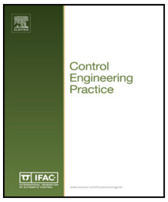
Control Engineering Practice 147 (2024) 105932

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac



Safe motion planner for autonomous driving based on LPV MPC and reachability analysis

Álvaro Carrizosa-Rendón [a,b], Vicenç Puig [a,b,*], Fatiha Nejjari [b]

[a] Institut de Robòtica i Informàtica Industrial CSIC-UPC, Carrer Llorens Artigas, 4-6, Barcelona, 08028, Spain
[b] Research Center for Supervision, Safety and Automatic Control (CS2AC), Universitat Politècnica de Catalunya (UPC), Rambla Sant Nebridi 22, Terrassa, 08222, Spain

ARTICLE INFO

Keywords:
Autonomous vehicles
Motion planning
Robust planner
Tubes
Safety
Coordination
Zonotopes
Constrained zonotopes
LPV

ABSTRACT

This article presents an innovative optimization-based solution to the collision avoidance challenge for autonomous vehicles. The presented approach consists in an online motion planner designed to define feasible and efficient paths able to deal with dynamic surroundings while implicitly ensure safety in the proposed maneuvers. The fact of considering moving obstacles inside the motion planner increases the complexity of the problem while forces it to be executed more frequently as others. To reduce this computational complexity, the approach presented counts with a two stages translation of the commonly used non-linear optimization-based structure into a QP formulation which can be easily solved. The first stage is based on the use of LPV matrices in the dynamic constraints of the vehicle. The second stage consists in performing a reachability analysis based on set propagation to obtain linear expressions of the permitted inputs and reachable states which guarantee safety conditions.

1. Introduction

Coordination of autonomous vehicles guaranteeing safety is a wide topic with different challenges to be solved, being collision avoidance one of its main aspects to guarantee safety. A common way to address the problem is through the motion planner (MP).

Many different motion planning methods have been developed and used along the years. As explained in Paden et al. (2016), it is important to consider the scenario where it would be used, being many of them not tractable in dynamic surroundings. This implies that are not able to deal with dynamic obstacles. Moreover, to deal with moving obstacles it is required to use time-dependent motion planners (trajectory planning) which generate references for the different states of the vehicle (kinematic and dynamics) instead of planning a spatial path time-independent with the locations the vehicle should track.

Therefore, the two main categories of motion planners compatible with dynamic surroundings are the sampling-based solutions, having two subcategories (graph and incremental search strategies) and optimization-based approaches. As explained in Manzinger (2021), other techniques based on machine learning, reinforcement learning or end-to-end learning are becoming popular, but their scalability to all possible scenarios and a verification of their capability to guarantee safety is still under study, making them nowadays unsuitable.

Considering all previous reasons, the approach presented in this paper follows one of the predominant lanes of research, which is the optimization-based approach. This sort of solution consists in formulating an optimization problem that includes a mathematical expression to evaluate the considered performance index while the dynamical, safety and physical limitations of the system are considered as a set of constraints to accomplish.

As detailed in the survey (Paden et al., 2016), several optimization-based approaches can be found in the literature which use model-based movement predictions to design the motion plan. Different formulations and objectives have been presented, having all in common the difficulty of skipping complex formulations to compute the optimal solution with low computational cost, able to operate in real-time. Consequently, the majority of the solutions proposed are focused on presenting approaches to simplify the problem or to reduce the computational complexity to make possible its real-time implementation.

A very extended way of addressing the problem of high computational costs is to design motion planners which deal with the complexity of non-linear expressions but avoid part of the computational cost by delegating the obstacles avoidance to the motion controller (MC). For example, Hegedüs et al. (2017) propose a non-linear formulation to find the optimal path using a dynamical model of the vehicle, while propose to check collision avoidance with a higher-level supervisor

[*] Corresponding author.
E-mail address: vicenc.puig@upc.edu (V. Puig).

https://doi.org/10.1016/j.conengprac.2024.105932
Received 19 November 2023; Received in revised form 28 March 2024; Accepted 28 March 2024
Available online 6 April 2024

outside the optimization problem. Moreover, the authors consider that the computational complexity is still significant and remark that the real-time execution is not viable with nowadays technology.

Another common manner of reducing the computational complexity is simplifying the expressions. One example of that is the MP for racing vehicles presented in Caporale et al. (2018). This solution uses dynamical models with a formulation based on a trade-off between the curvature of the path (to avoid slipping) and track length (to reach the goal as fast as possible). To reduce the computational cost, the authors propose to study the path as a sequence of linear segments, while use Taylor expansion to simplify the expression of the curvature. This kind of approaches allow the implementation but sacrificing the optimality. Additionally, this solution does not consider collision avoidance inside the MP, leading it to a hypothetical external module which provides the sufficient constraints to find a collision-free path.

Other approaches, such as the MP proposed in Liu et al. (2017), avoid the complexity of the problem using kinematic models. This work presents an interesting solution based on a mixed integer problem to decide the most suitable maneuver selecting lanes. Once the lane is selected, the potential field associated to the lane is computed to avoid getting close to obstacles. Moreover, the collision problem is addressed by approximating the vehicle and the different obstacles as polyhedra establishing a set of constraints to exclude solutions where polyhedra intersect. It is important to remark that many MC use dynamical reference, thus this methodology would be incompatible, but presents an interesting approach based on intersections of polyhedra to study the collision avoidance challenge.

Even more interesting is the proposal presented in Scheffe et al. (2022), where the authors remark the need of reaching high update rates and propose approaching the non-linear expression by convex approximations.

Meanwhile, the number of optimization-based solutions which apply set-theory are becoming more and more frequent. Between them, different approaches using different techniques and objectives have been presented. The approach presented in Danielson et al. (2020) computes robust positive-invariant sets to define the subset of states where it is safe to generate the MC reference. The solution proposes a methodology to deal with disturbances and parametric uncertainties. Even more interesting, the solution can cope with dynamic obstacles by bounding the time when the vehicle can transit between sets.

Also, with set-theory, different approaches propose solutions based on generating safety corridors or propagating the states to define regions where safety is guaranteed for allowed every vehicle movement. For example, Manzinger et al. (2021) propose a methodology to define corridors by reachability analysis and integrate them inside the MP. To reduce the computational complexity, some assumptions to achieve linear expressions for the kinematics aspects of the vehicle are done. Schäfer et al. (2023) also propose a solution with similar objectives. The authors describe a methodology to identify collision-free driving corridors using reachability analysis. Once developed, they are approximated by polyhedra to reduce the complexity of the solution by using them as constraints of the MP. The authors validated their methodology, combining it with already existing optimization-based MP.

Besides that, other approaches reduce the computational cost without linearizing or simplifying the model by using Linear Parameter Varying (LPV) matrices. For example, in Alcalá, Puig, and Quevedo (2020), the non-linearities are embedded inside a linear model with time-varying terms using an LPV state-space representation. By this way, the MP solves a problem where the dynamic constraints of the vehicle are linear. However, the MP can only deal with static obstacles by tightening the boundaries of the states to avoid collisions. Those techniques can be perfectly combined with tube-based solutions as in Alcalá et al. (2020), where LPV matrices and tubes are combined to solve a related problem associated to the MC of an autonomous vehicle, highlighting the utility that such a combination could have in the field of motion planning. Other example is the approach presented in Nezami et al. (2022) which proposes an optimization-based solution reducing the computational cost by means of LPV matrices. However, the approach focuses on the specific task of lane keeping and is not oriented on dealing with moving obstacles.

Thus, optimization-based non-linear solutions are not easily implementable due to computational costs and need to be simplified or adapted in case they are used on-line to deal with dynamic obstacles. Inside that, the use of LPV matrices is not extended but seems to offer interesting results while the inclusion of sets is also interesting for the side of guaranteeing safety.

On the basis of the aforementioned aspects, the aim of this paper is to present a novel motion planner for an autonomous vehicle able to generate references for a dynamic motion controller in dynamic surroundings. It consists in the generation of kinematic and dynamic references for the MC guaranteeing the avoidance of dynamic obstacles in real time, proposing a conversion of the non-linear optimization problem into a QP problem combining LPV state-space representation for the non-linear constraints associated to the dynamics of the vehicle, and a reachability analysis of the states to obtain linear expressions of the sets of states able to guarantee safety conditions. By this way, the complexity of the problem is reduced enabling its implementation in real time.

2. Problem statement

In this work, an autonomous vehicle with V2V communication systems and a level of autonomy 5 is considered. That means that the vehicle can drive autonomously without any human interaction and counts with a communication system to send and receive information from other vehicles such as their current and expected future location or the future maneuvers planned. The objective is to design an MP able to generate the kinematic and dynamic references for the MC guaranteeing collision avoidance.

Generally, the proposed optimization-based motion planners are non-linear and follow a structure based on an MPC without references, being the aim of the motion planner to design them. The structure is typically composed by a cost function, used to evaluate the performance of the path designed along a prediction horizon (H_p) with a determined sampling time (T_s), and a set of constraints, used to ensure feasibility by accomplishing the physical and dynamical limitations of the vehicle while guaranteeing not compromising safety conditions.

The aforementioned formulation of the optimization problem can be expressed with the following expressions:

$$\min_{u_0,\ldots,u_{H_p-1}} \quad J(x_k, u_k) \tag{1}$$

subject to the following discretized constraints:

$$x_{k+1} = f(x_k, u_k) \cdot T_s + x_k \tag{2}$$

$$g(x_k, u_k) \leq 0 \tag{3}$$

$$x_k \in [\underline{x}, \overline{x}] \tag{4}$$

$$u_k \in [\underline{u}, \overline{u}] \tag{5}$$

$$\Delta u_k \in [\underline{\Delta u}, \overline{\Delta u}] \tag{6}$$

Eq. (1) is a cost function with linear and quadratic terms rewarding or penalizing the values of the states, inputs and their slew rates. In comparison with a pure MPC controller, the MP does not count with references; it generates them. Expressions (4), (5) and (6) delimit the upper and lower bounds of the states, inputs and slew rates according to safety and physical limitations.

The first non-linearities come from Eq. (2), which are the constraints associated to the dynamic aspects of the vehicle. The second source of non-linearities is the inequalities (3), which are non-linear expressions used to ensure that there is no collision between the vehicle studied and the obstacles nearby.

The methodology proposed in this work consists in combining the constraints associated to upper and lower bounds (4)–(6) presented above with the obstacles avoidance challenge by performing a reachability analysis. In Althoff et al. (2021), the multiple applications that reachability analysis of dynamic systems are discussed, and different computational techniques are proposed, with special emphasis on the interest of methodologies based on set propagation, which is the one performed in the approach here presented.

Several studies have used these techniques for the specific case of autonomous driving. An example of this is the work performed in Söntges and Althoff (2017), where a continuous-time approach is proposed which performs an over-approximation of the reachability analysis, decoupling it in the different Cartesian axes. Subsequently, those locations prohibited due to the presence of obstacles are extracted from the generated tube. In this work, the same philosophy of performing an analysis that does not compute all the reachable states but those that, in addition to being reachable, meet certain specifications such as not exceeding the speed limits, being located inside the lane and not entering the safety region of other vehicles, is followed and performed.

The reachable set at time instant k can be defined as the set of all possible states ($\mathcal{X}_k$) that can be reached at that time instant starting from any state belonging to the set of initial states ($\mathcal{X}_0$) by applying all possible valid input trajectories ($u_{[0,\tau]}$), excluding those trajectories that involve states belonging to the set of forbidden states ($\mathcal{F}$) over the time interval studied (Liu et al., 2022).

$$\begin{aligned}\mathcal{R}_k(\mathcal{X}_0) := \{\mathcal{X}_k(x_o, u_{[o,k]}) | \exists x_0 \in \mathcal{X}_0, \forall \tau \in \{0, \dots, k\}, \\ \exists u_\tau \in \mathcal{U}_\tau : \mathcal{X}_\tau(x_0, u_{[0,\tau]}) \notin \mathcal{F}_\tau\}\end{aligned} \tag{7}$$

In the same manner, the set of forbidden states for a specific time instant ($\mathcal{F}_k$) is defined as all the feasible states for time instant k ($\mathcal{X}_k$) whose location $Q(x_k)$ does not intersect with the location of other agents or the safety region defined around them (O_k) (Liu et al., 2022).

$$\mathcal{F}_k := \{x_k \in \mathcal{X}_k | Q(x_k) \cap O_k \neq \emptyset\} \tag{8}$$

By this way, the optimization problem can be reformulated as follows: Keeping the objective function (9) as in the previous formulation and substituting the non-linear expressions of the constraints by the expressions of the sets of all applicable inputs (12) and all reachable safety states (11). The constraints associated to the dynamic model of the vehicle over the prediction horizon can be discretized and expressed in a state-space representation with the LPV matrices $A(\rho)$ and $B(\rho)$ as shown in (10). More details about the non-linear dynamic model chosen and its conversion into a linear state-space representation by embedding the non-linearities inside time-varying parameters ρ of the linear expressions and then discretizing the system will be given in the following sections.

$$\min_{u_0,\dots,u_{H_p-1}} J(x_k, u_k) \tag{9}$$

subject to:

$$x_{k+1} = A(\rho)_k \cdot x_k + B(\rho)_k \cdot u_k \tag{10}$$

$$x_k \in \mathcal{R}_k(\mathcal{X}_0) \subseteq [\underline{x}, \overline{x}] \tag{11}$$

$$u_k \in \mathcal{U}_k \subseteq [\underline{u}, \overline{u}] \tag{12}$$

The objective is to convert the initial optimization problem into a problem that can be solved by QP methods, which are popular due to their low computational cost. This requires a quadratic objective expression and a set of linear constraints. Therefore, the expressions used to define the sets have to be linear. In this work, a type of sets called zonotopes is used, since they are defined with linear expressions, which significantly reduces the computational cost of operating with them. By this way, the problem is successfully redefined as a QP problem that can be solved with significantly lower computational cost.

As a final remark, it is important to emphasize that the methodology proposed here is designed to be used in scenarios where vehicles drive on free-lanes roads and where there is a single road without the need to make decisions about which road to take. In the case of situations with multiple alternatives, such as splitting the road into two routes, or situations where overtaking can take place on both sides of the vehicle, other decision-making techniques will be necessary, which are not the subject of this study. Examples of this are solutions based on mixed logic as the methods studied in Ioan et al. (2021), or heuristic approaches as in Schäfer et al. (2023) or Manzinger et al. (2021) to choose a single driving corridor to evaluate.

2.1. Model of the vehicle

In this work, the vehicle has been modeled with a dynamical bicycle model with the following states: Three first states representing the linear, lateral and angular velocities of the vehicle ($v_x(t)$, $v_y(t)$ and $\omega(t)$). Three additional states inspired in the curvature-based model proposed at Verschueren et al. (2014) which are distance to the center of the road ($e_L(t)$), difference between the orientation of the vehicle and the curvature of the road ($e_\theta(t)$), and the distance traveled ($s(t)$) measured in the projection of the vehicle over the center of the road. The mathematical expressions of the different states are defined in Eqs. (13)–(18), while a more detailed description of the expressions and the reasoning behind them can be consulted in Alcalá, Puig, Quevedo, and Rosolia (2020).

$$\begin{aligned}\dot{v}_x(t) = a(t) - C_f \cdot \frac{\delta(t) \cdot sin(\delta(t))}{m} + \omega(t) \cdot v_y(t) - \mu \cdot v_x(t) \\ + C_f \cdot sin(\delta(t)) \cdot \frac{\omega(t) \cdot l_f + v_y(t)}{m \cdot v_x(t)}\end{aligned} \tag{13}$$

$$\begin{aligned}\dot{v}_y(t) = -\omega(t) \cdot \frac{C_f \cdot l_f \cdot cos(\delta(t)) - C_r \cdot l_r}{v_x(t) \cdot m} - v_x(t) \cdot \omega(t) \\ +C_f \cdot \frac{\delta(t) \cdot cos(\delta(t))}{m} - v_y(t) \cdot \frac{C_r + C_f \cdot cos(\delta(t))}{v_x(t) \cdot m}\end{aligned} \tag{14}$$

$$\begin{aligned}\dot{\omega}(t) = \frac{C_f \cdot \delta(t) \cdot cos(\delta(t)) \cdot l_f}{I} - v_y(t) \cdot \frac{C_f \cdot cos(\delta(t)) \cdot l_f}{v_x(t) \cdot I} \\ +v_y(t) \cdot \frac{l_r \cdot C_r}{v_x(t) \cdot I} - \omega(t) \cdot \frac{C_f \cdot cos(\delta(t)) \cdot {l_f}^2 + C_r \cdot {l_r}^2}{v_x(t) \cdot I}\end{aligned} \tag{15}$$

$$\dot{e}_L = v_x(t) \cdot sin(e_\theta(t)) + v_y(t) \cdot cos(e_\theta(t)) \tag{16}$$

$$\dot{e}_\theta(t) = \omega(t) + \kappa \cdot \frac{v_y(t) \cdot sin(e_\theta(t)) - v_x(t) \cdot cos(e_\theta(t))}{1 - e_L(t) \cdot \kappa} \tag{17}$$

$$\dot{s}(t) = \frac{v_x(t) \cdot cos(e_\theta(t)) - v_y(t) \cdot sin(e_\theta(t))}{1 - e_L(t) \cdot \kappa} \tag{18}$$

As explained in Verschueren et al. (2014), the main advantage of using a curvature-based model is the fact of obtaining spatial-dependent expression instead of time-dependent ones. This allows simpler expressions for the boundaries, especially for obstacle avoidance. Some other aspects that make this model suitable for the MP are the computation of the orientation of the vehicle in comparison with the curvature of the road and the distance traveled projected over the center of the road. In this way, it is possible to prevent the vehicle from taking very pronounced orientations that obstruct the passage of other vehicles, perform maneuvers that compromise the safety of others, or avoid undesired circumstances such as driving directly to the limits of the road. Even more interesting is the intrinsic computation of the distance traveled, which is a factor commonly used in many MP for the optimization criteria. In case it would not be used, it could be excluded of the model and computed outside the optimization problem, because, as it can be observed, its value does not infer in the other five states of the dynamic model.

Note that in order to be able to compute the spatial states associated with the central curve of the road, it is necessary that the distance to it ($e_L(t)$) is less than the radius of curvature (κ^{-1}) at all times, as explained in Verschueren et al. (2014). However, this is not expected to be the case when driving in urban environments or in conventional roads.

3. Methodology

In this section, the two stages followed to convert the non-linear problem into a QP problem are detailed.

3.1. LPV approach

The benefits of using LPV matrices is the achievement of linear expressions to describe the non-linear reality of the vehicle model defined by Eqs. (13)–(18). This LPV representation reduces the complexity of the problem without introducing the uncertainties associated to inaccurate simplifications of the model or linearization around operational points.

The key aspect of this procedure consists in embedding the non-linearities of the system inside time-varying parameters precomputed at each iteration for each time instant along the prediction horizon. Those parameters are introduced inside the linear expressions as varying terms of matrices A and B obtaining a set of linear constraints for the MP.

As the resulting state-space representation is not fully controllable for all state values, some adaptations and limitations have been performed to obtain a fully controllable system. On one hand, it has been considered that the MP will not be used with linear velocities around zero. Additionally, considering that values of the orientation difference between the vehicle and the curvature of the road ($e_\theta(t)$) are expected to be low, its sinus could be approximated by the angle $e_\theta(t)$ itself, while the cosine could be approximated by 1. In order to obtain more non-zero terms to get a fully controllable system and make a more accurate approximation of the sine and cosine of $e_\theta(t)$, the approximation here proposed is the one presented in Eqs. (19) and (20) which consists in dividing the sine or cosine in two terms which are the half of the sine or the cosine, then approximating just one of the two terms by the angle itself for the sine and 1 for the cosine. This simplification achieves the addition of new terms ρ_{ij} to matrices $A(\rho)$ solving controllability problems while introducing lower errors as other alternatives.

$$sin(e_\theta(t)) \simeq \frac{sin(e_\theta(t))}{2} + \frac{e_\theta(t)}{2} \tag{19}$$

$$cos\left(e_\theta(t)\right) \simeq \frac{cos(e_\theta(t))}{2} + \frac{1}{2} \tag{20}$$

Applying both expressions and discretizing via the Forward Euler Method, the LPV state-space representation obtained is the one described in (10) with the LPV matrices (21) and (22).

$$B(\rho) = \begin{bmatrix} T_s & -C_f \cdot \frac{sin(\delta(k))}{m} \cdot T_s \\ 0 & -C_f \cdot \frac{cos(\delta(k))}{m} \cdot T_s \\ 0 & C_f \cdot \frac{l_f \cdot cos(\delta(k))}{I} \cdot T_s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

$$A(\rho) = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} & 0 & 0 & 0 \\ \rho_{21} & \rho_{22} & \rho_{23} & 0 & 0 & 0 \\ 0 & \rho_{32} & \rho_{33} & 0 & 0 & 0 \\ \rho_{41} & \rho_{42} & 0 & 1 & \rho_{45} & 0 \\ \rho_{51} & \rho_{52} & T_s & 0 & \rho_{55} & 0 \\ \rho_{61} & \rho_{62} & 0 & 0 & \rho_{65} & 1 \end{bmatrix} \tag{22}$$

where the different time-varying terms ρ_{ij} are defined as:

$$\rho_{12} = C_f \cdot \frac{sin(\delta(k))}{v_x(k)m} \cdot T_s \qquad \rho_{13} = v_y(k) \cdot T_s + C_f \cdot \frac{l_f \cdot sin(\delta(k))}{v_x(k) \cdot m} \cdot T_s$$

$$\rho_{21} = -\omega(k) \cdot T_s \qquad \rho_{22} = -\frac{C_r + C_f \cdot cos(\delta(k))}{v_x(k) \cdot m} \cdot T_s + 1$$

$$\rho_{23} = \frac{C_r \cdot l_r - C_f \cdot l_f \cdot cos(\delta(k))}{v_x(k) \cdot m} \cdot T_s$$

$$\rho_{32} = -\frac{C_f \cdot l_f \cdot cos(\delta(k)) + C_r \cdot l_r}{I \cdot v_x(k)} \cdot T_s$$

$$\rho_{33} = -\frac{C_f \cdot {l_f}^2 \cdot cos(\delta(k)) + C_r \cdot {l_r}^2}{I \cdot v_x(k)} \cdot T_s + 1$$

$$\rho_{41} = \frac{sin(e_\theta)}{2} \cdot T_s \qquad \rho_{42} = cos(e_\theta(k)) \cdot T_s \qquad \rho_{45} = \frac{v_x(k)}{2} \cdot T_s$$

$$\rho_{51} = -\kappa \cdot \frac{cos(e_\theta(k))}{1 - e_L(k) \cdot \kappa} \cdot T_s \qquad \rho_{52} = \frac{1}{2} \cdot \frac{\kappa \cdot sin(e_\theta(k))}{1 - e_L(k) \cdot \kappa} \cdot T_s$$

$$\rho_{55} = \frac{1}{2} \cdot \frac{\kappa \cdot v_y(k)}{1 - e_L(k) \cdot \kappa} \cdot T_s + 1 \qquad \rho_{61} = \frac{cos(e_\theta(k))}{1 - e_L(k) \cdot \kappa} \cdot T_s$$

$$\rho_{62} = -\frac{1}{2} \cdot \frac{sin(e_\theta(k))}{1 - e_L(k) \cdot \kappa} \cdot T_s \qquad \rho_{65} = -\frac{1}{2} \cdot \frac{v_y(k)}{1 - e_L(k) \cdot \kappa} \cdot T_s$$

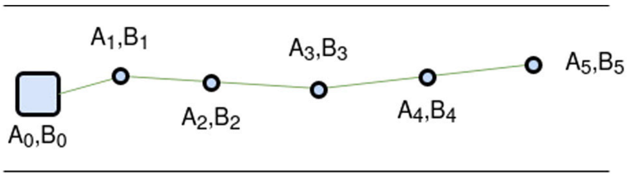


Fig. 1. Selection of the LPV matrices for the LPV-MP.

3.1.1. Selection of the LPV matrices

A critical aspect of the methodology proposed is the selection of the LPV matrices to obtain an accurate representation of the vehicle. In this work, it has been assumed that the surroundings will change progressively and no big changes that could happen suddenly are expected. By this way, the proposed path computed at each execution is expected to be similar to the path designed in the previous iteration adding a new further step. Consequently, the LPV matrices are computed based on the estimated location of the vehicle at each time instant extracted from the previous designed path. This procedure can be graphically seen in Fig. 1: The vehicle is represented as a big square while the green curve represents the future locations of the vehicle in case it follows the last path designed. The LPV matrices are computed according to the estimated values of the states at each time instant.

As it can be expected, this assumption may induce some errors in the model. Therefore, the MP should be combined with a closed-loop MC to minimize the error when tracking the path. Additionally, the MP has to be executed frequently in order to reduce the difference between expectations and reality, reacting rapidly to unexpected changes in the surroundings. In the same way, the time length of the path ($H_p{\cdot}T_s$) does not have to be excessively large as in typical off-line MP, because the uncertainty of the model is surely higher than the supposed one, while the dynamic surroundings do not ensure that the designed path would be optimal or even feasible in future iterations. Including uncertainties inside the model would also be interesting to guarantee robustness.

3.2. Reachability analysis

As expressed in Althoff et al. (2021), performing a reachability analysis is very useful for many aspects such as guaranteeing robustness of model predictive control certifying the systems do not reach unsafely regions. Therefore, at this approach a reachability analysis is performed to define the safety regions where the motion plan can be designed accomplishing safety conditions.

As a first step, the set of all applicable inputs $\mathcal{U}_k \subseteq [\underline{u}, \overline{u}]$ for each time instant along the prediction horizon, that is based on the last input applied, the slew rate and the upper and lower boundaries based on the dynamic limitations of the vehicle, is computed.

For analyzing the admissible values every state could take at each time instant, the current and expected locations of the obstacles nearby are analyzed and a safety region around them is defined, obtaining $\mathcal{F}_k$, the set of forbidden state values for each time instant along the prediction horizon as expressed in Eq. (8). Additionally, the states may also remain inside the set $\mathcal{S}_{x,k}$ which contains the generic boundaries of the states according to the dynamic surroundings, such as the maximum speed allowed or the maximum orientation or distance to the center of the road allowed taking into account the lane width and the curvature of the road. In case of static obstacles, the computation of the forbidden regions has to be analyzed just when the obstacle is detected and correctly localized. In that case, taking into account the obstacles is quite simple, for example tightening the width of the lane as has been done in Alcalá, Puig, and Quevedo (2020). However, when the agents in the environment are in motion, the complexity of the problem increases, as explained so far. On the one hand, it is necessary to receive information, estimate or predict the position and future maneuvers of the obstacles in the environment. In this example, the considered obstacles are other intelligent vehicles that communicate their position and future maneuvers. With these predictions, in each iteration the safe regions have to be calculated for each time instant as well as a new reachability analysis has to be performed, because the other agents may have varied their intention from one iteration to another. This dynamic reality of the environment forces the sampling period and the time between trajectory computations to be reduced, making it necessary to look for alternatives to reduce the computational cost of the problem performed in the proposed methodology.

A reachability analysis of the states is then performed to define the corridors through which the motion planner will look for the optimal trajectory. Firstly, the current state is read and represented as a set allowing the inclusion of uncertainties in measurements. In an iterative manner, this set gets propagated through time using the previously computed sets of applicable inputs $\mathcal{S}_{u,k}$ and the LPV matrices calculated in advance as expressed in Eq. (23).

$$\mathcal{X}_{k+1} = A(\rho)_k \cdot \mathcal{X}_k + B(\rho)_k \cdot \mathcal{U}_k \tag{23}$$

At each propagation step the resulting set is bounded with the states generic boundaries to exclude reachable states that exceed them. Also, states which imply entering the forbidden area around other obstacles are excluded from the set of reachable safety states $\mathcal{R}_k(\mathcal{X}_0)$. Therefore, the computation of the reachable set at a time instant k can be expressed as Eq. (24).

$$\mathcal{R}_k(\mathcal{X}_0) := \mathcal{X}_k \cap \mathcal{S}_{x,k} \cap \overline{\mathcal{F}}_k \tag{24}$$

In order to be able to propagate states in an analogous way to the evolution of dynamical systems with the state space representation, not all type of set representations are valid if a reduced computational cost is desired. As discussed in Zheng et al. (2022), representing is an excellent choice for being more accurate as other commonly used set representations such as ellipsoids, and are more computational efficient as other similar sets like polytopes, especially performing the Minkowski sum which is used for propagating the states at (23). As an aspect to be highlighted, the intersection of two zonotopes does not result in a zonotope. In the above-mentioned article, the authors propose to make an external approximation for the intersection of zonotopes that could be used as for example the one proposed in Althoff (2015b). However, performing an over-approach of the intersection would compromise the security intended to be achieved by exclusion of unsafe states at (24). This is the reason to make use at this approach of the constrained zonotopes presented in Scott et al. (2016), since constrained zonotopes maintain the simplicity in the mathematical operations for the propagation of sets, while being able to express the exact set resulting from the intersection of zonotopes as explained at Raghuraman and Koeln (2022). Additionally, in that same work, different methods are presented to eliminate redundancies in the generators, thus simplifying the expressions. Different internal and external approaches are also proposed, which will be considered in the implementation of the approach here presented. Last but not least, the article concludes with a practical example of a reachability analysis, reinforcing the idea of making use of constrained zonotopes for the approach proposed in this work.

With all this, the methodology proposed here is defined, with which it is possible to merge the constraints of both boundaries and collision avoidance into linear expressions. In Algorithm 1, a pseudocode with the different steps that have to be followed to compute the reachable set ($\mathcal{R}(\mathcal{X}_0)$) from the current states with the uncertainty of the measurements ($\mathcal{X}_0$), the LPV matrices ($A(\rho)$, $B(\rho)$) computed for each step k and the location of the obstacles nearby ($O_i(k)$) at every time-instant along the prediction horizon, is presented. Once described with Algorithm 1 the theoretical basis of performing a reachability analysis excluding forbidden states, in the implementation section the reader can consult two more specific algorithms showing how to implement both the reachability analysis, Algorithm 2, and the motion planner itself, Algorithm 3. For more detailed information, see the next sections.

Algorithm 1: Steps to perform a Reachability Analysis

```
Input Data: $\mathcal{X}_0, \mathcal{U}_{last}, \mathbf{A}, \mathbf{B}, O_i$
Output Data: $\mathcal{R}(\mathcal{X}_0)$

/* Initialization of both sets with last inputs
   applied and current states with
   uncertainties                                  */
$\mathcal{U}_{-1} \leftarrow \mathcal{U}_{last}$;
$\mathcal{R}_0(\mathcal{X}_0) \leftarrow \mathcal{X}_0$;

/* Computation of Unsafe Sets based on obstacles
   location at every time instant along the
   prediction horizon                             */
$k \leftarrow 1$;
while $k \leq H_p$ do
    $\mathcal{F}_k \leftarrow ComputationUnsafeRegions(O_{i,k})$;
    $k \leftarrow k + 1$;
end

/* Sets propagation excluding forbidden regions
   intersecting the reachable states, unsafe
   regions and constraints along the prediction
   horizon                                        */
$k \leftarrow 0$;
while $k < H_p$ do
    $\mathcal{U}_k \leftarrow (\mathcal{U}_{k-1} + \Delta\mathcal{U}) \cap \mathcal{S}_U$;
    $\mathcal{X}_{k+1} \leftarrow A(\rho)_k \cdot \mathcal{R}_k(\mathcal{X}_0) + B(\rho)_k \cdot \mathcal{U}_k$;
    $\mathcal{R}_{k+1}(\mathcal{X}_0) \leftarrow \mathcal{X}_{k+1} \cap \mathcal{S}_{x,k+1} \cap \overline{\mathcal{F}}_{k+1}$;
    $k \leftarrow k + 1$;
end
```

In Figs. 2–5, two examples of reachability analysis are shown. On the left side, a path is designed (green curve) after performing a

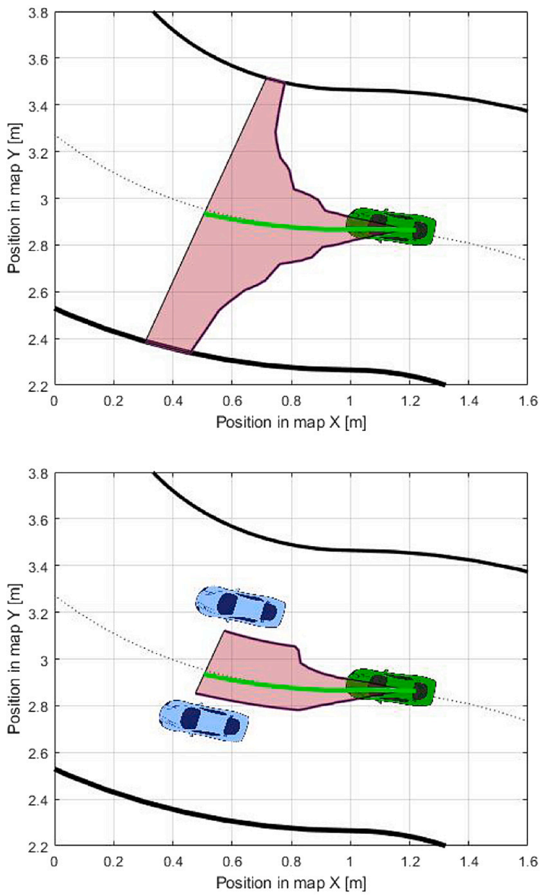


Fig. 2. Path planned for same scenario without and with obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

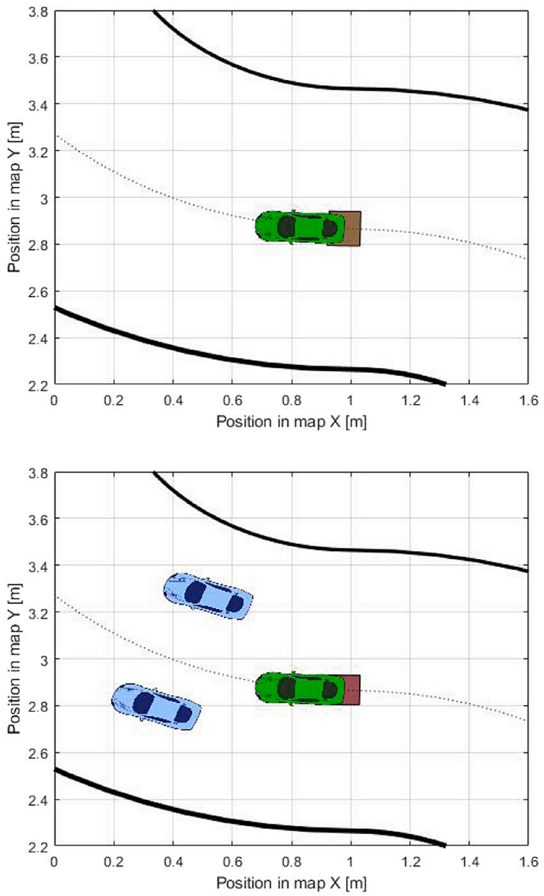


Fig. 3. Hull of the reachable set at t = 0.25 s without and with obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reachability analysis in a scenario where no obstacle is located nearby. On the contrary, the right images show the same scenario where the vehicle under study (green) has to drive between two obstacles (blue vehicles) which are driving at a constant velocity which is the half of the maximal speed the green vehicle can reach.

In Fig. 2, the path designed for both scenarios that consists in the motion plan for the 6 states through time projected over the Cartesian map, can be seen. This path is represented as a green curve over the road, while the shaded region represents for each point of the designed path what other distance from the center of the road the vehicle could have reached while complying with the safety criteria. The aim of this representation is to show that the motion planner adapts appropriately to the presence of obstacles by narrowing the feasible region, while in situations where there are no obstacles, the feasible region of the vehicle under study ends up converging with the entire lane.

To conclude, it is important to remark that the reachability analysis is performed along all the dynamical states of the vehicle: longitudinal and lateral velocity, angular velocity, distance to the center of the road, difference in the orientation of the vehicle with respect to the curvature of the road and distance traveled. Therefore, it is worth noting that the reachability analysis performed provides more information by refining all the states to ensure that safety requirements are met by complying with road traffic regulations. In fact, the results of the analysis is not the shadowed region presented in Fig. 2 but a set associated to every time instant studied along the prediction horizon with as many dimensions as states. The rectangular hulls of states $(e_L(t), s(t))$ projected over the road map are represented in Figs. 3–5. Each figure corresponds to three different time instants after Fig. 2 ($t = 0.25$ s, $t = 0.40$ s and $t = 0.50$ s, respectively).

4. Implementation

The methodology here presented has been verified implementing it in MATLAB 2021a and evaluated through simulations in different scenarios. For the implementation, the toolbox for rapid prototyping optimization problems YALMIP, Lofberg (2004), and a specific toolbox called CORA, useful for defining and dealing with set operations, have been used. More details about the toolbox CORA can be consulted in Althoff (2015a). The solvers chosen for the non-linear motion planner have been “fmincon” included in the MATLAB version cited above, while the motion planner with LPV matrices have been solved with “Gurobi” (Gurobi Optimization, 2021).

The simulations have been performed in different closed maps under different curvatures and locations of the obstacles, being the numerical results presented posteriorly those of the simulations performed in the map presented in Fig. 6, more details are given below. The vehicle has been defined with the model presented in the previous section using the parameters listed in Table 1, corresponding to a 1:10 scaled down RC car used in other works such as in Alcalá, Puig, Quevedo, and Rosolia (2020). It is important to remark that this work introduces a new methodology to design the path as well as the dynamic motion plan of a vehicle with dynamic surroundings considering safety. The accuracy of the trajectory followed, the robustness and the real performance quality will also vary depending on the motion controller or the path tracker used in combination with the motion planner here presented. For simplicity and to prove the wellness of the motion plan proposed in this work, a low-level controller has been implemented applying the first input proposed by the motion planner with a zero-order

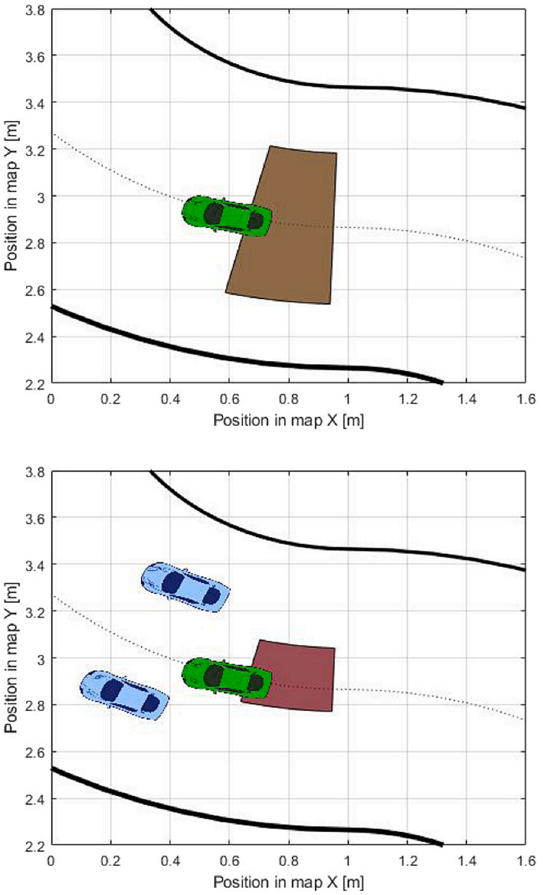


Fig. 4. Hull of the reachable set at t = 0.40 s without and with obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

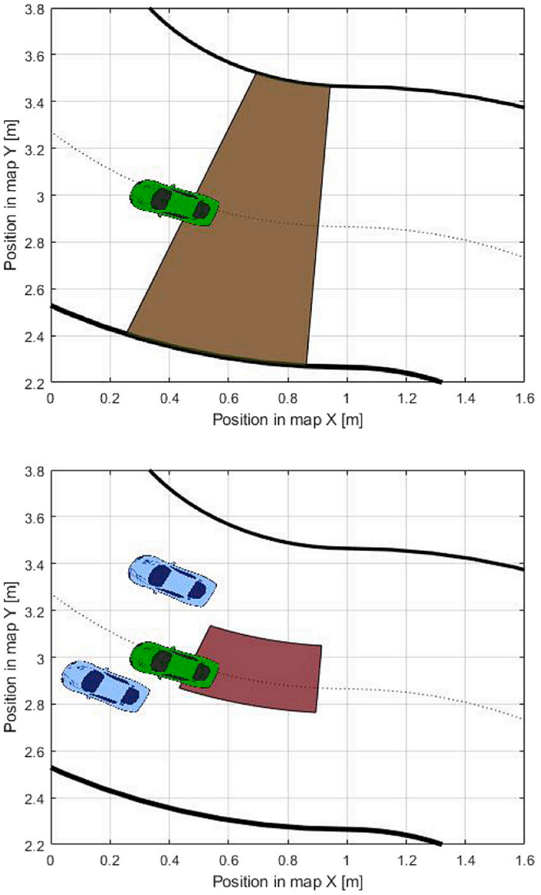


Fig. 5. Hull of the reachable set at t = 0.50 s without and with obstacles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Parameters values of the car-like robot.

Parameters	Value	Parameters	Value
l_r	0,125 m	l_f	0,125 m
C_r	65 N/rad	C_f	65 N/rad
μ	0,05	I	0,03 kg/m^2
m	1,98 kg	$\overline{v_x}$	1 m/s
$\underline{\delta}$	−0,36 rad	$\overline{\delta}$	0,36 rad
$\underline{a}$	−2,65 m/s^2	$\overline{a}$	1 m/s^2
$\underline{\Delta\delta}$	−2 rad/s	$\overline{\Delta\delta}$	2 rad/s
$\underline{\Delta a}$	−7,35 m/s^3	$\overline{\Delta a}$	7,35 m/s^3

hold, while the vehicle has been simulated with its ODEs. The motion planner here presented provides references to different dynamic states in addition to a trajectory on the road. Therefore, a great advantage of this motion planner is that it can be used in combination with multiple path trackers or trajectory controllers, from a basic controller to more complex techniques such as self-iterative learning methods, or MPC controllers.

In order to verify the performance of the methodology proposed here under different circumstances, a single complex scenario combining different challenging situations has been designed. With regard to the profile of the road, it counts with sections where the road is straight, sections with positive curvature, and sections with negative curvature, all of them with greater and lesser length. Similarly, two critical points are also considered where the change in curvature from positive to negative is of considerable magnitude.

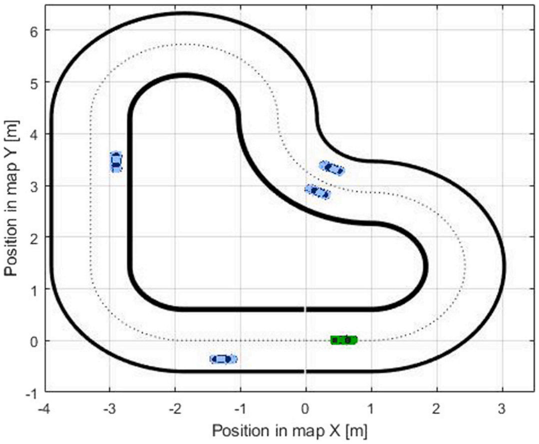


Fig. 6. Initial state of the vehicle (green car) and the obstacles (blue cars) over the map. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Additionally, different vehicles driving along the road, considered as dynamic obstacles, have been added to verify the capability of the MP to avoid them.

The motion planner here presented can deal with dynamic obstacles located at both sides of it by obtaining its motion plans in case they were other interconnected intelligent transportation agents, or by estimating and predicting their current locations and future movements thanks to the sensors and perception modules. In these simulations, it has been considered that all the agents around the vehicle are other intelligent vehicles sharing their current location and their future maneuvers planned with a certain accuracy. The safety region around the vehicles depends on the accuracy of the information offered by the agents nearby. For simplicity and to obtain a visually understandable evolution of the obstacles, the vehicles added are four vehicles driving at constant linear velocity (blue cars in Fig. 6) adapting their angular velocity according to the curvature of the road. They have been placed sufficiently far apart so that the vehicle under study encounters them at distinct times.

Firstly, two vehicles driving parallel to each other are added. This implies that the vehicle will have to drive between them in a narrow corridor, maximizing safety margins. Secondly, there will be a vehicle on the left-hand side of the road in a position where it is expected that the motion planner would propose a path over it, so it will have to change its trajectory to overtake it. Finally, the vehicle faces a final situation by a vehicle traveling slowly on the right-hand side of the road, which will not interrupt his intended path but will force the vehicle to modify its path to increase safety. Nevertheless, in future simulations as shown in Fig. 11 and Fig. 12, it can be observed how the motion planner is able to manage a dynamic environment with agents moving with different velocities and steering angles, as well as accelerations.

If a more realistic scenario were to be realized, different uncertainty magnitudes would have to be considered for the different vehicles in the surroundings. For this purpose, it would be sufficient to establish larger safety perimeters around them, tightening the safe region through which the vehicle can drive safely.

If it is desired to go a step further and work with non-intelligent vehicles that do not share their position, or the accuracy of the information provided is excessively low, it will not only be necessary to increase this safety margin, but also to perform a study of the possible maneuvers and regions that these vehicles could occupy. For this purpose, Lefkopoulos et al. (2021) and Zhou et al. (2022, 2023) suggested the use of a technique called Interaction-Aware methods whose objectives are to estimate the most probable positions or maneuvers to be performed by the surrounding vehicles as well as the probability of these maneuvers. Since these are probabilities, a trade-off between safety and performance have to be found to establish the occupancy probability threshold above which a region will be considered unsafe. To simplify this problem, there is a tendency to use roads with lanes, or to recreate virtual lanes on free-lane roads to delimit the possible maneuvers that other vehicles could perform, but this implies a regression in the optimality of the trajectory proposed by the motion planner, by substantially limiting the region through which the vehicle could drive safely, something that will not happen if the surrounding vehicles are also intelligent vehicles and share information with a certain degree of precision.

4.1. Approaches for order reduction

Analyzing the operations of constrained zonotopes, it is remarkable the simplicity of computing the reachability analysis in a very systematic manner. The use of a transformation and an addition for the propagation of sets as in Eq. (23) is required, which can be computed as in Eqs. (25) and (26), while the tightening of the set according to the desired states limitation and obstacle avoidance protocol as in Eq. (24) requires the use of sets intersections which can be computed as in Eq. (27).

$$R \cdot Z_i := \{R \cdot c_i, R \cdot H_i, A_{\alpha,i}, b_{\alpha,i}\} \tag{25}$$

$$Z_i + Z_j = \left\{ c_i + c_j, \quad [H_1|H_2], \quad \begin{bmatrix} A_{\alpha,i} & 0 \\ 0 & A_{\alpha,j} \end{bmatrix}, \quad \begin{bmatrix} b_{\alpha,i} \\ b_{\alpha,j} \end{bmatrix} \right\} \tag{26}$$

As can be observed, the larger the prediction horizon is, the larger the number of generators is. The same happens with the number of constraints used to define each reachable set.

$$Z_i \cap Z_j = \left\{ c_i, \quad [H_i|\mathbf{0}], \quad \begin{bmatrix} A_{\alpha,i} & 0 \\ 0 & A_{\alpha,j} \\ H_i & -H_j \end{bmatrix}, \quad \begin{bmatrix} b_{\alpha,i} \\ b_{\alpha,j} \\ c_j - c_i \end{bmatrix} \right\} \tag{27}$$

By this reason, it is recommendable to apply some simplifications and approaches to obtain sets simple enough for the motion planner to operate with.

4.1.1. Simplification of input sets

The different inputs are linear independent, for this reason, the set representing the applicable inputs can be studied as an interval which grows according to Eq. (28) until converging to $\mathcal{S}_U$ after some steps.

$$\mathcal{U}_k = (\mathcal{U}_{k-1} + \Delta\mathcal{U}) \cap \mathcal{S}_U \tag{28}$$

An interval can be represented as a zonotope (without constraints) in a very simple way by making use of a matrix of diagonal generators as in definition (29).

$$\mathcal{U}_k := \left\{ \begin{bmatrix} \frac{\overline{a}(k)+\underline{a}(k)}{2} \\ \frac{\overline{\delta}(k)+\underline{\delta}(k)}{2} \end{bmatrix}, \quad \begin{bmatrix} \frac{\overline{a}(k)-\underline{a}(k)}{2} & 0 \\ 0 & \frac{\overline{\delta}(k)-\underline{\delta}(k)}{2} \end{bmatrix} \right\} \tag{29}$$

Thus, performing the states propagation just add two new generators and does not imply any changes in the constraints.

4.1.2. Combining states limitations and collision avoidance

Another computation for reducing the complexity of the sets is by analyzing if the set of reachable states ($\mathcal{X}_k$) is a subset of the set with the states boundaries ($\mathcal{S}_{x,k}$). If it is, no intersection is needed. The same happens with the forbidden states, it is possible to study if the reachable states intersect with the forbidden states, otherwise, any tightening is needed.

In case the intersections were needed, another manner of reducing the complexity of the sets is performing a combination of the exclusion of unsafe and undesirable states with the avoidance of forbidden regions. By this way, the calculus of Eq. (24) can be performed through an intersection of the reachable states $\mathcal{X}_{k+1}$ and a previously computed zonotope $S_{k+1,const}$ (or constrained zonotope) which combines the boundaries desired for each state and the collision avoidance criteria, allowing the computation of an inner-approximation over it to reduce its complexity. This inner-approximation can be obtained before performing the intersection with the reachable sets by applying the techniques presented in Raghuraman and Koeln (2022). This procedure might imply obtaining a more conservative motion plan but may imply a significant reduction of the complexity of the optimization problem. During the implementation here presented, the combination of both sets has been computed modifying the set of the bounds of the states tightening the maximal allowed lateral distance e_L to avoid collisions with the obstacles detected nearby.

The idea of performing both simplifications is to bound and reduce the number of dimensions of the constraints and generator matrices being translated in a reduction of the complexity of the problem. Additionally, given the possibility of null generators, parallel generators or overly similar generators, it is convenient to apply order reductions as the one presented in Kopetzki et al. (2017) based on PCA or the methods presented in Althoff (2010). Those methods are included in CORA, being some of them and many others discussed and compared in Scott et al. (2016).

However, it is important to not forget the content of the set to which a reduction is being applied and what type of reduction it is. An inner-approximation of X_k excludes a feasible evolution of the system while just as over-approximations of $S_{k,const}$ could introduce unsafe states in the safety reachable set. On the contrary, performing an over-approach of X_k and an inner-approach of $S_{k,const}$ before computing the intersection of both to obtain the reachable set of safe states (24), does not compromise safety aspects and reduce the complexity of the optimization problem of the motion planner, but the computational cost of the simplification is higher. Additionally, it is also observable that the computation of the exact intersection between two constrained zonotopes as in Eq. (27) may lead to redundant constraints which can be deleted following a constraint reduction as presented in Raghuraman and Koeln (2022) which retains the exact original set. Therefore, the implementation of Algorithm 1 can be adapted to these simplifications as schematically summarized in Algorithm 2.

Algorithm 2: Implementation of the Reachability Analysis

```
Input Data: $\mathcal{X}_0, \mathcal{U}_{last}, A(\rho), B(\rho), O_i$
Output Data: $\mathcal{R}(\mathcal{X}_0)$
/* Initialization of the sets based on last
   input applied, current states and their
   uncertainties                                    */
$\mathcal{U}_{-1} \leftarrow \mathcal{U}_{last}$;
$\mathcal{R}_0(\mathcal{X}_0) \leftarrow \mathcal{X}_0$;
//
/* Compute Safety Regions based on Obstacles
   Location at every time instant along $H_p$       */
$k \leftarrow 1$;
while $k \leq H_p$ do
    $\mathcal{F}_k \leftarrow ComputationUnsafeRegions(O_{i,k})$;
    if $\mathcal{S}_{x,k} \cap \mathcal{F}_k$ then
        $\mathcal{S}_{k,const} \leftarrow OrderReduction(\mathcal{S}_{x,k} \cap \overline{\mathcal{F}_k})$;
    else
        $\mathcal{S}_{k,const} \leftarrow OrderReduction(\mathcal{S}_{x,k})$;
    end
    $k \leftarrow k + 1$;
end
/* Sets propagation excluding forbidden regions
   intersecting the reachable states, unsafe
   regions and constraints along the prediction
   horizon                                          */
$k \leftarrow 0$;
while $k < H_p$ do
    $\mathcal{U}_k \leftarrow ComputeInterval(\mathcal{U}_{k-1}, \Delta\mathcal{U}, \mathcal{S}_U)$;
    $\hat{\mathcal{X}}_{k+1} \leftarrow A_k(\rho) \cdot \mathcal{R}_k(\mathcal{X}_0) + B_k(\rho) \cdot \mathcal{U}_k$;
    $\mathcal{X}_{k+1} \leftarrow OrderReduction(\hat{\mathcal{X}}_{k+1})$;
    if $\mathcal{X}_{k+1} \subseteq \mathcal{S}_{k,const}$ then
        $\hat{\mathcal{R}}_{k+1}(\mathcal{X}_0) \leftarrow \mathcal{X}_{k+1}$;
    else
        $\hat{\mathcal{R}}_{k+1}(\mathcal{X}_0) \leftarrow \mathcal{X}_{k+1} \cap \mathcal{S}_{k+1,const}$;
    end
    $\mathcal{R}_{k+1}(\mathcal{X}_0) \leftarrow ConstraintsReduction(\hat{\mathcal{R}}_{k+1}(\mathcal{X}_0))$;
    $k \leftarrow k + 1$;
end
```

4.1.3. Numerical results

Finally, the resulting loop of the LPV motion planner with a reachability analysis avoiding forbidden obstacles can be summarized in the four steps described in Algorithm 3. The first step consists in obtaining information from the vehicle and the obstacles nearby. The second deals with computing the LPV matrices according to the previous path designed and performing the reachability analysis using these LPV matrices. Then, a new motion plan based in the computed LPV matrices and reachable sets is computed, and finally the variables are updated to execute it repetitively.

For the presentation of numerical results, two studies have been performed. In a first study, the results of a traditional non-linear motion planner and the same motion planner with LPV matrices have been compared. Those results can be consulted in Table 2.

Algorithm 3: Iterative Execution of the Motion Planner

```
while $TRUE$ do
    /* Read sensors and estimate vehicle location
       as well as obstacles location and possible
       maneuvers                                        */
    $\mathcal{X}_0, O_i \leftarrow SensorMeasurements()$;
    $O_{i,1-H_p} \leftarrow EstimateFutureLocation()$;

    /* Compute LPV Matrices and Reachable Sets          */
    $A(\rho), B(\rho) \leftarrow ComputeLPVMatrices(\mathcal{X}_0, last\_plan)$;
    $\mathcal{R}(\mathcal{X}_0), \mathcal{U} \leftarrow$
      $ReachabilityAnalysis(\mathcal{X}_0, U_{last}, A(\rho), B(\rho), O_{i,1-H_p})$;

    /* Execute Motion Planner                           */
    $new\_motion\_plan \leftarrow MotionPlanner(\mathcal{R}(\mathcal{X}_0), \mathcal{U}, A(\rho), \boldsymbol{B}(\rho))$;

    /* Update variables for next iteration              */
    $u_{last} \leftarrow u_{applied}$;
    $last\_plan \leftarrow new\_motion\_plan$;
end
```

Table 2
Results of NL-MP and LPV-MP - computation during 100 executions.

MP	H_p	T_s [s]	$H_p \cdot T_s$ [s]	T_{com} [s]
NL-MP	5	0,05	0,25	0,2155
NL-MP	7	0,05	0,35	0,2328
NL-MP	10	0,05	0,5	0,7832
NL-MP	15	0,05	0,75	2,2212
NL-MP	20	0,05	1	4,3297
NL-MP	30	0,05	1,5	8,6210
LPV-MP	5	0,05	0,25	0,0065
LPV-MP	7	0,05	0,35	0,0081
LPV-MP	10	0,05	0,5	0,0098
LPV-MP	15	0,05	0,75	0,0140
LPV-MP	20	0,05	1	0,0180
LPV-MP	30	0,05	1,5	0,0324

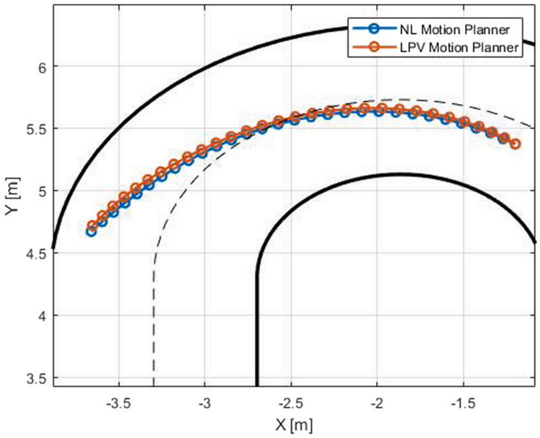


Fig. 7. Two paths designed by the NL-MP (blue) and the LPV-MP (red) with a T_s of 0,05 s and a H_p of 30 steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Analyzing the results of Table 2 it can be observed that the computational cost with a non-linear motion planner (NL-MP) without any safety verification has a significant increase when the prediction horizon gets increased. Additionally, it can be observed that when using a small discretization period to avoid uncertainties induced by

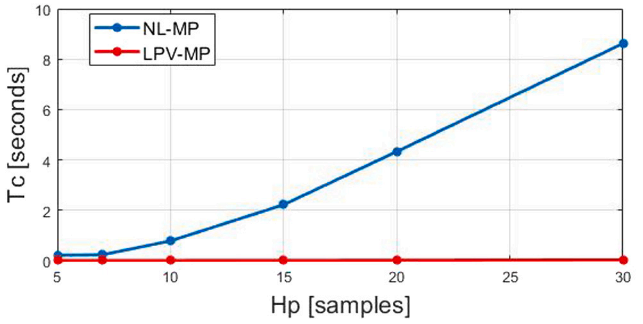


Fig. 8. Comparison of the mean computational cost of each iteration with a NL-MP (blue) and an LPV-MP (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

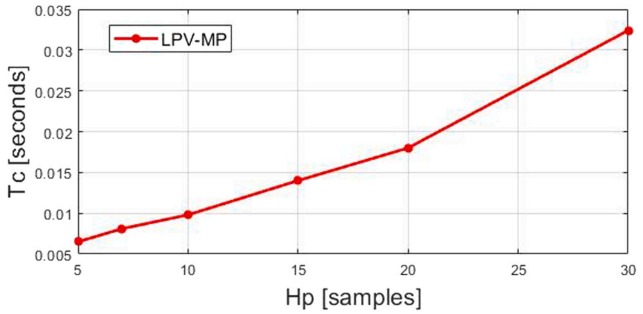


Fig. 9. Zoom of the mean computational cost of each iteration of the LPV-MP (red).

larger discrete periods, the duration of the motion plan designed is even smaller than the time needed to compute it. On the contrary, the computational cost of the same motion plan in the same circumstances but using LPV matrices gets reduced two orders of magnitudes, demonstrating the significant benefits of embedding the non-linearities of the dynamic model into an LPV representation of the system.

This substantial difference between the non-linear method and the novel one proposed in this work can be seen graphically in Fig. 8. Due to the significant difference of two orders of magnitude between both curves, an additional graph has been added in Fig. 9 to make observable the trend of the LPV-MP when the Hp is increased. Additionally, in Fig. 7, two motion plans designed under the same conditions using the non-linear motion plan (blue) and an LPV motion plan (red), are shown. As it can be observed, the trajectory proposed for a scaled-model vehicle coming from the right side to the left direction in a curvy-scenario is almost the same, having just a slight accumulative error noticeable at the last steps of the trajectory. The path designed by the NL-MP using a T_s of 0,05 s and a H_p of 30 samples. While the resulting path of the NL-MP has a length of 2,8471 m, the root mean squared error of the trajectory is just 0,0322 m. For this comparison, the distance between each (x,y) coordinate proposed for each time instant along the 30 samples, has been measured.

With these two comparisons it can be concluded that the LPV-approach allows the motion planner to obtain a valid motion plan with a much lower computational time without introducing a significant error in the trajectory.

Secondly, a study comparing the same non-linear MP with the LPV MP adding the computation of the reachability analysis to increase safety, has been performed. The results can be seen in Table 3. In the results shown in Table 3, it is observable that performing a reachability analysis through propagation and intersection of constrained zonotopes can result in complex expressions with numerous associated independent variables. This complexity might make it impossible to model with YALMIP as happened for prediction horizons longer than 8 samples in the previously described scenario. The complexity lies in the expressions used to define the problem constraints associated with the reachability analysis, not in the optimization problem itself. Furthermore, some of these constraints may be redundant or irrelevant to the problem definition. Alternatives for simplifying these expressions to obtain a tractable problem are discussed in later sections. Meanwhile, analyzing the remaining obtained results, it can also be observed that for shorter prediction horizons such as five samples, the computational cost is still lower than that of the NL-MP, so that the combination with LPV matrices ensures that safety criteria can be applied through a reachability analysis and still have a lower computational cost than the classical solution consisting of a NL-MP. Moreover, we can afford in the MP to make use of longer discretization times obtaining motion plans with a similar duration to those obtained with the NL-MP, but with a slightly lower computational cost, as can be seen in the last simulation where a motion plan is designed for 0.8 s, making use of a sampling time of 0.2 s with a computational cost of 0.1710 s. Comparable to the motion plan designed with the NL-MP with a duration of 0.75 s whose computational cost was 2.2212 s; almost thirteen times higher. However, it would be desirable to be able to make use of longer prediction horizons and to be able to design longer trajectories, so there is a need to explore alternatives that reduce the complexity of the problem while still contributing to increase safety.

Table 3
Results of NL-MP and R-LPV-MP - computation during 100 executions.

MP	H_p	T_s [s]	$H_p \cdot T_s$ [s]	T_{com} [s]
NL-MP	5	0,05	0,25	0,2155
NL-MP	7	0,05	0,35	0,2328
NL-MP	10	0,05	0,5	0,7832
NL-MP	15	0,05	0,75	2,2212
Z-R-MP	3	0,05	0,15	0,0193
Z-R-MP	4	0,05	0,20	0,0583
Z-R-MP	5	0,05	0,25	0,1632
Z-R-MP	6	0,05	0,30	0,4186
Z-R-MP	7	0,05	0,35	0,9116
Z-R-MP	8	0,05	0,40	Intractable
Z-R-MP	4	0,1	0,4	0,0592
Z-R-MP	5	0,2	0,8	0,1710

4.2. Approximating $\mathcal{R}(\mathcal{X}_0)$ with hulls

In order to decrease the computational cost of reachability analysis performed in combination with the LPV-MP, a trade-off between computational cost, efficiency of the trajectory and safety can be studied by simplifying the resulting reachable analysis before transmitting it to the optimization problem. That is, compute the reachability analysis as described in the previous section, but instead of defining the boundaries of the optimization problem based on the mathematical formulation of the constrained zonotope, define hulls for each constrained zonotope of the reachability analysis in which the set is contained. As this is an over-approach of the sets, unsafe solutions may be induced, but computing the hulls after having performed the complete reachability analysis is compatible with applying an extra step of safety verification in which it is confirmed that the solution obtained from the motion planner remains inside $\mathcal{R}(\mathcal{X}_0)$. The main advantage of this simplification is the avoidance of the excessive increase in the number of variables to define the constrained zonotopes that appears when using high prediction horizons, which on the other hand allow designing better optimized motion plans for looking further ahead. The following table shows the results having computed hulls based on computing for each k the smallest interval that contains the constrained zonotope of the safety reachable states associated to a particular time instant. The results are provided in Table 4.

As can be seen in Table 4, the computational cost of the reachability analysis with constrained zonotopes and its use inside the motion

Table 4
Results of different R-MP with zonotopic constraints and without hull-based constraints - computation during 500 executions of the MP.

MP	H_p	T_s [s]	$H_p \cdot T_s$ [s]	T_{com} [s]
Z-R-MP	2	0,1	0,2	0,0073
Z-R-MP	3	0,1	0,3	0,0197
Z-R-MP	4	0,1	0,4	0,0574
Z-R-MP	5	0,1	0,5	0,1625
Z-R-MP	6	0,1	0,6	0,4242
Hull-MP	2	0,1	0,2	0,0557
Hull-MP	3	0,1	0,3	0,0888
Hull-MP	4	0,1	0,4	0,1279
Hull-MP	5	0,1	0,5	0,1683
Hull-MP	6	0,1	0,6	0,2079

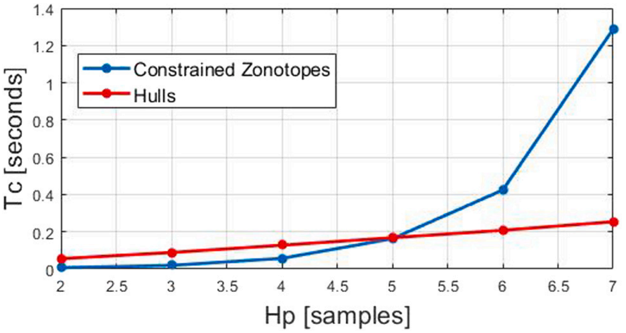


Fig. 10. Comparison of the mean computational cost of each execution of the whole process with (red) and without (blue) the hulls-approach. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

planner increases significantly when the prediction horizon increases, making difficult the implementation with large prediction horizons. However, if large paths are desired, larger discretization periods can be used, finding a trade-off between accuracy, computational time and path length. It is important to highlight that the complexity of the problem is independent of the discretization period, but the smaller T_s it is, the less abrupt the evolutions of the references designed in the motion plan are. Moreover, it is remarkable that with this methodology the trajectory that the vehicle should track is designed, having a controller at a lower level. Therefore, computation times close to the time with which the designed trajectory is sampled (T_s) are not an impediment for its implementation. The most relevant requirement is that the computation time is significantly lower than the duration of the designed trajectory so that it can be updated and continued multiple times before being concluded, adapting also to unexpected changes in the dynamic environment.

On the contrary, as it is observable in Fig. 10, the computational cost of the approach based in hulls increases linearly with a low slope in comparison with the solution based in constrained zonotopes. In this way, the usefulness of approximating sets by hulls is verified as the computational cost is significantly lower, especially for high prediction horizons where the first proposed method tends to very high values as it has already been commented.

4.2.1. Example of paths in other scenarios

Finally, it has been verified the possibility of using the designed MP over other scenarios and situations. An example of these is the simulation shown in Figs. 11 and 12, where a vehicle driving with an LPV-MP with reachability analysis can be observed. In this scenario, the vehicle was challenged to drive on a road section with a change of curvature, with a slow vehicle to be overtaken by another vehicle on the left side traveling at a higher velocity than the one allowed for the vehicle under study. To make matters more difficult, when the vehicle under study tries to overtake the vehicle on its right, the latter is about to be overtaken by the other vehicle on the left. As it can be consulted in both graphics, the vehicle adapts the path successfully to keep distance with them to ensure safety.

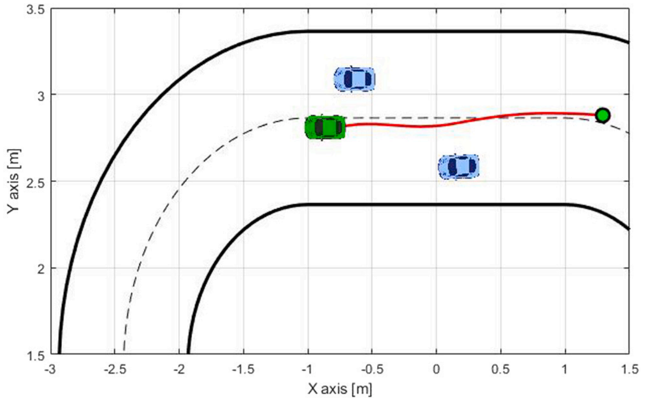


Fig. 11. Vehicle with a Z-R-MP avoiding obstacles (H_p 4 samples, T_s 0,15 s).

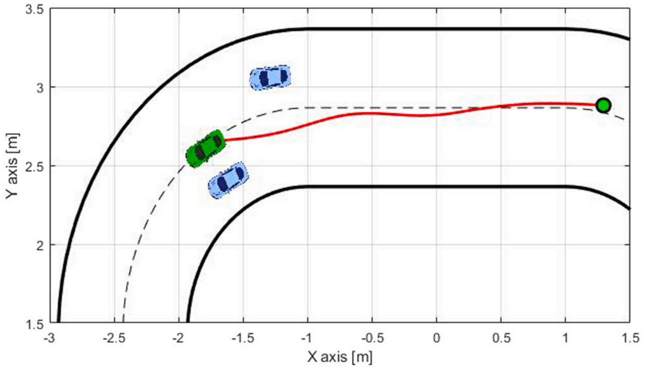


Fig. 12. Same simulation as Fig. 11 1,5 s later.

5. Conclusions and future work

Throughout the document, a methodology to reduce the computational cost of autonomous vehicle MP by considering the dynamics and an additional step to verify safety has been proposed. With this aim, the proposed approach consists in embedding the non-linearities of the vehicle dynamics in linear expressions by means of LPV matrices and including the performance of a reachability analysis which excludes forbidden states associated with obstacle avoidance using sets.

This reachability analysis presents different computational difficulties, being an interesting selection the use of constrained zonotopes, to have an accurate result with a low computational cost. However, it is evidenced that it is convenient to look for some set reduction technique to be able to design more complex motion plans. A first test has been made using hulls which shows that, indeed, the final computational cost could be significantly reduced.

Finally, as future work, a more exhaustive study of the most appropriate techniques and sets to reduce the computational cost by finding a balance between reducing the complexity and preserving the quality, could be considered.

CRediT authorship contribution statement

Álvaro Carrizosa-Rendón: Writing – review & editing. **Vicenç Puig:** Writing – original draft, Software, Investigation. **Fatiha Nejjari:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been co-financed by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SaCoAV (ref. MINECO PID2020-114244RB-I00) and by the DGR of Generalitat de Catalunya, Spain (SAC group ref. 2017/SGR/482).

References

Alcalá, E., Puig, V., & Quevedo, J. (2020). LPV-MP planning for autonomous racing vehicles considering obstacles. *Robotics and Autonomous Systems*, *124*, Article 103392. http://dx.doi.org/10.1016/j.robot.2019.103392.

Alcalá, E., Puig, V., Quevedo, J., & Rosolia, U. (2020). Autonomous racing using linear parameter varying-model predictive control (LPV-MPC). *Control Engineering Practice*, *95*, Article 104270.

Alcalá, E., Puig, V., Quevedo, J., & Sename, O. (2020). Fast zonotope-tube-based LPV-MPC for autonomous vehicles. *IET Control Theory & Applications*, *14*(20), 3676–3685.

Althoff, M. (2010). *Reachability analysis and its application to the safety assessment of autonomous cars* (Ph.D. thesis), Technische Universität München.

Althoff, M. (2015a). An introduction to CORA 2015. In *Proc. of the workshop on applied verification for continuous and hybrid systems* (pp. 120–151).

Althoff, M. (2015b). On computing the minkowski difference of zonotopes. arXiv preprint arXiv:1512.02794.

Althoff, M., Frehse, G., & Girard, A. (2021). Set propagation techniques for reachability analysis. *Annual Review of Control, Robotics, and Autonomous Systems*, *4*, 369–395.

Caporale, D., Fagiolini, A., Pallottino, L., Settimi, A., Biondo, A., Amerotti, F., Massa, F., De Caro, S., Corti, A., & Venturini, L. (2018). A planning and control system for self-driving racing vehicles. In *2018 IEEE 4th international forum on research and technology for society and industry* RTSI, (pp. 1–6). IEEE.

Danielson, C., Berntorp, K., Weiss, A., & Di Cairano, S. (2020). Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner. *IEEE Transactions on Automatic Control*, *65*(10), 4456–4463.

Gurobi Optimization, L. (2021). Gurobi optimizer reference manual.

Hegedüs, F., Bécsi, T., Aradi, S., & Gáspár, P. (2017). Model based trajectory planning for highly automated road vehicles. *IFAC-PapersOnLine*, *50*(1), 6958–6964.

Ioan, D., Prodan, I., Olaru, S., Stoican, F., & Niculescu, S.-I. (2021). Mixed-integer programming in motion planning. *Annual Reviews in Control*, *51*, 65–87. http://dx.doi.org/10.1016/j.arcontrol.2020.10.008.

Kopetzki, A.-K., Schürmann, B., & Althoff, M. (2017). Methods for order reduction of zonotopes. In *2017 IEEE 56th annual conference on decision and control* CDC, (pp. 5626–5633). IEEE.

Lefkopoulos, V., Menner, M., Domahidi, A., & Zeilinger, M. N. (2021). Interaction-aware motion prediction for autonomous driving: A multiple model Kalman filtering scheme. *IEEE Robotics and Automation Letters*, *6*(1), 80–87. http://dx.doi.org/10.1109/LRA.2020.3032079.

Liu, C., Lee, S., Varnhagen, S., & Tseng, H. E. (2017). Path planning for autonomous vehicles using model predictive control. In *2017 IEEE intelligent vehicles symposium* IV, (pp. 174–179). IEEE.

Liu, E. I., Würsching, G., Klischat, M., & Althoff, M. (2022). CommonRoad-reach: A toolbox for reachability analysis of automated vehicles. In *2022 IEEE 25th international conference on intelligent transportation systems* ITSC, (pp. 2313–2320). IEEE.

Lofberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *2004 IEEE international conference on robotics and automation (IEEE cat. no. 04CH37508)* (pp. 284–289). IEEE.

Manzinger, S. (2021). *Using reachability analysis for motion planning of autonomous vehicles in complex traffic situations* (Ph.D. thesis), Universität München.

Manzinger, S., Pek, C., & Althoff, M. (2021). Using reachable sets for trajectory planning of automated vehicles. *IEEE Transactions on Intelligent Vehicles*, *6*(2), 232–248. http://dx.doi.org/10.1109/TIV.2020.3017342.

Nezami, M., Abbas, H. S., Nguyen, N. T., & Schildbach, G. (2022). Robust tube-based LPV-MPC for autonomous lane keeping. *IFAC-PapersOnLine*, *55*(35), 103–108.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, *1*(1), 33–55.

Raghuraman, V., & Koeln, J. P. (2022). Set operations and order reductions for constrained zonotopes. *Automatica*, *139*, Article 110204.

Schäfer, L., Manzinger, S., & Althoff, M. (2023). Computation of solution spaces for optimization-based trajectory planning. *IEEE Transactions on Intelligent Vehicles*, *8*(1), 216–231. http://dx.doi.org/10.1109/TIV.2021.3077702.

Scheffe, P., Henneken, T. M., Kloock, M., & Alrifaee, B. (2022). Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing. *IEEE Transactions on Intelligent Vehicles*.

Scott, J. K., Raimondo, D. M., Marseglia, G. R., & Braatz, R. D. (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, *69*, 126–136.

Söntges, S., & Althoff, M. (2017). Computing the drivable area of autonomous road vehicles in dynamic road scenes. *IEEE Transactions on Intelligent Transportation Systems*, *19*(6), 1855–1866.

Verschueren, R., De Bruyne, S., Zanon, M., Frasch, J. V., & Diehl, M. (2014). Towards time-optimal race car driving using nonlinear MPC in real-time. In *53rd IEEE conference on decision and control* (pp. 2505–2510). http://dx.doi.org/10.1109/CDC.2014.7039771.

Zheng, H., Zheng, L., Li, Y., Wang, K., Zhang, Z., & Ding, M. (2022). Varying zonotopic tube RMPC with switching logic for lateral path tracking of autonomous vehicle. *Journal of the Franklin Institute*, *359*(7), 2759–2787.

Zhou, J., Olofsson, B., & Frisk, E. (2022). Interaction-aware moving target model predictive control for autonomous vehicles motion planning. In *2022 European control conference* ECC, (pp. 154–161). http://dx.doi.org/10.23919/ECC55457.2022.9838002.

Zhou, J., Olofsson, B., & Frisk, E. (2023). Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions. *IEEE Transactions on Intelligent Vehicles*, 1–14. http://dx.doi.org/10.1109/TIV.2023.3314709.