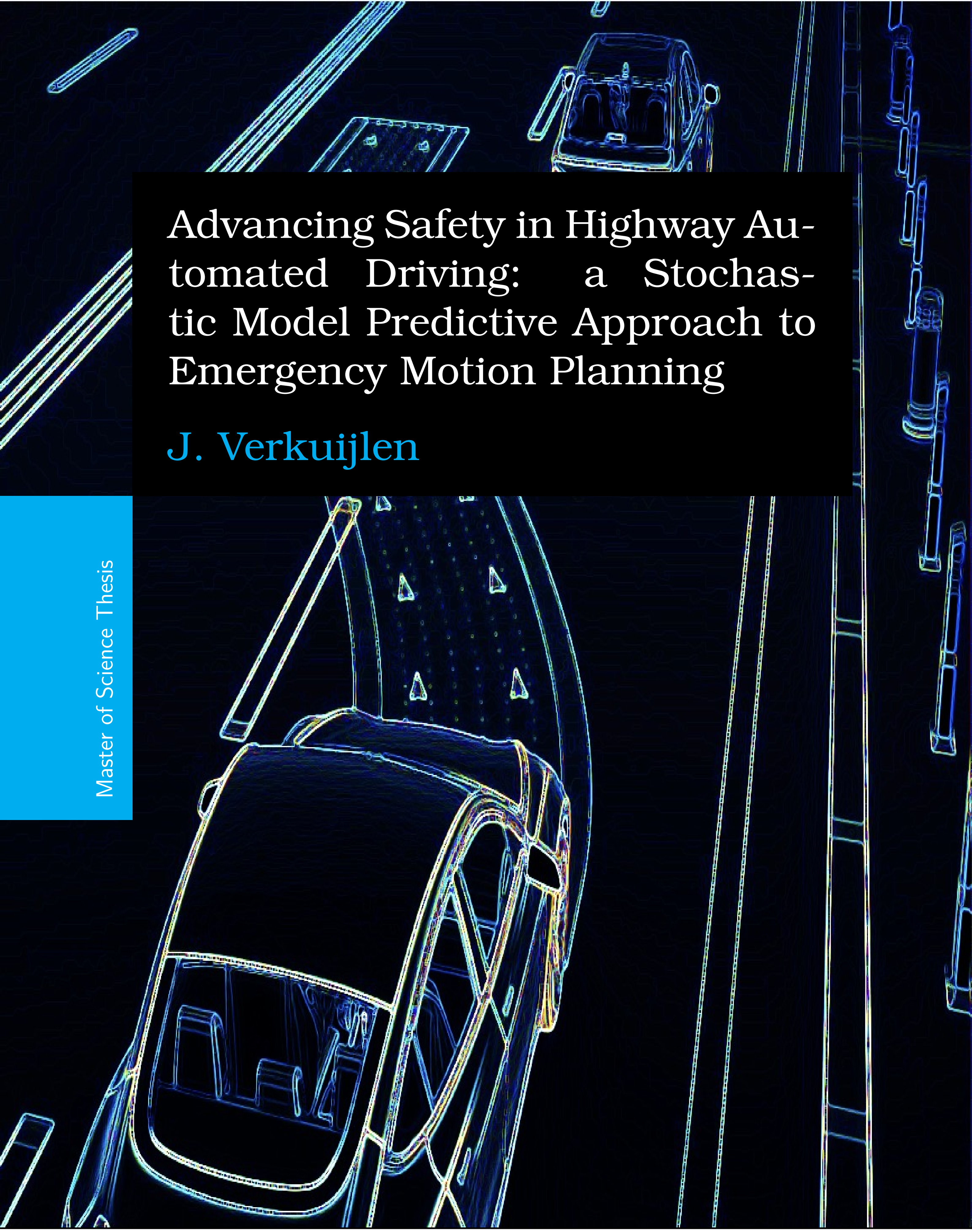


Advancing Safety in Highway Automated Driving: a Stochastic Model Predictive Approach to Emergency Motion Planning

J. Verkuijlen

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Advancing Safety in Highway Automated Driving: a Stochastic Model Predictive Approach to Emergency Motion Planning

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

J. Verkuijlen

April 10, 2024

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

This study addresses a critical aspect of automated driving: enhancing safety during highway emergency scenarios. This is achieved by formulating the problem within the framework of Stochastic Model Predictive Control (SMPC). In SMPC, safety constraints are designed such that a small chance of constraint violation is accepted, contrary to conventional Model Predictive Control (MPC), where hard constraints are imposed on the optimization problem that may never be violated. In the state-of-the-art, the exploration of SMPC for motion planning is minimal and focuses mostly on optimistic motion planning, depending on conservative backup solutions in emergencies. This research contributes to the state-of-the-art by designing a collision-avoiding, risk-averse, emergency motion planner by formulating the problem as SMPC, acknowledging the stochastic nature of real-world driving environments. By formulating the emergency planning problem specifically as SMPC, including risk minimization and providing a reference lane where the collision risk is minimal, the motion planner successfully navigates the automated vehicle through complex, dynamic emergency scenarios involving both static and dynamic obstacles without collision. In addition, the Interactive Multiple Model Algorithm (IMM) is used to estimate the maneuver intention of other vehicles (OVs). These maneuver estimates are used to formulate the probabilistic constraints in SMPC.

Furthermore, two novel backup controllers are designed to compute a fast, collision-avoiding trajectory when SMPC is infeasible. From the results, the backup controller based on MPC, designed such that it is always feasible and provides support in finding a feasible SMPC solution in the next iteration, showed the best performance.

The success of the proposed emergency motion planner is presented through simulation in 6 complex scenarios, where the planner safely navigates the automated vehicle through the emergencies without collision. The results are compared to related state-of-the-art, showing its effectiveness.

Contents

Preface **iv**

1 Introduction **1**

1-1 Motivation . . . 1

1-2 Challenges in Emergency Motion Planning . . . 2

1-3 State-of-the-art . . . 4

1-4 Thesis Contribution . . . 5

2 Motion Prediction of Other Vehicles **7**

2-1 Maneuver intention estimation . . . 10

2-1-1 Intention Estimation Methods . . . 10

2-1-2 Interactive Multiple Model Algorithm . . . 11

2-2 Trajectory Prediction and Uncertainty Propagation . . . 20

2-2-1 OV Trajectory Prediction . . . 20

2-2-2 Uncertainty Propagation . . . 21

3 Planning Architecture **23**

4 Stochastic Model Predictive Motion Planning **25**

4-1 Motion Planning Algorithms . . . 25

4-2 Mathematical Formulation of the Planning Optimization Problem . . . 28

4-3 Ego Vehicle Model . . . 29

4-3-1 Model Selection . . . 29

4-3-2 Single-track Bicycle model . . . 29

4-4 Cost Function Design . . . 32

4-5 Reference Trajectory Generation . . . 33

4-6 Safety Constraints Definition . . . 35

5 Backup Motion Planner **41**

5-1 Backup A: MPC based 42

5-2 Backup B: SMPC based 43

5-3 Comparison Backup Controllers 46

6 Simulation and Results **48**

6-1 Defined Emergency Scenarios 49

6-2 SMPC Performance 50

6-2-1 Scenario 1: single decelerating obstacle 50

6-2-2 Scenario 2: single static obstacle 53

6-2-3 Scenario 3: ambiguous OV behavior 54

6-2-4 Scenario 4: decelerating OV1 + overtaking OV2 54

6-2-5 Scenario 5: dynamic OV1 + static OV2 55

6-2-6 Scenario 6: two static OVs 56

6-3 Backup Controller Performance and Comparison 56

6-3-1 Backup A: MPC 56

6-3-2 Backup B: SMPC 57

6-3-3 Comparison to State-of-the-art 59

7 Discussion **61**

8 Conclusion **63**

A EV Bicycle Model **65**

Preface

When I began my Bachelor's in Mechanical Engineering at the University of Technology in Delft, my aspiration was clear: to acquire a comprehensive understanding of the field, paving my way towards a career in the automotive industry upon graduation. As I approach the conclusion of my academic journey, I find myself immersed in a project centered on automated driving, marking the final phase of my studies.

Throughout the past year, I have delved into this very interesting project: enhancing safety in automated driving by designing an emergency motion planner. Ever since the start of the project, I spent my car rides wondering what happened in my brain while driving, and thinking about how to translate this into an algorithm for motion planning. Turns out our brain is somewhat complex, and it is quite a challenge to copy-paste that into Matlab.

During the project, there were times when I felt lost, unsure of the next steps, and questioned whether I would ever find my way. However, I have come to realize that it was precisely these moments of introspection and doubt that ultimately guided me toward constructing a well-considered solution to the problem at hand.

Today, as I reflect on what this project has become, I am filled with a sense of pride.

I would like to express my gratitude to those who have contributed to the completion of this project. First, I would like to extend my sincere appreciation to Dr. Azita Dabiri for her supervision throughout this project. Her guidance and feedback were valuable in shaping the direction of my research.

Additionally, I am profoundly thankful to Leila Gharavi, PhD candidate, whose extensive support and assistance significantly enhanced the quality of this work. I deeply appreciate her high level of engagement in the project, and her positive reinforcement. It was reassuring to know that I could always reach out to her with any problems or questions, and her willingness to assist was very valuable to me.

Delft, University of Technology
April 10, 2024

J. Verkuijlen

Chapter 1

Introduction

1-1 Motivation

Autonomous driving is an extensively invested research area with many promising aspects. Besides improving mobility, the introduction of self-driving cars can significantly reduce the number of traffic accidents, of which 94% are now caused by human errors [1]. While human drivers can get tired or distracted, autonomous vehicles have the ability to constantly keep track of the environment and to react immediately to a hazardous situation to prevent collision. Car accidents have a wide range of negative effects, not only on the physical, mental, and financial health of the passengers but also on traffic flow. Accidents can lead to congestion, which is undesirable as it disrupts the daily lives of traffic participants and contributes to an increase in greenhouse gas emissions.

Even though self-driving cars have the potential to improve safety significantly compared to human-driven cars, public acceptance is still hard to obtain. Self-driving cars should be at least four times safer than human-driven cars in order for the risks to be tolerated by society [2]. The need for highly reliable self-driving cars that are able to prevent the majority of accidents stresses the relevance of developing robust controllers for motion planning in emergency situations.

Emergency situations could for example be caused by a suddenly appearing animal on the road or an unexpected maneuver of a surrounding vehicle. To avoid the obstacle, the motion planner has to react fast and compute a safe trajectory. These kinds of emergency maneuvers might require the vehicle to drive at its handling limits for a short period of time, which has nonlinear dynamics [3]. Besides, the vehicle operates in an environment that can present a range of challenges like unpredictable behavior of surrounding vehicles or sudden appearing obstacles. The combination of the short time available for computation and the highly nonlinear and unpredictable vehicle model in an uncertain, dynamic environment makes the autonomous driving problem for evasive maneuvers in emergencies a complicated task. Although many control strategies are proposed, emergency trajectory planning is still a challenging control problem that is very relevant for the acceptance of autonomous vehicles [4].

1-2 Challenges in Emergency Motion Planning

Figure 1-1 shows an overview of automated driving by highlighting its interdependent layers. It illustrates the sequential process of global route planning through local motion planning to trajectory tracking. First, a route from the starting point to the final destination of the vehicle is generated by a global route planner. Based on information like the final destination, available roads, and traffic conditions, the global planner generates a set of directions outlining the specific roads the vehicle must take to reach its intended endpoint. The details of this global planner are beyond the scope of this review since emergency situations are unforeseen, and cannot be accounted for by the high-level global planner.

At a lower level, a motion planner generates a path and velocity profile that the vehicle has to follow to achieve various goals like avoiding collision, obeying traffic rules, getting closer to its final destination, and maintaining comfort. While the output of the global planner outlines the general route that the vehicle will take throughout its entire journey, the local planner generates a motion reference for the short time horizon in the immediate future, emphasizing its crucial role in collision avoidance.

In the last layer, the tracking controller computes optimal control inputs for the automated vehicle, aiming to minimize the error between the optimal trajectory determined by the planner and the actual state of the vehicle. In the existing literature, various authors propose a solution to automated driving by integrating local motion planning and trajectory tracking to account for the tracking feasibility of the optimal trajectories computed by the planner [5]. In this research, however, the integration of a planner and tracker is not explored, as the focus remains on collision avoidance in an uncertain, dynamic environment. Hence, this research is centered around local motion planning.

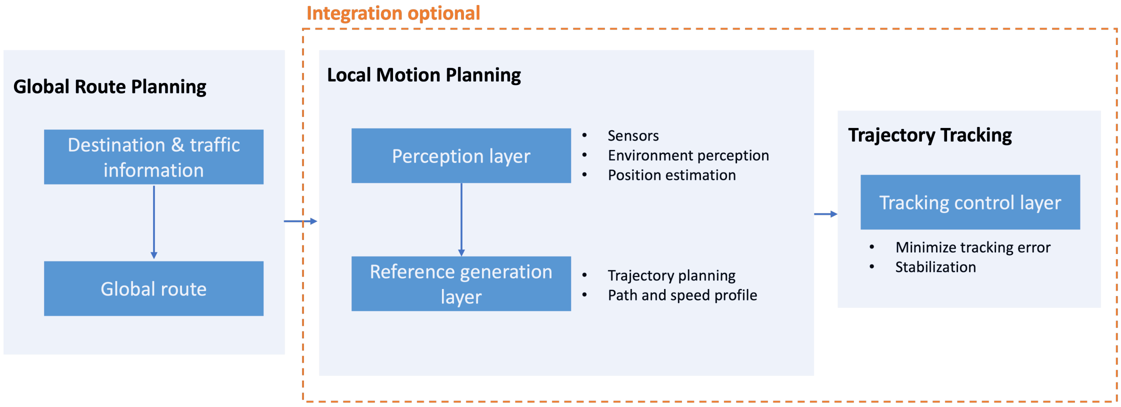


Figure 1-1: Hierarchical overview of automated driving - illustrating the sequential process of global planning through local planning to trajectory tracking. Created by Author.

This research aims to enhance safety in automated driving by improving emergency motion planning. For clarification, the definition of an emergency is given.

Definition 1-2.1 (Emergency)**.** In the context of automated driving, an emergency on the highway refers to any sudden, unexpected event or scenario that poses an immediate threat to the safety of the vehicle and its occupants, as well as other road users. The EV is required to perform an aggressive, obstacle-avoiding maneuver.

Aggressive maneuvers may require high lateral and longitudinal accelerations, which in turn, push the vehicle to its stability limits. Emergencies might be caused by the sudden appearance of obstacles on the road or radical behavior of Other Vehicles (OVs) like unexpected braking or lane changes. In these scenarios, other priorities apply than during general motion planning; for instance, collision avoidance (i.e. safety) should be preferred over other objectives such as passenger comfort or obeying traffic rules.

This research focuses on **local motion planning** in highway emergency scenarios. The complexity of this problem involves four different aspects: the uncertainty caused by the presence of other, human-driven vehicles on the road, the nonlinearity of the ego vehicle dynamics at its handling limits, conflicting objectives such as optimizing vehicle stability while ensuring safety, and lastly, the need for fast computations in real-time due to the limited window of opportunity to avoid a possible collision in highway emergency scenarios. Below, these four aspects are explained together with their respective requirements for the motion planner.

1. The assessment of trajectory optimality extends beyond safety considerations, it includes **various objectives, often conflicting** with collision-avoidance and each other. Examples of these objectives are following the center of a lane on the highway, operating strictly in the stability region of the EV, staying within road boundaries, and keeping the trajectory comfortable for passengers. When prioritizing collision avoidance, the planner might compute tightly uncomfortable trajectories that push the EV beyond its stability regions. On the other hand, when comfort is prioritized, the planner might fail to find collision-free trajectories in emergencies. Therefore, the planner should be able to carefully balance these conflicting objectives when computing trajectories.

2. The second challenge in designing an emergency motion planner is caused by the **nonlinear and unknown vehicle dynamics of the EV** [6]. The complexity arises from the unpredictable dynamic behavior of the vehicle, with its handling limits representing the boundaries within which stable motion is maintained. These handling limits are difficult to predict due to the increasing nonlinearity of the dynamics. In emergency situations where collision-avoiding maneuvers require high longitudinal and lateral accelerations, the EV might be pushed to its handling limits. In such scenarios, differences between the predicted and actual vehicle models could lead the planner to compute trajectories that exceed the vehicle's handling limits, risking instability. Conversely, underestimating vehicle dynamics may result in overly conservative behavior from the motion planner, potentially hindering its ability to generate collision-free trajectories and fully utilize the EV's capabilities.

3. To safely navigate the automated vehicle in an environment with other road users, **environmental uncertainty has to be considered**. It is assumed the uncertainty in the future motion of OVs is two-fold: on the one hand, the maneuver intention of the driver is unknown while on the other hand, it is uncertain how this maneuver is executed [7]. Therefore, the optimal control problem becomes stochastic and the planner must be able to handle probabilistic information from OVs. Moreover, a motion prediction method should be leveraged to predict OV states with related uncertainty.

4. There is one constraint that conflicts with almost all other objectives: the amount of **available time for computing the optimal trajectory.** The planner has to react quickly to the suddenly appearing obstacle to avoid a collision in emergencies. The limited computational time available in emergency scenarios, which can be as short as 10 ms [5], imposes a constraint on the level of complexity that can be accommodated by the optimal control problem. To find a balance between achieving an accurate optimal trajectory and minimizing computational time, a trade-off must be carefully considered.

These four requirements are used as key points to assess the available literature on motion planning, of which an overview is given in Section 4-1.

1-3 State-of-the-art

While a wide variety of solutions for emergency motion planning is proposed in the literature [5], MPC has proven to be a suitable control structure that meets all specified requirements when properly designed. MPC iteratively solves a Control Optimization Problem (COP) over a finite prediction horizon every time step. From the computed optimal input sequence over the prediction horizon, only the first input is applied to the system. Every time a new measurement becomes available, this process is repeated. Solving the COP involves minimizing a cost function while satisfying constraints. Within the cost function, objectives often include minimizing deviations from a reference trajectory and keeping inputs and input rates small. With constraints, boundaries can be put on the states of the automated vehicle due to road boundaries and, particularly relevant in emergency motion planning, OVs.

The state-of-the-art shows that MPC can be successfully implemented for motion planning, which is already widely explored [8, 9, 10, 11, 12]. However, to the best of our knowledge, fewer solutions are explored that define MPC with probabilistic chance constraints, as in SMPC.

Definition 1-3.1 (Chance constraints)**.** Chance constraints refer to probabilistic constraints that limit the maximum probability that a constraint is violated. Thereby allowing a small probability that constraints will not be satisfied [13].

While MPC imposes hard constraints on the vehicles' states, in SMPC the constraints are formulated as chance constraints. The probability of constraint violation is a design parameter and is defined by the safety factor $\beta \in (0, 1)$. For instance, with a safety factor of $\beta = 0.8$, the maximum probability a constraint is violated is $1 - \beta = 0.2$. Defining chance constraints in SMPC can significantly improve performance when driving in uncertain, dynamic environments where OVs with unpredictable future motion surround the automated vehicle. State-of-the-art solutions include the research by Brüdigam et al. [14], who combine SMPC with a reachability analysis-based Fail-Safe Trajectory planner (FTP) for conservative backup trajectories in emergencies. Benciolini et al. [15] propose a framework to reduce conservatism in urban trajectory planning by estimating the maneuver intentions of other road users with the IMM algorithm, and designing constraints accordingly within SMPC. This IMM algorithm can be understood as an extension of the Kalman Filter and provides a structure to not only estimate the OV state but also the model used to update the state at that time step. A similar approach is exploited by Carvalho et al. [16], who combine IMM with SMPC for urban automated driving.

Some promising designs of SMPC for motion planning have been proposed in the literature. However, the formulation of chance constraints with SMPC is mostly utilized for optimistic motion planning, since allowing a small probability of constraint violation leads to less conservative behavior of the automated vehicle.

Definition 1-3.2 (Conservatism)**.** Conservatism refers to a risk-averse approach where the system tends to be overly cautious, often leading to suboptimal or overly constrained trajectories.

In emergencies, collision avoidance should be prioritized compared to optimistic planning. For that reason, [14] combines SMPC with FTP as a backup controller to account for emergencies. However, FTP, which relies on worst-case reachability analysis, may fail and run into infeasibility when dealing with late-detected or static obstacles. Since trajectory planning with SMPC provides optimistic trajectories with relaxed constraints, in emergencies, if SMPC cannot find a path satisfying all constraints, imposing even tighter constraints with FTP would rarely result in feasible solutions. In the case of infeasible FTP, the authors suggest using the shifted input sequence computed in the previous iteration, thereby ignoring the newest measurement. Through these limitations, a gap in the existing literature can be recognized.

1-4 Thesis Contribution

This research makes distinctive contributions to the existing literature, by proposing a framework to compute safe trajectories in emergencies by solving the planning problem with SMPC. With SMPC, trajectories with the lowest chance of collision can be computed effectively, even in scenarios involving static or late-detected obstacles. Furthermore, by designing chance constraints in SMPC, collision avoidance can be guaranteed with a probability level β, considering the uncertain behavior of OVs. By incorporating SMPC into emergency motion planning strategies, the chance of collision can be actively minimized, filling a gap in the existing literature where such an approach has not been explored. This approach aligns with the fundamental goal of ensuring safety in hazardous scenarios.

To ensure the computation of a collision risk-minimizing trajectory at all time steps, a fast, reliable backup controller is designed if SMPC fails to compute a solution that satisfies all constraints within the assigned maximum time. The designed backup controller shows a significant improvement compared to the state-of-the-art: while [14] tightens constraints in emergencies, resulting in unfeasible backup solutions, our proposed backup controller is always feasible and, moreover, significantly faster. Our proposed backup controller does not explicitly guarantee collision avoidance at all time steps where [14] does provide such a guarantee. However, they rely on unrealistic assumptions of OV behavior, therefore raising concerns regarding their practical applicability. In real-life emergencies, it is impossible to always guarantee safety since situations might occur in which a collision-avoiding trajectory does not exist, for instance, when a sudden and unexpected obstacle, such as a fallen tree or debris from a previous accident, obstructs the roadway with insufficient time for vehicles to react.

These unpredictable events show that fail-safe collision avoidance strategies are ineffective, underscoring the need for our proposed backup system, which prioritizes adaptability and real-time responsiveness over adherence to theoretical constraints.

Furthermore, this research introduces a contribution by incorporating a 'helping reference trajectory' into SMPC, where deviations from a predefined reference trajectory are penalized in the cost function. This innovative approach leverages threat assessment, specifically identifying the safest possible lane on the highway, thereby helping SMPC to find collision-free trajectories. To account for the two-fold uncertainty in future OV motion, the IMM algorithm is used for highway maneuver intention estimation, inspired by [15]. Lastly, this research introduces an innovative approach to managing chance constraints within SMPC, offering flexibility in determining the desired level of certainty for constraint satisfaction by tuning the safety factor β. Uniquely, to prioritize constraints in the earlier time steps of the prediction horizon, the safety factor β is dynamically adjusted as a variable, gradually decreasing over the prediction horizon.

The contributions of this work are summarized as follows:

- SMPC specifically designed for evasive maneuvers in emergencies concerning both static and dynamic obstacles, leveraging the IMM algorithm for maneuver prediction of OVs. This approach actively reduces the likelihood of collision by minimizing it within the cost function, ensuring collision avoidance with a safety factor β.
- A novel design combining SMPC with a fast, real-time responsive backup controller to ensure a collision risk-minimizing solution based on the most recent available measurements. The backup controller is only applied when SMPC fails to find a solution satisfying all specified constraints within the maximum computational time.

The remainder of the thesis is built up as follows. First, it explains how the future motion of OVs is predicted utilizing the IMM algorithm for maneuver intention estimation. Then, in Chapter 3 the planning architecture is explained, highlighting the coordination between IMM, SMPC, and the backup controller. The design choices of SMPC are presented and discussed in Chapter 4. In Chapter 5, two proposed backup controllers are presented, and compared with the state-of-the-art. Also, conclusions are drawn on the hypothetical performance of the backup controller. Chapter 6, shows results that indicate the effective performance of SMPC and the backup controllers, compared to the state-of-the-art. The results are discussed and future work is suggested in Chapter 7. Finally, the research is concluded in Chapter 8.

Chapter 2

Motion Prediction of Other Vehicles

In the rapidly advancing field of autonomous driving, one of the pivotal challenges lies in ensuring the safe and efficient integration of the ego vehicle into complex and dynamic environments, particularly on highways. A fundamental aspect of achieving this integration is the accurate anticipation of the future motions of surrounding vehicles. The ability to predict the trajectories and behaviors of these vehicles is crucial for an automated system to make informed decisions, proactively respond to potential hazards, and safely navigate through traffic.

The objective is to compute a sequence of OV states $\boldsymbol{\tilde{z}^j}$ over the prediction horizon N_p at every timestep k:

$$\boldsymbol{\tilde{z}_k} = \left[\tilde{z}_k \cdots \tilde{z}_{k+N_p}\right] \tag{2-1}$$

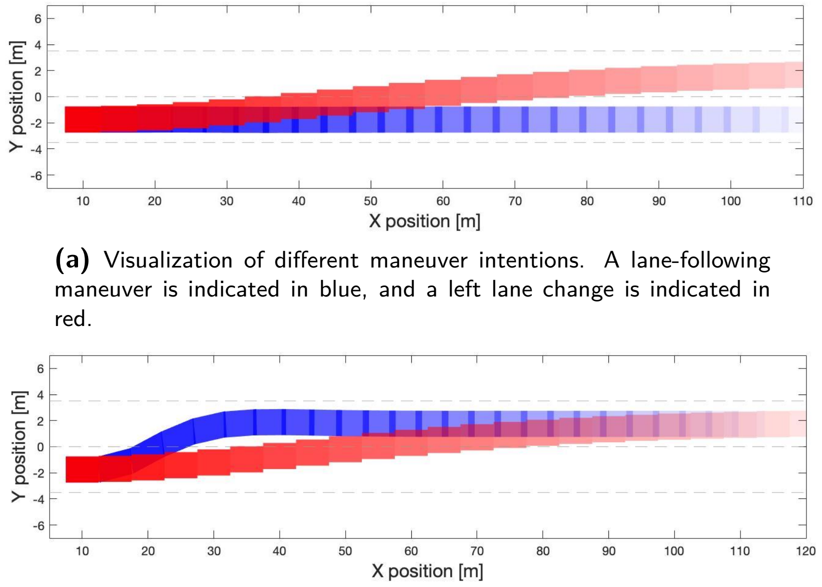
The available information is a noisy measurement γ of the $x-$ and $y-$ position of the OV at every timestep k.

$$\gamma = \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \tag{2-2}$$

In this research, the uncertainty in driver behavior is modeled upon the assumption that this uncertainty is twofold, building upon the approach of [16], in which a framework integrating environment modeling and SMPC is proposed. In the field of autonomous driving, this assumption about twofold human-driver uncertainty is frequently incorporated and implemented [7, 17]. The two contributors to the uncertainty are listed below.

- On the one hand, the high level **maneuver intention** of the driver is unknown. Maneuvers are defined as a sequence of actions or behaviors, which could e.g. represent a left or right lane change, or taking a turn at an intersection. In Figure 2-1a, differences in maneuver intentions are visualized. While the red vehicle keeps its lane, the blue vehicle performs a left lane change maneuver.

- On the other hand, uncertainty in future OV trajectories is caused by variability inherent in the **execution of maneuvers**. A driver can for example aggressively change a lane (indicated in blue in Figure 2-1b) or adopt a more gradual approach to execute a smooth maneuver (the red vehicle in Figure 2-1b). For this form of uncertainty, the propagation of OV states with their covariance plays a crucial role.



(a) Visualization of different maneuver intentions. A lane-following maneuver is indicated in blue, and a left lane change is indicated in red.

(b) Visualization of different maneuver executions. The red vehicle performs a smooth lane change contrary to the vehicle indicated in blue.

Figure 2-1: Visualization of Uncertainty in Maneuver Intentions and Maneuver Executions. Grey dashed lines indicate road boundaries. Created by author.

To be able to find the probability of maneuver intentions and make an estimation of the OV trajectories, all OVs are modeled with a dynamic model. The goal of this vehicle model is to give a good representation of the possible intentions and executions of maneuvers. Therefore, the linear point-mass model already contains sufficient information on the dynamics to estimate the future OV states. Implementing a higher fidelity model could be beneficial because the heading angle of the OVs would also be known. However, the added computational complexity associated with a higher fidelity model would add unnecessary computational burden to the optimal control problem.

The state matrices of the OVs consist of the positions and velocities in x- and y-direction: $[s_x, v_x, s_y, v_y]^T$. The x-direction corresponds to the longitudinal, road aligned, direction, while the y-direction corresponds to the lateral direction. Figure 2-2 shows a graphic representation of the kinematic point-mass model. The orange line is road-aligned, and it is assumed the curvature on the highway is negligible. In the figure, s_x represents the longitudinal position of the OV along the road, while s_y indicates the lateral position of the OV perpendicular to the road. The velocity vector v consists of a component in x-direction and y-direction, $v_x = v\cos(\phi)$ and $v_y = v\sin(\phi)$.

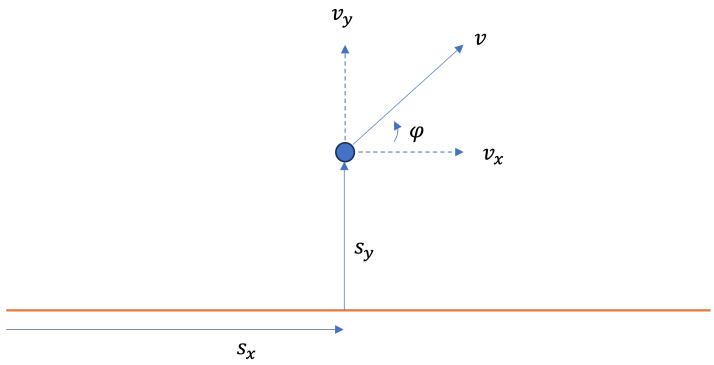


Figure 2-2: Scheme of kinematic point mass model in road aligned coordinates. Created by author.

The input is defined by acceleration in both x- and y-direction: $u = [a_x, a_y]^T$. In Equation 2-3 the state space representation of the open loop point-mass model is given.

$$\begin{aligned} z_{k+1} &= Az_k + Bu + \omega \\ \gamma &= Hz_k + \nu \end{aligned} \tag{2-3}$$

The state matrices A and B arise from basic physics, and they both depend on the sampling time h. Furthermore, it is assumed that at every time step, the x- and y- positions of the OV, subject to perturbation by sensor noise, are available through measurements $\gamma \in \mathcal{R}^2$. The matrices A, B H are given in Equation 2-4.

$$A = \begin{bmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5h^2 & 0 \\ h & 0 \\ 0 & 0.5h^2 \\ 0 & h \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2-4}$$

ω is the process noise, caused by the uncertainty in driver behavior, and is defined by $\omega \sim \mathcal{N}(0, \Sigma_\omega)$. The diagonal of the covariance matrix contains the variance in the OV x- and y-position. ν is the sensor noise and is defined by $\nu \sim \mathcal{N}(0, \Sigma_\nu)$, with Σ_ν diagonal. Furthermore, ω and ν are independent, which is an assumption that should hold when using Kalman Filtering in IMM:

$$\mathbb{E}\left[(z_k - \hat{z}_{k|k-1})[\nu^T \; \omega^T]^T\right] = 0 \tag{2-5}$$

The remainder of this chapter is built up as follows: it is divided into two sections, elaborating on the two contributors to the stochasticity of the problem: maneuver intention, and maneuver execution uncertainty. Section 2-1 explains how the IMM algorithm is implemented to estimate the OV state and to compute the probabilities of the OV executing certain highway lane-following maneuvers. Section 2-2 elaborates on the propagation of the OV state and its uncertainty over the prediction horizon. It explains how the OV trajectories are computed based on the state estimation by IMM, and how the estimation uncertainty increases with further steps in the prediction horizon. For clarification, a distinction is made between the terms estimation and prediction:

Definition 2-0.1 (Estimation)**.** Estimation refers to the process of determining or approximating parameters at the current time step. In the context of this chapter, the current vehicle state and the maneuver probabilities are estimated.

Definition 2-0.2 (Prediction)**.** Prediction involves forecasting or anticipating future outcomes, events, or states based on current information. In the context of this chapter, it refers to predicting the future states of OVs over the prediction horizon.

2-1 Maneuver intention estimation

2-1-1 Intention Estimation Methods

(S)MPC operates in a receding horizon fashion, where it continually optimizes control inputs over a finite time horizon called the prediction horizon. Based on OV measurements obtained at the current time step k, predictions have to be made of their future positions over the prediction horizon for the SMPC to plan a collision-free trajectory. By estimating the maneuver intention of a driver, the possibilities of future OV states are narrowed down, allowing for more focused and efficient anticipation of the vehicle's actions. For instance, if the probability of a left lane change is significantly large, concerns about potential maneuvers in the right lane can be alleviated.

In the literature, various solutions to the challenge of intention estimation of OVs can be found. In the review paper by Lefèvre et al. [18], the concept of maneuver-based estimation is explored, and one approach involves leveraging prototype trajectories. The fundamental idea behind this method is to categorize trajectories into different clusters, each representing a specific maneuver. When given a partial OV trajectory, the estimation process involves determining the trajectory cluster to which it best aligns. However, due to the need for vehicle data, this approach is not used for this research. The same limitation applies to learning-based maneuver estimation methods like support vector machines (SVM) [19].

Scenario-based model predictive control (SCMPC) is an analytical method considering intention uncertainty proposed by Schildbach et al. [20] and leveraged by for example Brüdigam et al. [7]. In SCMPC, the maneuver of the OV driver is estimated by drawing samples from the probability distribution P_T describing the distribution of maneuver probabilities. Each sample represents a maneuver, and from this sampled set the most probable maneuver is estimated and used over the prediction horizon.

Another similar method is combining MPC with a maneuver prediction algorithm called the IMM Algorithm. This novel approach is proposed in [15]. The algorithm returns a state estimate of the OV together with the probabilities μ_j that it is currently executing maneuver j for all n_m maneuvers: $\{\mu_j\}_{j=1}^{j=n_m}$. Where scenario-based MPC only considers the most probable scenario, IMM provides a maneuver probability distribution at every time step instead of removing scenarios, which is a practical characteristic in emergencies where collision avoidance is prioritized. Therefore, the IMM algorithm is leveraged for this collision-avoiding motion planner. A detailed description of the IMM algorithm is given in the next section.

2-1-2 Interactive Multiple Model Algorithm

Overview of the key concepts of IMM

The IMM Algorithm is proposed in [21], and provides a structure for state estimation of objects with changing dynamic models. It can be understood as an extension of the well-known Kalman Filter [22]. In addition to estimating the OV state and covariance from noisy sensor data at the current time step k, it returns an estimate of the probability that the state is updated according to the dynamic model j. The algorithm is used for moving target tracking by for example [23], which provides a structure track maneuvering targets by fusing sensor data. They implemented two distinct dynamic models, a constant velocity model and a constant turn rate model, between which the target dynamics switch over time. The use of these two models for target tracking with IMM is seen more often, an example is [24], where the authors aim to track human motion leveraging the constant velocity and constant turn rate models in IMM. In [15] dynamic obstacles are tracked in an urban environment, with each dynamic model representing a traffic participant like a cyclist or pedestrian. In Algorithm 1, an overview of the mathematical expression of IMM is given. In the subsequent part of this section, each element will be explained in more detail.

Algorithm 1 Interactive Multiple Model Algorithm

```
1: Input: $\{\mu_j\}_{j=1}^{j=n_m}$, $\{\hat{z}_j\}_{j=1}^{j=n_m}$, $\{\hat{P}_j\}_{j=1}^{j=n_m}$, $\gamma$, $\Pi$
2: State Interaction
3: $c_{0j} = \sum_{i=1}^{n_m} \mathbf{\Pi}_{ij}\hat{\mu}_i$
4: $\mu_{i|j} = \Pi_{ij}\hat{\mu}_i(c_j)^{-1}$
5: $z_{0j} = \sum_{i=1}^{n_m} \mu_{i|j}\hat{z}_i$
6: $P_{oj} = \sum_{i=1}^{n_m} \mu_{i|j}[\hat{P}_i + (\hat{z}_i - z_{0j})(\hat{z}_i - z_{0j})^T]$
7: Kalman Filtering
8: $\tilde{z}_j = F_j z_{0j} + G_j \eta_j$
9: $\tilde{P}_j = F_j P_{0j} F_j^T + \Sigma_\omega$
10: $\epsilon = \gamma - H\tilde{z}_j$
11: $S_j = H\tilde{P}_j H^T + \Sigma_\nu$
12: $L_j = \tilde{P}_j H^T (S_j)^{-1}$
13: $\hat{z}_j = \tilde{z}_j + L_j \epsilon_j$
14: $\hat{P}_j = (I - L_j H)\tilde{P}_j$
15: Model Probability Update
16: $\Lambda_j = \frac{\exp(-0.5\epsilon_j^T (S_j)^{-1}\epsilon_j)}{\sqrt{\det(2\pi S_j)}}$
17: $c = \sum_{i=1}^{n_m} \Lambda_i c_{0i}$
18: $\mu_j = \Lambda_j c_{0j} (c)^{-1}$
19: State Estimate Combination
20: $\hat{z} = \sum_{i=1}^{n_m} \hat{z}_i \hat{\mu}_i$
21: $\hat{P} = \sum_{i=1}^{n_m} \hat{\mu}_i[\hat{P}_i + (\hat{z}_i - \hat{z})(\hat{z}_i - \hat{z})^T]$
22: Output: $\{\mu_j\}_{j=1}^{j=n_m}$, $\hat{z}$, $\hat{P}$
```

A) In line 1 of the algorithm, the **input** is given. The first element consists of the model probabilities from the previous time step $\{\mu_j\}_{j=1}^{j=n_I}$. The number of predefined vehicle models is n_m, and for every vehicle model j, the number μ_j indicates the probability that the previous state was updated according to that predefined dynamic model. Next, $\{\hat{z}_j\}_{j=1}^{j=n_m}$ and $\{\hat{P}_j\}_{j=1}^{j=n_m}$ are the state and covariance respectively of the OV state if it would be maneuvering according to dynamic model j. These are both internal parameters and in the next steps, it will become evident how these parameters are obtained and how they influence the estimation algorithm. γ is the noisy measurement of the OV state. The last parameter is the state switching matrix Π. This is a design parameter that is determined a priori. It is a matrix in which every element Π_{ij} represents the probability of the dynamic obstacle to change from dynamic model i to dynamic model j. Important properties of this matrix include its size: $\Pi \in \mathbb{R}^{n_m x n_m}$ and the characteristic that the sum of every row equals one: $\sum_{j=1}^{j=n_m} \Pi_{ij} = 1 \ \forall i$.

B) The first step is the **state interaction**. In this step, the model states and covariances are mixed based on the state switching matrix Π and the model probabilities from the previous time step. In Figure 2-3, a schematic overview of the algorithm is given in the case of two distinct filter models.

Definition 2-1.1 (Filter models)**.** In the context of IMM, filter models refer to the mathematical representations or descriptions of the observed system. This model typically consists of equations that describe the dynamics of the system, including its state variables, inputs, outputs, and the relationships between them.

$\hat{z}_1$ and $\hat{z}_2$ are the state estimates from the previous cycle with model 1 and model 2 respectively. To include the probability of the object switching dynamics, these estimated states are updated using conditional probabilities obtained from Π and the previous model probabilities μ_1 and μ_2. The same is done for the covariances $\hat{P}_1$ and $\hat{P}_2$. The updated states and covariances are indicated as $\hat{z}_{01}$, $\hat{z}_{02}$, $\hat{P}_{01}$, and $\hat{P}_{02}$.

C) The updated states from the interaction are then filtered with a **Kalman Filter**, starting from step 7. For a more detailed explanation of the Kalman Filter, the reader is referred to [22]. For each model, the updated state is filtered by first performing the time update using the predefined dynamic models in steps 8 and 9. Then, γ is used for the measurement update. In the measurement update of the Kalman filter, the predicted state is adjusted based on the difference between the actual measurement and the predicted measurement. This adjustment, weighted by the Kalman gain, refines the state estimate by incorporating the new measurement information. The parameters $\hat{z}_1$, $\hat{z}_2$, $\hat{P}_1$ and $\hat{P}_2$ are the result of the filtering step with two models as shown in Figure 2-3. Note that during the filtering step, the time is updated and the estimates now apply to the current time step. Furthermore, it is worth mentioning that the outputs of the Kalman Filters are the inputs of the State Interaction step at the next cycle.

D) In steps 15 till 18, the **model probabilities are updated** using information from the Kalman Filtering step and the model probabilities from the previous cycle. The parameter Λ is designed such that the further away the predicted state, based on model j, is from the measurement, the lower the probability the state is updated according to that model. The outputs of this step are μ_1 and μ_2 in Figure 2-3.

E) In the final step, the new model probabilities and **state estimates are combined**. The higher the probability the object is moving with dynamic model j, the closer the final estimation $\hat{z}$ will be to the predicted state $\hat{z}_j$. The outputs are $\hat{z}$ and $\hat{P}$.

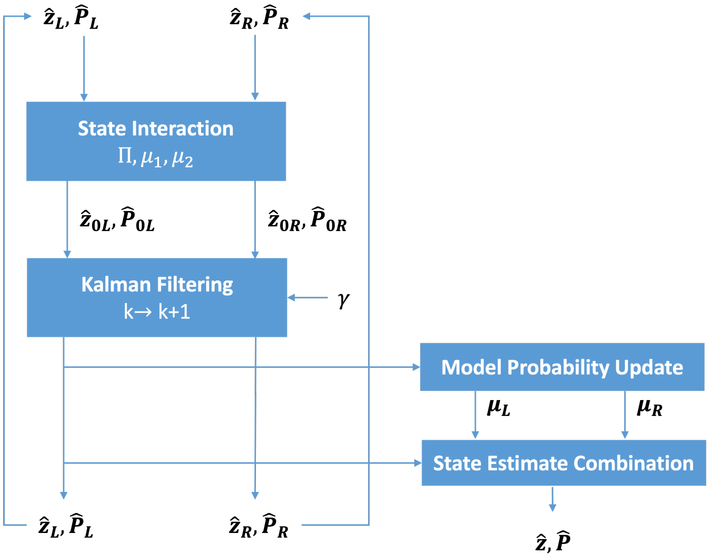


Figure 2-3: Block diagram of the Interactive Multiple Model Algorithm with two different filter models.

Highway maneuver definition and intention estimation by IMM

IMM is used to estimate the maneuver intentions of OVs. To serve this purpose, filter models are designed that represent highway maneuvers. For each lane center, a closed-loop lane-following model is designed. The size of the set of maneuvers is equal to the number of lanes on the highway, and the closed-loop reference tracking models are designed as follows.

The closed-loop reference tracking models contain feedback with gain K, to track different reference trajectories $\{\boldsymbol{z}_k^{*j}\}_{j=1}^{j=n_I}$, each corresponding with a predefined maneuver. Reference trajectories are defined as:

$$\boldsymbol{z}_k^{*j} = [z_k^{*j} \cdots z_{k+N_p}^{*j}] \tag{2-6}$$

with

$$z_k^{*j} = [s_{x,k}^{*j}, v_{x,k}^{*j}, s_{y,k}^{*j}, v_{y,k}^{*j}] \tag{2-7}$$

Implementing this feedback gain K and reference z_k^{*j}, the OV input at time step k becomes:

$$u_k = K(z_k - z_k^{*j}) \tag{2-8}$$

In Equation 2-8, z_k is the state at k of the OV and z_k^{*j} is the reference state corresponding with the j^{th} maneuver. The feedback gain K is obtained by solving an LQR problem. The theory about LQ optimal control is applied as described in the book by Astrom and Murray [25]. LQ optimal control involves minimizing a cost function as shown in Equation 2-9 using positive semi-definite weighting matrix $Q \in \mathcal{R}^{4x4}$, $Q = Q^T$, and positive definite weighting matrix $R \in \mathcal{R}^{2x2}$, $R = R^T$, that can both be tuned intuitively, in contrary to other methods like pole-placement. The input sequence u is found by minimizing the discrete cost function that penalizes large inputs and deviations from the reference state.

$$J = \sum_{k=1}^{\infty} (z_k - z_k^{*j})^T Q (z_k - z_k^{*j}) + u_k^T R u_k \tag{2-9}$$

With u as defined in Equation 2-8, the optimal feedback gain K can be obtained by solving the Ricatti equation and subsequently computing:

$$K = -(B^T P B + R)^{-1} B^T P A \tag{2-10}$$

With P the unique solution of the Ricatti equation, given in Equation 2-11.

$$P = A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A \tag{2-11}$$

The closed loop system matrices with the obtained matrix K are given in Equation 2-12.

$$z_{k+1}^j = (A + BK) z_k^j - BKz^{*j} + \omega \tag{2-12}$$

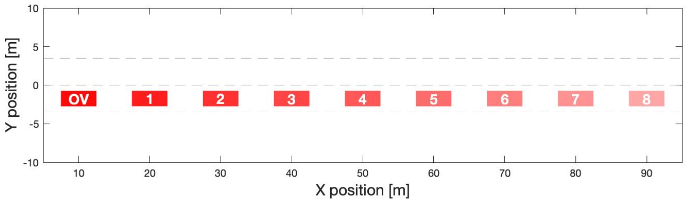
A more concise description can be given by defining the matrices $F := (A + BK)$, $G := B$, $\eta^j := -Kz^{*j}$, and $H := C$ resulting in:

$$\begin{aligned} z_{k+1}^j &= F z_k^j + G \eta^j + \omega \\ \gamma_k &= H z_k + \nu \end{aligned} \tag{2-13}$$

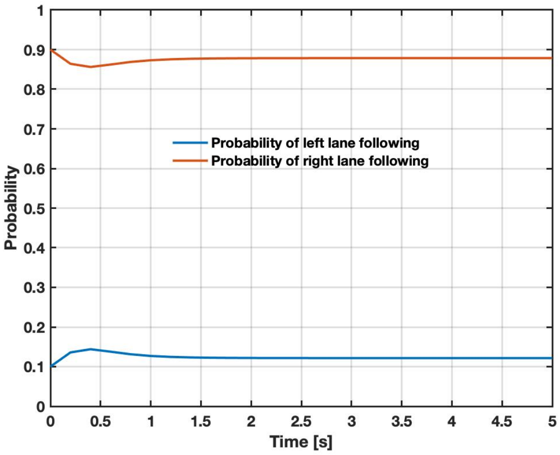
From Equation 2-13 it becomes evident that only η is dependant on j, and characterizes the maneuver to which the model aligns since it contains the reference z^{*j}. The reference trajectories $\{\boldsymbol{z}^{*j}\}_{j=1}^{j=n_I}$ are characterized by constant states throughout the prediction horizon, meaning the reference state remains consistent across the entire duration. Given that the motion planner is designed for highway scenarios, each reference trajectory aligns with tracking a specific lane. For instance, on a three-lane highway, three distinct reference trajectories are generated. The difference between these references is emphasized by the y-position, aligning with the lane center that the vehicle must track: $s_y^* = y_{lc}$. Furthermore, it is assumed that the OV aims to obtain zero lateral velocity $v_y^* = 0$, and that the longitudinal velocity remains unchanged $v_x^* = v_{x,k}$. A reference position s_x^* is not taken into account, penalizing deviations from the time-independent longitudinal velocity is adequate to achieve the desired forward velocity of the OV.

In this research, two distinct vehicle models are used since the controller is tested in a simulation environment with two lanes. When driving in an environment with a higher number of lanes, the algorithm can be adjusted accordingly by adding lane-following models.

In Figure 2-4 and 2-5, visualizations of the evolution of the maneuver probabilities over time in different scenarios are given. The probabilities associated with following a left and right lane are referred to as μ_L and μ_R respectively. In Figure 2-4a an OV executing a right lane-keeping maneuver is presented. The x- and y-axis represent the OV positions along and perpendicular to the road respectively. The grey dashed lines indicate road boundaries. As time progresses the opacity of the OV, indicated with a red rectangle, increases. In the second subFigure 2-4b, the evolution of μ_L and μ_R over time are shown for the maneuver indicated in 2-4a. Since the OV keeps tracking the right lane center, μ_R convergences to a high probability level of around 0.9, and μ_L converges to a probability of approximately 0.1. The values to which the probabilities converge when the maneuver intention stays constant are chosen as the probabilities with which the IMM is initialized. When the OV is initially in the right lane, the initial values are $\mu_L = 0.1$ and $\mu_R = 0.9$. When the first measurement indicates the OV is in the left lane, the IMM is initiated with $\mu_L = 0.9$ and $\mu_R = 0.1$.



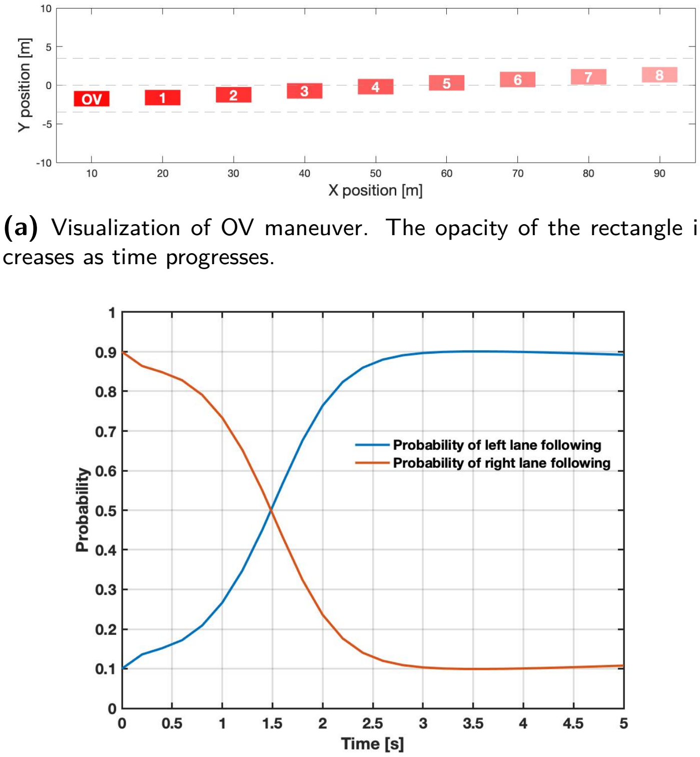
(a) Visualization of OV maneuver. The opacity of the rectangle increases as time progresses.



(b) Evolution of model probabilities μ_L and μ_R over time. The probability of a right-lane maneuver is significantly larger.

Figure 2-4: Evolution of the model probabilities μ_L (left lane following) and μ_R (right lane following) over time when OV keeps tracking the right lane.

In Figure 2-5, the same two subfigures are shown for an OV maneuvering from the right to the left lane. In Figure 2-5a the OV maneuver is shown, and in Figure 2-5b the evolution of μ_L and μ_R are presented. The probability of following the right lane decreases as time progresses, while μ_L increases. Note that the $\sum_{i=1}^{i=n_m} = 1$ at all times.



(a) Visualization of OV maneuver. The opacity of the rectangle increases as time progresses.

(b) Evolution of model probabilities μ_L and μ_R over time. The probability of following a left lane maneuver increases as μ_R decreases.

Figure 2-5: Evolution of the model probabilities μ_L (left lane following) and μ_R (right lane following) over time when OV changes from the right to the left lane.

In Figure 2-6, the y position of the OV (red) is plotted together with the probabilities μ_L and μ_R (blue) in a solid and dashed line respectively. On the left side, the y-axis indicates probability, while the right y-axis indicates position in meters. This figure illustrates the proactive behavior of the IMM algorithm, already taking into account the beginning of the OV maneuver by decreasing the probability of the OV following its current lane. This is a useful characteristic to accurately predict the OV positions and refine the optimization of the EV trajectory, which will be explained in detail in Chapter 4.

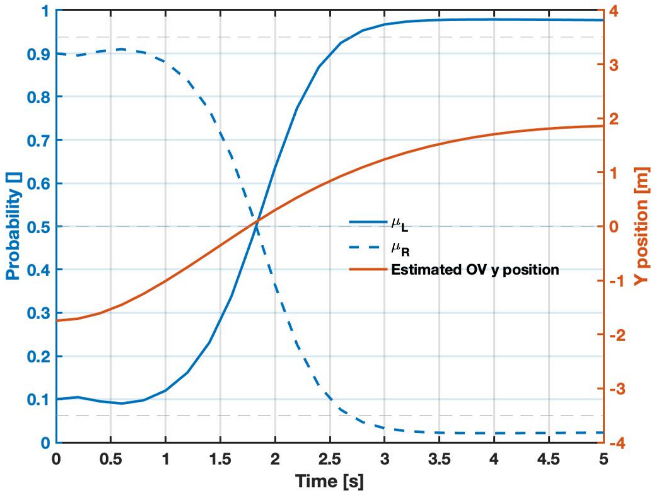


Figure 2-6: Evolution of model probabilities μ_L (left lane following) and μ_R (right lane following) over time together with the y position of the OV when it changes from the right to the left lane.

State-switching matrix Π

Another design parameter in IMM is the time-invarying switching matrix Π. The diagonal elements of the matrix represent the probability of the OV following its lane, π_{LF}, while the off-diagonal elements represent the probability of the OV changing lane π_{LC}. Higher values are selected for the diagonal elements as compared to the off-diagonal elements, due to the infrequent occurrence of lane changes by the vehicle. In Equation 2-14 the matrix Π with its elements is given.

$$\Pi = \begin{bmatrix} \pi_{LF} & \pi_{LC} \\ \pi_{LC} & \pi_{LF} \end{bmatrix} \tag{2-14}$$

The influence of Π can be understood by analyzing Figure 2-7. For the same scenario as in Figure 2-5, the probabilities μ_L and μ_R are computed over time with a different probability switching matrix Π. The result in Figure 2-5b was obtained by choosing Π as $\text{diag}(0.99, 0.01, 0.01, 0.99)$. In 2-7, Π was set to $\text{diag}(0.8, 0.2, 0.2, 0.8)$. The result of lower diagonal elements π_{LK} is a faster convergence of μ_L and μ_R to probabilities that are closer to each other. After tuning, the probability switching matrix is set to $\Pi = \text{diag}(0.99, 0.01, 0.01, 0.99)$. This decision is based on two factors: first, implementing lower values on the off-diagonal entries contributes to less conservative behavior. This is because a reduced likelihood of executing a lane change allows the model to prioritize adhering to the current lane, making it less sensitive to minor deviations. Second, higher diagonal entries establish a more pronounced contrast between the model probabilities. This is particularly advantageous when the system needs to distinguish between different maneuvers. This will become clear in Chapter 4 where the model probabilities μ_L and μ_R will be used in different parts of the optimization algorithm.

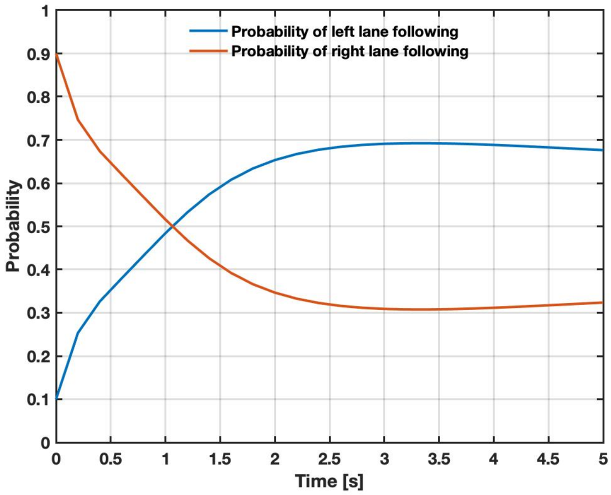
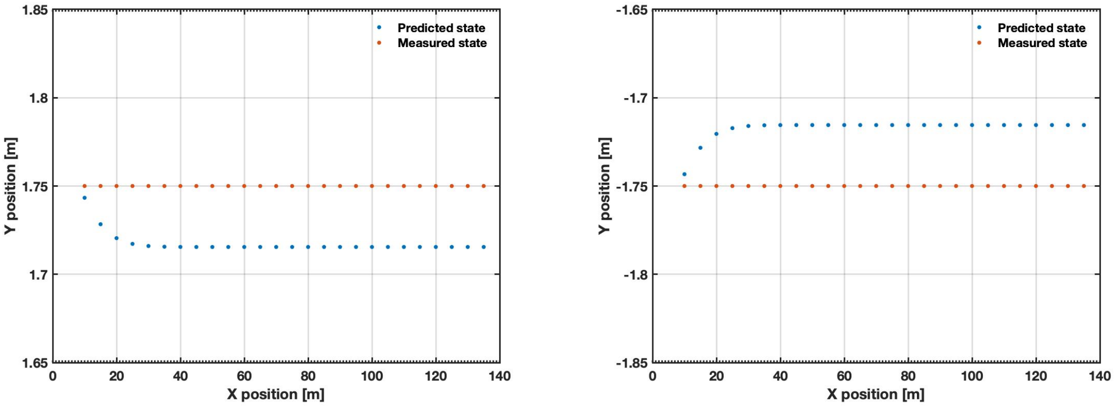


Figure 2-7: Evolution of μ_L and μ_R over time with lower values of π_{LK} and higher values of π_{LC} in the switching matrix Π.

OV state estimation with IMM

Next to the model probabilities, the IMM algorithm computes the estimated state $\hat{z}$ with covariance $\hat{P}$. In algorithm 1, from step 7, these steps are mathematically explained. By combining the estimated states from both filter models, the predicted state is computed based on the model probabilities. In Figure 2-8, two plots are shown of the measured and estimated OV positions over the simulation horizon. In subFigure 2-8a, the OV is following the left lane and its y-position is constant at $y = 1.75$. In subFigure 2-8b the same plot is shown in the case of the OV following the right lane with $y = -1.75$ as the center. In both cases, it is clear the predicted state has a constant offset from the measured state. This is because IMM combines the predicted states with both models to estimate the OV state. In this example, IMM predicts the next step if it is executing a right-lane maneuver and when it is executing a left-lane maneuver. Then, based on the probability of each maneuver, the final estimated state is computed as in step 20 in algorithm 1. When tracking the left lane, the predicted state consistently shows a leftward offset, whereas a rightward offset is observed when tracking the right lane. This is caused by the small probability of executing a lane change, which is never entirely zero. Even though the likelihood of adhering to the current lane is significantly higher, the estimate is influenced by the potential lane change scenario. Consequently, the estimated state is always a combination of the current lane-following prediction and the potential lane-change prediction, explaining the offsets.

In Figure 2-9, the measured and estimated states of the OV are shown when the OV's position changes from the right to the left lane. Again, the offsets are caused by the different model probabilities μ_L and μ_R. In all three examples, the mean estimation error is approximately 0.01 meters, which is nearly entirely caused by the offset in the y-direction since both prediction models use the same reference velocity in the x-direction.



(a) Left lane following.

(b) Right lane following.

Figure 2-8: Measured (red) and predicted (blue) OV states when the OV keeps its lane.

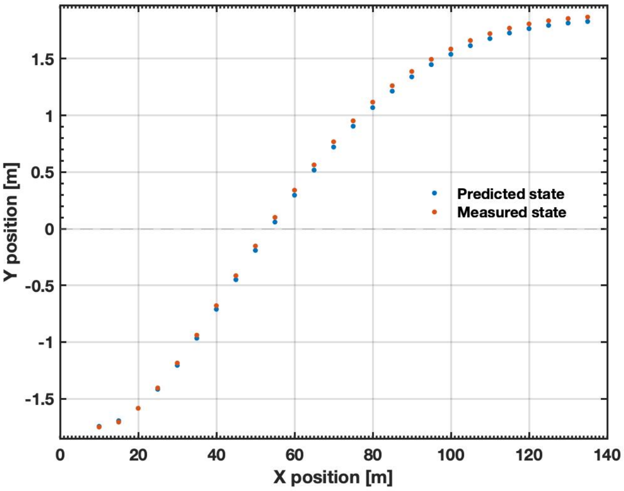


Figure 2-9: Mesured (red) and predicted (blue) OV states when the OV changes from the right to the left lane.

2-2 Trajectory Prediction and Uncertainty Propagation

In Section 2-1, it is explained how the Interactive Multiple Model algorithm is used to estimate the intention of other vehicles on the highway. At every timestep k of the simulation horizon, the IMM algorithm returns the following parameters:

- The estimated OV state at the current timestep k, $\hat{z}_k$.
- The covariance of the OV position estimation at timestep k, $\hat{P}_k$
- The probability of the OV following the left lane μ_L and the probability the OV is following the right lane μ_R.

SMPC works in a receding horizon fashion, which means that at every timestep in the simulation horizon, a prediction is made of the environment over the shorter prediction horizon N_p. Based on the IMM outputs, the future OV states can be predicted with a certain probability.

This section initiates with a summary of existing methods for motion prediction and uncertainty propagation, after which it will be explained how in this research *A)* OV states are predicted over the prediction horizon, and *B)* how the uncertainty of this estimation evolves.

2-2-1 OV Trajectory Prediction

Several methods for OV state prediction over the prediction horizon N_p are proposed in the literature, and their complexity varies. Motion prediction methods are commonly divided into three categories: physics-based, maneuver-based, and interaction-aware methods, increasing in complexity.

Methods that require low computational effort and are easy to implement use a predefined, **physics-based** dynamic model [26]. Widely used models are for example the constant velocity, constant acceleration, constant turn rate and velocity, and constant turn rate and acceleration models, of which the first two only consider forward motion, and the other methods also consider angular movement. As the name implies, it is assumed that certain OV states will stay constant over the prediction horizon. More complex models are for example the constant steering angle and velocity model, which uses the kinematics of the vehicle to calculate the trajectory when the steering angle stays constant instead of the yaw rate. The main advantage of physics-based motion models is that they are computationally efficient and can provide good results for short periods. However, the accuracy of the predicted trajectory is highly dependent on the initial conditions obtained by sensors that can be noisy. Furthermore, physics-based predictions are limited to short-term periods and are unable to predict changes in the vehicles' motion due to a particular maneuver or behavior of surrounding vehicles. Nevertheless, it is a popular method and widely used in the literature for motion planning with MPC or other receding horizon controllers. In [27] a comprehensive review of these physics-based models is given.

maneuver-based methods aim to predict the maneuver that is executed by another driver, thereby providing more accurate estimations than physics-based methods. These models include possible maneuvers of surrounding vehicles. Where physic-based methods assume that certain parameters will stay the same, maneuver-based models try to predict the maneuver that a vehicle is about to make and adjust the predicted future OV states accordingly. For example, when it is given that a vehicle will make a left lane change, it is assumed it will follow a certain reference path, with the middle of the left lane as the reference y-position. This method is used in for example [7], and in the papers that combine (S)MPC with IMM [15, 28, 26].

More complex motion prediction methods are **interaction-aware methods**. These methods aim to predict future OV movements by considering interactions between vehicles on the highway. For example, [29] proposes an interaction-aware prediction model for urban autonomous driving by penalizing certain maneuvers of other traffic participants that would result in a collision. [30] utilized Scenario-Based Model Predictive Control for highway motion planning while accounting for interaction-aware movements of other vehicles. Even though interaction-aware prediction models offer a more detailed understanding of the behavior of surrounding vehicles, it is not utilized because of the added computational complexity.

For this research, a maneuver-based motion prediction method is implemented by utilizing the output of the interactive multiple-model algorithm. At every time step, n_m predictors are initialized, with n_m the number of maneuvers as explained in the previous section. For every OV in the presence of the EV, all the trajectories associated with the predefined maneuvers are computed. This means that at every time step, $n_m * n_{OV}$ trajectories are generated. Each maneuver is initialized with the same initial state $\hat{z}_k$ as computed by the IMM algorithm. Then, the OV states are computed by iterating the closed-loop reference tracking dynamics as given in Equation 2-12 for all the maneuvers. Each maneuver is associated with a reference trajectory z^{j*}, guiding the vehicle to follow the center of the reference lane with zero lateral, and a positive longitudinal velocity This yields:

$$\tilde{\boldsymbol{z}}_{\boldsymbol{k}}^{\boldsymbol{j}} = \left[\tilde{z}_k^j \cdots \tilde{z}_{k+N_p}^j\right] \qquad \forall j \in \mathbb{I}(1, n_m). \tag{2-15}$$

How these predicted OV states would be implemented in the Stochastic Model Predictive Control algorithm will be explained in Chapter 4. Note that the hat symbol is utilized to denote the predicted or estimated state obtained through the Kalman Filter, representing the assimilation of measurements and correction. On the other hand, the tilde symbol is employed to signify states computed using the dynamic model alone, without the incorporation of measurements.

2-2-2 Uncertainty Propagation

When given the predicted current state of the OV $\hat{z}_k$ together with its covariance $\hat{P}_k$, the future trajectory of this OV over a time horizon with N_p time steps $\tilde{\boldsymbol{z}}_{\boldsymbol{k}}$ can be predicted with an uncertainty $\tilde{\boldsymbol{P}}_{\boldsymbol{k}}$. With a linear vehicle model of the OVs, the way this covariance matrix evolves can be recursively obtained with a Kalman Filter (KF) as proposed in multiple papers like [18], [15], and [14]. Other methods like chaos-based approaches [31] are proposed by the literature as well. Chaos-based approaches represent a random variable as a polynomial expansion to efficiently represent the uncertainty in a system and approximate the response of the system to variations in uncertain parameters. However, uncertainty analysis based on the Kalman Filter is more intuitive and computationally efficient than chaos-based approaches, which is why it is leveraged for uncertainty propagation. For a more detailed overview of uncertainty propagation methods, the reader is referred to [32].

With $F := (A + BK)$, $\tilde{\boldsymbol{P}}_{\boldsymbol{k}}$ can be obtained by iteratively solving:

$$P_{k+1} = F P_{k+1} F^T + \Sigma_\omega \tag{2-16}$$

Another possibility is to update the covariance without the closed-loop dynamics. However, the uncertainty then increases significantly in further time steps and the distribution will eventually become flat. By including the feedback, the covariance stays within bounds. The initial uncertainty P_k^j is set to the covariance computed by the IMM, and Σ_ω is the process noise with $\omega \sim \mathcal{N}(0, \Sigma_\omega)$. As can be seen from Equation 2-16, the evolution of P is not dependent on the trajectory j since all predictors are initialized with the same $P_0 = \hat{P}$. Therefore:

$$\tilde{\boldsymbol{P}}_{\boldsymbol{k}} := \left[\tilde{P}_k \cdots \tilde{P}_{k+N_p} \right] \; \forall j \in \mathbb{I}(1, n_m). \tag{2-17}$$

The diagonal elements of P_k represent the uncertainty in the x- and y-direction of the OV position at time step k. Only position uncertainty is considered, and since off-diagonal elements will develop during the computation, the covariance matrix is reduced to $\text{diag}(\sigma^2_{\text{x,k}}, \sigma^2_{\text{y,k}})$ at all time steps k, as proposed and implemented by Brüdigam et al. [14]. In Chapter 4 it is explained how the position uncertainty is used in the SMPC optimal control problem.

Chapter 3

Planning Architecture

In this chapter, the planning architecture is discussed. The proposed method for emergency maneuver motion planning utilizes a combination of the following three components:

1. The IMM algorithm for OV motion prediction.
2. SMPC as the primary motion planner.
3. The backup controller.

The details of these three features are discussed in the designated chapters. This chapter describes how these three features are combined and how each contributes to the trajectory optimization method.

Figure 3-1 shows an overview of the control structure including IMM, SMPC, and the backup MPC. The three features are shown in orange, blue squares indicate functions and in the ellipses, parameters are given. The figure shows the planning structure when n OVs are present: $\{\mathrm{OV1} \cdots \mathrm{OVn}\}$. For each OV, at every time step, noisy measurements of its position are received and used as input for the IMM algorithm. IMM can be understood as an extension of the Kalman Filter, it returns not only the estimated OV state $\hat{z}_k$ and covariance $\hat{P}_k$ but the probabilities of high-level maneuver intentions $\{\mu^j\}_{j=1}^{j=n_\mathrm{m}}$ of the associated OV as well. In this research, the maneuvers are related to following a certain lane center. For instance, on a two-lane highway, the probability indicates the likelihood of the OV following either the left or right lane. The two outputs of IMM serve the following purposes:

- From the *estimated state and covariance*, possible trajectories of the OV are generated using multimodal trajectory prediction, representing trajectories of all possible maneuvers of the OV. The predicted trajectories are used for constraint generation in SMPC.
- The *probability distribution over the possible maneuvers* indicates the likeliness of the OV following a certain predefined maneuver. The maneuver probabilities are used for both

OCPs: SMPC and the backup MPC. For SMPC, the maneuver probabilities add to the chance-constraint generation. Maneuvers with a higher probability are more carefully assessed than unlikely maneuvers. Furthermore, if an OV shows ambiguous behavior, tighter constraints are applied, resulting in more conservative behavior. Furthermore, the model probabilities are utilized in the generation of the reference trajectory that is used in the cost function of both SMPC and the backup MPC. The details of constraint generation and reference lane selection are given in further sections in this chapter.

If SMPC is feasible, the optimal input sequence corresponds to the one prescribed by SMPC, denoted as $u_{\text{opt}} = u_{\text{smpc}}$.

Definition 3-0.1 (Feasibility)**.** Feasibility is characterized by the capability of SMPC to find a solution within the designated computational time that meets all specified chance constraints. Note that feasibility is unrelated to the optimality of the solution.

To account for the possible infeasibility of SMPC, a backup controller is designed. The backup controller should be able to compute collision risk-minimizing trajectories within a short computational time. For that purpose, two backup controllers were designed of which the backup MPC showed the best performance. The details are described in Chapter 5. The backup OCP is only solved when SMPC is infeasible, and the optimal input sequence is then $u_{\text{opt}} = u_{\text{backup}}$.

From the optimal input sequence, only the first input is applied to the EV. Moreover, the input sequence is utilized as an initial guess for the SMPC optimal control problem. An initial guess in an optimization algorithm represents the starting point for an optimization process, aiding the algorithm in its search for the optimal solution. A well-chosen initial guess can contribute to faster convergence and better solutions, while a poor guess may lead to longer optimization times. Therefore, the initial guess is chosen as the optimal control input u_{opt}, shifted over one time step. Using this smart initial guess reduced the computational times of SMPC significantly. Furthermore, using the optimal input sequence provided by the backup MPC in case of infeasible SMPC 'helps' SMPC with finding a feasible solution in the next iteration, which is an addition to the performance of the backup MPC.

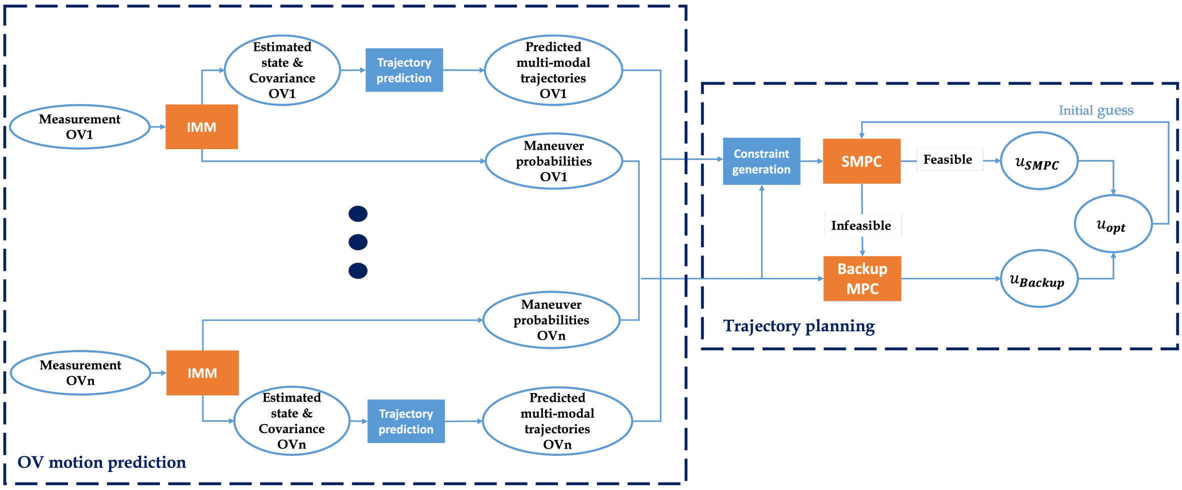


Figure 3-1: Contol structure IMM + SMPC + Backup MPC with n OVs in the presence of the EV.

Chapter 4

Stochastic Model Predictive Motion Planning

This chapter explains the design choices of trajectory planning with SMPC. It starts with an overview and comparison of trajectory planning algorithms, leading to the motivation behind the employment of SMPC. Then, an overview is shown of the control architecture, highlighting the role of SMPC within the full control structure including IMM and the backup controller. Then, the mathematical formulation of the SMPC COP is given, and the separate parts are then systematically explained in subsequent sections.

4-1 Motion Planning Algorithms

Trajectory planning for autonomous driving is a widely studied research area, and a variety of solutions can be found in the literature. In the review paper by Claussmann et al., different families of motion planning algorithms are defined and examined [5]. Below, the five families are listed.

- Space configuration analysis
- Pathfinding algorithms
- Curve-based methods
- Attractive and repulsive forces
- Optimization-based
- Artificial intelligence

Table 4-1 presents an overview of how well these planning algorithm families align with the specified requirements of the planner derived from the problem description in Section 1-2. An elaboration of this table will be given in the following paragraphs, concluding with the reasoning behind the choice to use the optimization-based Stochastic Model Predictive Control Algorithm for the defined problem.

Space configuration methods define the configuration space, representing all possible configurations or positions of a robot or vehicle within its environment. These methods aim to find collision-free paths by considering the obstacles and constraints present in the configuration space. However, these methods need an additional algorithm to find a feasible motion within that space. In [33] a space configuration method is used in combination with model predictive control to design a low-complexity trajectory controller. However, space configuration methods face limitations in the context of trajectory planning for highway emergency situations. The dynamic and unpredictable nature of emergency scenarios, coupled with the need for real-time response, makes it challenging to rely solely on predefined configuration spaces. For this reason, space configuration methods are not considered as suitable.

Space configuration methods define the configuration space to find collision-free paths for robots or vehicles within their environment. Despite being combined with model predictive control for low-complexity trajectory control in [33], these methods are inadequate for highway emergency scenarios due to their reliance on predefined configuration spaces. The dynamic and unpredictable nature of emergencies necessitates real-time response, making such methods unsuitable. Therefore, alternative approaches are required for effective trajectory planning in emergency situations. **Pathfinding algorithms** aim to find optimal paths in a given environment. Although they use efficient algorithms like Dijkstra's or A* algorithm, they fail to account for the high nonlinearity of EV dynamics. In [34] a pathfinding algorithm for long-term path planning is used. However, due to the need for rapid decision-making and adaptability in dynamic scenarios, these methods are computationally inefficient and unsuitable for addressing the specific requirements of high-stakes driving environments. Therefore, pathfinding algorithms are not viable options for trajectory planning in such contexts. **Curve-based** methods utilize mathematical curves, such as splines or Bézier curves, to represent paths or trajectories. These methods allow for smooth and continuous motion planning, providing desirable characteristics for tasks that require precise and fluid movements. These methods are suitable when the objective is to find fluid and smooth motion, which is not the case in emergencies with dynamic obstacles.

Methods that rely on **attractive and repulsive forces**, such as the Artificial Potential Field method (APF), incorporate forces to guide the ego vehicle towards the goal while avoiding obstacles. Although widely researched, APF is sensitive to local minima, as addressed in [35], where improvements for collision-avoidant motion planning are proposed. However, a critique highlighted in [36] is that APF neglects vehicle dynamics during planning, leading to the computation of unfeasible trajectories. Due to this limitation in accommodating vehicle dynamics, APF is deemed unsuitable for the proposed problem of trajectory planning in the given context.

Utilizing **Artificial Intelligence** (AI) in autonomous driving is gaining popularity, particularly in dynamic and uncertain environments, where learning-based methods excel in generalizing learned rules to new situations [37]. Learning-based models offer advantages over traditional methods by autonomously adapting and improving performance over time through real-world data training. However, challenges outlined in [38] include the need for substantial data to train robust planners and ensure safety during data collection. Training AI models for emergencies proves suboptimal due to safety concerns and the rarity of emergency occurrence during data collection. While some promising approaches like deep reinforcement learning-based motion planning exist [39], scarcity of data led to the decision against deploying AI for motion planning in the given context.

The last methods discussed are **based on optimization**, with MPC being the most commonly used. MPC formulates the motion planning problem as an optimization task to minimize or maximize a specified objective function while adhering to system constraints. [40] exemplifies this by optimizing trajectories while minimizing various factors like travel time and deviation from the road descriptor path through a carefully designed cost function. MPC, being model-based, utilizes system dynamics to predict behavior over a prediction horizon N, allowing the inclusion of nonlinear EV dynamics to prevent infeasible trajectories. [11] demonstrates how the estimation of vehicle models in MPC prevents infeasible trajectories in hierarchical control structures. MPC employs a receding horizon strategy, finding optimal trajectories over a prediction horizon and applying only the first input to the system, enabling proactive consideration of future obstacles for collision avoidance in dynamic environments, particularly leveraging predictions of other vehicle motions [11].

Since the future of motion of OVs is uncertain, and motion prediction methods are always probabilistic, it is worth exploring **SMPC**, a method that not only incorporates a receding horizon and optimizes trajectories but also accommodates uncertainty in OV behavior [13]. SMPC extends MPC by explicitly considering the stochastic nature of the problem. The stochasticity of the problem can be incorporated using *chance constraints*. Where the classic MPC approach minimizes a cost function subject to deterministic constraints, SMPC finds the best possible solution with constraints on the probability that certain limits are satisfied. For example, a chance constraint could state that the probability that the distance between the vehicle and an obstacle is larger than the safety distance has to be greater than a certain value between 0 and 1, which is done in [16]. Since SMPC is a useful method to account for conflicting objectives, nonlinear EV dynamics, and navigating in dynamic, uncertain environments, it is chosen to further employ this control structure. However, a limitation of this method is imposed by the computational complexity of the SMPC COP, leading to high computational times. While various successful approaches to keep computational times within bounds can be found in the literature, such as defining the COP as a quadratic programming problem [41, 3] or decreasing control frequency [42], it remains a challenging aspect. Computational complexity therefore has to be closely considered when designing the motion planner.

	Conflicting Objectives	Nonlinear EV dynamics
Space configuration analysis	- -	- -
Pathfinding algorithms	- -	-
Curve-based methods	-	+
Attractive and repulsive forces	+	0
Artificial Intelligence	++	+
Optimization-based	++	++

(a) Assessment for Conflicting Objectives and Nonlinear EV Dynamics

	Dynamic environment	Computational time
Space configuration analysis	-	++
Pathfinding algorithms	-	+
Curve-based methods	0	+
Attractive and repulsive forces	+	+
Artificial Intelligence	++	+
Optimization-based	++	-

(b) Assessment for Dynamic Environment and Computational Time

Table 4-1: Assessment of six trajectory planning algorithm families based on the four key requirements that arose from the problem description. '- -' indicates a significant limitation, '-' represents a moderate limitation, '0' denotes a neutral performance, '+' signifies a moderate strength, and '++' indicates a significant strength.

4-2 Mathematical Formulation of the Planning Optimization Problem

In Section 4-1 the reasoning is given behind choosing stochastic model predictive control as the control structure for motion planning. SMPC operates in a receding horizon fashion, and iteratively solves a control optimization problem over a finite time horizon. The result is an optimal input sequence, of which only the first input is applied to the system. At the next time step, the process is repeated. This section shows a mathematical overview of the COP of SMPC as illustrated in the context of the motion planner under consideration. All separate elements will be explained in detail in the following sections.

In Equation 5-4, the cost function J is given. This function indicates the quality of the optimal solution: the lower the cost function, the better the planned trajectory. It consists of multiple terms with associated weighting matrices: Q penalizes deviations from a reference trajectory, R penalizes large inputs, S input rates, and T penalizes outputs of the problem-specific risk function. Section 4-3 elaborates on the design of the cost function.

The constraints are given in Equation 5-5a - 5-5d. The first constraint imposes boundaries on the planned trajectories imposed by the vehicle dynamics, explained in Section 4-3. Equation 5-5b and 5-5c describe the set of allowed inputs and states respectively. In Equation 5-5d the formulation of the chance constraints is given. As explained, the probability that a certain constraint holds should be larger than the safety factor β. This factor β is a design parameter and can be adjusted to the planner's objective. Section 4-6 provides details on the constraints.

$$J = \min_{u} \sum_{i=1}^{N_p} \mathbb{E}\left[||\Delta \xi_i||_Q^2 + ||u_i||_R^2 + ||\Delta u_i||_S^2 + ||\mathcal{R}||_T^2 \right], \tag{4-1}$$

s.t.

$$\begin{aligned}
\xi_{i+1} &= f(\xi_i, u_i), && \forall i \in \{1, ..., N\}, && \text{(4-2a)}\\
u_i &\in \boldsymbol{\mathcal{U}}, && \forall i \in \{1, ..., N\}, && \text{(4-2b)}\\
\xi_i &\in \boldsymbol{\Xi}, && \forall i \in \{1, ..., N\}, && \text{(4-2c)}\\
P(q_{i,j} \leq 0) &\geq \beta, && \forall i \in \{1, ..., N\}, \forall j \in \{1, ..., n_{\text{ov}}\}, && \text{(4-2d)}
\end{aligned}$$

4-3 Ego Vehicle Model

The SMPC algorithm requires a system model for the ego vehicle to make predictions of the vehicles' behavior over the prediction horizon N_p. It aims to find the control input $u_{1:N_p}$ for the defined vehicle model, generating vehicle states that minimize the cost function while satisfying the (chance) constraints.

4-3-1 Model Selection

As outlined in the problem description, predicting the highly nonlinear dynamics of the ego vehicle at its handling limits presents challenges. To balance model fidelity and computational efficiency, a predefined kinematic vehicle model is chosen over learning from data due to data scarcity and safety concerns [12]. Common models include the point-mass model, the bicycle model (single-track), and the double-track model. The point-mass model simplifies dynamics, making it computationally efficient [41, 11]. The bicycle model captures translational and rotational dynamics, aiding in yaw stabilization. The double-track model extends this by considering lateral motion, providing a more detailed representation [43]. While the double-track model is more accurate, its computational demands are higher. [44] addresses computational load challenges, finding the single-track bicycle model sufficient for critical scenarios. [45] demonstrates that increased model complexity has minimal impact on planned trajectories, but rotational dynamics are crucial for extreme maneuvers. [43] confirms the efficiency of the single-track model in computing feasible trajectories, especially with careful lateral acceleration bounds. Consequently, the kinematic bicycle model is adopted to balance computational efficiency and essential kinematic considerations.

4-3-2 Single-track Bicycle model

In figure 4-1 a schematic drawing is shown of the kinematic bicycle model within the road-aligned frame. The model is based on the book by Rajesh Rajamani [6], and is widely adopted in the literature. s is the distance traveled by the EV along the path indicated in orange. This path can be either straight or curved. d is the lateral distance from the path to the center of gravity, and ϕ indicates the yaw angle of the vehicle with respect to the reference path. l_f

and l_r are the distances from the center of gravity to the front and rear wheel respectively, and δ is the steering angle. Furthermore, v represents the velocity vector aligned with the angle α relative to the vehicle frame.

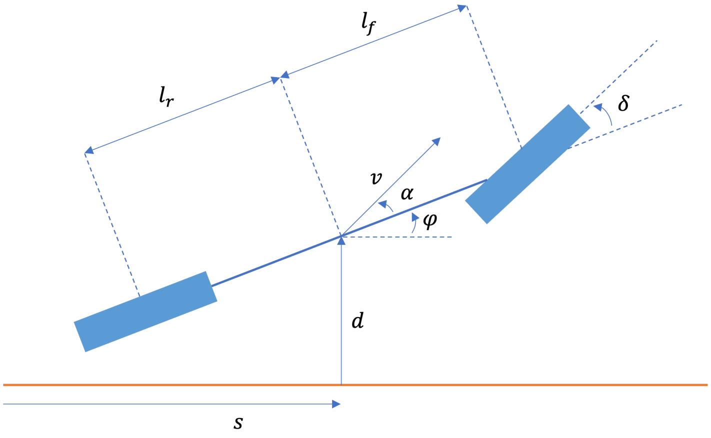


Figure 4-1: Scheme of kinematic bicycle model in road aligned coordinates. Created by author, based on [6].

The vehicle's state, denoted by $\xi = [s, d, \phi, v]^T$, includes its position along the longitudinal (s) and lateral (d) axes, as well as its heading angle (ϕ) and velocity (v). The continuous, nonlinear dynamics of the vehicle are represented by $\dot{\xi} = f_c(\xi, u)$, where $\dot{\xi} = [\dot{s}, \dot{d}, \dot{\phi}, \dot{v}]^T$. The input vector $u = [a, \delta]^T$ comprises the acceleration (a) and the steering angle (δ). Equation 4-3 shows the relation between $\dot{\xi}$, ξ, and u, providing a concise expression for the kinematic bicycle model dynamics.

$$\begin{aligned}\dot{s} &= v\cos(\phi + \alpha)\\ \dot{d} &= v\sin(\phi + \alpha)\\ \dot{\phi} &= \frac{v}{l_r}\sin(\alpha)\\ \dot{v} &= a\end{aligned} \tag{4-3}$$

In Equation 4-3, α is the slip angle. It is dependent on the geometry of the vehicle, and the steering angle δ, as shown in Equation 4-4.

$$\alpha = \arctan(\frac{l_r}{l_r + l_f}\tan(\delta)) \tag{4-4}$$

Linearization and Discretization

At every time step in the simulation horizon T, the COP is initialized by linearizing the bicycle model around a chosen operating point (ξ^*, u^*), using the theory from the book 'Nonlinear Control' by Hassan Khalil [46]. The linearized model is used to predict the behavior of the EV over the prediction horizon N to solve the SMPC COP. The choice of operating point around which the system is linearized significantly affects the accuracy of the model estimation. Since the goal is to find a local approximation, the operating state is chosen as the current EV state: $\xi^* = \xi_k$. However, linearizing around the current input u_k results in large prediction errors in further time steps in the prediction horizon N. A solution to this problem that is often seen in the literature is using the current state and zero input as operating point, since this significantly reduces the prediction error. To validate this theory, the vehicle dynamics are estimated with linearization around the current input,$u^* = u_k$, and zero input, $u^* = [0, 0]^T$, in the scenario shown in figure 4-2. In this scenario the EV, in blue, overtakes the preceding OV, in red, that starts decelerating.

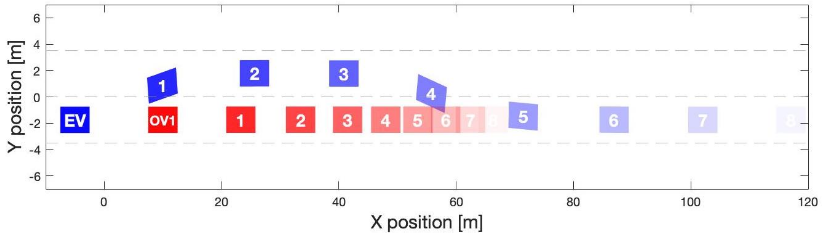


Figure 4-2: Scenario in which linearization points are compared: EV, in blue, overtakes decelerating, preceding OV, in red. Opacity increases as time progresses. Grey dashed lines indicate road boundaries.

During this simulation, the prediction error of the linearized model with $u^* = u_k$ was 15 times higher than the model with $u^* = [0, 0]^T$. In figure 4-3, a visualization is shown of the predicted EV states over the prediction horizon for the nonlinear system (blue), the linearized system around the current input (red), and the linearized system around zero input (yellow). At this point, the EV just passed the OV and aims to return to the right lane.

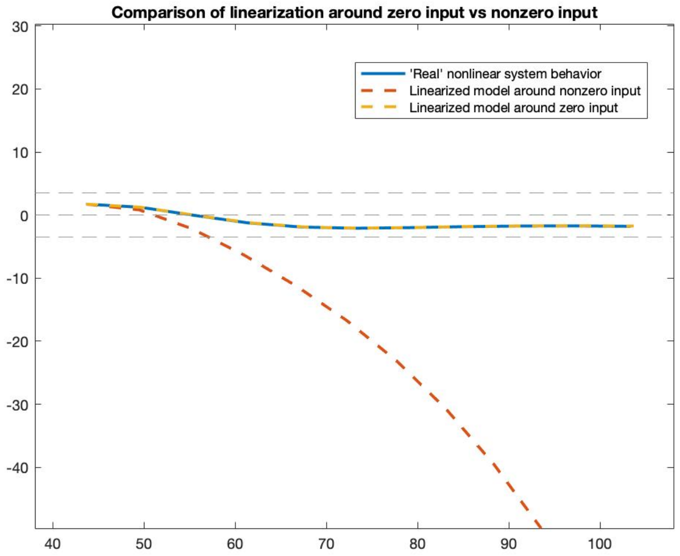


Figure 4-3: Comparison of linearization around the current input (red) vs zero input (yellow). Nonlinear dynamics were used as a benchmark (blue). Grey dashed lines indicate road boundaries.

From the figure, it can be seen that the prediction error of the model using the current state as the operating point increases significantly as time progresses within the prediction horizon while using zero input as the operating point results in an accurate estimation of the real model. Therefore, the operation point around which the bicycle model is linearized is chosen as the current state ($\xi^* = \xi_k$ with zero input $u^* = [0, 0]^T$. This yields the dynamics as stated in Equation 4-5.

$$\dot{\xi} = f_c(\xi^*, u^*) + A_1(\xi - \xi^*) + B_1 u \tag{4-5}$$

With A_1 and B_1 the Jacobian matrices of the nonlinear function f_c to the state ξ and the input u respectively.

$$A_1 = \frac{\delta f_c}{\xi}\bigg|_{(\xi^*, u^*)} \qquad B_1 = \frac{\delta f_c}{u}\bigg|_{(\xi^*, u^*)} \tag{4-6}$$

The planner receives measurements from its environment in discrete time and applies control input in discrete time. It is therefore evident that the linearized, continuous model in Equation 4-5 has to be discretized to facilitate its implementation in the motion planner, ensuring compatibility with the discrete nature of the process. Sampling the continuous system with zero-order-hold is a common method to discretize continuous systems, as explained in the book 'Computer-Controlled Systems' by Aström and Wittenmark [47]. Discretizing with zero-order-hold entails holding the signal obtained from the continuous system until a new value is commanded after the sampling time h. The discretized system is given in Equation 4-7. The indices i indicate different steps in the prediction horizon N, while k indicates the time step in the simulation horizon T.

$$\xi_{i+1} = \xi^* + f_c(\xi^*, u^*) + \Phi(\xi_i - \xi^*) + \Gamma u_i \tag{4-7}$$

The discrete system matrices Φ and Γ can be obtained with Equation 4-8, as explained in the book by Aström and Wittenmark [47].

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} ds B \end{aligned} \tag{4-8}$$

The obtained system matrices Φ and Γ can be found in appendix A.

4-4 Cost Function Design

This section explains the structure of the cost function, given in Equation 4-9.

$$J = \min_u \sum_{i=1}^{N_p} \mathbb{E}\left[||\Delta\xi_i||_Q^2 + ||u_i||_R^2 + ||\Delta u_i||_S^2 + ||\mathcal{R}||_T^2\right] \tag{4-9}$$

The first term penalizes deviations from a reference trajectory ξ_i^*, with $\Delta\xi_i = \xi_i^* - \xi_i$. The reference state ξ_i^* is constant over the prediction horizon:

$$\xi_i^* = [d^*, \phi^*, v^*] \tag{4-10}$$

For the longitudinal position s no reference is determined since it would be time-dependent. Instead, a forward reference velocity $v^* > 0$ is set to reach the desired velocity. The lateral position d is set to one of the lane centers on the highway: $d^* = y_{LC}$. The selection of this reference lane is based on threat assessment and is further explained in Section 4-5. The reference heading angle ϕ^* is set to zero. The weighting matrix $Q \in \mathbb{R}^{4x4}$ is diagonal. The second term in the cost function penalizes large inputs with the diagonal matrix $R \in \mathbb{R}^{2x2}$. Furthermore, input rates are penalized with diagonal matrix $S \in \mathbb{R}^{2x2}$. These two terms are introduced in the cost function to aim for trajectories that are comfortable for the passengers and do not consume unnecessary energy. However, collision avoidance is prioritized in the emergency motion planner, so relatively lower costs are assigned to these terms.

The last term presents the risk function $\mathcal{R}$, penalized with the diagonal matrix $T \in \mathbb{R}^{1x1}$. In addition to incorporating chance constraints into the COP to prevent collisions with OVs, a dedicated risk function is formulated and integrated into the cost function. This inclusion aims to minimize the associated risk, providing an approach to further ensure safety in the trajectory planning process. For the definition of the risk function, the information provided by the IMM is exploited. With the predicted states of the OV $\tilde{\boldsymbol{z}}^{\boldsymbol{j}}$ with $\tilde{\boldsymbol{P}}^{\boldsymbol{j}}$, a probability density function (PDF) can be computed around the OV at every time step in the prediction horizon N. The PDF can be directly used as a risk function in the COP:

$$\mathcal{R}_i = \frac{1}{2\pi\sqrt{\det(q\bar{P}_i)}} \exp(-\frac{1}{2}(\bar{\xi}_i - \bar{z}_i)^T q\bar{P}_i^{-1}(\bar{\xi}_i - \bar{z}_i)) \tag{4-11}$$

In this equation, $\bar{\xi}_i \in \mathbb{R}^{2x1}$ and $\bar{z}_i \in \mathbb{R}^{2x1}$ are the reduced state matrices that contain the positions of the EV and OV at time step i respectively. Moreover, $\bar{P}_i$ is the variance at time step i around the predicted OV position $\bar{z}_i$. However, the PDF does not consider the size of both vehicles, and its values decrease significantly with higher values of $(\bar{\xi}_i - \bar{z}_i)$, causing it to be near zero at the boundaries of the OV shape. Therefore, the variance is scaled with a factor q. Scaling the variance makes the PDF more flat, increasing its values at positions further away from the predicted OV state. Since the function is used for minimization, scaling the PDF does not influence the optimization outcome, but it enhances the exploration of the solution space. Note that in the risk function, in contrast to the constraints, only the most probable maneuver is considered, i.e. the maneuver with the highest model probability μ computed by the IMM. Minimizing the collision risk for all possible OV maneuvers results in unnecessary conservative EV trajectory planning.

4-5 Reference Trajectory Generation

In the cost function of the COP for both SMPC and the backup controller, deviations from the reference trajectory ξ^* are penalized with weighting matrix Q. In all scenarios, the reference state sequence is set to a heading angle of zero relative to the road alignment: $\phi^* = 0$.

Reference velocity selection

The reference velocity is always set to a value greater or equal to zero, $v^* \geq 0$. Recognizing that a reference position along the road s is time-dependent, for the sake of simplicity, only a forward reference velocity is set, which is also sufficient for specifying the reference state in the longitudinal direction. By default, the reference velocity is set to the designated speed limit $v^* = v_{\text{sl}}$. In some cases, however, the reference velocity is set to a lower value. When the model probabilities of an OV do not give a clear conclusion about the OV maneuver, in other words, when μ_L and μ_R are close to each other, the EV is encouraged not to overtake that OV. Then, the reference velocity v^* is set to a value lower than the concerned OV $0 \leq v^* \leq v_{\text{ov}}$.

Reference lane selection through threat assessment

The reference lateral position d^* is set to one of the lane centers on the highway, $d^* = y_{\text{LC}}$. y_{LC} is set to the lane center of the safest possible lane at time step k, which is an improvement compared to choosing the current lane as reference lane [15, 17] or choosing the right lane as reference lane at all times [8]. By setting the reference lateral position to the center of the safest possible lane, the cost function decreases when SMPC explores solutions of trajectories in the direction of a safer lane. For instance, in the event of an accident ahead of the EV, the reference lane shifts to an adjacent, safer lane, causing the cost to decrease when SMPC explores trajectories in that direction. Setting the reference lane to the current lane would lead to an increased cost when SMPC explores trajectories in the direction of another lane, working against finding a safe trajectory. The safest possible lane is determined by applying situational threat assessment [48, 49] based on the information obtained by the IMM algorithm. Based on the threat metrics, a cost is assigned to each lane, and the one with the lowest cost is selected as the safest possible lane. The center of this lane is selected as the lateral reference $d^* = y_{LC}$. When any OVs are present in the proximity of the EV, they contribute to the overall cost of the lanes. The total cost is calculated by summing up the individual costs associated with each OV. The cost imposed by one OV is computed by adding two risk metrics, balanced with weighting matrices. A cost is only assigned to a lane if the following holds for the OV:

- The probability that an OV is following lane j is higher than a minimum value: $\mu^j \geq \mu_{\text{min}}$.
- The distance between the EV and the OV is smaller than a maximum value: $|\Delta x| \leq x_{\text{min}}$.

In Equation 4-12 the cost assigned to a lane related to a single OV is given. If multiple OVs are present, the costs for lane j are added. The cost for lane j imposed by vehicle k is determined as

$$J^{j,l} = W_1 J_1^{j,l} + W_2 J_2^{j,l} \tag{4-12}$$

The first metric J_1^j is statistic-based. Typically, statistic-based risk metrics require high computational times, but since the IMM algorithm already provides the maneuver probabilities for the chance constraints, they are also leveraged in the threat assessment. Therefore, the first part of the cost function, J_1 increases with a higher model probability:

$$J_1^{j,l} = \mu^{j,l} \tag{4-13}$$

The second term is based on the time to collision (TTC), a risk metric widely used in the literature for collision avoidance. The TTC is the time required before the distance between two vehicles is zero and a collision occurs: $\{t|d_r^t = 0\}$. The higher the TTC between two vehicles, the lower the collision risk. The time to collision is defined as follows [49]:

$$TTC = \{t|d_r^t = 0\} \begin{cases} -\frac{d_r}{v_r}, & \text{if } v_r < 0, \\ \text{undefined}, & \text{if } v_r \geq 0, \end{cases} \tag{4-14}$$

In this equation, $d_r = d_l - d_f$ and $v_r = v_l - v_f$ with d_r the relative position, v_r the relative velocity. The subscripts l and f indicate the leading and following vehicle respectively. In the multimodal trajectory prediction, it is assumed the OV will maintain a constant velocity throughout the prediction horizon. Therefore acceleration is not taken into account for the computation of the TTC. For the cost function, the inverse of the TTC is computed, and scaled with weighting matrix W_2:

$$J_2^{j,l} = TTC^{-1} \tag{4-15}$$

The total cost for a lane is computed by adding the costs imposed by the total number of OVs n_{ov}.

$$J^j = \sum_{l=1}^{l=n_{\text{ov}}} J^{j,l} \tag{4-16}$$

The reference lane is set to the lane with the lowest associated cost. In case the cost of both lanes is zero, the reference lane is set to the right lane, to adhere to traffic rules.

4-6 Safety Constraints Definition

In this section, it is explained how the state constraints are designed and implemented in the COP. First, a brief recap is given of the deterministic constraints, then the chance constraints will be described in detail. The mathematical formulation of the deterministic constraints are repeated below:

$$\begin{aligned} \xi_{i+1} &= f(\xi_i, u_i), & \forall i \in \{1, ..., N\}, \quad &\text{(4-17a)} \\ u_i &\in \boldsymbol{\mathcal{U}}, & \forall i \in \{1, ..., N\}, \quad &\text{(4-17b)} \\ \xi_i &\in \boldsymbol{\Xi}, & \forall i \in \{1, ..., N\}, \quad &\text{(4-17c)} \\ & & &\text{(4-17d)} \end{aligned}$$

Equation 4-17a defines the constraint imposed by the dynamic model of the EV. Equation 4-17b and 4-17c describe the boundaries on the input and state sequence respectively. The state and input constraints in this context are deterministic. State constraints establish limits on lateral positions, due to road limits, and forward velocity, ensuring it remains greater than zero to maintain forward motion. On the other hand, input constraints impose boundaries on maximum acceleration, deceleration, and steering angle, reflecting their limitations arising from vehicle dynamics and mechanical restrictions.

The chance constraints, on the contrary, are probabilistic and are defined based on a safety factor β as in Equation 4-18.

$$P(q_{i,j} \leq 0) \geq \beta, \qquad \forall i \in \{1, ..., N\}, \forall j \in \{1, ..., n_{\text{ov}}\}, \tag{4-18}$$

Instead of putting hard boundaries on the control problem as in robust MPC, the chance that a constraint will be violated is bounded with $1 - \beta_i$, dependent on the safety factor β at time step i. The chance constraints are leveraged to account for collision avoidance: for every OV in the presence of the EV, a chance constraint on the position of the EV relative to the OV is defined at every time step in the prediction horizon. Because directly implementing chance constraints in an COP is impossible, deterministic constraints must be designed to achieve the same result. In the following, the design of chance constraints based on the output of the IMM algorithm and the multi-model predictions will be elaborated.

At every timestep $i \in \mathbb{I}(1, N)$, n_{m} position estimations for the OV are available: one for every possible maneuver. Around those positions, ellipsoidal safety areas are created based on the uncertainty of the position estimation, the size of the vehicles, and the timestep i. EV trajectories should be planned such that the center of the EV stays outside the ellipsoidal areas around the OVs. Figure 4-4 shows a graphic representation of an ellipse around an OV with semi-major axis a and semi-minor axis b.

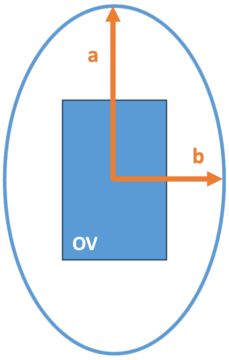


Figure 4-4: Safety ellipsoidal area around the OV with semi-major axis a and semi-minor axis b. Created by author

The value of a and b are built up from the following elements:

$$\begin{aligned} a_i &= l_{\text{veh}} + \delta_x + a_{\beta,i} \\ b_i &= w_{\text{veh}} + \delta_y + b_{\beta,i} \end{aligned} \tag{4-19}$$

l_{veh} and w_{veh} represent the vehicle length and width, respectively. By utilizing the full length and width, both the size of the OV and EV are taken into consideration, ensuring that the center of the EV remains outside the ellipse. The second terms, δ_x and δ_y, denote additional safety margins that can be introduced to address geometric disparities between the OV and an ellipse. The third terms, $a_{\beta,i}$ and $b_{\beta,i}$, depend on the IMM output and represent the components accounting for the uncertainty in OV maneuver executions. Using the estimated OV position with its variances in longitudinal and lateral directions, a multivariate normal probability distribution function can be formulated as

$$P(\bar{z}_i^j = \boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\bar{z}_i^j - \boldsymbol{x})^T(\bar{P}_i^j)^{-1}(\bar{z}_i^j - \boldsymbol{x})\right)}{2\pi\sqrt{\det(\bar{P}_i^j)}} \tag{4-20}$$

with $\boldsymbol{x}$ some predicted OV position and $\bar{P}_i^j$ the covariance of the prediction. In figure 4-5 a graphic representation of a multivariate normal distribution is shown with uncertainty in the x- and y- position of the OV. On the PDF, a black dashed line in the shape of an ellipse is indicated. This ellipse can shift over the PDF and encircle an area of possible OV positions. The larger the ellipse, the bigger the area that it contains. The size of the ellipse relates to the cumulative probability of the OV being within it, leveraging the PDF characteristics. In other words, avoiding collision with a safety factor β translates to maintaining lateral and longitudinal distances from the OV center.

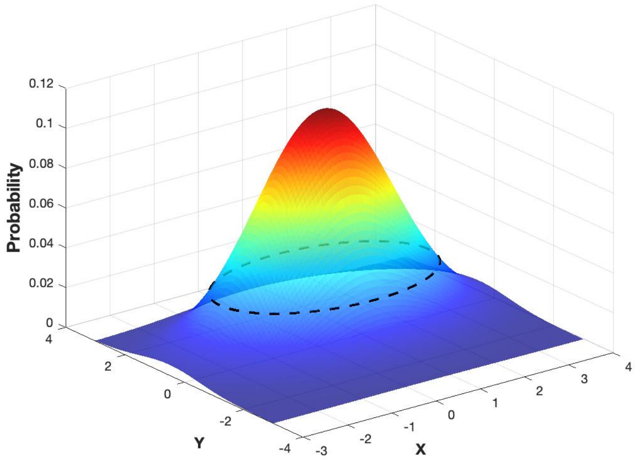


Figure 4-5: Multivariate normal distribution with probability level β indicated by the dotted line.

The integral of this 2-dimensional normal distribution is equal to 1, accounting for all possible OV positions at a certain time step. It is possible to find the size of the ellipse that contains the desired probability level $\beta \leq 1$. This is accomplished by leveraging the properties of the cumulative distribution function $F(\kappa, n)$ of the chi-square distribution χ_n^2. The subscript n is the dimension of the normal distribution, which is in this case equal to 2. With $F(\kappa, 2)$, it is possible to find the area underneath the PDF when it is cut off by the ellipse. A chi-square random variable is characterized by the summation of the squares of standard normal random variables, each distributed independently. This additive property accounts for the behavior of independent chi-square random variables. The probability distribution of a chi-square random variable is modeled by a gamma probability density function, meaning its range is in the right half plane only, since the random variables are squared. For a more detailed explanation of the chi-square distribution, the reader is referred to [50]. If the ellipse is defined as

$$\kappa = (\bar{z}_i - \boldsymbol{x})^T(\bar{P}_i^j)^{-1}(\bar{z}_i - \boldsymbol{x}) \tag{4-21}$$

κ can be found by using the inverse of the cumulative chi-square distribution:

$$\kappa = F^{-1}(\beta, 2) \tag{4-22}$$

Recall that

$$\hat{P}_i = \mathrm{diag}(\sigma_{sx,i}^2, \sigma_{sy,i}^2) \tag{4-23}$$

Then from Equation 4-21, the semi-major and semi-minor axes at time step i are

$$\begin{aligned} a_{\beta,i} &= \sigma_{sx,i}\sqrt{\kappa} \\ b_{\beta,i} &= \sigma_{sy,i}\sqrt{\kappa} \end{aligned} \tag{4-24}$$

$a_{\beta,i}$ and $b_{\beta,i}$ depend on i since the covariance matrix $\bar{P}_i$ changes over time. The further away in the prediction horizon, the larger the uncertainty of the OV prediction $\bar{z}_i$. This also becomes evident from the method that is used for uncertainty propagation, stated in Equation 2-16. In figure 4-6 the influence of the time step i is visualized. The dots indicate the predicted OV states $\bar{z}_k^j$ over the prediction horizon for one maneuver: following the right lane. Around the dots, the ellipses are plotted, and it is clear the size of the ellipses increases over time as the variance increases.

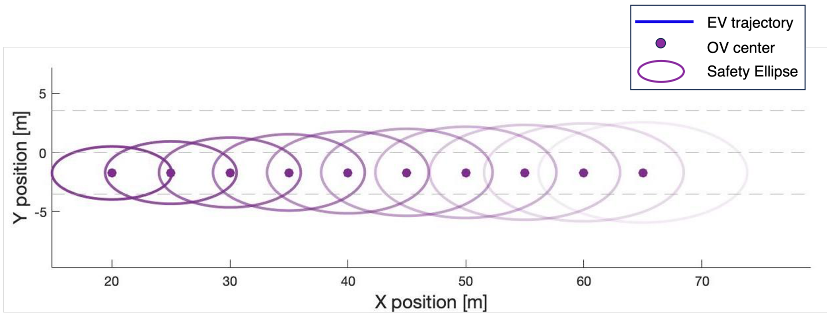


Figure 4-6: Safety ellipses around predicted OV states over the prediction horizon with constant safety level β. Ellipses include sizes of both the EV and the OV, meaning the center of the EV should stay outside the ellipses.

Even though this phenomenon is intuitive since the further away in time, the larger the area where the OV can be with the same certainty level β, adhering to these constraints results in excessively conservative behavior. This approach, when applied, significantly reduces the likelihood of an SMPC solution in which the EV overtakes the OV. Besides being influenced by the increasing covariance, $a_{\beta,i}$ and $b_{\beta,i}$ also depend on β, the safety level. As β increases, the size of the safety ellipses expands as well. To account for the problem imposed by the increasing covariance, the safety level can be decreased over the time horizon to prioritize earlier time steps and to stimulate more optimistic trajectory planning. The result is shown in figure 4-7. This is an improvement to the approach as opposed by Benciolini et al. [15], who implemented constant safety levels β since overly conservative behavior is prevented.

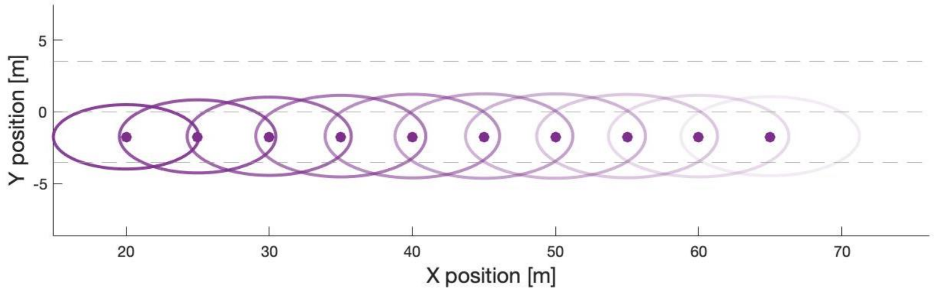


Figure 4-7: Safety ellipses around predicted OV states over the prediction horizon with decreasing safety level β. Ellipses include sizes of both the EV and the OV, meaning the center of the EV should stay outside the ellipses.

The examples in figure 4-6 and 4-7 show the safety regions for one predicted OV trajectory. Recall that at every time step in the simulation horizon k, n_m distinct OV trajectories are computed, and for both trajectories, safety constraints are computed. However, these are highly dependent on the model probability μ^j. In principle, more likely maneuvers should be prioritized over less likely maneuvers. This is done by changing the size of the ellipses based on the model probability in the following way:

$$\begin{aligned} a_i^j &= (l_\text{veh} + \delta_x + a_{\beta,i})\mu_j \\ b_i^j &= (w_\text{veh} + \delta_y + b_{\beta,i})\mu_j \end{aligned} \tag{4-25}$$

By scaling the semi-major and semi-minor axes of the safety ellipse, the constraints relax if a maneuver has a low probability and increase if the IMM returns a high value of μ. Moreover, if the maneuver probability is below a certain level $\mu \leq \mu_\text{low}$, the constraints for that maneuver are not considered. For instance, if the IMM returns the probability distribution $[\mu_L, \mu_R] = [0.1, 0.9]$ for following the left and right lane, constraints are only considered for the OV executing a right lane following maneuver. Compared to the paper by Benciolini et al. [15] who propose a similar constraint design without removing the constraints at low probabilities, this approach prevents the SMPC from planning trajectories around small ellipses around non-relevant OV positions. With the final definition of the safety ellipsoids, the safety constraints $q_{i,j}$ can be defined:

$$q_{i,j} = 1 - \frac{x_{EV,i} - \hat{x}^j_{OV,i}}{(a_i^j)^2} - \frac{y_{EV,i} - \hat{y}^j_{OV,i}}{(b_i^j)^2} \tag{4-26}$$

The number of constraints per OV varies from the number of time steps N_p to $n_\text{m} * N_p$, depending on the model probabilities μ^j. The trajectory planner is designed to avoid static obstacles as well. When an obstacle is detected with near zero velocity in the x- and y-direction, it is considered a static obstacle, and the IMM algorithm is not applied. Instead, a constant ellipse is created around the obstacle

Chapter 5

Backup Motion Planner

The SMPC motion planner is designed for emergency trajectory planning. However, in extreme cases, SMPC might be infeasible, e.g. due to the sudden appearance of static obstacles on the road, resulting in highly complex cases in which the relative velocity of the EV compared to the OV is high. Here, SMPC is considered infeasible if it fails to find a solution that satisfies all constraints within the maximum computational time, recall Definition 3-0.1. In these cases, it is possible to use the shifted optimal input sequence computed in the previous iteration. However, shifting the previous input sequence implies that the newest measurement is neglected, and crucial information is overlooked, potentially leading to a collision. Therefore, a backup controller is designed. The idea of a backup controller is based on [14], where a model predictive motion planner is designed, based on worst-case reachability analysis to serve as a backup mode. The principal idea is to be extremely conservative in emergency scenarios. However, a limitation of this approach is that constraints are further tightened when SMPC fails to find a feasible solution, increasing the challenge of finding a feasible trajectory. At time steps where an FTP solution proves infeasible, the controller resorts to the shifted input from the previous iteration, resulting in a loss of information as explained.

For this research, two backup controllers are designed, with the following objectives:

- Computational times should be low. If SMPC is infeasible, a fraction of the available computational time is already used for SMPC. Therefore, the backup controller should be able to compute collision-avoiding trajectories quickly.
- The solution of the backup controller should be collision risk-minimizing.
- The backup controller should always be feasible, to avoid falling back to the control input computed in the previous iteration.
- The optimal input sequence computed by the backup MPC is used as an initial guess for SMCP in the next iteration. A well-chosen initial guess can significantly accelerate the convergence of SMPC. Therefore, the output of the backup controller should help SMPC with finding a solution in the next iteration.

In the following, the two backup controllers will be discussed. One is based on conventional MPC, the other on SMPC. Both have advantages and disadvantages, which will be compared at the end of this chapter. Validation of the expected behavior of the backup controllers is presented in Chapter 6.

5-1 Backup A: MPC based

The first backup controller is in the form of a conventional MPC, without chance constraints. The mathematical formulation is given below.

$$J = \min_{u} \sum_{i=1}^{N} \mathbb{E}\left[||\Delta\xi_i||_Q^2 + ||u_i||_R^2 + ||\Delta u_i||_S^2 + ||\mathcal{R}||_T^2 \right] \tag{5-1}$$

s.t.

$$\xi_{i+1} = f(\xi_i, u_i) \quad \forall i \in \{0, ..., N\} \tag{5-2a}$$
$$u_i \in \boldsymbol{\mathcal{U}} \quad \forall i \in \{0, ..., N\} \tag{5-2b}$$
$$\xi_i \in \boldsymbol{\Xi} \quad \forall i \in \{0, ..., N\} \tag{5-2c}$$
$$\tag{5-2d}$$

Contrary to SMPC, the computationally demanding chance constraints are not applied in the COP. Instead, the cost function is designed to navigate the EV away from the critical obstacle. The cost function penalizes the error between the reference state ξ_i^* and the planned EV state ξ_i, weighted by Q. The input and input rates are penalized similarly to the SMPC COP. The control structure of trajectory planning with the backup MPC is shown in figure 5-1.

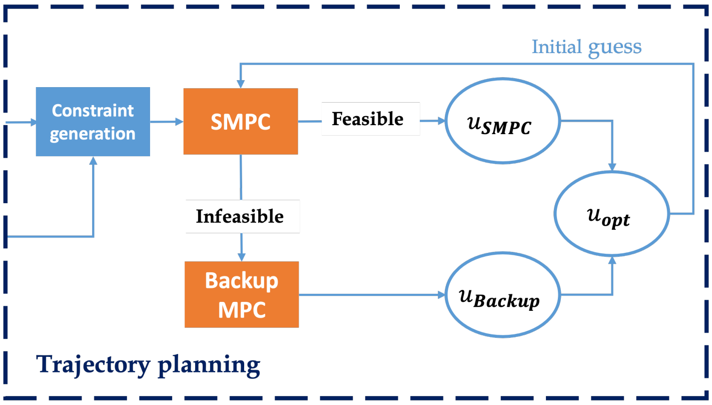


Figure 5-1: Control structure of trajectory planning with a backup MPC in case of infeasibility of the primary SMPC.

The backup solution is recalled only when SMPC cannot find a feasible solution, which only occurs in hazardous situations. This could for instance be caused by a suddenly appearing static obstacle or late detection of an obstacle on the highway. In those scenarios, it is intuitive to fully brake, and steer to the safest possible lane, which can be achieved by setting the reference trajectory as follows:

$$\{\xi_i^*\}_{i=k}^{i=k+N} = [d^*, \phi^*, v^*] = [y_{\mathrm{LC}}, 0, 0]; \tag{5-3}$$

As can be seen from the reference trajectory, the reference velocity is set to zero. It is not always the best solution to come to a full standstill in a dynamic environment with high-velocity dynamic obstacles. However, the emergency backup MPC is only used at time steps in which SMPC is infeasible, and will rarely result in a full stop. Another benefit of applying a braking maneuver during the backup MPC is that it provides more time for the SMPC to compute feasible trajectories again. Furthermore, the shifted optimal control sequence computed by the backup MPC is used as an initial guess for the SMPC COP in the next time step. This is beneficial because it accelerates the convergence of the SMPC, leveraging the insights gained from the backup MPC's solution. Since no computationally demanding chance constraints are added to the MPC COP, the solution is computed quickly and efficiently.

To add an extra safety factor to the MPC COP and to push it to safe solutions, the risk function $\mathcal{R}$ is added to the cost function equivalent to SMPC, penalized with the matrix T. Even though no constraints are added related to the OVs, the risk function prevents solutions with trajectories that bring the system too close to the OVs, mitigating the chance of collision and ensuring a safer path.

The backup MPC requires minimal computational time due to the elimination of the chance constraints. However, a limitation of the backup MPC applies. This backup controller can not guarantee safe trajectories but only assures solutions that minimize the chance of collision.

5-2 Backup B: SMPC based

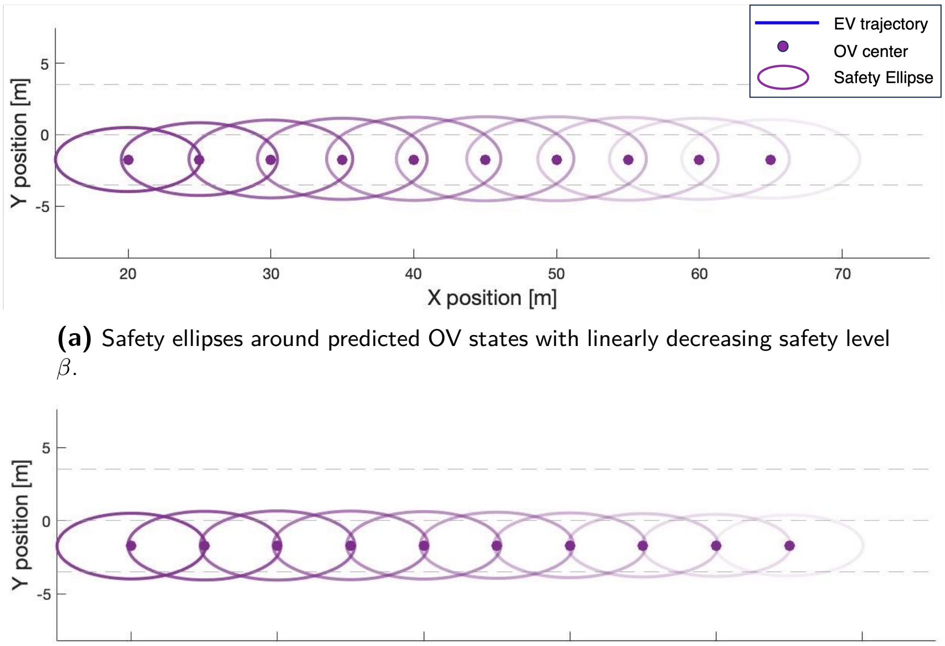
The second backup controller is SMPC-based, very similar to the primary SMPC motion planner. The idea behind this controller is to relax the constraints even further over the prediction horizon to prioritize earlier time steps and to help SMPC find feasible solutions. The mathematical formulation is similar to the COP of the primary SMPC, but it is given for completeness.

$$J = \min_{u} \sum_{i=1}^{N} \mathbb{E}\left[||\Delta\xi_i||_Q^2 + ||u_i||_R^2 + ||\Delta u_i||_S^2 + ||\mathcal{R}||_T^2 \right] \tag{5-4}$$

s.t.

$$\begin{aligned}
\xi_{i+1} &= f(\xi_i, u_i) && \forall i \in \{0, ..., N\} && \text{(5-5a)}\\
u_i &\in \boldsymbol{\mathcal{U}} && \forall i \in \{0, ..., N\} && \text{(5-5b)}\\
\xi_i &\in \boldsymbol{\Xi} && \forall i \in \{0, ..., N\} && \text{(5-5c)}\\
P(q_{i,j} \leq 0) &\geq \beta && \forall i \in \{0, ..., N\}, \forall j \in \{0, ..., n_{ov}\} && \text{(5-5d)}
\end{aligned}$$

When SMPC is infeasible, it is impossible to find a solution that satisfies all constraints in the prediction horizon. Therefore, relaxing constraints in time steps further in the prediction horizon could solve the problem of infeasibility at that time step. As explained in Section 4-6, the safety factor β is already linearly decreased over the prediction horizon to prevent overly conservative behavior. In the backup SMPC, this safety factor β is reduced exponentially, to enhance the effect. Figure 5-2 shows the effect of reducing the safety factor exponentially on the safety ellipses around the OV. At time steps further in the prediction horizon, the ellipse only consists of the vehicles' sizes.



(a) Safety ellipses around predicted OV states with linearly decreasing safety level β.

(b) Safety ellipses around predicted OV states with exponentially decreasing safety level β.

Figure 5-2: Safety ellipses around predicted OV states over the prediction horizon. Transparency increases as time progresses. Ellipses include sizes of both the EV and the OV, meaning the center of the EV should stay outside the ellipses.

Since the safety constraints are still present in the COP of the backup SMPC, it can still run into infeasibility. Therefore, the backup MPC is added as an extra backup. Since the computational times of the backup MPC are very low, this extra backup can be included without adding significant computational complexity. The control structure is shown in figure 5-3.

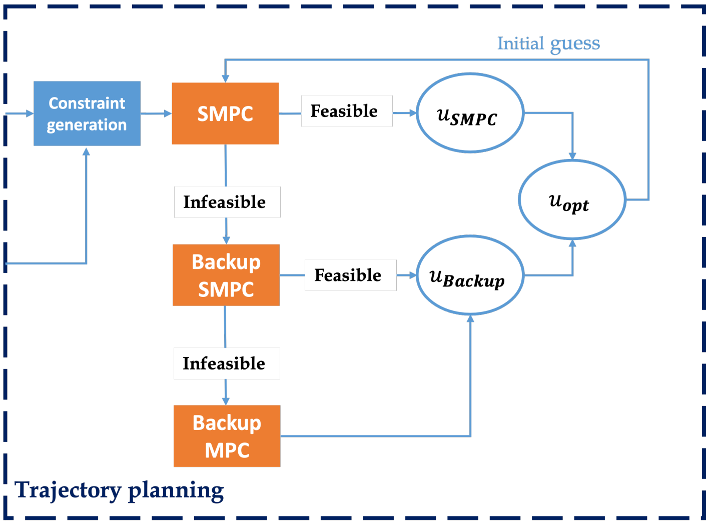


Figure 5-3: Control structure of trajectory planning with a backup SMPC in case of infeasibility of the primary SMPC, and a backup MPC in case of an infeasible backup SMPC.

The major improvement of this backup controller compared to the backup MPC is that the constraints are still present, and safety can therefore be guaranteed with a certain probability level, similar to the primary SMPC. However, one of its limitations is that it is computationally demanding. In the next section, the advantages and disadvantages of both controllers will be listed and explained.

5-3 Comparison Backup Controllers

The performance of the two distinct backup controllers differs across various criteria. To compare their performance, they are evaluated based on the four objectives outlined earlier in this chapter: computational efficiency, collision avoidance, feasibility, and initial guess support. Furthermore, both controllers are compared with the state-of-the-art solution by Brüdigatm et al. [14], who propose FTP as a backup controller. Their fail-safe trajectory planner is based on worst-case reachability analysis. Predictions are made of the OV position if it would apply the maximal or minimal inputs. It imposes tight constraints on the EV trajectory to avoid the total region that could be occupied by an OV, resulting in conservative, careful behavior. Constraints are put on the lateral and longitudinal positions of the EV, dependent on the scenario. Examples of FTP will be given in the results. In figure 5-4 the control structure of SMPC with FTP is shown.

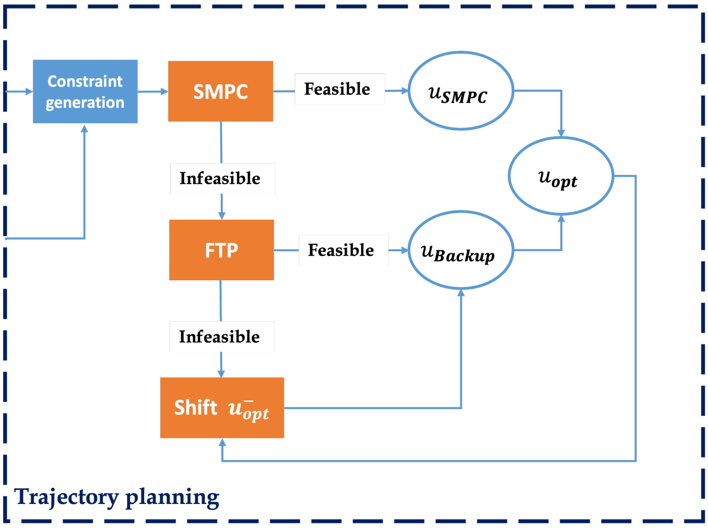


Figure 5-4: Control structure of trajectory planning with FTP as a backup for SMPC, and a shifted input sequence of previous iteration when FTP is also infeasible. State-of-the-art solution by Brüdigam et al. [14].

Table 5-1 provides an overview of how the backup controllers perform with respect to each defined objective.

The primary objective is computational efficiency. Given that SMPC already demands significant computational resources, the backup controller should be fast. In comparison, the backup MPC outperforms the backup SMPC in this aspect, due to the simplicity of its COP. While the computational times of the backup SMPC are similar to those of the primary SMPC, the backup MPC yields solutions within timeframes with a factor of 0.15 compared to SMPC. FTP has the largest computational time. The paper states that the computational times of FTP are on average 1.3 times higher than the computational times of SMPC.

The second objective is collision avoidance. As explained, with the backup SMPC a probability can be guaranteed that a trajectory will be collision-free. With the backup MPC, this is not the case, only the chance of collision is minimized. Arguably, in real life, collision-free trajectories can not always be ensured. Situations may occur in which the distance from the

EV to the obstacle is too small, or in which there is no space for an evasive maneuver, and no feasible solution exists, independent of the active motion planner. Nevertheless, the backup SMPC controller scores higher than the backup MPC controller. FTP is based on worst-case reachability analysis, it accounts for all possible OV maneuvers and executions. In theory, solutions provided by FTP could never result in a collision.

When assessing the feasibility of both backup controllers, the backup MPC scores higher than the backup SMPC. The backup MPC is always feasible, due to the absence of chance constraints. The backup SMPC can still be infeasible within the allowed computational time in the same way the primary SMPC can be infeasible. The main limitation of FTP is its property to impose tighter constraints compared to the primary SMPC. When SMPC struggles to find a solution even with relaxed constraints, the prospect of finding a solution with even tighter constraints becomes significantly more challenging and often infeasible. In the results, examples are shown of this effect.

The last objective is the initial guess support, which refers to the backup controller's capability to provide an initial estimate or prediction that aids the primary controller in generating a feasible solution in subsequent iterations. The backup MPC solves a different COP compared to the primary SMPC and consistently applies a full braking strategy. Therefore, it may explore a solution space that SMPC could not converge to in the previous iterations within the maximum computational time, but might converge to in subsequent ones. The backup SMPC, on the other hand, solves a very similar COP as the primary SMPC, but with relaxed constraints in further time steps. Solutions computed by the backup SMPC are therefore often infeasible for the primary SMPC, resulting in a weak initial guess for the next iteration. In the paper proposing FTP, it is not stated if the backup solution is used as initial guess support. However, it can be reasoned that the FTP would provide a good initial guess for the primary SMPC since the constraints are very conservative and different from SMPC's constraints.

Based on these findings, using backup A: MPC, is preferred over backup B: SMPC.

	Computational efficiency	Collision avoidance
Backup A: MPC	++	0
Backup B: SMPC	-	+
FTP [14]	- -	++

(a) Assessment for computational time and collision-avoidance

	Feasibility	Initial guess support
Backup A: MPC	++	++
Backup B: SMPC	-	- -
FTP [14]	- -	+

(b) Assessment for feasibility and initial guess support

Table 5-1: Assessment of the two backup trajectory planning algorithms based on the four objectives. '- -' indicates a significant limitation, '-' represents a moderate limitation, '0' denotes a neutral performance, '+' signifies a moderate strength, and '++' indicates a significant strength.

Chapter 6

Simulation and Results

In this chapter, the results of the proposed method combining IMM with SMPC and the backup MPC are presented. The motion planner is tested in emergencies with both static and dynamic obstacles, and the scenarios are carefully selected to test its performance. Firstly, we provide the scenarios for testing the motion planner and provide details of the simulation setup. Subsequently, the resulting trajectories of the EV within these defined scenarios are presented and discussed.

The following results are presented:

- The performance of the standalone SMPC motion planner without activating a backup controller, tested in all 6 defined scenarios.
- In scenario 1, two extra results are shown: the effectiveness of the helping reference, and the result of linearly decreasing the safety factor β over the prediction horizon.
- The standalone performance of backup controller A and the performance of SMPC with backup controller A, both in scenario 5a.
- The performance of SMPC with backup controller B in scenario 5a.
- Comparison of the motion planner with the state-of-the-art solution in [14].

The simulations are executed in MATLAB using the nonlinear solver `fmincon`. The optimization algorithm is the interior-point algorithm, which is an iterative optimization method that seeks the solution to linear and nonlinear programming problems by iteratively navigating the feasible region. The simulations are run on a computer with a Dual-Core Intel Core i5 processor. The setup parameters are as follows. For both the SMPC and the backup MPC, the sampling time is set to $h = 0.2$. The prediction horizon for SMPC and the backup MPC consists of 10 time steps, $N_p = 10$, and simulations run for 5 seconds.

In this chapter, various figures like Figure 6-1 are shown. To ensure a correct interpretation of these figures, they are explained. In these figures, the rectangles represent the position of the vehicles at different time steps. On the first rectangle, it is shown what rectangle color represents what vehicle. The EV is always shown in blue. The subsequent rectangles are numbered to indicate the vehicles' positions relative to each other at a certain time step. Collision-free trajectories can be recognized by the absence of overlapping rectangles with the same number. In the example, the 4th blue rectangle coincides with the 5th and 6th red rectangle. However, these are vehicle positions at different time steps and therefore not indicate collision. The vehicle positions were plotted once every three-time steps, and sequential numbers (1, 2, 3, 4, etc.) were assigned to differentiate between different snapshots. However, it's essential to note that these assigned numbers do not align with consecutive time steps in the simulation; instead, they correspond to time steps separated by three intervals (1, 4, 7, etc.). All simulations are on a 2-lane highway, the grey dashed lines correspond with the road boundaries. Note that the axes are not equal.

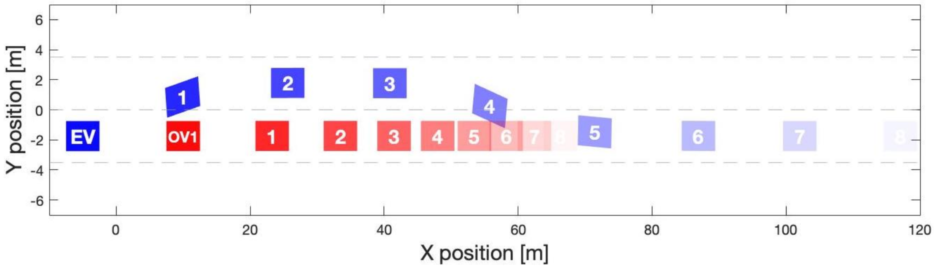


Figure 6-1: Example of results figure

6-1 Defined Emergency Scenarios

To assess the performance of the motion planner, it is tested across various scenarios on a 2 lane highway. All emergency scenarios are intentionally designed to be challenging but not impossible. This ensures that within each scenario, a feasible trajectory exists for the EV allowing safe navigation, thereby avoiding collisions with OVs, and adhering to the EV dynamics. The complexity of scenarios increases significantly when the relative velocity between the EV and obstacles is large. This arises from the necessity to plan and execute an evasive maneuver within a limited space and time frame. Consequently, all scenarios incorporate one or more slow-moving or static obstacles to simulate real-world challenges and evaluate the planner's performance under varying degrees of difficulty. In all scenarios, the initial velocity of the EV is set to $25m/s$, the lateral position to one of the lane centers, this depends on the scenario. The following 6 scenarios are designed for the assessment of the motion planner:

- Scenario 1: large deceleration of an OV in front of the EV on the same lane. This situation occurs when an OV suddenly applies a full braking maneuver.
- Scenario 2: a static obstacle suddenly appears in front of the EV. When detected late, braking is not sufficient, and overtaking is required.
- Scenario 3: an OV driver may lose consciousness, resulting in arbitrary, unpredictable, and decelerated movements.

- Scenario 4: one OV decelerates in front of the EV like in scenario 1, but the other lane is not free due to another OV overtaking the EV.
- Scenario 5a & 5b: one static OV in front of the EV like in scenario 2, but another OV blocks the optimal trajectory.
- Scenario 6: two static obstacles on the road.

6-2 SMPC Performance

In this section, the results are shown when only SMPC is utilized to clearly indicate its performance. The backup controller is in these examples not activated. To obtain these results, the limits that are put on the maximum computational effort are relaxed, so the chance of feasibility is high. In all scenarios, SMPC succeeds in finding a collision-free trajectory, without the use of a backup controller. All scenarios are discussed separately.

6-2-1 Scenario 1: single decelerating obstacle

The first scenario involves an OV that suddenly starts decelerating in front of the EV. In figure 6-2, the trajectories of the EV and OV are shown. The EV immediately starts moving to the left lane to avoid collision with OV1 without overly decelerating. Since the risk function $\mathcal{R}_i$ is minimized at every time step i in the prediction horizon, the optimal EV trajectory converges to the left lane instead of staying behind the OV. Another reason for the EV to move to the left lane is the reference lane, determined through threat assessment. As soon as the EV passes the OV, the threat on the right lane is zero again, and it returns to the right lane center.

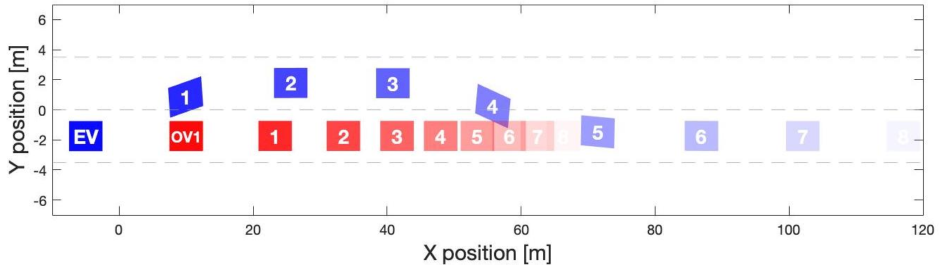


Figure 6-2: Scenario 1: preceding OV in front of the EV braking aggressively. The EV overtakes.

Helping reference performance

To indicate the performance of the helping reference, as explained in Section 4-5, the result of the EV trajectory in scenario 1 is shown when the reference is chosen as the center of the lane the EV is currently in. Figure 6-3 shows the EV and OV trajectory when the helping reference is not applied. In this case, the EV stays behind the OV instead of overtaking.

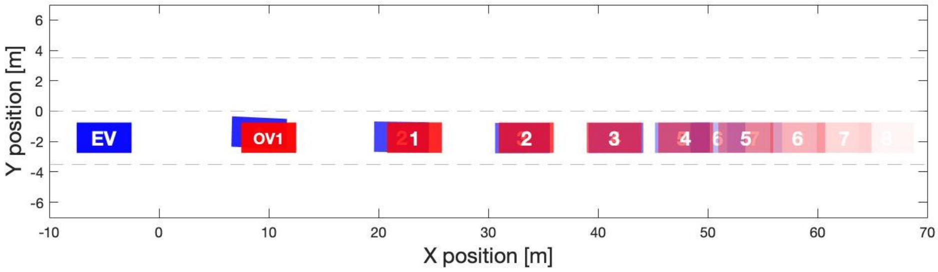


Figure 6-3: Scenario 1 without helping reference: preceding OV in front of the EV braking aggressively, the EV stays behind and applies a full braking maneuver.

Table 6-1 illustrates the reference lane at different time instances under two scenarios: 1 when the assisting reference is applied and 2 when the EV is encouraged to maintain its lane. For the sake of simplicity, the reference lanes are only shown once every three time steps, therefore corresponding with the numbers indicated at the rectangles in figure 6-2 and 6-3. In the table, R and L indicate that the reference is set to the right and left lane center respectively.

Time instance	1	2	3	4	5	6	7	8
Helping reference	L	L	L	R	R	R	R	R
Keep current lane	R	R	R	R	R	R	R	R

Table 6-1: Comparison of reference lane when the helping reference is applied or the EV is encouraged to keep its current lane. Time instances coincide with numbers on the rectangles in figure 6-2 and 6-3.

Effect of varying safety factor β

Scenario 1 is used to give an example of the functionality of the decreasing safety factor β over the time horizon, as explained in Section 4-6. Figure 6-4 shows the trajectory in scenario 1c if the safety factor β is not linearly decreased over the prediction horizon N. The safety ellipses around the estimated OV position in further time steps become too large to plan an overtaking trajectory.

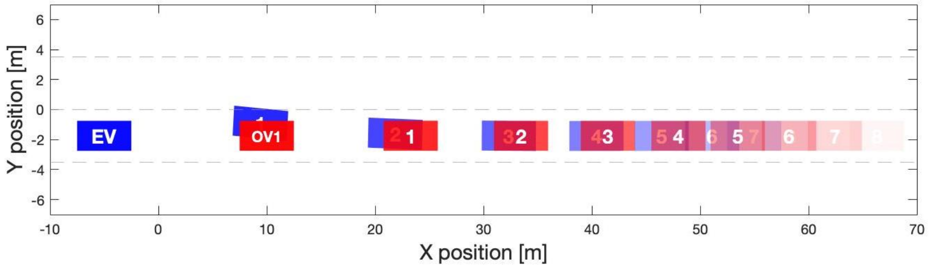
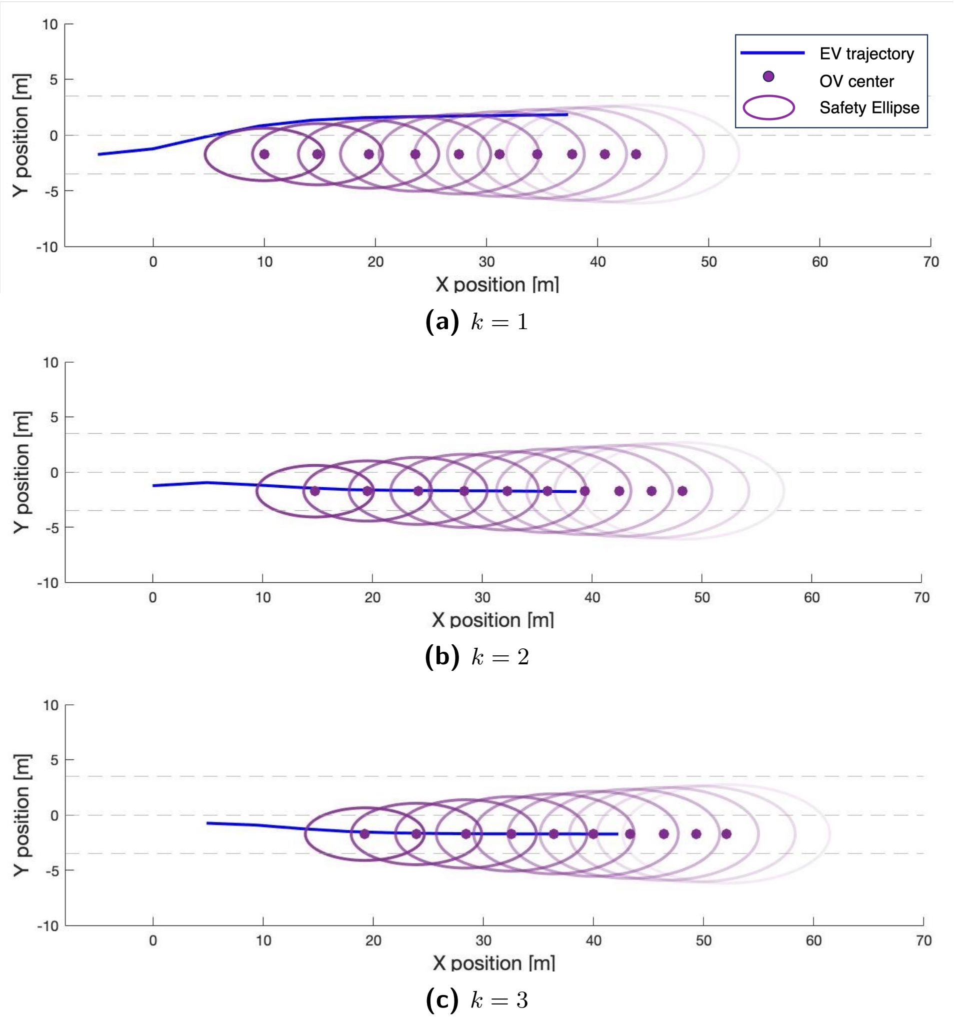


Figure 6-4: Scenario 1 without linearly decreasing safety level β: preceding OV in front of the EV braking aggressively, the EV stays behind and applies a full braking maneuver.

It can be seen that at first, the EV tries to move to the left lane, but then recognizes there is no feasible path overtaking the OV, decelerates, and moves back to the right lane. Due to the aggressive braking of the EV, the TTC with the OV becomes undefined as explained in 4-5, stimulating the reference lane to be adjusted to the right lane. To indicate what happens inside the planning algorithm at every time step, the planned EV trajectory over the prediction horizon at the first three time steps is shown together with the safety ellipses it should avoid. As stated, the ellipses include the EV and OV size, so the center of the EV should stay outside the ellipses. Figure 6-5 shows the planned EV trajectory over the prediction horizon at $k = 1$, $k = 2$, and $k = 3$ when β is not decreased over time. The blue line indicates the planned EV path. Collision avoidance can be recognized by looking at the first and last point of the EV trajectory, which should be outside the first and last ellipse with the lowest and highest opacity respectively. To maintain plot simplicity, all time steps in the planned EV trajectory are uniformly represented.



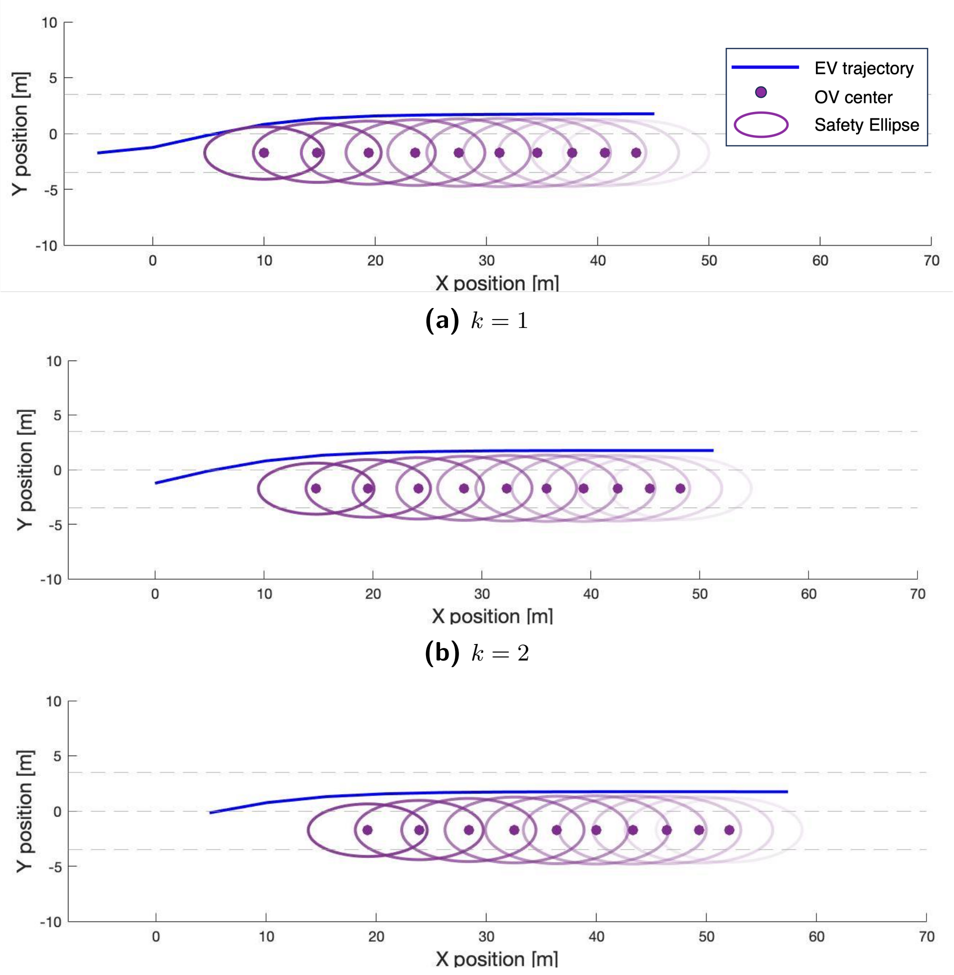
(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

Figure 6-5: Planned EV Trajectories over prediction horizon with constant β at the first three time steps. The EV trajectory is indicated in blue. Safety ellipses around OV are shown for every time step with decreasing opacity as time progresses.

Figure 6-6 shows the intermediately planned EV trajectories at the first three time steps when β is linearly decreased over time. An overtaking maneuver is found and executed.



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

Figure 6-6: Planned EV Trajectories over prediction horizon with linearly decreasing β at the first three time steps. The EV trajectory is indicated in blue. Safety ellipses around OV are shown for every time step with decreasing opacity as time progresses.

6-2-2 Scenario 2: single static obstacle

In figure 6-7 the scenario is shown when a static obstacle appears in front of the EV. In this case, there is no other option for the EV than to move to the left lane to avoid collision, since the initial distance between the OV and the EV is smaller than the minimal braking distance of the EV. From the figure, it can be seen that the EV successfully avoids collision with the static obstacle. As it passes the static OV, it returns to the right lane. The figure illustrates the successful avoidance of collision by the EV with the static obstacle. After the EV passes the static OV, it promptly returns to the right lane. The figure demonstrates the immediate initiation of an evasive maneuver by the EV upon detection of the OV within the prediction horizon.

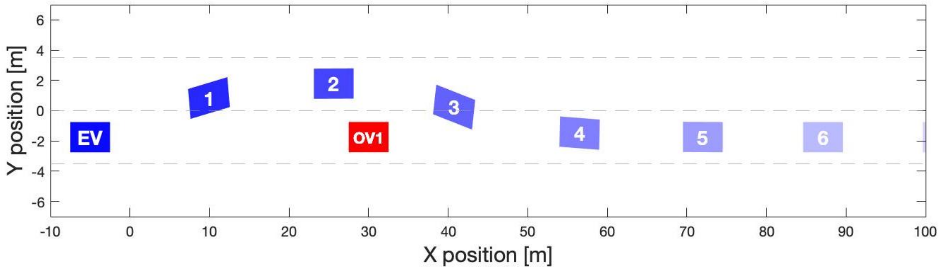


Figure 6-7: Scenario 2: static OV in front of EV. The EV performs an evasive maneuver and overtakes.

6-2-3 Scenario 3: ambiguous OV behavior

This scenario may occur when the driver of an OV loses consciousness or control of the vehicle due to other reasons. Consequently, the OV begins to move unpredictably and decelerates. In such instances, the driver's intentions are unclear, and the IMM algorithm provides no clear maneuver intention. Within the SMPC OCP, the strategy is to maintain a position behind an OV exhibiting ambiguous behavior, as explained in Subsection 4-5. This strategy is illustrated in Figure 6-8. Instead of actively seeking an overtaking trajectory as in scenarios 1 and 2, the EV decelerates and strategically positions itself behind the ambiguously behaving OV. The average computational time of SMPC with one OV present is 0.55 seconds.

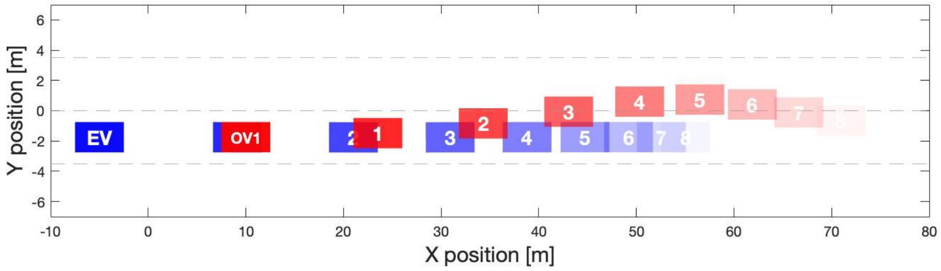


Figure 6-8: Scenario 3: OV with ambiguous behavior driving in front of the EV. The EV strategically positions itself behind the OV and avoids overtaking.

6-2-4 Scenario 4: decelerating OV1 + overtaking OV2

The next scenarios involve multiple OVs. Figure 6-9 shows an extension of scenario 12 and introduces an additional OV executing an overtaking maneuver on the EV. Unlike scenario 1, where it was safe to navigate to the left lane, in this instance, the EV must brake and wait before overtaking OV1 to avoid a potential collision with OV2. The figure illustrates the sequence of actions, wherein the EV initially decelerates and follows behind OV1, waiting for OV2 to complete its overtaking maneuver. Once moving to the left lane is safe, the EV proceeds to overtake OV1. Adding an extra OV in SMPC increases the computation time, which is now 0.82 seconds.

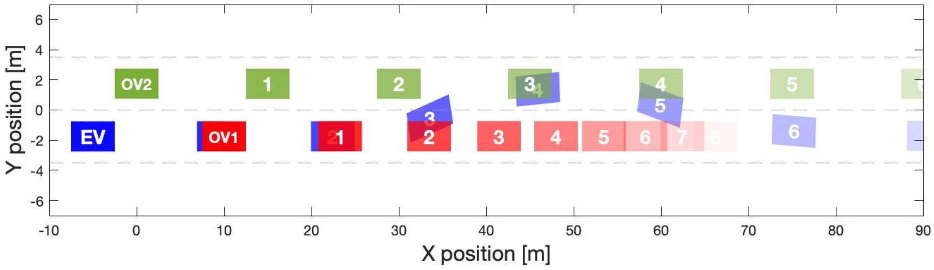
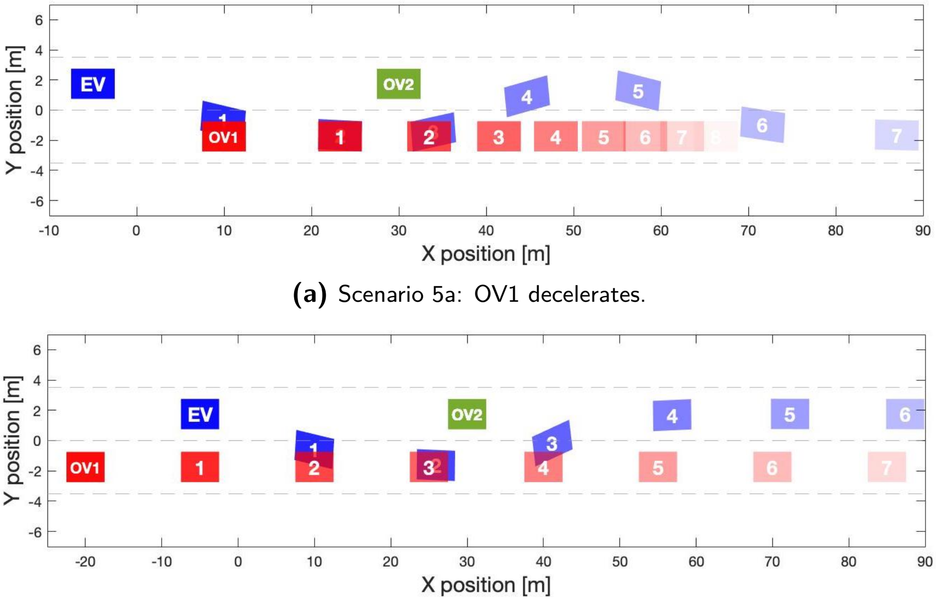


Figure 6-9: Scenario 4: OV1 decelerates in front of the EV while OV2 overtakes the EV.

6-2-5 Scenario 5: dynamic OV1 + static OV2

In the following, two scenarios are designed with a dynamic and a static OV. Similar to scenario 2, the EV has to move around OV1, since the distance between the EV and OV1 is smaller than the minimal braking distance. In both scenarios, the EV has to perform an evasive maneuver to avoid collision with OV2 but adjust the maneuver according to the moving OV1. The first sub-scenario, 5a, is presented in figure 6-10a, where OV1 decelerates and the EV slows down while performing the evasive maneuver. Figure 6-9 shows scenario 5b, the case in which OV1 approaches with high velocity, and the EV has to quickly shift to the right lane to stay in front of OV1 and avoid collisions with both OVs.



(a) Scenario 5a: OV1 decelerates.

(b) Scenario 5b: OV1 approaches with high velocity.

Figure 6-10: Scenario 5: OV1 is dynamic and OV2 is static.

6-2-6 Scenario 6: two static OVs

In the last scenario, 2 static OVs are presented to the EV. The EV immediately starts navigating to the right lane to avoid OV2 and then moves back to the left lane to avoid OV1. After both obstacles are passed, it moves to the right lane again, to adhere to general traffic rules.

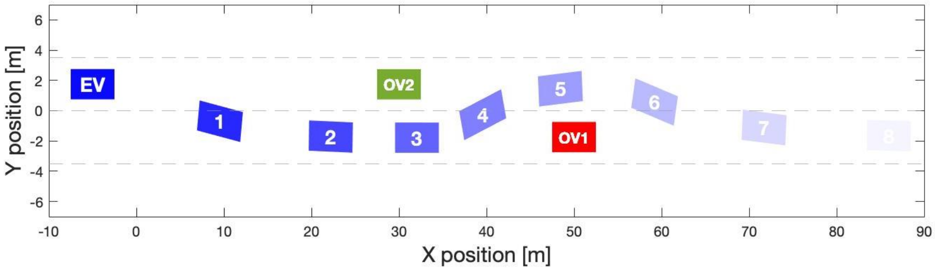


Figure 6-11: Scenario 6: both OV1 and OV2 are static.

6-3 Backup Controller Performance and Comparison

This section shows the performance of the designed backup controllers. Across all presented scenarios, the SMPC motion planner consistently achieved feasible solutions within the allocated computational time. To intentionally activate the backup controller, the maximum computational time of SMPC is decreased. Then, the likelihood of SMPC running into infeasible solutions increases. The results will be shown separately in Subsection 6-3-1 and 6-3-2. Then, the performance of both backup controllers is evaluated in comparison to each other and the current state-of-the-art. All backup controllers are tested in scenario 5a.

6-3-1 Backup A: MPC

The backup MPC is a fast solution that provides solutions that steers the EV in the direction of the safest possible lane while fully braking. Figure 6-12 shows the optimal trajectory if *only* the backup MPC is utilized for motion planning. From the figure, it can be seen that the backup MPC navigates the EV to the safest possible lane at every time step k and decelerates. It succeeds in providing a collision-free trajectory for the EV and eventually comes to a standstill.

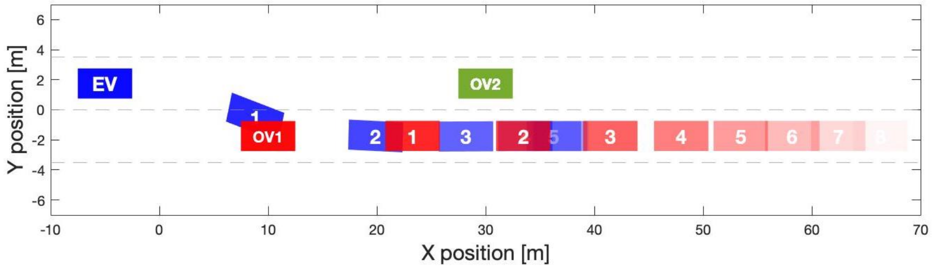
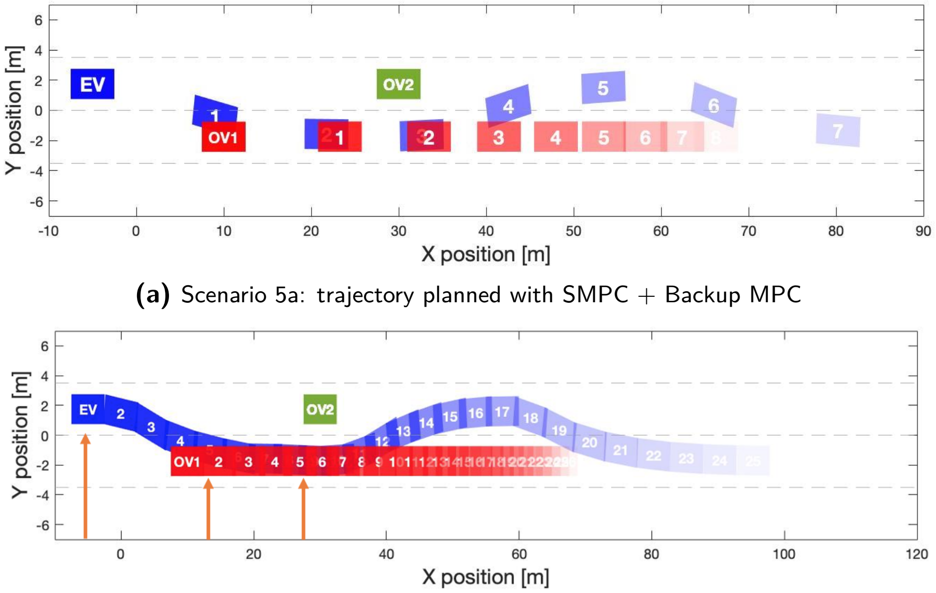


Figure 6-12: Scenario 5a: trajectory planning by the backup MPC only.

In figure 6-13a the optimal EV trajectory is shown when SMPC is active, and the backup MPC is activated at 3 time instances in the simulation horizon, at $k = 1$, $k = 5$, and $k = 8$. Figure 6-13b shows the snapshots of the vehicles at every time step in the simulation horizon. In this figure, the numbers in the rectangles indicate the related time step. The orange arrows indicate the positions of the EV at the instances at which the backup MPC is activated. Compared to the performance of the SMPC planner only in figure 6-10a, it can be seen that due to the property of the backup MPC slowing down, the EV passes OV2 with a slightly lower velocity. In both cases, collision is avoided successfully. The computational times of the backup MPC are significantly lower than the computational times of the primary SMPC: 0.07 seconds.



(a) Scenario 5a: trajectory planned with SMPC + Backup MPC

(b) Scenario 5a: SMPC + Backup MPC. Orange arrows indicate time instances at which the backup MPC is activated: at $k = 1$, $k = 5$, and $k = 8$.

Figure 6-13: Scenario 5a when the backup MPC is activated in time steps 1, 5, and 8.

6-3-2 Backup B: SMPC

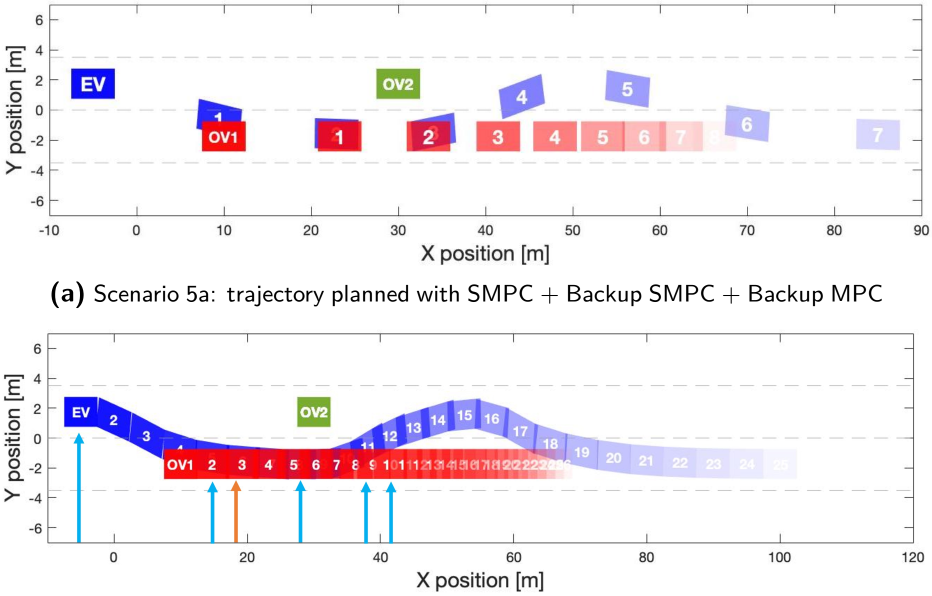
The second backup controller is based on SMPC. If the primary SMPC runs into infeasibility, the backup SMPC is activated and solves the same optimization problem with relaxed constraints in further time steps in the prediction horizon. Since there is a possibility the backup SMPC runs into infeasibility as well, the backup MPC can be called if both SMPC OCPs are infeasible within the maximum amount of time.

In figure 6-14, the EV trajectory in scenario 5a is shown when the primary SMPC is combined with the backup SMPC and the backup MPC. Figure 6-14a does not show major differences from figure 6-13a. However, by the position of the EV at the 7th snapshot, it can be seen the backup SMPC provides a trajectory with higher velocities. This is due to the property of the backup MPC to slow down the vehicle and apply a full braking maneuver.

Figure 6-14b shows when the backup SMPC and the backup MPC are called with two different arrows:

- The blue arrow indicates that at that time step, the primary SMPC was infeasible, and the backup SMPC succeeded in computing a feasible solution instead.
- During time steps marked by the orange arrow, both the primary SMPC and the backup SMPC were unable to produce feasible solutions, activating the backup MPC to compute the solution instead.

In steps $k = 1$, $k = 5$, $k = 11$, and $k = 12$, the backup SMPC was activated and able to find a feasible solution. In time step $k = 6$, both SMPCs were infeasible, and the backup MPC was activated.



(a) Scenario 5a: trajectory planned with SMPC + Backup SMPC + Backup MPC

(b) Scenario 5a: SMPC + Backup SMPC + backup MPC. Blue arrows indicate time instances at which the backup SMPC is activated and feasible $k = 1$, $k = 5$, $k = 11$, $k = 12$. Orange arrows indicate time instances where both SMPCs are infeasible, and the backup MPC is applied $k = 6$.

Figure 6-14: Scenario 5a when the backup MPC is activated in time steps 1, 5, and 8.

By comparing figure 6-13b and 6-14b, conclusions can be drawn regarding the initial guess support for the primary SMPC in the next iteration. With only the MPC motion planner as backup, the primary SMPC ran into infeasibility 3 times. With the backup SMPC however, a backup was needed at 6 time steps in the simulation horizon, which is double compared to the backup MPC. Therefore, it can be concluded that the backup MPC provides better initial guess support than the backup SMPC, as expected according to the theoretical analysis in Section 5-3. The difference between the computation time of the primary SMPC and the backup SMPC is minimal.

An overview of the computation times of the controllers is given in table 6-2. Please note that with faster solvers and a more advanced processor, it is possible to reduce the computation times significantly [14].

Controller	SMPC with 1 OV	SMPC with 2 OVs	Backup MPC	Backup SMPC
Computation time	0.55s	0.82s	0.07s	0.53s

Table 6-2: Computation times of the designed controllers

6-3-3 Comparison to State-of-the-art

In this section, a comparison will be made with the state-of-the-art solution by Brüdigam et al. [14]. It will be shown that in scenario 5a, their proposed controller would not be able to find a feasible solution.

In the paper by Brüdigam et al., the fail-safe trajectory planner is proposed as a backup controller based on the fail-safe motion planner in [51]. The purpose of this controller is to safely navigate the EV through the dynamic environment in emergencies when SMPC fails to compute a solution that is safe according to their definition. In FTP, lateral and longitudinal constraints are designed based on reachability analysis. The authors propose constraints based on the position of the EV with respect to the OV and define distinct constraints for each case. For the complete overview of constraints, the reader is referred to the associated paper [14]. If FTP were applied in time step 1 of scenario 5a, the constraints imposed by OV1 and OV2 would look like figure 6-15.

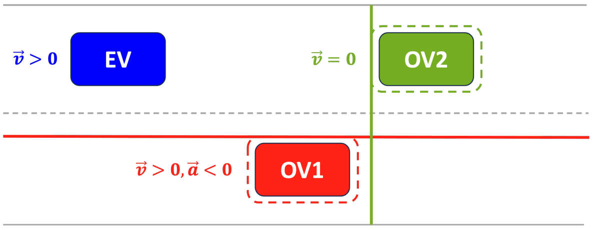


Figure 6-15: Constraints on the EV trajectory imposed by OVs when using FTP of [14].

The dynamic, decelerating OV1 would impose a lateral constraint on the EV and the static OV2 would impose a longitudinal constraint. However, with the initial velocity of the EV and the longitudinal distance between the EV and OV2, finding a trajectory that satisfies the constraints at all steps in the prediction horizon is not possible. Consequently, FTP is infeasible and according to [14], the first entry of the shifted optimal input sequence of the previous iteration is applied. However, it is assumed that OV2 is only detected at the current time step, and the optimal input sequence of the previous iteration does not consider the obstacle. The previously computed safe input sequence is designed such that the first input corresponds with the one computed by SMPC, followed by a braking sequence to bring the EV to a standstill:

$$
\begin{aligned}
\boldsymbol{U}_{\text{safe}} &= [\boldsymbol{U}'_{\text{FTP}}\ , \boldsymbol{U}_{\text{brake}}\ ] \\
\boldsymbol{U}_{\text{brake}} &= \left[\begin{bmatrix} a_{\min} \\ 0 \end{bmatrix}, \begin{bmatrix} a_{\min} \\ 0 \end{bmatrix}, \ldots \right]
\end{aligned}
\tag{6-1}
$$

Even though this safe input sequence would guarantee safety in an environment without static or very slow-moving obstacles, it is not sufficient to ensure safety in this scenario. In the presented case, FTP stays infeasible and the planner keeps applying the shifted previous input sequence. Even though the EV fully brakes, the braking distance is too short, and eventually a collision is caused. This is shown in figure 6-16.

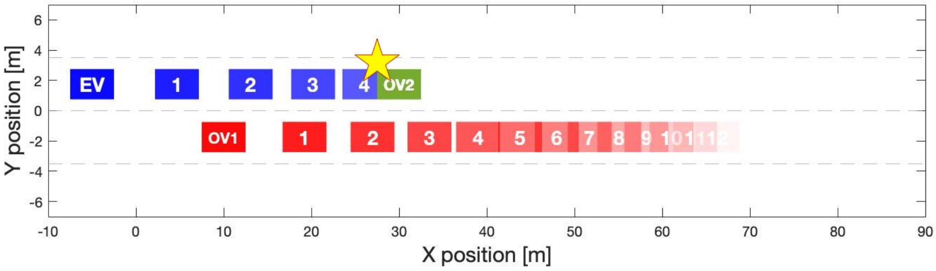


Figure 6-16: Scenario 5a: SMPC combined with Fail-safe trajectory planning as proposed by [14]. At all time steps, SMPC and FTP are infeasible, and the optimal input of the previous iteration is applied. Eventually, the EV collides with OV2, indicated by the yellow star.

The failure of FTP shows the improvement of our proposed method compared to the prior state-of-the-art work, where FTP takes over SMPC in emergencies. Figure 6-16 highlights the shortcomings of SMPC + FTP in scenario 5a, and the same reasoning can be used to argue FTP fails to compute collision-free trajectories in scenarios 2, 5b, and 6.

Chapter 7

Discussion

Distinctive Features and Advantages of the Proposed Method

The proposed motion planner, incorporating the IMM algorithm, SMPC, and the backup MPC, consistently generates collision-free trajectories across various challenging emergency scenarios. It adeptly navigates through environments with both static and dynamic obstacles, even under conditions where obstacles are detected late or initial distances between the EV and obstacles are minimal. The ability of the designed SMPC controller to safely navigate the EV in a wide variety of emergency scenarios highlights the first contribution of this paper which states that SMPC is specifically designed for evasive maneuvers in emergencies. By minimizing the collision risk and smartly adapting to the environment with the use of IMM, the SMPC motion planner succeeds in finding collision-avoiding solutions in all designed scenarios.

Furthermore, the results show a significant improvement in the designed backup controller compared to the state-of-the-art. By looking at the resulting trajectories of the backup controllers in the same scenario, it is clear that both designed backup controllers succeed in finding collision-free trajectories where FTP fails. This success highlights a notable improvement compared to the prior state-of-the-art work, where FTP takes over SMPC in emergencies [14]. Subsection 6-3-3 highlights the shortcomings of SMPC + FTP in scenario 5a, and the same reasoning can be used to argue FTP fails to compute collision-free trajectories in scenarios 2, 5b, and 6. In all instances where the relative distance and velocity concerning the preceding OV indicate that a complete braking maneuver alone is insufficient to prevent a collision, FTP would be infeasible. Consequently, the shifted previously computed input sequence in which the obstacle was not accounted for yet, is applied and the vehicles collide. This analysis proves the second contribution of this paper: to design a fast, reliable backup controller that ensures risk-minimizing solutions.

Limitations

However, some limitations of the proposed method apply. The backup controller can not guarantee safe trajectories but only assures solutions that minimize the chance of collision.

Arguably, in real life, collision-free trajectories can not always be ensured. Situations may occur in which the distance from the EV to the obstacle is too small, or in which there is no space for an evasive maneuver, and no feasible solution exists, independent of the active motion planner. Yet, to account for this limitation, a second backup controller based on SMPC was designed that could ensure collision-free trajectories with a safety level β. Still, Section 5-3 and 6-3-3 reason why the backup MPC is preferred over the backup SMPC: compared to the backup SMPC: it provides risk-minimizing trajectories with the lowest computational effort, provides the best initial guess support, and is always feasible.

Additionally, the computational times of SMPC exceed the prescribed maximum of 0.2 seconds, the sampling time. Yet, faster solvers exist that can improve computational times significantly. Furthermore, the processor used for the simulation limits the computational time as it is not designed for complex optimization tasks.

Suggestions for future work

For future work, different backup controller designs would be exploited that could guarantee safety in the majority of emergencies. Research into backup controllers that effectively balance computational efficiency and safety guarantees would be relevant.

Furthermore, further research could be done exploring the enhancement of the computational efficiency of the motion planner. An inspiring paper is [52], where max-min-plus-scaling (MMPS) approximations of the nonlinearities of the EV dynamics are made to avoid conservatism and improve computational efficiency. In their paper, MMPS is also utilized to approximate the nonlinear density function around predicted OV positions, which might reduce computational times for our proposed motion planner as well.

In this research, it is assumed the fidelity of the kinematic bicycle model is sufficient for the prediction of the EV dynamic behavior. However, future work could assess whether the trajectories computed by the designed motion planner are feasible for a tracker, recall figure 1-1. Although the bicycle model was implemented to estimate the vehicle's limits, no explicit validation or verification was conducted. This could be addressed in future studies by incorporating checks for tracker feasibility. Additionally, note that considering a variable tire-road friction was neglected in this research. While this work primarily focused on maneuvering in an uncertain environment, future investigations could explore integrating friction estimation into the modeling framework.

Chapter 8

Conclusion

In conclusion, this research presented a framework to safely navigate an automated vehicle in a dynamic, uncertain environment in highway emergencies. It presents a novel design of SMPC for collision-avoiding motion planning, and a backup controller to provide solutions when SMPC is infeasible. On one hand, this research contributed to the existing literature in exploring SMPC designs specifically for emergency motion planning, by prioritizing collision avoidance and minimizing risk. On the other hand, the design of a backup controller complements the contributions of this research by providing fast, feasible solutions and a supportive initial guess to SMPC in subsequent iterations. It thereby outperforms the state-of-the-art solution of Brüdigam et al. [14], whose proposed method combining SMPC with FTP falls short in complex emergency scenarios.

Based on the findings in this research, the following conclusions regarding the primary SMPC controller can be drawn.

A) In addition to the state-of-the-art application of SMPC where it is employed for optimistic motion planning for general urban [15] and highway [14] automated driving, SMPC can successfully be utilized for emergency motion planning.

B) The proposed motion planner can safely navigate the EV in a wide variety of complex emergency scenarios. Scenarios include static obstacles, dynamic obstacles, OVs with ambiguous behavior, and late-detected obstacles. The results in Section 6-2 show its success.

C) By providing a helping reference to the motion planner based on threat assessment, the chance of finding collision-free trajectories is increased. With a proper chosen reference trajectory, the cost function decreases when the optimization algorithm explores the space of the safest possible lane, resulting in a faster convergence to collision-avoiding trajectories. Furthermore, this approach shows its success in backup controller A, which highly depends on the reference trajectory.

D) The safety level β is linearly decreased over the time horizon to prioritize earlier time steps and to stimulate optimistic trajectory planning. This positively affects the feasibility of collision avoidance since it expands the search space of the optimization algorithm with a minimal increase in collision risk.

The following conclusions can be drawn through the exploration of two different backup controllers and a comparison to the state-of-the-art.

E) Based on the four identified objectives of a backup controller in Chapter 5 and the results shown in Section 6-3-3, it can be concluded backup A, based on MPC, outperforms backup B, based on SMPC, and is the better option as a backup for the designed primary SMPC motion planner.

F) Backups A and B are a superior choice to the state-of-the-art solution of employing FTP. By relaxing constraints instead of tightening constraints, the feasibility of the designed backup controllers is enhanced significantly. Section 6-3-3 highlights one of the failure cases of FTP in a complex scenario with a late-detected static, and a dynamic obstacle.

To conclude, the successful design and implementation of SMPC for emergency motion planning have yielded promising results in simulation. While numerous challenges remain on the journey towards safer roads, this study represents a meaningful step forward in the collective pursuit of enhancing mobility and decreasing the frequency of traffic accidents.

Appendix A

EV Bicycle Model

The linearized, discretized system matrices of the ego vehicle are given in equation A-1. Linearization is done around the current operating point ξ_k and zero input $u = [0, 0]^T$. The method used for discretizing is the zero-order-hold method.

$$\begin{aligned}
\mathbf{\Phi} &= \begin{bmatrix} 1 & 0 & -Tv\sin p_1 & T\cos p_1 - \frac{p_2 \sin p_1}{2p_4} \\ 0 & 1 & Tv\cos p_1 & T\sin p_1 + \frac{p_2 \cos p_1}{2p_4} \\ 0 & 0 & 1 & \frac{T\tan\delta}{p_4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{\Gamma} &= \begin{bmatrix} \frac{T^2\cos p_1}{2} & -\frac{T^2 v p_7 \sin p_1}{2} - \frac{p_8 \sin p_1}{p_9} \\ \frac{T^2\sin p_1}{2} & \frac{T^2 v p_7 \cos p_1}{2} + \frac{p_8 \cos p_1}{p_9} \\ \frac{T\tan\delta}{2p_4} & Tp_7 \\ T & 0 \end{bmatrix}
\end{aligned} \tag{A-1}$$

$$
\begin{aligned}
p_1 &= \phi + \arctan\left(\frac{l_\mathrm{r}\tan\delta}{l_\mathrm{r}+l_\mathrm{f}}\right)\\
p_2 &= T^2 v \tan\delta\\
p_3 &= \left(l_\mathrm{r}\tan\delta\right)^2\\
p_4 &= \left(l_\mathrm{r}+l_\mathrm{f}\right)\left(\frac{p_3}{\left(l_\mathrm{r}+l_\mathrm{f}\right)^2}+1\right)^{\frac{1}{2}}\\
p_5 &= v\left(\left(\tan\delta\right)^2+1\right)\\
p_6 &= \left(l_\mathrm{r}+l_\mathrm{f}\right)^3\left(\frac{p_3}{\left(l_\mathrm{r}+l_\mathrm{f}\right)^2}+1\right)^{\frac{3}{2}}\\
p_7 &= \frac{p_5}{p_4}-\frac{p_3 p_5}{p_6}\\
p_8 &= T l_\mathrm{r} p_5\\
p_9 &= \left(l_\mathrm{r}+l_\mathrm{f}\right)\left(\frac{p_3}{\left(l_\mathrm{r}+l_\mathrm{f}\right)^2}+1\right).
\end{aligned}
\tag{A-2}
$$

Bibliography

[1] National Highway Traffic Safety Administration and Us Department of Transportation. *TRAFFIC SAFETY FACTS Crash • Stats Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. Tech. rep.

[2] P. Liu, R. Yang, and Z. Xu. "How Safe Is Safe Enough for Self-Driving Vehicles?" In: *Risk Analysis* 39.2 (2018), pp. 315–325.

[3] J. Funke et al. "Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios". In: *IEEE Transactions on Control Systems Technology* 25.4 (2017), pp. 1204–1216.

[4] S.O. Hansson, M. Belin, and B. Lundgren. "Self-Driving Vehicles—an Ethical Overview". In: *Philosophy &amp Technology* 34.4 (2021), pp. 1383–1408.

[5] L. Claussmann et al. "A Review of Motion Planning for Highway Autonomous Driving". In: *IEEE Transactions on Intelligent Transportation Systems* 21.5 (2020), pp. 1826–1848.

[6] R. Rajamani. *Vehicle dynamics and control*. University of Minnesota: Springer, 2006.

[7] T. Brüdigam et al. *Collision Avoidance with Stochastic Model Predictive Control for Systems with a Twofold Uncertainty Structure*. 2021.

[8] M. Ammour, R. Orjuela, and M. Basset. "A MPC Combined Decision Making and Trajectory Planning for Autonomous Vehicle Collision Avoidance". In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022). ISSN: 1558-0016.

[9] B. Brito, B. Floor, and Javier Ferranti L.and Alonso-Mora. "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4459–4466.

[10] A. Arab et al. "Motion Planning and Control of Autonomous Aggressive Vehicle Maneuvers". In: *IEEE Transactions on Automation Science and Engineering* (2023), pp. 1–13.

[11] P. Falcone et al. "A hierarchical Model Predictive Control framework for autonomous ground vehicles". In: *2008 American Control Conference*. IEEE, 2008.

[12] M. Rokonuzzaman et al. "Model Predictive Control With Learned Vehicle Dynamics for Autonomous Vehicle Path Tracking". In: *IEEE Access* 9 (2021), pp. 128233–128249.

[13] A. Mesbah. "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research". In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44.

[14] T. Brudigam et al. "Stochastic Model Predictive Control With a Safety Guarantee for Automated Driving". In: *IEEE Transactions on Intelligent Vehicles* 8.1 (2023), pp. 22–36.

[15] T. Benciolini, D. Wollherr, and M. Leibold. "Non-Conservative Trajectory Planning for Automated Vehicles by Estimating Intentions of Dynamic Obstacles". In: *IEEE Transactions on Intelligent Vehicles* 8.3 (2023), pp. 2463–2481.

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

[17] T. 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.

[18] S. Lefèvre, D. Vasquez, and C. Laugier. "A survey on motion prediction and risk assessment for intelligent vehicles". In: *ROBOMECH Journal* 1.1 (2014). ISSN: 2197-4225.

[19] P. Kumar et al. "Learning-based approach for online lane change intention prediction". In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. 2013, pp. 797–802.

[20] G. Schildbach et al. "The scenario approach for Stochastic Model Predictive Control with bounds on closed-loop constraint violations". In: *Automatica* 50.12 (2014). ISSN: 0005-1098.

[21] A.F. Genovese. "The interacting multiple model algorithm for accurate state estimation of maneuvering targets". In: *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)* 22 (2001), pp. 614–623.

[22] M. Verhaegen and V. Verdult. *Filtering and System Identification, A Least Squares Approach*. Cambridge University Press, 2012.

[23] Z. Ding et al. "Unscented Information Consensus Filter for Maneuvering Target Tracking Based on Interacting Multiple Model". In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018, pp. 1535–1540.

[24] D. Lee et al. "Parallel Interacting Multiple Model-Based Human Motion Prediction for Motion Planning of Companion Robots". In: *IEEE Transactions on Automation Science and Engineering* 14.1 (2017), pp. 52–61.

[25] K.J. Astrom and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2009.

[26] L. Zhang et al. "Surrounding Vehicles Motion Prediction for Risk Assessment and Motion Planning of Autonomous Vehicle in Highway Scenarios". In: *IEEE Access* 8 (2020), pp. 209356–209376.

[27] R. Schubert, E. Richter, and G. Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking". In: 2008, pp. 1–6. ISBN: 978-3-8007-3092-6.

[28] L. Wang et al. “Motion Planning Framework for Autonomous Vehicle with Surrounding Vehicle Motion Prediction in Highway Scenarios”. In: *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. 2022, pp. 1–6.

[29] J. Li et al. “An interaction-aware predictive motion planner for unmanned ground vehicles in dynamic street scenarios”. In: *International Journal of Robotics and Automation* 34 (2019).

[30] R. Kensbock, M. Nezami, and G. Schildbach. “Scenario-Based Decision-Making, Planning and Control for Interaction-Aware Autonomous Driving on Highways”. In: *2023 IEEE Intelligent Vehicles Symposium (IV)*. 2023, pp. 1–6.

[31] Y.K. Nakka and S. Chung. “Trajectory Optimization of Chance-Constrained Nonlinear Stochastic Systems for Motion Planning Under Uncertainty”. In: *IEEE Transactions on Robotics* (2023).

[32] Y. Luo and Z. Yang. “A review of uncertainty propagation in orbital mechanics”. In: *Progress in Aerospace Sciences* 89 (2017), pp. 23–39. ISSN: 0376-0421.

[33] J. Nilsson, J. Fredriksson, and E. Coelingh. “Trajectory planning with miscellaneous safety critical zones * *This work was supported by FFI - Strategic Vehicle Research and Innovation.” In: *IFAC-PapersOnLine* 50.1 (2017), pp. 9083–9088. ISSN: 2405-8963.

[34] D. Kim et al. “A Hierarchical Motion Planning Framework for Autonomous Driving in Structured Highway Environments”. In: *IEEE Access* 10 (2022), pp. 20102–20117.

[35] S. Feng, Y. Qian, and Y. Wang. “Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm”. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 235.14 (2021), pp. 3416–3430.

[36] S. Pendleton et al. “Perception, Planning, Control, and Coordination for Autonomous Vehicles”. In: *Machines* 5.1 (2017), p. 6.

[37] V. Rausch et al. “Learning a deep neural net policy for end-to-end control of autonomous vehicles”. In: *2017 American Control Conference (ACC)*. 2017, pp. 4914–4919.

[38] S. Kuutti et al. “A Survey of Deep Learning Applications to Autonomous Vehicle Control”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.2 (2021), pp. 712–733.

[39] Y. Qian et al. “Obstacle avoidance planning of autonomous vehicles using deep reinforcement learning”. In: *Advances in Mechanical Engineering* 14.12 (2022).

[40] V.A. Laurense and J.C. Gerdes. “Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity”. In: *IEEE Transactions on Control Systems Technology* 30.1 (2022), pp. 166–179.

[41] J.K. Subosits and J.C. Gerdes. “From the Racetrack to the Road: Real-Time Trajectory Replanning for Autonomous Driving”. In: *IEEE Transactions on Intelligent Vehicles* 4.2 (2019), pp. 309–320.

[42] J. Wurts, J.L. Stein, and T. Ersal. “Design for Real-Time Nonlinear Model Predictive Control With Application to Collision Imminent Steering”. In: *IEEE Transactions on Control Systems Technology* 30.6 (2022), pp. 2450–2465.

[43] P. Polack et al. “The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?” In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017.

[44] V. Bithar, P. Tulpule, and S. Midlam-Mohler. “Online Robust Model Predictive Control Based Emergency Maneuvering System for Autonomous Vehicles”. In: *Journal of Autonomous Vehicles and Systems* 2.2 (2022).

[45] J.K. Subosits and J.C. Gerdes. “Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers”. In: *IEEE Transactions on Intelligent Vehicles* 6.3 (2021), pp. 546–558.

[46] H.K. Khalil. *Nonlinear Control.* Pearson Education Limited, 2015.

[47] K.J. Aström and B. Wittenmark. *Computer-Controlled Systems, theory and design.* Tsingua University Press, 1997.

[48] J. Dahl et al. “Collision avoidance: A literature review on threat-assessment techniques”. In: *IEEE Transactions on Intelligent Vehicles* 4.1 (2019), pp. 101–113.

[49] Y. Li et al. “Threat Assessment Techniques in Intelligent Vehicles: A Comparative Survey”. In: *IEEE Intelligent Transportation Systems Magazine* 13.4 (2021), pp. 71–91.

[50] H. Lancaster and E. Seneta. “Chi-square distribution”. In: *Encyclopedia of biostatistics* 2 (2005).

[51] S. Magdici and M. Althoff. “Fail-safe motion planning of autonomous vehicles”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 452–458.

[52] L. Gharavi et al. *Proactive Emergency Collision Avoidance for Automated Driving in Highway Scenarios*. 2023.