Convex MPC behavior planner with collision avoidance guarantees in mixed traffic

Author: Menno van Zutphen

Supervisors: Jochem Brouwer (TNO), Merlijne Geurts, Prof. Maurice Heemels, Dr. Erjen Lefeber, and Dr. Emilia Silvas.

MSc Graduation Project DC 2022.049

***abstract*—Behavior planning for autonomous vehicles is subject to the challenges of uncertainty and non-convexity. Mixed-Integer Quadratic Programming (MIQP) has been proposed as a suitable approach capable of efficiently producing optimal solutions to such non-convex problems. However, the existing formulations assume the availability of perfect predictions of the behavior of environment vehicles. As perfect prediction methods do not exist for mixed traffic scenarios, the application of the MIQP methods in practice can still lead to unsafe behavior. In this paper, we propose an extension to the MIQP framework that employs legal reachable set prediction-based emergency maneuvers to guarantee safety under uncertainty. Our algorithm guarantees collision avoidance in scenarios with law-abiding road users, independent of prediction quality. The approach is tested on a large volume of randomly generated merging scenarios containing unpredictable environment vehicle trajectories and it is shown that our algorithm successfully resolves the safety issues of previous work.**

I INTRODUCTION

A. Background

In a report by the American National Highway Traffic Safety Administration [1], the *critical reason* (i.e., the immediate reason that made the crash imminent) for 95% of the investigated motor vehicle crashes between July 2005 and December 2007 in the United States could be attributed to driver error. Although the report emphasises that the critical reason is not always the same as the cause of the crash, it nonetheless demonstrates that automating the driving task can play an important role in preventing crashes and improving driver safety in general. Additionally, by automating the driving task, the driver of the vehicle could save time by spending the time ordinarily spent driving on other productive tasks. Furthermore, entrusting full control of the autonomous vehicle (AV) to its control systems allows future implementations to optimize all aspects of transportation, such as road throughput, sustainability, ride comfort and travel time.

In realizing these benefits, the vehicles should ideally have reached at least the fourth level of driving autonomy, as defined by the SAE Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles [2]. This means the vehicle (i) actively monitors its environment in real-time, and (ii) is able to coordinate its actions safely and efficiently without requiring human monitoring. To ensure the coordination of actions results in safe and efficient behavior in practice, an advanced decision making*, i.e.,* control process is crucial. Fig. 1 shows an overview of how the AV control process (in orange) [3] interacts with other components of the proposed AV architecture. Within this controller module architecture, the *routing module* provides high-level goals (*e.g.*, "merge onto the highway" or "take the next exit") for which the *behavior planner* generates an efficient trajectory that achieves the high-level goal while avoiding collisions with other traffic. This trajectory is followed through the low-level *tracking controller*'s high frequency feedback actuation of the physical AV (plant).

As the behavior planner module is responsible for safety and efficiency of the trajectory, which often passes through a non-convex free space with uncertain future predictions, its development is not only complex [4], but also central to the AV control task. Additionally challenging is that the behavior planner must ensure that the trajectory it generates is followable by the tracking controller, as the tracking controller cannot realize every continuous curve imaginable as it is constrained by the properties of the physical vehicle.

B. Related literature

Common approaches for the design of the behavior planner in literature focus on finding ways around*, e.g.,* the non-convexity of the problem by exploiting properties of a specific scenario [5]. In order to design a provably safe behavior planner for a

Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, P.O. Box 513, 5600 MB, The Netherlands
CORRESPONDING AUTHOR: M.J.T.C. van Zutphen (e-mail: m.j.t.c.v.zutphen@student.tue.nl)
This work was completed in collaboration with the Netherlands Organisation for Applied Scientific Research TNO.

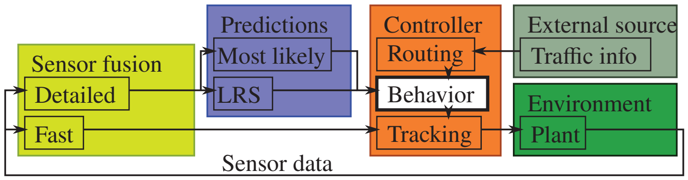


Figure 1: Proposed AV architecture. Sensor data is processed by two sensor fusion modules (yellow), a fast module for high-frequent low-level feedback and a more extensive, or 'detailed', module supplying the semantic environment information required for the predictions modules. The 'Most likely' sub-component of the predictions module (blue) produces a most likely scenario prediction of future environment vehicle occupancies. The legal reachable-set (LRS) sub-component conversely predicts the set of all possible locations each environment vehicle could reach while satisfying the laws of physics and the law of the road. The sub-components of the controller module (orange) subsequently use this data, together with publicly available traffic information, to actuate the ego vehicle (plant) safely and efficiently.

highway take over scenario, the controller by Naumann *et al.* [6] makes use of the explicit definition of a (highway-specific) *longitudinal safety gap*. Additionally, their overtaking trajectory is heuristically constrained to the sequence: constant acceleration, constant velocity, constant acceleration. Similarly, the provably robust highway merge controller by Cao *et al.* [7] employs a method that finds the best ego vehicle trajectory curve through parameter tuning of a certain parametric curve, which has to be designed for each scenario specifically. Other examples of scenario dependency in existing methods are the pre-, peri- and post-regions in [8], the follower/predecessor- and same-lane-or-not classification of neighboring vehicles required for the methods in [9] and [10], the overtaking window and critical-zone in [11], and the lane-dependent vehicle avoidance triangles described in [12], all of which, due to these scenario-specific solutions, are exclusively applicable to highway overtaking scenarios.

To overcome this problem of limited generalizability, the methods in [13] and [14] use the direct encoding of all obstacles (and future obstacle-scenario samples) into the environment. Dealing with the resulting non-convexity of the feasible space is subsequently done by removing feasible space from the scope until the remaining space forms a convex set (*i.e.,* to solve the problem merely within a convex local part of the complete non-convex space). In [13], this is achieved by fitting obstacle-touching squares around the predicted future ego vehicle positions and using these as the feasible space around the respective ego vehicle position at each time step. In [14], it is done by defining a tangential half-plane constraint, perpendicular w.r.t. the respective predicted ego-location, for each obstacle (Monte Carlo) scenario sample. This causes all predicted scenario samples to lie within the resulting infeasible region, while using only linear (convex) constraints. The main downside of such approaches is that the optimization problem only becomes convex because it is being constrained to find a solution locally. This has the potential to increase the speed of convergence, but does not eliminate non-convexity from the main problem, merely from the one presented to the solver, thus resulting in local minima. Additionally, due to the planned locations of the ego vehicle affecting the construction of the constraints, and the constraints affecting these planned locations in return, the algorithm must be ran sequentially, losing some of the speed the linear constraints were able to gain.

An alternative approach to handling the non-convexity of the feasible region generated by a direct encoding of all obstacles into the optimization problem is described in [15] and [16]. These formulations explicitly encode the non-convex feasible spaces into an optimization solver that is able to handle non-convex (in these cases also non-linear) problems. Formulating the optimization problem with a non-convex feasible space in this direct way has the same downside as the aforementioned convex reduction of the non-convex space, namely that a solver generally cannot guarantee its solution to be a global optimum.

In solving the local-optima limitation, the only optimal control method that has been able to describe the environment as directly (without the necessity for scenario-specific elements) without losing global optimality guarantees is the *"generic mixed integer quadratic programming* (MIQP) *model predictive control* (MPC)*"* method described by Esterle *et al.* in [17]. The method encodes the non-convex feasible space as the union of a finite set of convex spaces, which together cover the original feasible space without introducing conservatism. Solving the planning problem for each permutation of the convex feasible regions and returning the best one, the solver is guaranteed to return the globally optimal solution. This would result in an NP-hard problem, if not for the fact that the structure of the problem enables industrial solvers to leave sets of permutations unsolved if they can be shown to either (i) have properties which prevent them from achieving better outcomes than ones already found or (ii) are infeasible w.r.t. the other constraints.

The main remaining downside of the method in [17], is that it assumes the availability of perfect predictions. This would make the application of this method unsafe in practice, as provably correct prediction methods do not exist yet [18] (and can be conjectured to be impossible for mixed-traffic).

C. Contribution

To address this safety issue, we propose a new safe convex behavior planning (SCBP) method that is able to plan safely despite the inevitable imperfections of the most likely scenario predictions module. Instead of computing probability distributions and avoiding regions with a probability of occupancy above a certain threshold [14, 15] (or ignoring the uncertainty altogether [17]), the SCBP method uses *legal reachable set* predictions of other road users [19, 20] to monitor the safety of the optimal path [21, 22] generated by the MIQP-MPC method w.r.t. the most likely scenario prediction. The legal reachable set of another road user is defined here as (an overestimation of) the set of all points that this vehicle could occupy within a future window of time, given that the vehicle (i) abides to the law of the road and (ii) the laws of physics. While the probability distribution methods as in*, e.g.,* [14, 15], merely limit the chance of collision, this alternative approach can provide safety guarantees. Indeed provided the initial state of the ego vehicle does not make collision unavoidable, the technique achieves recusive feasibility (and thereby, safety) through the planning of an emergency intervention in case unexpected environment vehicle maneuvers occur. In order to reduce the amount of safety intervention events caused by poor-quality predictions (or simply hard to predict environments), the SCBP method offers the additional functionality of being able to incorporate information from the legal reachable set into the original optimal path problem.

We show how the proposed method executes merging and overtaking maneuvers autonomously while preserving sufficiently large distance from environment vehicles, all without explicit instruction. Additionally, we show that the new SCBP method is able to maneuver collision free through traffic-law-respecting environments by means of Monte Carlo simulation of unpredictable environment vehicle trajectories. Lastly, we demonstrate that the inclusion of legal reachability set information into the optimal path problem can improve SCBP performance.

D. *Structure of the paper*

In Sec. II, we present the intended use cases, define the properties of the environment, and formulate the behavior planner problem. Next, the proposed solution — the safe convex behavior planner — is first outlined and then elaborated upon in detail in Sec. III. In Sec. IV, the results of simulation experiments comparing the proposed SCBP to the original MIQP formulation performance are presented. Finally, the conclusion and recommendations for future work are provided in Sec. V.

II PROBLEM FORMULATION

The safe behavior planning trajectory generation problem revolves around two fundamental goals.

- *Safety*: ensure that the ego vehicle does not risk colliding with obstacles or road boundaries, irrespective of the (unpredicatable) decisions made by other road users.
- *Efficiency*: whenever there is more than one way execute a maneuver, ensure the one that yields the best cost realization is selected.

In order to outline the problem formulation, firstly the use cases, the properties of and requirements for the environment and the properties of the ego vehicle are discussed.

A. *Use cases*

The method developed in this paper is designed to be applicable to a wide range of road scenarios. The properties the scenarios are required to posses to enable the application of the SCBP method are listed below. For any scenario to be suitable for application of the SCBP in its current form, it is required that

R1: the maximum turning radius of the *ego vehicle* (cf. Sec. II-B1) is known,

R2: a *lanelet-map* (cf. Sec. II-B2) of the environment is available,

R3: the initial states and dimensions of all *environment vehicles* (cf. Sec. II-B3) are known,

R4: the scenario does not contain any traffic lights,

R5: an upper-bound on the acceleration of all environment vehicles is known and

R6: the speed limit holding on each road section is known.

It is further assumed that in these scenarios, only the ego vehicle is controlled by the SCBP, while the environment vehicles are controlled by either human drivers or unidentified control algorithms.

Two scenarios that satisfy these requirements are selected to illustrate and test the proposed behavior planner approach, cf. Fig. 2: (i) a curved merging scenario and (ii) a straight merging scenario, both subject to mixed traffic. The curved merging scenario is used as an example on which to illustrate the qualitative aspects of the algorithm. The straight merging use case is used for Monte Carlo experiments that aim to quantify (i) performance under uncertainty and (ii) the effect of legal reachable set information incorporation into the original planning problem.

The curved merging scenario, cf. Fig. 2A, consists of three parallel lanes, the left lane of the main road (L-lane), right lane of the main road (R-lane), and merging lane (M-lane) connected to the right border of the right lane of the main road. The R-lane center line is a curve with a radius of 200 (m). Two environment vehicles indicated by $i \in \{1, 2\}$ with respective velocities $v^i(t) \in \mathbb{R}^2$ defined w.r.t. the global inertial (x_1, x_2)-frame drive on the R-lane with constant velocity $(\|v^1(t)\|_2, \|v^2(t)\|_2) = (40, 44)$ (km/h) for all $t \in \mathbb{R}_{\geq 0}$. The scenario starts with the ego vehicle on the M-lane, 60 (m) before it ends, directly next to environment vehicle 1. Environment vehicle 2 starts 40 (m) ahead of environment vehicle 1. The ego vehicle is instructed to track the center line of the R-lane with a velocity of 80 (km/h). The scenario is simulated for 8 (s).

The straight merging scenario, illustrated in Fig. 2B, also consists of the L-, R- and M-lanes but with a non-curved road. The ego vehicle is initialized on the M-lane, 75 (m) from its end with velocity $v(t_0) = \begin{bmatrix} 80 & 0 \end{bmatrix}^\top$ (km/h). Three environment vehicles $i \in \{1, 2, 3\}$ are randomly placed, each $x_2^i(t_0)$-coordinate on the center line of either the R-lane or L-lane with probability

$$\Pr(x_2^i = x_2^{\text{R-lane}}) = 0.5 = \Pr^{\text{C}}(x_2^i = x_2^{\text{L-lane}}).$$

Their x_1^i-coordinate is randomly sampled from a uniform distribution between 0 and 100 (m) (as indicated as a grey area in Fig. 2B) for all $i \in \{1, 2, 3\}$ with the constraint that the next vehicle on the same lane must have at least 33 (m) distance from the previously placed environment vehicle. Each time an x_1^i sample does not satisfy this condition, a new x_1^i *and* x_2^i sample is generated until the new position does satisfy this constraint. The environment vehicles generally travel with a constant v_1^i velocity that is randomly sampled from the uniformly distributed range of $[56, 80]$ (km/h). The sampled v_1^i values are constrained to be increasing when sorted on the basis of increasing x_1^i coordinate to prevent rear-ending. In this scenario, all environment vehicles initialized on the R-lane reduce their velocity by 20% for two consecutive time steps, starting at a time t^i_{merge} sampled uniformly from $[0, t_{\text{scen}}]$, where the scenario simulation lasts $t_{\text{scen}} = 8$ (s). After this *brake maneuver*, the vehicles are controlled back to their original velocity. Further, the velocity in the lateral direction, $v_2^i(t)$ is subject to a normally distributed random force disturbance that is rejected by a proportional lateral acceleration controller penalizing error $\varepsilon = x_2^i - x_2^{\text{R-lane}}$.

During simulations, as the most likely scenario prediction method, a heuristic method that assumes lane-keeping and constant acceleration is employed. This enables the simulation of events where the most likely scenario predictor fails to anticipate on the brake maneuver of an environment vehicle, which can happen in practice. This way, the ability of the emergency maneuver method to neutralize this safety hazard is put to the test.

B. Definitions

The environment in which the SCBP operates consists of a road populated with environment vehicles and the ego vehicle itself. Before articulating the problem formulation, these terms are more clearly defined below.

1) Ego Vehicle

The state of the ego vehicle at time t is fully defined w.r.t. a global inertial Cartesian $(x_1, x_2) \in \mathbb{R}^2$-coordinate system by its location $s(t) \in \mathbb{R}^2$, velocity $v(t) \in \mathbb{R}^2$ and orientation $\theta(t) \in (-\pi, \pi]$, see Fig. 3. The dimensions of the ego vehicle $2w \times (d_r + l_r + l_f + d_f)$ represent, respectively, its width $2w$ and length $d_r + l_r + l_f + d_f$ composed of the distance from rear axle to rear bumper d_r, distance from rear axle to center of mass l_r, distance from center of mass to front axle l_f and distance from front axle to front bumper d_f. At time t, the area occupied by the ego vehicle body is denoted by $O(t) \subset \mathbb{R}^2$.

The ego vehicle in practice is a passenger car subject to complex vehicle dynamics governing its velocity and heading angle response to (accelerator- $e(t) \in \mathbb{R}$ and steering angle $\delta(t) \in \mathbb{R}$) inputs. Given that its state at time t is defined by $z(t) = \begin{bmatrix} s(t)^\top & v(t)^\top & \theta(t) \end{bmatrix}^\top$, the vehicle dynamics can be described as

$$\dot{z}(t) = f_{\text{ego}}(z(t), e(t), \delta(t)),$$

for an appropriate function f_{ego}.

The main goal of this paper is to propose an SCBP framework that is able to compute safe, comfortable, optimal and followable trajectories. The minimum level of model complexity required to demonstrate the ability of the algorithm to respect vehicle (non-holonomic) dynamics constraints is the kinematic bicycle model [23]. This is not to say that taking into account more intricate properties of complex dynamical (tyre slip-)vehicle models can not be beneficial to the implementation of a behavior planner in practice, merely that the consideration of these properties is outside the scope of this work.

The kinematic bicycle model [24] simplifies the kinematics of a four-wheel front-steered vehicle by lumping its front- and rear wheels into a single front-, respectively rear wheel. As the vehicle still rotates around its instantaneous center of rotation due to the kinematic non-slip assumption, the model output does not deviate from a four-wheel kinematic model. The non-slip assumption — the key difference between kinematic- and dynamic models — serves to greatly reduce the degrees of freedom, but does not significantly affect the behaviour results in the linear range [23].

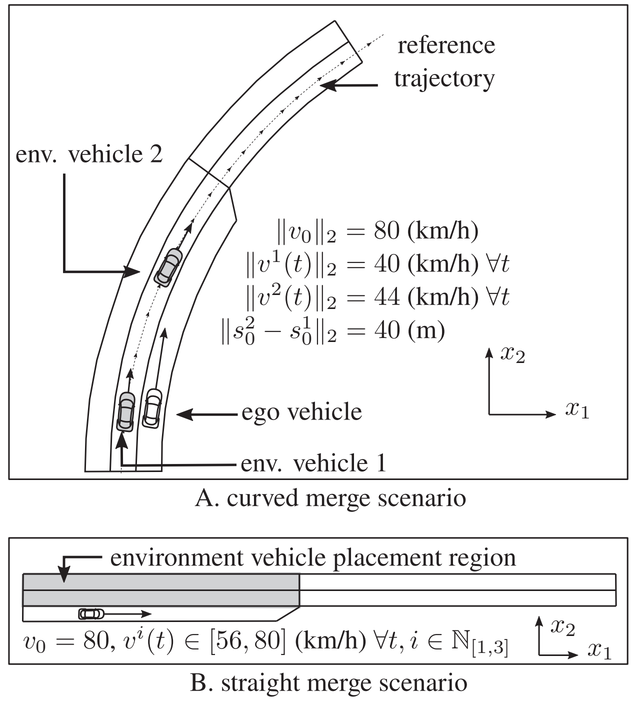


Figure 2: The use cases designed for experimenting and demonstrating the algorithm.

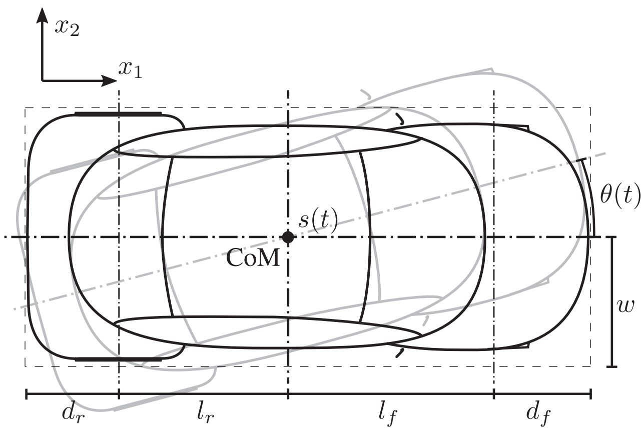


Figure 3: The ego vehicle geometry: the distance of its rear-end to the rear-axle d_r, the axle-distances from the CoM, l_r, l_f, and the distance between its front-end and front-axle d_f. Additionally, the CoM location $s(t)$, width $2w$ and rotation $\theta(t)$ and the cartesian global inertial frame (x_1, x_2) are illustrated.

The front wheel is subject to the *steering angle* input $\delta(t)$ defined as the difference in heading angle of the front wheel and of the the vehicle chassis, cf. Fig. 4. The entire vehicle is additionally subject to the *accelerator* control input $e(t)$, governing the absolute velocity of the model. The dynamic equations for this model can be derived as

$$\begin{aligned}
\begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} &= \nu(t) \begin{bmatrix} \cos(\theta(t) + \beta(\delta(t))) \\ \sin(\theta(t) + \beta(\delta(t))) \end{bmatrix}, \\
\dot{\theta}(t) &= \frac{\nu(t)}{l_r} \sin\left(\beta\left(\delta(t)\right)\right), \quad \dot{\nu}(t) = e(t) \\
\beta\left(\delta(t)\right) &= \arctan\left(\tan\left(\delta(t)\right) \frac{l_r}{l_f + l_r}\right)
\end{aligned} \tag{1}$$

where $\beta\left(\delta(t)\right)$ is the slip angle at the center of gravity.

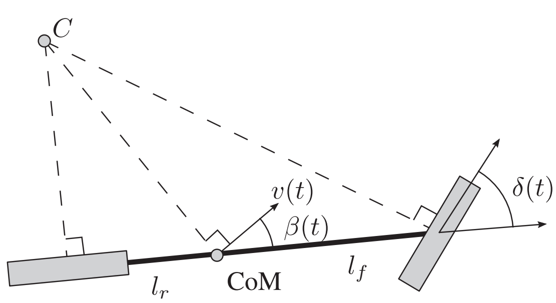


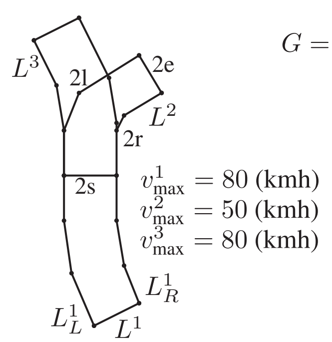
Figure 4: The kinematic bicycle model with (front wheel) steering angle $\delta(t)$, sideslip angle $\beta(t)$, velocity vector $v(t)$, instantaneous rotation center C and dimensions l_r, l_f.

2) Road Modeling

The environment is described by road maps that are generated offline. These maps (cf. Fig. 5) consist exclusively of lanelet [25] elements, each denoted L^j for $j \in \{1, \ldots, N_L\}$, $N_L \in \mathbb{N}_{\geq 0}$. A lanelet element L^j is a small section of road encoded through a set of left- and right $L^j = (L^j_L, L^j_R) \in \mathbb{R}^{2 \times N_L} \times \mathbb{R}^{2 \times N_L}$ polyline road boundaries, defined in the global cartesian inertial $(x_1, x_2) \in \mathbb{R}^2$ coordinate frame. Each lanelet L^j has a single associated speed limit $v^j_{\text{max}} \in \mathbb{R}_{>0}$. A road section subject to a changing speed limit, *e.g.,* is therefore described using at least as many atomic lanelet pieces as speed limits, one per unique speed limit. The relative location of the left L^j_L- and right L^j_R road boundary polylines is used to indicate driving direction.

Using a directional graph, described by adjacency matrix $G \in \{0, 1\}^{N_G}$, the legal traversals of lanelets are encoded. Since every lanelet L^j has four potential connection sites: its start, its end, its left- and its right border (denoted js, je, jr, and jl respectively for $j \in \{1, \ldots, N_L\}$ in Fig. 5), the adjacency matrix G has dimension $N_G = (4 \times N_L) \times (4 \times N_L)$. Note that the adjacency matrix is inherently sparse as connections such as, *e.g.,* end-to-end, start-to-start, right-to-start etc. never occur.

This yields the road environment $\mathcal{R}$ fully described by $\mathcal{R} = (\mathcal{L}, \mathcal{V}, G)$, with $\mathcal{L} = \{L^1, L^2, \ldots, L^{N_L}\}$ and respectively $\mathcal{V} = \{v^1_{\text{max}}, v^2_{\text{max}}, \ldots, v^{N_L}_{\text{max}}\}$.



$$G = \begin{array}{c} \\ 1s \\ 1e \\ 1l \\ 1r \\ 2s \\ 2e \\ 2l \\ 2r \\ 3s \\ 3e \\ 3l \\ 3r \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1s & 1e & 1l & 1r & 2s & 2e & 2l & 2r & 3s & 3e & 3l & 3r \end{array} \\ \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \end{array}$$

Figure 5: A simple example of a road-map $\mathcal{R} = (\mathcal{L}, \mathcal{V}, G)$ that is made up of three lanelets $\mathcal{L} = \{L^1, L^2, L^3\}$ with corresponding velocity limits $\mathcal{V} = \{v^1_{max}, v^2_{max}, v^3_{max}\}$ and adjacency matrix G, indicating all legal transitions (in this case: end of L^1 to start of L^2, and end of L^1 to start of L^3). Note that the driving direction can be derived from the definition of the left L^j_L- and right L^j_R polylines making up L^j.

3) Environment Vehicles

The ego vehicle shares the road with $N_O \in \mathbb{N}_{\geq 0}$ other (AV and non-AV) road users, referred to as *environment vehicles*. The superscript $\cdot^i$ is used to indicate when a variable belongs to environment vehicle $i \in \{1, 2, \cdots, N_O\}$. Variables describing the ego vehicle simply omit the superscript. The state of the environment vehicle i at time t is described equivalently to those of the ego vehicle through the tuple

$$(s^i(t), v^i(t), \theta^i(t), O^i(t)) \in \mathbb{R}^2 \times \mathbb{R}^2 \times (-\pi, \pi] \times P(\mathbb{R}^2),$$

with $P(\mathbb{R}^2)$ denoting the power set of $\mathbb{R}^2$. Requirements R2 and R5, respectively, state that this state information, in combination with the dimensions $2w^i \times (d_r^i + l_r^i + l_f^i + d_f^i)$, and an upper-bounding acceleration value $a^i_{\text{max}} \in \mathbb{R}_{\geq 0}$ are available to the algorithm for each of the environment vehicles at the time of planning the trajectory. This value a^i_{max} is at least as high as the highest achievable acceleration magnitude of the environment vehicle in both the lateral- *and* longitudinal directions, *i.e.*,

$$a^i_{\text{max}} \geq \max(a^i_{\text{max,long}}, a^i_{\text{max,lat}}).$$

C. Problem Statement

A general discrete-time finite horizon $h \in \mathbb{N}_{\geq 1}$ constrained optimization problem is to be formulated. Its objective is to generate an optimal trajectory (denoted using $\cdot^*$) described by $S^* = (s(0), s^*(1), \ldots, s^*(h))$ with associated velocities $V^* = (v(0), v^*(1), \ldots, v^*(h))$ starting from the current position s_0 and velocity state v_0 of the ego vehicle, that achieves the lowest cost w.r.t. finite horizon cost function $\hat{J}(S, V)$. Here, notation $\cdot(k)$ describes the value of the variable at $t = t_0 + k\tau$, with $\tau \in \mathbb{R}_{>0}$ the time step size. Since the predicted future positions S are fully determined by the initial position s_0 and all velocities $V = (v_0, V_{[1,h]})$, notation $\hat{J}(s_0, v_0, V_{[1,h]})$ will be adopted from this point on.

The objective is for the resulting trajectory to be comfortable and — most fundamentally — to make progress along the route defined by the routing module. Therefore, the cost function $\hat{J}(s_0, v_0, V_{[1,h]})$ is made up of the weighted sum of these two components as

$$\hat{J}(s_0, v_0, V_{[1,h]}) = \hat{J}_{\text{p}}(s_0, v_0, V_{[1,h]}) + w\hat{J}_{\text{c}}(V_{[1,h]}), \tag{2}$$

with input variables s_0, v_0, optimization variables $V_{[1,h]}$, weight factor $w \in \mathbb{R}_{>0}$, comfort $\hat{J}_{\text{c}}$- and $\hat{J}_{\text{p}}$ progress terms.

Adding the safety (defined through a no-collision-constraint w.r.t. environment vehicles $O(t) \cap O^i(t) = \emptyset$ and satisfaction of road boundaries $O(t) \subseteq \mathcal{L}$ condition $\forall t \in [t_0, t_0 + h\tau]$) aspect to this problem through constraints results in the optimization problem formulation

$$
\begin{aligned}
V^*_{[1,h]} = \arg \min_{V_{[1,h]}} \; & \hat{J}(s_0, v_0, V_{[1,h]}) && \text{(3a)}\\
\text{s.t. } & \|v(t)\| \leq v^j_{\text{max}} \text{ if } O(t) \cap L^j \neq \emptyset && \text{(3b)}\\
& O(t) \cap O^i(t) = \emptyset && \text{(3c)}\\
& O(t) \subseteq \mathcal{L} && \text{(3d)}\\
& \dot{z}(t) = f_{\text{ego}}(z(t), e(t), \delta(t)) && \text{(3e)}\\
& e(t) \in [e_{\text{min}}, e_{\text{max}}],\ \delta(t) \in [\delta_{\text{min}}, \delta_{\text{max}}] && \text{(3f)}\\
& \forall i \in \{1, \ldots, N_O\},\ \forall j \in \{1, \ldots, N_L\} && \text{(3g)}\\
& \forall t \in [t_0, t_0 + h\tau]. && \text{(3h)}
\end{aligned}
$$

The method by which this thus far conceptually formulated optimization problem will be made computable and solved in receding horizon fashion to produce the behavior planner, is subject of the next section.

Given the above described environment, the ego vehicle, the environment vehicles and a given target location or trajectory that can be encoded as $\hat{J}_p(s_0, v_0, V_{[1,h]})$, the problem of solving (3a-3h) can be stated as:

Given ego vehicle dynamics (1), initial state x_0 that does not make collision unavoidable, a road map $\mathcal{R}$ and regularly updated road occupation information $O^i(t_0)$ for $i = 1, \ldots, N_O$, develop a receding horizon behavior planner that iteratively generates a (i) followable trajectory, (ii) guarantees law-abiding-obstacle avoidance for $t \geq t_0$, (iii) minimizes safety interventions, (iv) maximizes comfort and (v) does not suffer from sub-optimal trajectories within the planning horizon h.

III METHOD

In solving the problem formulated in Sec. II, the SCBP algorithm is developed. This section provides an overview of the high-level workings of the algorithm, after which it goes into detail on the computational methods required.

A. Main Algorithm Overview

The most intuitive, and most researched method of behavior planner design is to make a *most likely scenario* prediction of the future occupancies of the other road users and plan a trajectory that avoids these obstacle occupancy predictions [7, 9, 10, 12, 17], resulting in what this work refers to as the *ideal trajectory*. Provided (i) the planning horizon is sufficiently long and (ii) the most likely scenario prediction is 100% accurate in predicting future road user occupancies, the direct application of such ideal trajectory through a receding horizon scheme can form a safe behavior planner.

The significant levels of uncertainty in the most likely scenario predictions [18] unfortunately make the direct application of ideal trajectories prone to safety issues in practice. This, as the predicted trajectories used to avoid obstacles can deviate from the true paths the vehicles take. This is why the method proposed in this work couples the globally optimal ideal trajectory generation method [17] with an emergency maneuver technique [21]. This way, the resulting architecture is able to guarantee collision avoidance in environments with law-abiding road users, without requiring perfect scenario predictions.

1) Ideal Trajectory

The ideal trajectory $T^* = \begin{bmatrix} x_1^{*\top} & x_2^{*\top} & \cdots & x_h^{*\top} \end{bmatrix}^\top$, where x_k^* is the state at $t = t_0 + k\tau$ described by $x^*(t) = \begin{bmatrix} s_1^*(t) & v_1^*(t) & s_2^*(t) & v_2^*(t) \end{bmatrix}^\top$, is generated at $t = t_0$ using the mixed integer quadratic programming (MIQP) model predictive control (MPC) method developed by Esterle *et al.* [17]. The method uses a convex partitioning of the often non-convex feasible configuration space of the ego vehicle w.r.t. environment vehicle occupancy predictions to guarantee global optimality w.r.t. its planning horizon $h \in \mathbb{N}_{\geq 1}$. The integer variables this method introduces, enable the encoding of non-linear constraints such as obstacle avoidance and the non-holonomic constraints of the ego vehicle model through local linearizations.

2) Emergency Maneuver

After the ideal trajectory T^* is computed for the current state of the ego vehicle $x(0)$, the emergency maneuver T_E^* is planned from the first predicted ideal trajectory state x_1^* to state $x_E^*(h_E)$ at the emergency maneuver horizon $h_E \in \mathbb{N}_{[1,h]}$, see Fig. 6. Thus, the emergency maneuver is a trajectory computed at the current time t_0 to be applied one time step into the future $t_0+\tau$, *i.e.,* after execution of a single ideal trajectory step. Note that at this time $t = t_0 + \tau$, the emergency maneuver is ideally not executed at all. Instead, the first step of the next ideal trajectory T^*, computed at $t = t_0 + \tau$ alongside a new emergency trajectory T_E^* is executed whenever the planning of both T^* and T_E^* is successful. The emergency maneuver is constrained to avoid the *legal reachable set* predictions of all environment vehicles. The legal reachable set of an environment vehicle is the set of all states it could reach at a certain moment in the future, which is constrained by the traffic laws and the laws of physics. Avoiding these regions guarantees that the emergency trajectory cannot cause collision without other road users breaking the law, thereby providing a safety from legal liability.

The main purpose of this emergency trajectory T_E^* is to have a guaranteed safe back-up plan in case the next planning cycle returns infeasible. When infeasibility of either of the planning problems (T^* and or T_E^*) occurs at t_0, due to, *e.g.,* the occurance of environment vehicle moves that were not anticipated by the most likely scenario prediction, the emergency maneuver computed at the previous planning cycle $t = t_0 - \tau$ can be executed, as it has been confirmed to be safe w.r.t. the worst-case scenario environment vehicle trajectories (provided they abide to the traffic laws).

Due to the necessity of the emergency maneuver to avoid the legal reachable set of all environment vehicles, the emergency trajectory is generally highly conservative and therefore much less desirable than the ones computed as the ideal trajectory. This is why the frequency of emergency maneuver executions should be minimized to yield the most desirable paths in practice.

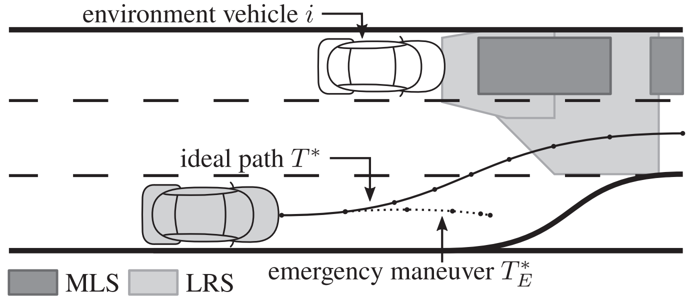


Figure 6: The workings of the SCBP algorithm visualized: illustrating the optimal path and the emergency maneuver (which can be seen to originate from the optimal path's first-planned state x_1^*). Additionally, two prediction steps are visualised from the most likely scenario (MLS)- and legal reachable set (LRS) prediction algorithms respectively.

B. Ideal Trajectory

Following the method in [17], a number of techniques are introduced that enable the path planner to avoid predicted occupancies and ensure its trajectory is followable w.r.t. the ego vehicle dynamic model (1).

1) Prerequisites

a. Coupling of States: To enable the use of $s(k)$, $v(k)$, and acceleration vector $a(k-1) \in \mathbb{R}^2$ in the optimization formulation for all $k \in \{1, \dots, h\}$, their relation is encoded through equality constraints. As an appropriate model relating these variables, the integrator model is selected for its ability to be directly encoded into the final problem through a linear constraint. The linear integrator model in continuous time is described by

$$\frac{d}{dt}\begin{bmatrix} s_\ell \\ v_\ell \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} s_\ell \\ v_\ell \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_\ell, \tag{4}$$

for the two inertial frame directions $\ell \in \{1, 2\}$. By defining $x = \begin{bmatrix} s_1 & v_1 & s_2 & v_2 \end{bmatrix}^\top$ and $u = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^\top$, (4) is written for $\ell \in \{1, 2\}$ using the block-diagonal set (A_c, B_c) as

$$\dot{x} = A_c x + B_c u. \tag{5}$$

As the input u is piece-wise constant, *i.e.,* $u(t) = u_k$ for $t \in [t_0 + k\tau, t_0 + (k+1)\tau)$, exact discretization of (5) yields the discrete dynamic representation

$$x_{k+1} = Ax_k + Bu_k, \text{ where } \begin{cases} A = e^{A_c\tau}, \\ B = A\int_0^\tau e^{-A_c t}\mathrm{d}t B_c. \end{cases} \tag{6}$$

Indirect access to the jerks j_ℓ is enabled through $j_\ell(k) \propto a_\ell(k+1) - a_\ell(k)$. Note that this dynamical model lacks the ability to accurately represent many of the unique physical properties of the AV model (1). This loss of analogue is later made up for through the application of a set of tailored constraints on the curvature of the trajectory.

Although the equality constraints (6) could be provided to the solver explicitly, we opted to substitute them into the cost function and constraints to eliminate the explicit dependency on states $X_k = \begin{bmatrix} x_1^\top & \cdots & x_h^\top \end{bmatrix}^\top$ from the optimization problem. Through the equation in $U_k = \begin{bmatrix} u_0^\top & \cdots & u_{h-1}^\top \end{bmatrix}^\top$,

$$X_k = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^h \end{bmatrix}}_{\Phi} x_0 + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{h-1}B & A^{h-2}B & \dots & B \end{bmatrix}}_{\Gamma} U_k, \quad (7)$$

the occurrences of X_k can be eliminated in favor of the more fundamental initial position and planned accelerations (x_0, U_k). This exposes the accelerations in U_k as the true optimization variables.

b. Indicator Constraints: The inclusion of binary variables in the problem formulation enables the use of indicator constraints. Indicator constraints are constraints of the form

$$\bar{\delta} = 1 \Rightarrow \bar{A}\bar{x} \leq \bar{b},$$

which states that when binary variable $\bar{\delta} \in \{0,1\}$ is set to one $\bar{\delta} = 1$, it ‘indicates’ that the $\bar{n}$ optimization variables in $\bar{x} \in \mathbb{R}^{\bar{n}}$ have to satisfy the $\bar{m}$ linear constraints defined by $(\bar{A}, \bar{b}) \in \mathbb{R}^{\bar{m}\times\bar{n}} \times \mathbb{R}^{\bar{m}}$. This indicator action ‘$\Rightarrow$’ can be encoded into a mixed-integer problem by adding the binary variable $\bar{\delta}$ to the optimization variables and adapting the linear constraints $\bar{A}\bar{x} \leq \bar{b}$ through the *big-M method* [26] as

$$\bar{A}\bar{x} \leq \bar{b} + M(1 - \bar{\delta}), \quad (8)$$

where, by ensuring the elements of $M \in \mathbb{R}^{\bar{m}}$ (indicated using index $\cdot_{\bar{r}}$) satisfy

$$\bar{b}_{\bar{r}} + M_{\bar{r}} > \max_{\bar{x}} \bar{A}_{\bar{r}}\bar{x}, \quad \forall \bar{r} \in \{1, \dots, \bar{m}\},$$

the setting of $\bar{\delta} = 0$ then effectively disables the $(\bar{A}, \bar{b})$-constraint on $\bar{x}$ (8). Some solvers, such as the solver used for the simulations in this work, Gurobi [27], enable the encoding of indicator constraints explicitly, as well as through the algebraic big-M method. Both methods, when applied correctly, should yield the same solutions.

c. Velocity Region Indicator Variables: The non-linear properties of the AV behavior planning problem — such as collision avoidance requiring trigonometry and followability ensurance requiring non-holonomy — cannot be modeled directly in an MIQP formulation. Esterle *et al.* [17] therefore develop a subdivision-and-linearization method to enable the approximate encoding of non-linear relations through a piece-wise linear approximation.

The method creates a subdivision of the (v_1, v_2)-plane, cf. Fig. 7, through a set of $N_r \in \mathbb{N}_{\geq 1}$ connected, conic regions for all $r \in \{1, \dots, N_r\}$ of equal proportion radiating from the origin $(v_1, v_2) = (0, 0)$ as

$$\mathcal{V}_r = \{(v_1, v_2) \in \mathbb{R}^2 \ : \ \alpha_r v_2 \geq \beta_r v_1, \ \gamma_r v_2 \leq \varepsilon_r v_1\},$$

for region-defining constants $(\alpha_r, \beta_r, \gamma_r, \varepsilon_r) \in \mathbb{R}^4$. Next, based on in which of these regions r velocity vector $v(k) = \begin{bmatrix} v_1(k) & v_2(k) \end{bmatrix}^\top$ at timestep k lies, a different (local) linearization of the non-linear constraints is activated to best approximate their original non-linear form.

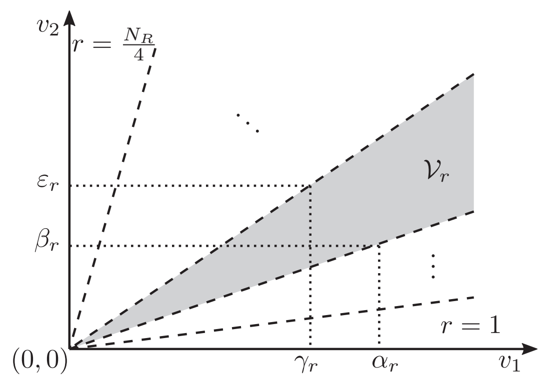


Figure 7: The regions r visualised for $r = \{1, \dots, N_R/4\}$ with all velocities belonging to region r marked in grey v_r. Figure adapted from [17].

This activation is realized by firstly defining a set of binary variables $\delta_r \in \{0,1\}$ for $r \in \{1, \dots, N_r\}$ which are linked to their respective velocity region $\mathcal{V}_r$ through indicator constraints [17] as

$$\delta_r = 1 \Rightarrow \begin{matrix} \alpha_r v_2 \geq \beta_r v_1 \\ \gamma_r v_2 \leq \varepsilon_r v_1 \end{matrix}, \quad \text{for } r = \{1, \dots, N_R\}, \quad (9a)$$

which are constrained to satisfy

$$\sum_{r=1}^{N_R} \delta_r = 1. \quad (9b)$$

This ensures one of the regions is always activated, *i.e.,* its indicator variable δ_r set to one. Note that these constraints (9) are defined for every $k \in \{1, \dots, h\}$, whereby the active region at timestep k is found as $r(k) = \{r \in \mathbb{N}_+ \ : \ \delta_r(k) = 1\}$, or equivalently $r(k) = \{r \in \mathbb{N}_+ \ : v(k) \in \mathcal{V}_r\}$.

Now, locally-linearized non-linear constraints can be encoded to only act in region r by simply defining indicator constraints

$$\delta_r = 1 \Rightarrow A_{\text{LL},r} U_k \leq b_{\text{LL},r}, \ \text{ for } r = \{1, \ldots, N_R\},$$

where $A_{\text{LL},r}$, $b_{\text{LL},r}$ describe the local linear approximation of a certain non-linear constraint, valid over region r.

d. Fitting of Linear Parameters: The process of computing these regional linear approximation constraint expressions $A_{\text{LL},r}, b_{\text{LL},r}$, that enable the linear encoding of non-holonomic and trigonometric constraints is also described in [17]. In short, all non-linear equations $f(s(k), v(k), a(k))$ are linearized over region $\mathcal{V}_r$ to yield the linear expression

$$f_r^{\text{lin}}(s(k), v(k), a(k)) = p_{0,r} + \begin{bmatrix} p_{s,r}^\top & p_{v,r}^\top & p_{a,r}^\top \end{bmatrix} \begin{bmatrix} s(k) \\ v(k) \\ a(k) \end{bmatrix},$$

for $r = \{1, \ldots, N_R\}$, where the parameters $p_{0,r} \in \mathbb{R}$, $p_{s,r} \in \mathbb{R}^2$, $p_{v,r} \in \mathbb{R}^2$, $p_{a,r} \in \mathbb{R}^2$ are computed through a least-square fit on the error between it and the original function over a grid of $N_s \in \mathbb{N}_{\geq 3}$ points $v_{r,s} \in \mathcal{V}_r$ sampled from its respective velocity region r. In the case of the curvature terms required for the non holonomic constraint, this is done while constraining the resulting planes to form a (i) non-negative (ii) lower bound on the function evaluation samples. The reason for this becomes clear when considering the details of this particular constraint in Sec. III-B3.

As an example, the linearization of the non-linear relation between the direction of travel (v_1, v_2) and the rotation $R(\theta) \in SO(2)$ of the vehicle body volume, which is assumed to be described sufficiently well through linear combinations of $\cos(\text{atan2}(v_2, v_1)) \approx \cos(\theta)$ and $\sin(\text{atan2}(v_2, v_1)) \approx \sin(\theta)$, is worked out. Firstly, the finite grid of N_s sample points $v_{s,r} \in \mathcal{V}_r$ for $s = \{1, \ldots, N_s\}$, $r \in \{1, \ldots, N_R\}$ is defined. The target non-linear equation is then evaluated on these points, *e.g.,*

$$\bar{f}_{r,s} = \cos(\text{atan2}(v_{2,r,s}, v_{1,r,s})) \ \text{ for } \ s = \{1, \ldots, N_s\},$$

after which matrix A_r and vector b_r are constructed as

$$A_r = \begin{bmatrix} 1 & v_{1,r,1} & v_{2,r,1} \\ 1 & v_{1,r,2} & v_{2,r,2} \\ \vdots & \vdots & \vdots \\ 1 & v_{1,r,N_s} & v_{2,r,N_s} \end{bmatrix}, \quad b_r = \begin{bmatrix} \bar{f}_{r,1} \\ \bar{f}_{r,2} \\ \vdots \\ \bar{f}_{r,N_s} \end{bmatrix}.$$

By defining vector of coefficients $x_r = \begin{bmatrix} p_{0,r} & p_{v_1,r} & p_{v_2,r} \end{bmatrix}^\top$ that describes the parameters of the polynomial

$$f_{r,s}(v \in \mathcal{V}_r) \approx p_{0,r} + v_1 p_{v_2,r} + v_2 p_{v_2,r},$$

the square of the error $A_r x_r - b_r$ can be minimized through the QP (least-squares) problem

$$\min_{x_r} \ (A_r x_r - b_r)^\top (A_r x_r - b_r).$$

The application of the constraints $-A_r x_r \leq -b_r$ and $A_r x_r \leq b_r$ are respectively used to yield an upper- or lower bound of the function in this region, while $-A_r x_r \leq 0$ can be applied in the event where the non-negativity of the original function is to be retained.

2) Predicted Occupancy Avoidance and Satisfaction of Road Boundaries

Since vehicles, road boundaries and static obstacles are to be avoided, a method for doing so has to be included in the MIQP-based optimal trajectory planner. Polygons are used to describe every obstacle, road boundary and predicted occupancy region, which allows for doing so up to an arbitrary level of accuracy.

To encode the boundaries of the drivable space into the optimization problem as constraints, the following set of steps are executed; note that the simplified adjacency matrix G_s is used, which in contrast to G does not discern between *types* of connections, but simply describes to which lanelets one is allowed to move from a certain lanelet, cf. Fig. 8.

1. Find what set of lanelets $\mathcal{L}_E(t_0) \subseteq \mathcal{L}$ the ego vehicle (partially) occupies at time $t = t_0$, *i.e.,* compute the set
$$\mathcal{L}_E(t_0) = \{L^j \ : \ O(t_0) \cap L^j \neq \emptyset\}.$$
2. Compute the ego vehicle *reachable* lanelet set $\mathcal{L}_{E,R}(t_0)$ for $t \in [t_0, t_0 + h\tau]$ by forming the union of all lanelets that can be reached from $\mathcal{L}_E(t_0)$ (w.r.t. directional graph G_s), with the set $\mathcal{L}_E(t_0)$ itself.
3. The legal road space polygon $\mathcal{P}_R$ is subsequently found as the union of the lanelet polygons in $\mathcal{L}_{E,R}(t_0)$, cf. Fig. 8.
4. The corresponding occupancy prediction polygons for each future time step k are then removed from $\mathcal{P}_R$, to yield h polygons (generally non-convex with holes) $\mathcal{P}_{R,k}$ for $k \in \{1, \ldots, h\}$ fully describing the unoccupied section of the reachable road space , cf. Fig. 10.

a. Modeling of the ego vehicle volume The ego vehicle area is a polygon. Computing whether a polygon is fully inside the feasible reachable space $\mathcal{P}_{R,k}$ requires non-linear (algorithmic) computations that are not straightforwardly encoded as linear constraints [28]. Much like [17, 29], the vehicle body is instead encoded through the union of a set of circles $c \in \{1, \ldots, N_C\}$ with radius r_{circles}, see Fig. 9, that each have a center with constant offset $s_{c,\text{loc}} \in \mathbb{R}^2$ from the ego vehicle center of mass $s(t)$ (constant only w.r.t. the local vehicle reference frame). The circles must cover the area of the ego vehicle polygon completely. The circle centers $s_c(k) \in \mathbb{R}^2$ as observed from the global (inertial) frame of reference move with the ego vehicle as

$$s_c(k) = s(k) + R(v(k)) s_{c,\text{loc}} \ \text{ for } \ c \in \{1, \ldots, N_C\}. \quad (10)$$

Shrinking the feasible reachable space $\mathcal{P}_{R,k}$ by radius r_{circles} then yields a slightly conservative approximation to the feasible

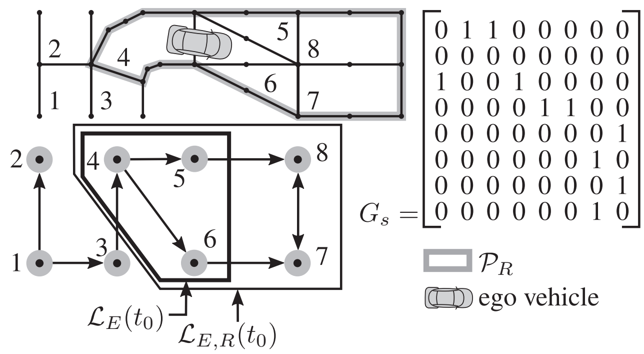


Figure 8: An example of a simple lanelet map and its corresponding transition graph in a visual representation and as encoded in the simplified adjacency matrix G_s. Note that for the ego vehicle-occupied lanelets set $\mathcal{L}_E(t_0)$ to be legal, the sub-graph it spans must be connected. Note also how the reachable lanelet set $\mathcal{L}_{E,R}(t_0)$ (and by extension the reachable lanelet polygon $\mathcal{P}_R$) is found by forward propagation through G_s from $\mathcal{L}_E(t_0)$.

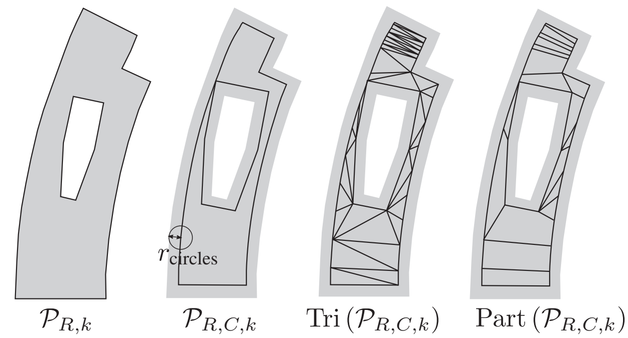


Figure 10: The configuration space partitioning algorithm applied to a part of the curved merge scenario. From left to right: the feasible part $\mathcal{P}_{R,k}$ of the road at timestep k, determined by subtracting all relevant obstacle predictions from the road surface; the feasible configuration space $\mathcal{P}_{R,C,k}$, i.e., the feasible part $\mathcal{P}_{R,k}$ reduced by buffer width $r_{circles}$; the triangulation of the buffered feasible part; and the final convex partition of the feasible configuration space.

configuration space of these circle centers $s_c(k)$, given that their circumference must remain inside $\mathcal{P}_{R,k}$. By ensuring the centers of these circles, $s_c(k)$ for $c \in \{1, \dots, N_C\}$ remain inside this configuration space estimate $\mathcal{P}_{R,C,k}$, cf. Fig. 10, it is ensured the entire ego vehicle polygon remains inside $\mathcal{P}_{R,k}$.

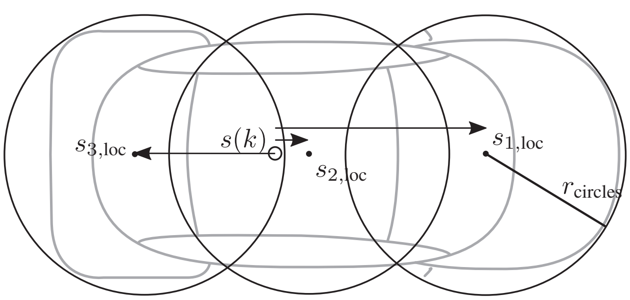


Figure 9: The circles at $s_{1,loc}$, $s_{2,loc}$ and $s_{3,loc}$ with radius $r_{circles}$ inscribing the ego vehicle at $s(k)$.

Encoding the constraints that force the circle centers $s_c(k)$ to remain in the feasible configuration space $\mathcal{P}_{R,C,k}$, requires the non-convex polygon $\mathcal{P}_{R,C,k}$ to be represented by the union of a set of convex polygons first. This process is described next.

b. Splitting of non-convex polygons Given one of the non-convex feasible configuration space polygons $\mathcal{P}_{R,C,k}$ with holes, cf. Fig. 10, the polygon is partitioned into a set of convex polygons $\mathcal{P}_{R,\text{conv},k}$ through the application of the Hertel-Melhorn partitioning algorithm [28]. Simply put, this algorithm first triangulates the polygon $\mathrm{Tri}\,(\mathcal{P}_{R,C,k})$ through, *e.g.,* ear-clipping [28], after which it merges the resulting neighboring polygons (initially all triangles) if the merge results in a convex polygon, repeating until no more neighbor-pairs are left that could yield a convex polygon when merged.

c. Restriction to convex polygons Given this finite set of convex polygons $\mathcal{P}_{R,\text{conv},k}$ for every $k \in \{1, \dots, h\}$, the last step is to restrict the centers of the ego vehicle circles $s_c(k)$ to always satisfy $s_c(k) \in \mathcal{P}_{R,\text{conv},k}$.

Constraining a point $s_c(k)$ to lie in one of the convex polygons p is done through a set of linear half-plane constraints $A_{p,k} s_c(k) \leq b_{p,k}$, cf. Fig. 11. The ability for the point to move from polygon to polygon is subsequently enabled using indicator constraints by setting

$$\delta_{p,k} = 1 \Rightarrow A_{p,k} s_c(k) \leq b_{p,k}, \text{ and } \sum_{p=1}^{N_{p,k}} \delta_{p,k} = 1. \qquad (11)$$

for all $k \in \{1, \dots, h\}$, *i.e.,* $s_c(k)$ needs to only satisfy the constraints of a single polygon $p \in \{1, \dots, N_{p,k}\}$ at a time.

The last step that remains is writing (11) linearly in terms of U_k, in order for it to be directly encoded into the solver. Noting that $s_c(k)$ expressed in terms of the states of the ego vehicle yields the non-linear trigonometric equation (10), we use the local linearization method to approximate $\sin(\theta(k))$ and $\cos(\theta(k))$ in region r as respectively

$$\delta_r = 1 \Rightarrow \begin{cases} \sin(\theta(k)) \approx p_{0,r}^{\sin} + p_{v_1,r}^{\sin} v_1(k) + p_{v_2,r}^{\sin} v_2(k), \\ \cos(\theta(k)) \approx p_{0,r}^{\cos} + p_{v_1,r}^{\cos} v_1(k) + p_{v_2,r}^{\cos} v_2(k), \end{cases}$$

for $r \in \{1, \dots, N_R\}$ and $(v_1(k), v_2(k)) \in \mathcal{V}_r$. Substitution into (10) yields the expression

$$s_c(k) \approx p_c^{0,r} + p_c^{v_1,r} v_1(k) + p_c^{v_2,r} v_2(k) + s(k), \qquad (12a)$$

for $c = \{1, \dots, N_c\}$, $(v_1(k), v_2(k)) \in \mathcal{V}_r$ and

$$p_c^{0,r} = \begin{bmatrix} p_{0,r}^{\cos} s_{c,\text{local}}^1 - p_{0,r}^{\sin} s_{c,\text{local}}^2 \\ p_{0,r}^{\sin} s_{c,\text{local}}^1 + p_{0,r}^{\cos} s_{c,\text{local}}^2 \end{bmatrix}, \tag{12b}$$

$$p_c^{v_1,r} = \begin{bmatrix} p_{v_1,r}^{\cos} s_{c,\text{local}}^1 - p_{v_1,r}^{\sin} s_{c,\text{local}}^2 \\ p_{v_1,r}^{\sin} s_{c,\text{local}}^1 + p_{v_1,r}^{\cos} s_{c,\text{local}}^2 \end{bmatrix}, \tag{12c}$$

$$p_c^{v_2,r} = \begin{bmatrix} p_{v_2,r}^{\cos} s_{c,\text{local}}^1 - p_{v_2,r}^{\sin} s_{c,\text{local}}^2 \\ p_{v_2,r}^{\sin} s_{c,\text{local}}^1 + p_{v_2,r}^{\cos} s_{c,\text{local}}^2 \end{bmatrix}. \tag{12d}$$

Substituting this expression (12) into the convex polygon constraint (11) yields the linear form that must be encoded for all circles describing the ego vehicle $c \in \{1, \dots, N_c\}$, all regions over which the non-linear equations are linearized $r \in \{1, \dots, N_R\}$, all time-steps $k \in \{1, \dots, h\}$ and all polygons within the respective timestep $p \in \{1, \dots, N_p(k)\}$.

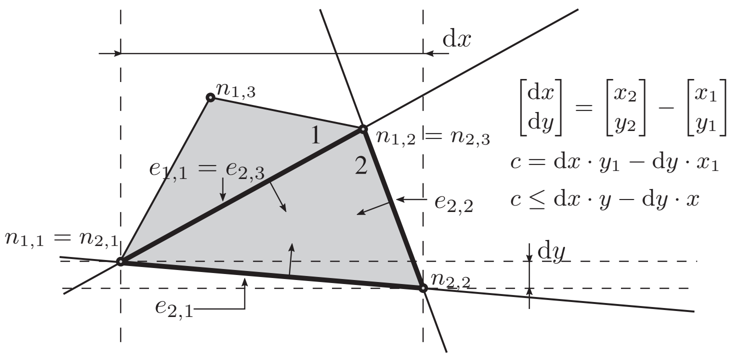


Figure 11: An example feasible space consisting of two polygons, 1 and 2, where the encoding of one of the half-plane constraints composing convex polygon 2 is described by the included equations, resulting in the linear constraint $c \leq \mathrm{d}x \cdot y - \mathrm{d}y \cdot x$. Nodes are labeled through notation $n_{p,i}$, with p the polygon index and i the node index, edges equivalently through $e_{p,i}$. Note that all polygon nodes are defined in counter- clockwise direction, this ensures the half-plane constraints satisfy the presented equation.

3) *Non-Holonomic Constraint*

Since the dynamic model in the MPC architecture lacks the ability to describe the non-holonomic character of a passenger car, this constraint on the trajectory is encoded through the locally linearized implementation of the curvature κ constraint. The curvature of the trajectory (of any line) is described by

$$\kappa = \frac{v_1 a_2 - v_2 a_1}{\sqrt{v_1^2 + v_2^2}^3}.$$

Constraining the trajectory to satisfy curvature constraint $\kappa_{\text{min}} \leq \kappa \leq \kappa_{\text{max}}$, where $\kappa_{\text{min}} < 0 < \kappa_{\text{max}}$, is achieved by substituting the expression for κ and rewriting to yield

$$\kappa_{\text{min}} \sqrt{v_1^2 + v_2^2}^3 \leq v_1 a_2 - v_2 a_1, \tag{13a}$$

$$\kappa_{\text{max}} \sqrt{v_1^2 + v_2^2}^3 \geq v_1 a_2 - v_2 a_1, \tag{13b}$$

which remains valid due to $\sqrt{v_1^2 + v_2^2}^3 \geq 0$. Linearization of these two inequalities is done through the linear approximation of $\sqrt{v_1^2 + v_2^2}^3$ in v_1 and v_2 as

$$\sqrt{v_1^2 + v_2^2}^3 \approx p_{0,r}^{\text{curve}} + p_{v_1,r}^{\text{curve}} v_1 + p_{v_2,r}^{\text{curve}} v_2,$$

and a linear approximation of the right-hand side through $(v_1, v_2) \approx (v_{1,\text{avg}_r}, v_{2,\text{avg}_r})$ for $(v_1, v_2) \in \mathcal{V}_r$, yielding

$$\kappa_{\text{min}}(p_{v_1,r}^{\text{curve}} v_1 + p_{v_2,r}^{\text{curve}} v_2) - v_{1,\text{avg}_r} a_2 + v_{2,\text{avg},r} a_1 \leq -\kappa_{\text{min}} p_{0,r}^{\text{curve}}, \tag{14a}$$

$$\kappa_{\text{max}}(p_{v_1,r}^{\text{curve}} v_1 + p_{v_2,r}^{\text{curve}} v_2) - v_{1,\text{avg}_r} a_2 + v_{2,\text{avg},r} a_1 \geq -\kappa_{\text{max}} p_{0,r}^{\text{curve}}, \tag{14b}$$

for all $r \in \{1, \dots, N_R\}$. Note that the least-squares fit that computes the parameters $(p_{0,r}^{\text{curve}}, p_{v_1,r}^{\text{curve}}, p_{v_2,r}^{\text{curve}}) \in \mathbb{R}^3$ is constrained to be (i) non-negative and (ii) a lower-bound. This as (i) the division required to yield (13) relies on the non-negativity of $\sqrt{v_1^2 + v_2^2}^3$ and (ii) the replacement of $\sqrt{v_1^2 + v_2^2}^3$ by a function that evaluates to lower-or-equal values to the original function guarantees satisfaction of $\kappa_{\text{min}} \leq \kappa \leq \kappa_{\text{max}}$.

4) *Cost Function*

The cost objectives as described conceptually in Sec. II are encoded through the techniques described below.

a. Path following Deviation from reference trajectory $\hat{X}_k = \begin{bmatrix} \hat{x}_1^\top & \cdots & \hat{x}_h^\top \end{bmatrix}^\top$ set by the routing module, with corresponding $\hat{U}_k = \begin{bmatrix} \hat{u}_0^\top & \cdots & \hat{u}_h^\top \end{bmatrix}^\top$ is to be penalized. To this end, $J_\text{p}(X_k, U_k)$ is formulated as

$$J_\text{p}(X_k, U_k) = \sum_{k=0}^{h-1} \|x_{k+1} - \hat{x}_{k+1}\|_Q^2 + \|u_k - \hat{u}_k\|_R^2, \tag{15}$$

where the $\|\cdot\|_A^2$ notation is defined as $\|x\|_A^2 = x^\top A x$ and $Q = Q^\top \in \mathbb{R}^{n \times n}$, $R = R^\top \in \mathbb{R}^{m \times m}$, $Q \succ 0$ and $R \succeq 0$. Which can be written without the summation as

$$J_\text{p}(X_k, U_k) = (X_k - \hat{X}_k)^\top \Omega (X_k - \hat{X}_k) + (U_k - \hat{U})^\top \Psi (U_k - \hat{U}_k),$$

where $\Omega = \text{Diag}\,(Q, Q, \dots, Q)$, $\Psi = \text{Diag}\,(R, R, \dots, R)$ of appropriate sizes. Eliminating X_k through the substitution of relation (7) yields the quadratic form in U_k as

$$J_\text{p}(x_0, U_k) = U_k^\top H_\text{p} U_k + 2 f_\text{p}^\top U_k + C_\text{p}, \tag{16}$$

where

$$H_\text{p} = \Gamma^\top \Omega \Gamma + \Psi, \tag{17a}$$

$$f_\text{p} = ((\Phi x_0 - \hat{X}_k)^\top \Omega \Gamma - \hat{U}^\top \Psi)^\top, \tag{17b}$$

$$C_\text{p} = (\Phi x_0 - \hat{X}_k)^\top \Omega (\Phi x_0 - \hat{X}_k) + \hat{U}^\top \Psi \hat{U}. \tag{17c}$$

b. Comfort The physical comfort experience of a human passenger is largely invariant of velocity and location, as a different magnitude of a constant vehicle velocity, when, *e.g.,* reading a book, is only discernable through a potential change in noise caused by drag. Similarly, the effect of a relatively constant vehicle acceleration, although directly detectable by a human, is quickly mitigated by a change of posture.

Conversely, rapid changes in the acceleration vector do compromise the comfort of passengers by not allowing them time to adapt to a new acceleration level. Using this observation, the comfort of passengers can be understood to correlate inversely with the jerk values of the vehicle trajectory. The comfort objective J_c is therefore designed to penalize large jerk values through the cost-term

$$J_\mathrm{c}(U_k) = \sum_{\nu=1}^{H-1} \|u(\nu) - u(\nu-1)\|_V^2,$$

where V is a positive-definite $m \times m$ matrix of real values. This objective, through the use of U_k, can alternatively be written as

$$J_\mathrm{c}(U_k) = (\Delta U_k)^\top \Pi \Delta U_k, \tag{18}$$

where $\Pi = \mathrm{Diag}\,(V, V, \ldots, V)$ and

$$\Delta = \begin{bmatrix} I & -I & 0 & \cdots & 0 \\ 0 & I & -I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & I & -I \end{bmatrix},$$

of appropriate size.

c. Merging the cost terms To facilitate the balancing of comfort vs. progress in the final cost function $J(x_0, U_k)$, the comfort-biasing scalar weighting parameter $w \in \mathbb{R}_{>0}$ is introduced to form the complete cost formulation that combines (16) and (18) as

$$\begin{aligned} J(x_0, U_k) &= J_\mathrm{p}(x_0, U_k) + wJ_\mathrm{c}(U_k), \\ &= U_k^\top (H_\mathrm{p} + w\Delta^\top \Pi \Delta) U_k + f_\mathrm{p}^\top U_k + C_\mathrm{p}. \end{aligned} \tag{19}$$

5) Optimal Control Problem Formulation

The MIQP problem posed conceptually in Sec. II yielding the ideal trajectory T^* is thereby computed through the optimization of objective function

$$U_k^* = \arg\min_{U_k} \,(19),$$

while subject to constraints

$$(7), (9), (11), (12) \text{ and } (14),$$

yielding T^* through the post-processing conversion step $T^* = \Phi x_0 + \Gamma U_k^*$.

C. Emergency Maneuver

The emergency maneuver planning is subject to the same cost-function and constraint architecture as the ideal trajectory planner. The differences between them arise in the components of

1. cost function tuning,
2. initial state,
3. predictions they avoid,
4. reference trajectory.

The cost function of the ideal trajectory planning is tuned to yield a comfortable (low in jerk and acceleration) controller that tracks the trajectory well, *i.e.,* a balance is found between comfort and trajectory tracking. In the emergency maneuver planning cost function, more emphasis is put on the tracking of the (zero-velocity) trajectory, *i.e.,* the weight factor w described in (19) of the two planners satisfy $w_{\text{ideal trajectory}} > w_{\text{emergency maneuver}}$. Additionally, the weight penalizing deviation from the reference locations $\|s(k) - s_\mathrm{ref}\|^2_{Q_{\{1,3\}}}$, *i.e.,* the first- and third diagonal elements of Q are set to zero in the emergency maneuver planning problem as only the velocity reduction of the vehicle matters for the emergency maneuver, in contrast to the positive weights on spacial error in the ideal trajectory planning problem.

The initial state used to plan the ideal trajectory is simply $x(0)$, while the emergency maneuver problem starts planning from the first-planned ideal trajectory state: $x^*(1)$.

As for the prediction avoidance constraints (11), (12), the emergency maneuver planning is constrained to avoid the legal reachable set prediction polygons, as avoiding these regions guarantees the (legal) safety of all its future planned states at the moment of planning. In contrast, in the original formulation of the fail-safe trajectory planning protocol [21], the ideal trajectory planning is expected to plan using the most likely scenario prediction. In this work, the ideal trajectory planning does not plan using the most likely scenario prediction exclusively, but instead uses a mix of legal reachable set- and most likely scenario predictions. The prediction set considered by the ideal trajectory planning problem is constructed by taking the first $h_\text{form} \in \{0, \ldots, h-1\}$ legal reachable set predictions and appending the most likely scenario predictions for the rest of the horizon $k \in \{h_\text{form}+1, \ldots, h\}$. Tuning the length of this *formal horizon* h_form results in a trade-off between conservativeness of the resulting path and frequency of emergency maneuver interventions, two undesirable traits of the algorithm.

Lastly, the reference trajectories. The reference trajectory of the ideal trajectory planning is constructed as a maximum acceleration-to-desired-velocity trajectory along the centerline of the desired lanelet, originating from the point on this centerline closest to the ego vehicle. In contrast, the emergency maneuver planning trajectory only contains a set of 0-velocities as its spacial deviation is, as mentioned before, not penalized by the cost-function, *i.e.,* the trajectory tracking component is repurposed to penalize any velocity magnitudes away from zero. This is done as bringing your vehicle to a standstill is assumed a universally safe action [30].

D. Legal Reachable Set

The legal reachable set prediction, which the emergency maneuver uses to guarantee its safety, is an essential part of this behavior planner. By confirming that a trajectory does not cause the ego vehicle to intersect the legal reachable sets of any of the environment vehicles, one can guarantee that it is impossible for the vehicles to collide during the execution of this trajectory when all road users abide to the law. The legal reachable set describes the complete set of points that *could* be occupied by an environment vehicle at a future moment in time, given that it complied to the (i) law of the road and the (ii) laws of physics.

The legal reachable set prediction is generated using the method developed in [30] with a small number of adaptations. The legal reachable set method constructs an over-estimate of the physically reachable set w.r.t. the environment vehicle dynamics and subsequently reduces this set through environmental/legal considerations such as road boundaries, speed limits and safe merging distance to yield a legal reachable set over-estimation. Over-estimations are employed as the computation of exact reachable sets of non-linear dynamical systems such as those of the traffic participants is impossible within the constraints of a live behavior planner [31].

The approach, using polygonal reachable set over-estimations $\hat{\mathcal{R}}_\alpha$, was first developed in [20]. Given two over-estimations, *e.g.,* one that overestimates the amount of acceleration that can be achieved in all directions $\mathcal{R} \subset \hat{\mathcal{R}}_{a_{\max}}$ and one that notes the reachable set must be a sub-set of the entire road-network $\mathcal{R} \subset \hat{\mathcal{R}}_{\text{road}}$, taking their intersection yields a less conservative estimation that is still guaranteed to be an over-estimation

$$\mathcal{R} \subset \hat{\mathcal{R}}_{a_{\max}} \cap \hat{\mathcal{R}}_{\text{road}}.$$

Similarly, a region $\hat{\mathcal{R}}_\alpha$ can be subtracted from the legal reachable set estimate $\hat{\mathcal{R}}$ by ensuring the subtracted region is an under-estimation of the true region that cannot be reached. The legal reachable set applied in this work is constructed by the intersection, and subtraction of the following over- and under-estimations, see Fig. 12

1. maximum acceleration (in all directions),
2. set of legal lanes w.r.t. current environment vehicle location,
3. speed limit on the road (plus a certain percentage),
4. the duty to switch lanes only when a sufficiently, large safety-gap is sustained,
5. no backwards motion,
6. avoidance of the ego vehicle predecessor.

One important adaptation has been made to the algorithm for legal reachable set prediction [30] to enable it to be used in the context of this work. The merging of the legal reachable set polygons that are defined per lanelet in [30], are instead merged into a single polygon through the use of its outer-boundaries. In the original algorithm, the occupancy of every lanelet is defined through a unique polygon. The imperfections of road boundary data causes small holes to occur on the boundary between two lanelets when doing a direct merge in such scenario. When combined with the configuration space method (shrinking feasible space by r_{circles}, Sec. III-B2), this would result in large erroneous holes in the feasible space.

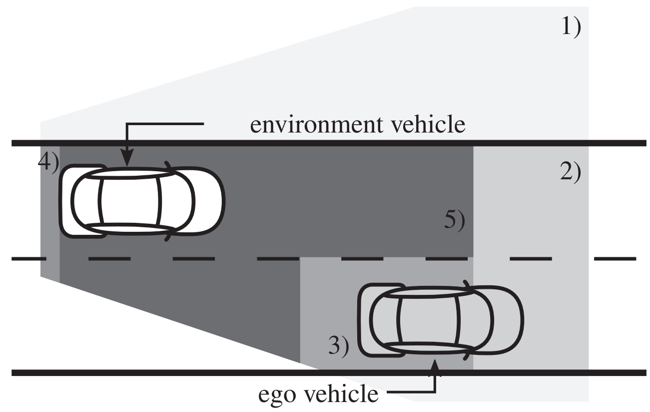


Figure 12: Illustration visualizing five of the six reachable-set shrinking steps. 1) the maximum acceleration in all directions overestimation; 2) the reduction to the set of all legal lanes; 3) the satisfaction of the speed limit (with a slight margin); 4) the illegality of switching lanes without leaving a safety-gap; 5) no backwards motion; 6) avoidance of the ego vehicle predecessor, similar to 5) but not displayed as this only occurs when the ego vehicle shares a lane with the environment vehicle.

IV RESULTS

In this section, the proposed SCBP method is validated using the application scenarios described in Section II-A. First, on the straight lane merging scenario, an example of failure of the original MIQP method due to most likely scenario prediction uncertainty is provided. Next, the application of the SCBP on the same scenario is shown to result in collision free motion. This straight lane merging scenario is then simulated another 1000 times subject to randomly determined environment vehicle trajectories. This is used to quantify the difference between collision avoidance efficacy in the original MIQP and novel SCBP method over varying conditions.

Additionally, the abilities of the SCBP method are demonstrated qualitatively on a curved lane merging scenario. It is shown that the method is able to autonomously (i) merge onto a road, (ii) pass environment vehicles, (iii) handle non-straight roads, and (iv) keep an appropriate amount of safety distance, while existing methods are generally only able to perform a strict sub-set of these. Finally, the running time of the method is analysed.

A. Collision Example

To provide a benchmark for comparison, consider the example in Fig. 13A. This straight lane merge scenario as introduced in Sec. II-A is simulated for hyper parameters $(N_R, \tau, h, h_E, h_{\text{form}}) = (4, 1/3, 8, 5, 0)$. The ego vehicle is tasked with following the center-line of the middle lane with 80 (km·h^{-1}). Vehicle 1, occupying this middle lane, unexpectedly reduces its velocity by 20% for two consecutive time-steps at $k \in \{8, 9\}$, after which it accelerates back to its target velocity of 75 (km·h^{-1}). Using the original MIQP-MPC method [17], a trajectory is generated in receding horizon that fully relies on the most likely scenario prediction. Since the most likely scenario prediction assumes constant acceleration, the brake maneuver is not anticipated. The ego vehicle does not maintain enough distance from its predecessor, vehicle 1, to respond when vehicle 1 activates its brakes. This results in infeasibility of the MIQP-MPC planning problem, *i.e.,* the ego vehicle finds itself in an unavoidable-collision state. The simulation is terminated at this failure.

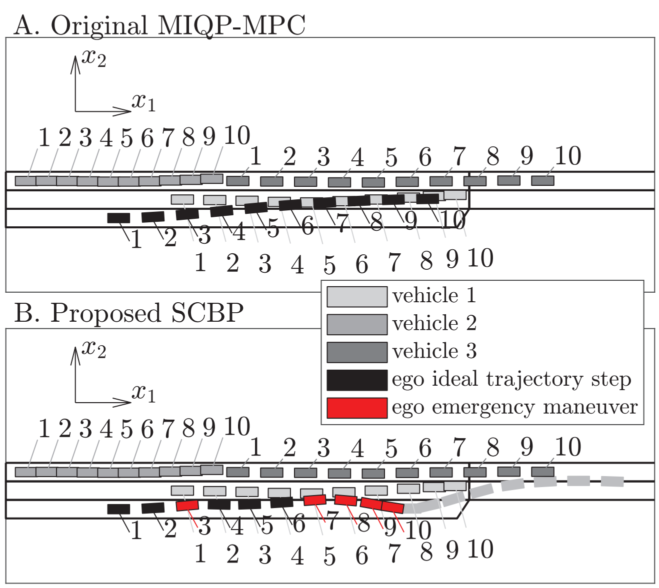


Figure 13: A: the original MIQP-MPC planner becomes infeasible (encounters unavoidable collision) in this randomly generated simulation scenario, which terminates the simulation at $k = 10$. B: the proposed SCBP method uses emergency maneuver steps to anticipate on possible brake maneuvers by environment vehicles. It manages able to safely navigate this scenario.

B. Collision Prevention - SCBP Approach

Trajectory planning through the proposed SCBP method is compared to the original MIQP-based benchmark example described above. Consider the same example scenario as simulated in Fig. 13A. Fig. 13B describes the trajectory resulting from the application of the SCBP controller. Through the four emergency-maneuver-activation events at $k \in \{7, 8, 9, 10\}$, the SCBP first preventively and later reactively slows the ego vehicle down as it takes into consideration the possibility of unexpected brake maneuvers. This enables the SCBP method to successfully reach the target lane without collision, illustrating the ability to preserve safety in contrast to the MIQP-MPC benchmark example.

a. SCBP Safety The original MIQP-MPC method has no effective way of maintaining spacial safety buffers. This can cause it to favour almost touching predicted future occupancy locations, as it often results in the most efficient path. Unexpected environment vehicle maneuvers occurring during such close encounters thereby have the potential to result in collision. The SCBP is implicitly constrained to maintain a safety buffer as it is required to have a feasible safety maneuver planned at each time step. Moving the ego vehicle too close to an environment vehicle can prohibit the existence of a feasible emergency maneuver. Steps that compromise the spacial safety buffer are thereby prevented from executing in favor of a previously computed emergency maneuver.

This effect can be observed in large-volume simulation. Out of 1000 randomly generated straight lane merge simulations scenarios, 24.7% resulted in failure when employing the original MIQP-MPC method. Failure meaning the method either caused termination of the simulation through an observed collision (14.4%) or infeasibility of the planning problem (10.3%). By exchanging the original MIQP- for the SCBP method, this failure percentage on the same 1000 scenarios drops to a mere 0.5%. All of them collisions caused by the simulated environment vehicles not being responsive to the ego vehicle actions, *i.e.,* the emergency braking of the ego vehicle resulted in collision as environment vehicle followers did not slow down in response. This means the ego vehicle was not responsible for any of the recorded collisions in the SCBP simulations.

C. Curved Lane Merge

To illustrate the extensive ability of the SCBP algorithm to make decisions and cope with environmental factors, the behavior planner is simulated on the example scenario in Fig. 14. This curved lane merge scenario as introduced in Sec. II-A is simulated for hyper parameters $(N_R, \tau, h, h_E, h_{\text{form}}) = (8, 1/3, 6, 5, 2)$. The ego vehicle is initialized with initial velocity $\|v(0)\|_2 = 80$ (km·h^{-1}) on the center-line of the *on-ramp* (*right lane*) in the direction of its center-line. It is tasked with following the center-line of the *middle lane* with target velocity $\|v\|_2 = 80$ (km·h^{-1}). Two environment vehicles drive respectively $\|v^1(t)\|_2 = 40$ and $\|v^2(t)\|_2 = 44$ (km·h^{-1}) on the center-line of the middle lane for the entire duration of the simulation. Optimizing its objective requires the ego vehicle to merge onto the middle lane while avoiding collision with the environment vehicles, and subsequently either following its predecessor or passing him.

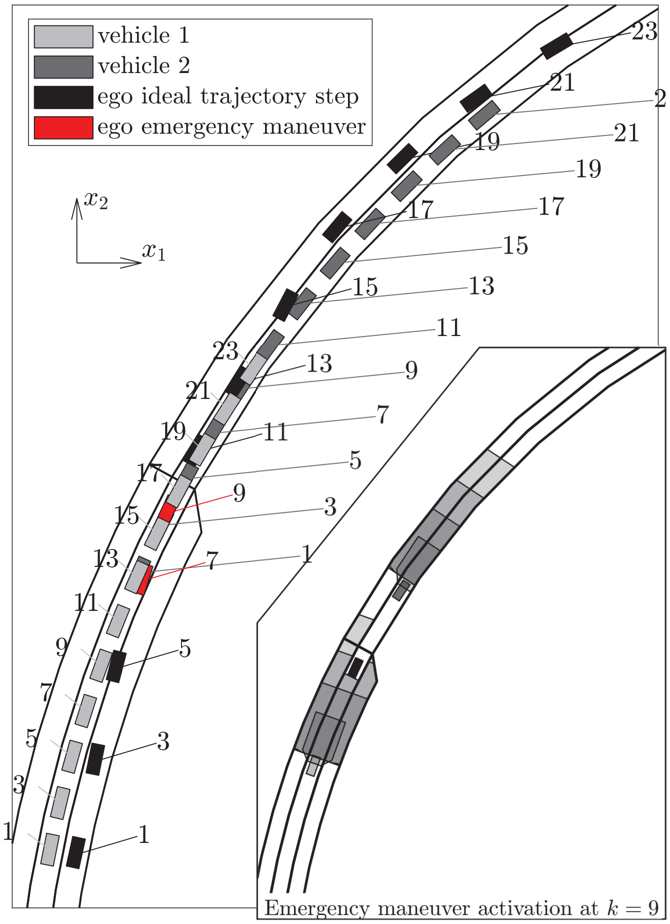


Figure 14: The curved merge scenario with $\|v^1\|_2 = 40$ (km·h^{-1}), $\|v^2\|_2 = 44$ (km·h^{-1}); simulated for $(N_R, \tau, h, h_E, h_{form}) = (8, 1/3, 6, 5, 2)$. The emergency maneuver was activated at $k \in \{6, 7, 9\}$. A visualization of the scenario that lead to this decision at $k = 9$ is displayed in the insert.

As vehicle 1 has a lower velocity than the ego vehicle, the ego vehicle decides to first pass him and merge in front. As vehicle 2 starts out ahead of the ego vehicle, once on the middle lane, the ego vehicle is forced to either (i) slow down to adapt its velocity to its predecessor or (ii) pass him on the left. As can be seen in the simulation results in Fig. 14, the ego vehicle autonomously makes the decision to pass vehicle 2 on the left as this optimizes its combined objective of following the centerline of the middle lane and making progress along the road with 80 (km·h^{-1}) over the planning horizon h.

The curvature of the resulting path is plotted in Fig. 15 over the time steps $k \in \{1, \ldots, 23\}$ corresponding to Fig. 14. The fact that the line described by the trajectory curvature $\kappa(k)$ remains within the bounds $\kappa_{\text{max}} \geq \kappa(k) \geq \kappa_{\text{min}}$ for $k \in \{1, \ldots, 23\}$ indicates that the trajectory is trackable for the ego vehicle model (1). The same holds for the acceleration and velocity magnitudes and their limits, visualized in Fig. 16.

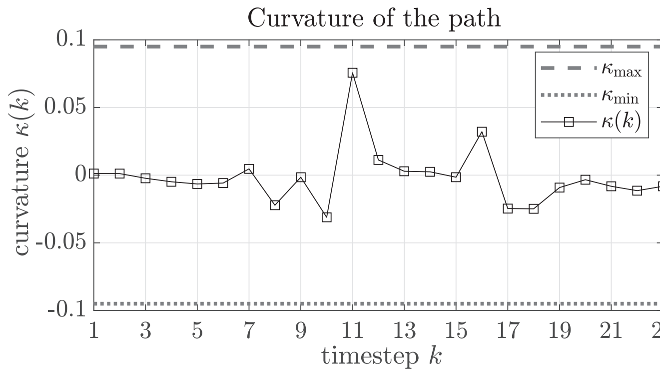


Figure 15: The curvature of the trajectory traversed in the curved merge scenario.

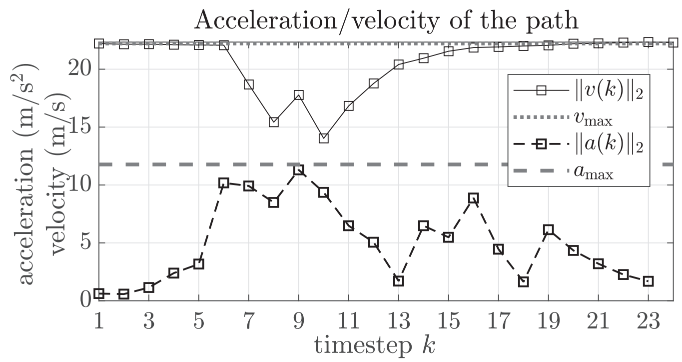


Figure 16: The acceleration and velocity magnitudes of the trajectory traversed in the curved merge scenario.

a. Safety Buffer While executing its behavior planning, the SCBP method is able to adaptively keep a safe amount of distance from environment vehicles. Note how the algorithm maintains a safety margin w.r.t. vehicle 2 when passing at timestep $k = 15$, cf. Fig. 14. The ego vehicle can be seen to move to the left lane early, leaving a buffer of space between it and vehicle 2, which is not explicitly demanded in the behavior planner formulation but a result of the non-zero formal horizon setting $h_{\text{form}} = 2$. Such distance would, in case of $h_{\text{form}} = 0$, be created through emergency maneuver activations, cf. Fig. 13B.

Note that the decisions made at time steps $k \in \{6, 7, 9\}$ were emergency maneuvers, in all of these cases caused by the impossibility of a safe emergency maneuver at their newly planned ideal trajectory state $T^*_{x(1)}$. To illustrate the decision making process, the planning problem at $k = 9$ is visualized in the insert of Fig. 14. As the ideal trajectory is almost unobstructed after $k = h_{\text{form}}$, it plans its next step close to (80 km·h^{-1}). The feasible space left after the $1 + h_E = 6$ legal reachable set prediction steps is subsequently insufficient to enable a new emergency maneuver plan.

D. Computational Time

The straight lane road encoding requires significantly less convex polygons (2 vs. 18 when compared to the curved lane merge). This has the amount of indicator constraints in the problem reduced by an almost proportional amount.

The computational time of a planning cycle is an essential property of the algorithm and must remain sufficiently low to be applicable in practice. The straight lane merge scenario has been initialised randomly one additional time, after which this set up has been simulated for a grid of hyper parameter values constructed by $h \times N_R = \{3, \ldots, 8\} \times \{4, 8, 12, 16\}$, where the emergency maneuver planning horizon h_E is set to $h_E = h - 1$ and the discrete time step size to $\tau = 1/3$ (s). The average time required for a planning cycle under these conditions has been recorded and visualized in Fig. 17.

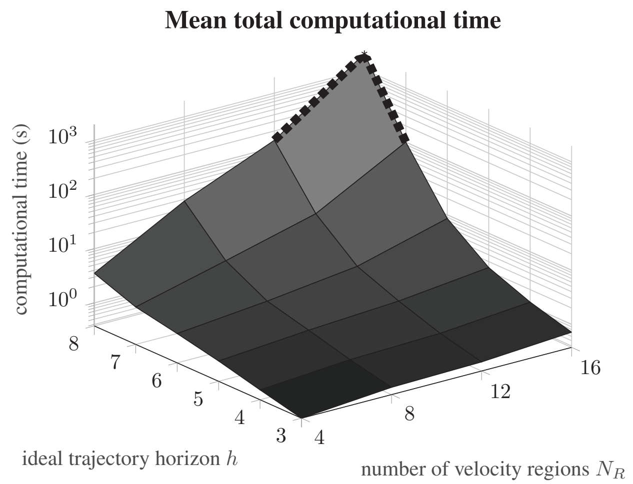


Figure 17: The average computational time of one planning cycle as a function of the hyper parameters $h \in \{3, \ldots, 8\}$, $N_R \in \{4, 8, 12, 16\}$. The computation time displayed at $(h, N_R) = (8, 16)$ is theoretical, it was generated by extrapolating the rest of the data and was not actually computed as the 18 cycles required to run this simulation scenario are expected to take $18 \cdot 33 = 594$ (min), which is 9.9 (h).

Here, on the logarithmic z-axis of the surface plot, the exponential increase of computational time with increasing hyper parameters becomes clear. This exponential nature can be traced back to the emergency maneuver generation. In Fig. 18 and 19, the computational time of the individual algorithm components are displayed separately. It becomes clear that the ideal trajectory planner scales polynomially with the hyper parameters, which is observed on the log-scaled axis as an increase with a linear character. The computational time of the emergency maneuver scales more exponentially than the combined running time, cf. Fig. 18.

The reason for this could be the fact that the emergency maneuver uses the legal reachable set until its last prediction step, in contrast to the ideal trajectory that in this case only uses

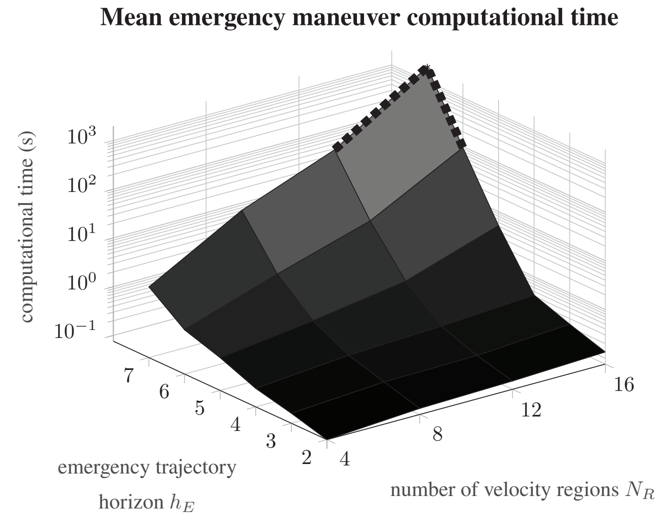


Figure 18: The average computational time of one planning cycle as a function of the hyper parameters $h_E \in \{2, \ldots, 7\}$, $N_R \in \{4, 8, 12, 16\}$. The computation time displayed at $(h_E, N_R) = (7, 16)$ is theoretical, it was generated by extrapolating the rest of the data.

the first $h_{\text{form}} = 2$. The legal reachable set grows with every predicted time step, reducing the feasible space to plan in to the point of (almost) filling the entire road. As the uncertainty, and therefore the legal reachable set, grows endlessly with time, the application of this method loses much of its value for moments far into the future. A solution could be a more tailored formulation of the emergency maneuver optimization problem that aims to bring the vehicle to $v = 0$ in as little time steps as possible and then terminates, disregarding the irrelevant tail that is subject to huge legal reachable sets constraining its feasible space.

As for the lower hyper parameter value runs, the fastest at $(h, N_R) = (3, 4)$ had an average cycle time of 0.38 (s) on an Intel(R) Core(TM) i7-4710MQ CPU 2.50 (GHz), which makes the method too slow for safety critical implementation in its current form. One of the main downsides of the algorithm is the necessity to run the prediction modules, ideal trajectory planner and emergency trajectory planner in sequence. This makes acceleration through parallelisation impossible.

V CONCLUSION

In this paper, the safe, convex behavior planner algorithm is developed for road scenarios populated with mixed traffic environment vehicles. The main idea was to integrate an emergency maneuver safety mechanism into the global optimum yielding MIQP behavior planning architecture proposed by Esterle *et al.* [17]. This modification has enabled the MIQP behavior planning method to be applied safely under uncertainty. We have demonstrated in simulation that over 1000 randomized merge scenarios, not a single collision was caused by the ego

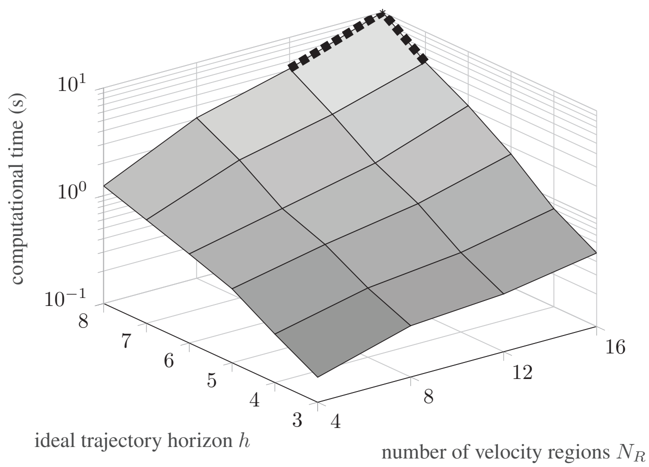


Figure 19: The average computational time of one planning cycle as a function of the hyper parameters $h \in \{3, \ldots, 8\}$, $N_R \in \{4, 8, 12, 16\}$. The computation time displayed at $(h, N_R) = (8, 16)$ is theoretical, it was generated by extrapolating the rest of the data.

vehicle when subject to the proposed SCBP algorithm, while previous work only achieved collision free motion in 75.3% of these scenarios.

The addition of the safety method in its current form has come at a higher processing cost, which has thus-far resulted in prohibitively high computation times, although not by a large margin. Through algorithm acceleration efforts, the running time could potentially be improved to the point of real-time capability, as many such opportunities were likely left on the table.

The current architecture activates a non-adaptive environment maneuver whenever the ego vehicle finds itself in dangerous situations. This prevents the vehicle, in the event of unavoidable collision scenarios caused by dangerous maneuvers by environment vehicles, from actively minimizing impact of collision.

Additionally, the emergency maneuver planning is currently used as a method by which to judge the safety of the next ideal trajectory step retroactively, *i.e.,* after it is already planned. Attempts at making the ideal trajectory take this safety into account while planning thus far resulted in the formulation of the formal horizon h_{form}. A more integrative method might be found in the formulation of the SCBP problem into a single optimization problem. The ideal trajectory could then be generated while constrained to have a feasible emergency maneuver from its first-planned point.

Future work should aim at making the method safer in events where other road users cause dangerous situations. Where the current method merely avoids legal responsibility, it would be more desirable to additionally avoid damage when involved in an accident. Additionally, for the method to become real-time capable, acceleration of the algorithm is required.

REFERENCES

[1] U.S. Department of Transportation, National Highway Traffic Safety Administration, "National motor vehicle crash causation survey," NHTSA Report DOT HS 811 059, Washington, DC: USA, 2008. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059

[2] ISO/SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," ISO/SAEPA Report S22736, 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[3] S. Waslander. Introduction to self-driving cars – module 2: Self-driving hardware and software architectures. Coursera. [Online]. Available: https://www.coursera.org/learn/intro-self-driving-cars?specialization=self-driving-cars

[4] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–23, 05 2019.

[5] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, 04 2016.

[6] M. Naumann, H. Königshof, and C. Stiller, "Provably safe and smooth lane changes in mixed trafic," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1832–1837, 2019.

[7] W. Cao, M. Mukai, and T. Kawabe, "Merging trajectory generation method using real-time optimization with enhanced robustness against sensor noise," *Artificial Life and Robotics*, vol. 24, pp. 527 – 533, 2019.

[8] C. Miller, C. Pek, and M. Althoff, "Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1954–1961, 2018.

[9] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1253–1258, 2013.

[10] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Engineering Practice*, vol. 41, pp. 124–133, 2015.

[11] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 644–649, 2015.

[12] X. Qian, F. Altché, P. Bender, C. Stiller, and A. de La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An

miqp perspective," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 205–210, 2016.

[13] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4459–4466, 2019.

[14] O. de Groot, B. Brito, L. Ferranti, D. M. Gavrila, and J. Alonso-Mora, "Scenario-based trajectory optimization in uncertain dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, pp. 5389–5396, 2021.

[15] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 2994–3008, 2018.

[16] H. Andersen, J. Alonso-Mora, Y. H. Eng, D. Rus, and M. H. Ang, "Trajectory optimization and situational analysis framework for autonomous overtaking with visibility maximization," *IEEE Transactions on Intelligent Vehicles*, vol. 5, pp. 7–20, 2020.

[17] K. Esterle, T. Kessler, and A. Knoll, "Optimal behavior planning for autonomous driving: A generic mixed-integer formulation," *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1914–1921, 2020.

[18] A. Stockem Novo, M. Krüger, M. Stolpe, and T. Bertram, "A review on scene prediction for automated driving," *Physics*, vol. 4, no. 1, pp. 132–159, 2022. [Online]. Available: https://www.mdpi.com/2624-8174/4/1/11

[19] M. Althoff and J. Dolan, "Online verification of automated road vehicles using reachability analysis," *Robotics, IEEE Transactions on*, vol. 30, pp. 903–918, 08 2014.

[20] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Transactions on Intelligent Vehicles*, vol. PP, pp. 1–1, 10 2016.

[21] 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.

[22] S. Magdici, Z. Ye, and M. Althoff, "Determining the maximum time horizon for vehicles to safely follow a trajectory," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, 2017.

[23] P. Polack, F. Altché, B. Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" 06 2017, pp. 812–818.

[24] R. Rajamani, *Vehicle Dynamics and Control*, 01 2006.

[25] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 420–425, 2014.

[26] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Autom.*, vol. 35, pp. 407–427, 1999.

[27] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com

[28] J. O'Rourke, *Computational Geometry in C (2nd Ed.)*. USA: Cambridge University Press, 1998.

[29] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha — a local, continuous method," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 450–457, 2014.

[30] M. Koschi and M. Althoff, "Spot: A tool for set-based prediction of traffic participants," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1686–1693.

[31] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," in *Dissertation*. Technische Universität München, 2010. [Online]. Available: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4