *IEEE Transactions on Intelligent Transportation Systems* 24.4 (2023), pp. 3999–4015. ISSN: 1524-9050 1558-0016. DOI: 10.1109/tits.2022.3230680.

[32] L. Magni and R. Scattolini. “Robustness and Robust Design of MPC for Nonlinear Discrete-Time Systems”. In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Ed. by R. Findeisen, F. Allgöwer, and L.T. Biegler. Vol. 358. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer, 2007. DOI: 10.1007/978-3-540-72699-9_19. URL: https://doi-org.tudelft.idm.oclc.org/10.1007/978-3-540-72699-9_19.

[33] Baha Zarrouki, Chenyang Wang, and Johannes Betz. *A Stochastic Nonlinear Model Predictive Control with an Uncertainty Propagation Horizon for Autonomous Vehicle Motion Control*. Oct. 2023. URL: https://arxiv.org/abs/2310.18753.

[34] Marco Gabella. "Variance of Fluctuating Radar Echoes from Thermal Noise and Randomly Distributed Scatterers". In: *Atmosphere* 5.1 (2014), pp. 92–100. ISSN: 2073-4433. DOI: 10.3390/atmos5010092. URL: https://www.mdpi.com/2073-4433/5/1/92.

[35] Tim Brüdigam. *Safety and Efficiency in Model Predictive Control for Systems with Uncertainty*. Apr. 2022.

[36] Marcelo Saval-Calvo et al. "A Review of the Bayesian Occupancy Filter". In: *Sensors* 17.2 (2017). ISSN: 1424-8220. DOI: 10.3390/s17020344. URL: https://www.mdpi.com/1424-8220/17/2/344.

[37] Sascha Steyer et al. "Grid-Based Object Tracking With Nonlinear Dynamic State and Shape Estimation". In: *IEEE Transactions on Intelligent Transportation Systems* 21.7 (2020), pp. 2874–2893. DOI: 10.1109/TITS.2019.2921248.

[38] Changliu Liu et al. "Convex feasible set algorithm for constrained trajectory smoothing". In: *2017 American Control Conference (ACC)*. 2017, pp. 4177–4182. DOI: 10.23919/ACC.2017.7963597.

[39] Hans Pacejka. *Tire and vehicle dynamics*. Elsevier, Apr. 2012.

[40] Ali MesbAh. "An overview and perspectives for future research". In: *IEEE CONTROL SYSTEMS MAGAZINE* (Nov. 2016).

[41] Jack E Bresenham. "Algorithm for computer control of a digital plotter". In: *Seminal graphics: pioneering efforts that shaped the field*. 1998, pp. 1–6.

[42] Tim Brüdigam et al. "Combining Stochastic and Scenario Model Predictive Control to Handle Target Vehicle Uncertainty in an Autonomous Driving Highway Scenario". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 1317–1324. DOI: 10.1109/ITSC.2018.8569909.

A

Control Models

Point-Mass Model

The state vector x_k is defined as:

$$x_k = \begin{bmatrix} x_k \\ v_{x,k} \\ y_k \\ v_{y,k} \end{bmatrix}$$

where x_k and y_k represent the longitudinal and lateral positions of the vehicle, $v_{x,k}$ and $v_{y,k}$ represent the longitudinal and lateral velocities respectively.

The control input u_k is:

$$u_k = \begin{bmatrix} a_{x,k} \\ a_{y,k} \end{bmatrix}$$

where $a_{x,k}$ and $a_{y,k}$ are the longitudinal and lateral accelerations.

The system is described by the following equations:

$$x_{k+1} = Ax_k + Bu_k \tag{A.1}$$

where A and B are the system matrices given by:

$$A = \begin{bmatrix} 1.0 & \Delta t & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & \Delta t \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5\Delta t^2 & 0.0 \\ \Delta t & 0.0 \\ 0.0 & 0.5\Delta t^2 \\ 0.0 & \Delta t \end{bmatrix}$$

where Δt is the time step between successive states.

Kinematic Bicycle Model

The system is described by the following equations:

$$\begin{aligned}
\dot{x} &= v\cos(\psi + \alpha) \\
\dot{y} &= v\sin(\psi + \alpha) \\
\dot{\psi} &= \frac{v}{l_\mathrm{r}}\sin(\alpha) \\
\dot{v} &= a \\
\alpha &= \arctan\left(\frac{l_\mathrm{r}}{l_\mathrm{r} + l_\mathrm{f}}\tan(\delta)\right)
\end{aligned} \tag{A.2}$$

where x and y represent the longitudinal and lateral positions of the vehicle, v is the velocity, ψ is the heading angle, and α is the body slip angle. The parameters l_r and l_f represent the distances to the rear and front axles, respectively, and δ is the steering angle. a is the longitudinal acceleration.

Dynamic Bicycle Model, Linear Tire Model

The system is described by the following equations:

$$\begin{aligned}
\dot{x} &= v_x\cos(\psi) - v_y\sin(\psi) \\
\dot{y} &= v_x\sin(\psi) + v_y\cos(\psi) \\
\dot{\psi} &= r \\
\dot{v}_x &= \frac{-F_{yf}\sin(\delta) + F_{xf}\cos(\delta)}{m} + r \cdot v_y \\
\dot{v}_y &= \frac{F_{yf}\cos(\delta) + F_{xf}\sin(\delta) + F_{yr}}{m} - r \cdot v_x \\
\dot{r} &= \frac{l_f F_{yf}\cos(\delta) + l_f F_{xf}\sin(\delta) - l_r F_{yr}}{I_{zz}} \\
\dot{F}_{xf} &= m\dot{a}_x \\
\dot{\delta} &= \dot{\delta}
\end{aligned} \tag{A.3}$$

where v_x and v_y are the longitudinal and lateral velocities, respectively, r is the heading rate, F_{yf} and F_{xf} are the lateral and longitudinal tire forces at the front wheels, F_{yr} is the lateral tire force at the rear wheels, m is the vehicle mass, and I_{zz} is the yaw moment of inertia. $\dot{a}_x$ is the longitudinal jerk and $\dot{\delta}$ is the steering rate.

Sideslip angles are given by:

$$\alpha_f = \tan^{-1}\left(\frac{v_y + l_f r}{v_x}\right) - \delta \tag{A.4}$$

$$\alpha_r = \tan^{-1}\left(\frac{v_y - l_r r}{v_x}\right) \tag{A.5}$$

The lateral tire forces are given by the linear tire model:

$$F_y = -C_\alpha * \alpha \tag{A.6}$$

where C_α is the tire stiffness coefficient and α is the slip angle.

Dynamic Bicycle Model, Fiala Tire Model

The system is described by the following equations:

$$\begin{aligned}
\dot{x} &= v_x \cos(\psi) - v_y \sin(\psi) \\
\dot{y} &= v_x \sin(\psi) + v_y \cos(\psi) \\
\dot{\psi} &= r \\
\dot{v}_x &= \frac{-F_{yf} \sin(\delta) + F_{xf} \cos(\delta)}{m} + r \cdot v_y \\
\dot{v}_y &= \frac{F_{yf} \cos(\delta) + F_{xf} \sin(\delta) + F_{yr}}{m} - r \cdot v_x \\
\dot{r} &= \frac{l_f F_{yf} \cos(\delta) + l_f F_{xf} \sin(\delta) - l_r F_{yr}}{I_{zz}} \\
\dot{F}_{xf} &= m \dot{a}_x \\
\dot{\delta} &= \dot{\delta}
\end{aligned} \tag{A.7}$$

where v_x and v_y are the longitudinal and lateral velocities, respectively, r is the heading rate, F_{yf} and F_{xf} are the lateral and longitudinal tire forces at the front wheels, F_{yr} is the lateral tire force at the rear wheels, m is the vehicle mass, and I_{zz} is the yaw moment of inertia. $\dot{a}_x$ is the longitudinal jerk and $\dot{\delta}$ is the steering rate.

Sideslip angles are given by:

$$\alpha_f = \tan^{-1}\left(\frac{v_y + l_f r}{v_x}\right) - \delta \tag{A.8}$$

$$\alpha_r = \tan^{-1}\left(\frac{v_y - l_r r}{v_x}\right) \tag{A.9}$$

The maximum lateral tire force is given by:

$$F_{y,\max} = \sqrt{\left(\mu F_z\right)^2 - F_x^2} \tag{A.10}$$

The lateral tire force is given by:

$$F_y = \begin{cases} -C_\alpha \tan\alpha + \frac{C_\alpha^2}{3F_{y,\max}} |\tan\alpha| \tan\alpha - \frac{C_\alpha^3}{27\left(F_{y,\max}\right)^2} \tan^3\alpha, & |\alpha| < \tan^{-1}\left(\frac{3F_{y,\max}}{C_\alpha}\right) \\ -F_{y,\max} \operatorname{sgn}\alpha, & \text{otherwise} \end{cases} \tag{A.11}$$

where μ is the tire-road friction coefficient, F_z is the vertical load on the tire, F_x is the longitudinal force, and $F_{y,\max}$ is the maximum lateral tire force.

The vertical forces at the front and rear tires are given by:

$$\begin{aligned}
F_{zf} &= \frac{1}{L}\left(mgl_r - h_{CoG} F_{xf}\right) \\
F_{zr} &= \frac{1}{L}\left(mgl_f + h_{CoG} F_{xf}\right)
\end{aligned} \tag{A.12}$$

where h_{CoG} is the height of the center of gravity, L is the total wheelbase length, and F_{xf} is the longitudinal force at the front wheels.