European Journal of Control 74 (2023) 100847

Contents lists available at ScienceDirect

European Journal of Control

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

# Frenet-Cartesian model representations for automotive obstacle avoidance within nonlinear MPC

Rudolf Reiter[a,*], Armin Nurkanović[a], Jonathan Frey[a], Moritz Diehl[a,b]

[a] Department of Microsystems Engineering, University of Freiburg, Georges-Köhler-Allee 102, Room 00-075, Freiburg im Breisgau 79110, Baden-Württemberg, Germany

[b] Department of Mathematics, University of Freiburg, Ernst-Zermelo-Straße 1, Freiburg im Breisgau 79104, Baden-Württemberg, Germany

ARTICLE INFO

*Article history:*
Received 14 May 2023
Accepted 8 June 2023
Available online 15 June 2023

Recommended by Prof. T Parisini

*Keywords:*
Motion planning
Obstacle avoidance
Automotive control
Frenet transformation
Curvilinear coordinate frame

ABSTRACT

In recent years, nonlinear model predictive control has been extensively used for solving automotive motion control and planning tasks. In order to formulate the nonlinear model predictive control problem, different coordinate systems can be used with different advantages. We propose and compare formulations for the nonlinear MPC related optimization problem, involving a Cartesian and a Frenet coordinate frame in a single nonlinear program. We specify costs and collision avoidance constraints in the more advantageous coordinate frame, derive appropriate formulations and compare different obstacle constraints. With this approach, we exploit the simpler formulation of opponent vehicle constraints in the Cartesian coordinate frame, as well as road-aligned costs and constraints related to the Frenet coordinate frame. Comparisons to other approaches in a simulation framework highlight the advantages of the proposed methods.



## 1. Introduction

Trajectory optimization with obstacle avoidance is a major challenge of motion planning and control in autonomous driving. Trajectories need to be feasible to kinodynamic equations and avoid collisions with objects that are often hard to predict. Collision avoidance and the related generation of a reference trajectory or collision avoidance as part of the controller, e.g., nonlinear model predictive control (NMPC), is often formulated as a nonlinear optimal control problem [3,9,12,17]. Through a carefully chosen nonlinear programming (NLP) formulation and by using dedicated real-time optimization solvers [23,25], the problem can be solved efficiently. The transformation of the dynamics into a road-aligned coordinate frame (CF), namely the Frenet CF (FCF), has shown many advantages, such as the simplification of references and road boundaries [9,19,28]. Nevertheless, the transformed coordinates also come with the disadvantage of transformed geometric obstacle shapes [29], cf. Section 2.3. Typical convex geometric shapes, such as boxes, ellipses or circles, are easier to describe in the Cartesian reference CF and become nonconvex after transformation into the FCF. The shapes of objects in both frames are shown in Fig. 1. In nonlinear optimization, "lifting" is a technique where the optimization problem is formulated and solved in a higher dimensional space, which offers advantages regarding convergence rates and the region of attraction [1]. We contribute by an approach to extend and lift the state space of the vehicle model by including both CFs and formulate constraints and costs in the more appropriate CF. We show an increase of the overall performance due to the improved description of the obstacle shapes with various deterministic obstacle avoidance formulations in simulation. Despite the increased state dimensions, even the computation time can be lowered compared to a pure FCF representation. Additionally, references can be set in any of the two CFs, which allows for flexible combinations with planning modules that use either CF, e.g., Vázquez et al. [24].

### 1.1. Related work

The effectiveness of NMPC using the FCF related to automotive tasks was shown in numerous works [5,9,13,17,19,21,22,24,26–28]. None of them explicitly considers the shape transformation of objects, which are rather over-approximated with convex shapes in the FCF. Convex obstacle shapes in the Cartesian CF (CCF) are considered in Rasekhipour et al. [18] with potential fields, in Wang et al. [27], Ziegler et al. [30] with covering circles and Euclidean

* Corresponding author.
*E-mail addresses:* rudolf.reiter@imtek.uni-freiburg.de (R. Reiter), armin.nurkanovic@imtek.uni-freiburg.de (A. Nurkanović), jonathan.frey@imtek.uni-freiburg.de (J. Frey), moritz.diehl@imtek.uni-freiburg.de (M. Diehl).

https://doi.org/10.1016/j.ejcon.2023.100847
0947-3580/

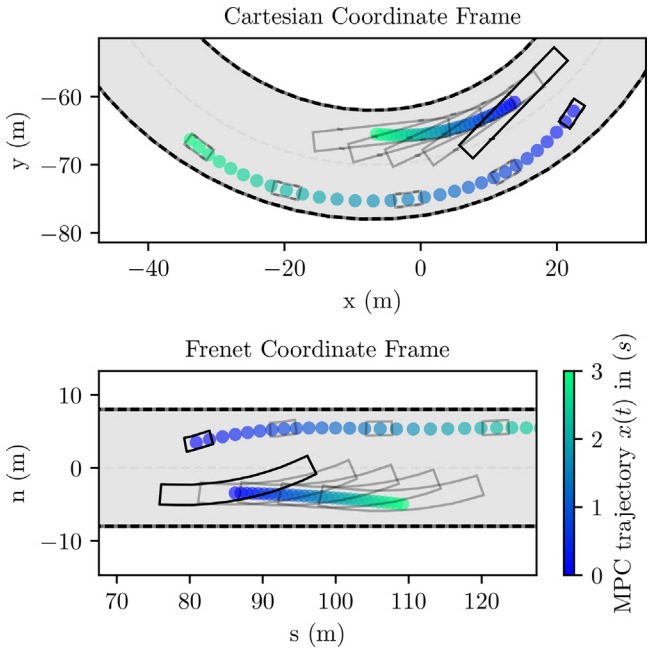


**Fig. 1.** Simulated overtaking of the same maneuver shown in the Cartesian (top plot) and the Frenet CF (bottom plot). Planned trajectories plotted with $\Delta t = 0.1$ s and snapshots of boundary box alignments every 0.7 s.

distance constraints, in Brito et al. [3] with ellipses, in Brossette and Wieber [4], Nair et al. [16] with separating hyperplanes and in Evens et al. [6], Sathya et al. [23] with a formulation related to a conjunction of convex planes covering the obstacle. The most prominent variants are compared within this paper in the Frenet and the lifted formulation. More importantly, the shape-fitting problem with transformed objects in the FCF and an approach with surrogate representations in both CFs were recently considered in a related way in Xing et al. [29]. However, Xing et al. [29] focuses on linear MPC and does not consider dedicated obstacle formulations. Furthermore, it integrates an approximation of the transformation itself into the model, i.e., an approximation of a differential algebraic equation (DAE). In contrast, in our formulation, we provide a reduced index formulation which constitutes an ordinary differential equation (ODE), cf. Section 2.4. Secondly, we eliminate algebraic variables directly. Another variant of tracking along a reference path stems from Lam et al. [11] and was extended to vehicles in Liniger et al. [12]. It uses a method called *contouring control*, which uses a state on a path-length parameterized reference curve and an additional state for its path position. Similarly to Xing et al. [29], it considers the transformation implicitly, which involves approximating a bi-level optimization problem for finding the closest point on the reference curve.

### 1.2. Contribution

We propose novel NMPC formulations that extend the state space to two CFs and allow for more efficient consideration of the occurring costs and constraints. Thereby, the usually convex and simple obstacle shapes in the CCF can be directly used in the NMPC formulation. We show in simulation that we outperform the conventional approach of over-approximation [13,17,22,26] in terms of computation time and performance. Furthermore, the obstacle shapes are independent of the states and the road (up to Euclidean transformations), which is not true in a conventional Frenet representation. Additionally, we contribute with an extensive comparison of common obstacle avoidance formulations in the proposed formulations.

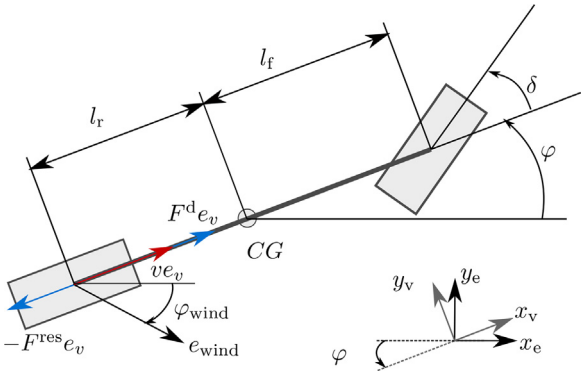


**Fig. 2.** Kinematic vehicle model including wind drag in wind direction $e_{\text{wind}}$.

## 2. Vehicle models

In order to formulate the NMPC problem, we use a rear axis referenced kinematic vehicle model of Kloeser et al. [9], Reiter and Diehl [19], shown in Fig. 2. Given a high enough sampling time and excluding highly dynamic maneuvers (e.g., emergency turns), kinematic vehicle models perform similarly to dynamic models for many automotive motion planning problems [10]. The model comprises three states $x^{\text{c}}$ related to the CF. Particularly, we use $x^{\text{c,C}} = [x \quad y \quad \varphi]^\top \in \mathbb{R}^3$ for the Cartesian states and $x^{\text{c,F}} = [s \quad n \quad \alpha]^\top \in \mathbb{R}^3$ for the Frenet states. We use the Cartesian (earth) position states $x_{\text{e}}$, $y_{\text{e}}$ and the heading angle $\varphi$. Similarly, in the FCF we use longitudinal and lateral position states $s$ and $n$, together with the difference angle $\alpha$ (cf. Section 2.2 and Fig. 1). The FCF position states are curvilinear coordinates along the reference road. Further states $x^{\neg\text{c}} = [v \quad \delta]^\top \in \mathbb{R}^2$ are used for both, CCF and FCF, where $v$ is the absolute value of the velocity at the rear axis and $\delta$ is the steering angle. For the full CCF model we use the state $x^{\text{C}} = [x^{\text{c,C}\top} \quad x^{\neg\text{c}\top}]^\top$ and for the FCF model we use the state $x^{\text{F}} = [x^{\text{c,F}\top} \quad x^{\neg\text{c}\top}]^\top$. We assume a rear wheel drive force $F^{\text{d}}$ as input, including the acceleration and braking force, which is a valid approximation for small steering angles (cf. Kloeser et al. [9], Reiter and Diehl [19]). The most prominent resistance forces for wind $F^{\text{wind}}(v,\varphi) = c_{\text{air}} v_{\text{rel}}(v,\varphi)^2$ and the rolling resistance $F^{\text{roll}}(v) = c_{\text{roll}}\text{sign}(v)$ are included, with the total resistance force $F^{\text{res}}(v,\varphi) = F^{\text{wind}}(v,\varphi) + c_{\text{roll}}\text{sign}(v)$. The air drag depends on the vehicle speed $v$ in relation to the wind speed $v_{\text{wind}}$ with the air friction parameter $c_{\text{air}}$. The rolling resistance is proportional to $\text{sign}(v)$ by the constant $c_{\text{roll}}$. We drop the sign function since we only consider strictly positive speeds. We model the relative speed related to the air drag, which we assume constant and known by $v_{\text{rel}}(v,\varphi) = v - v_{\text{wind}}\cos(\varphi - \varphi_{\text{wind}})$, where $\varphi_{\text{wind}}$ is the angle of the direction $e_{\text{wind}}$ in which the wind asserts force and $\varphi$ is the heading of the vehicle in the CCF. The wind speed was included in recent works [14,15], particularly when it comes to energy-efficient trajectory planning. Real-time wind data can be obtained by weather service providers, such as shown in Meshginqalam and Bauman [14]. The wind speed demonstrates an influence that can be easily modeled in the FCF, but is difficult to model in the CCF.

The input of our model is given by $u = [F^{\text{d}} \quad r]^\top \in \mathbb{R}^2$, where $r = \frac{\text{d}\delta}{\text{d}t}$ denotes the steering rate. The dynamics of the coordinate unrelated states are written as

$$\begin{aligned}\dot{x}^{\neg\text{c}} &= f^{\neg\text{c}}(x^{\neg\text{c}}, u, \varphi)\\ &= \begin{bmatrix} \frac{1}{m}(F^{\text{d}} - F^{\text{wind}}(v,\varphi) - F^{\text{roll}}(v)) \\ r \end{bmatrix},\end{aligned} \tag{1}$$

where $m$ denotes the vehicle mass. The lateral acceleration $a_{\text{lat}}(x^{\neg c})$ at the rear wheel axis is given by

$$a_{\text{lat}}(x^{\neg c}) = \frac{v^2 \tan(\delta)}{l}, \tag{2}$$

where $l$ is the total wheelbase length of the vehicle. The wheelbase length can be expressed as the sum of the distanced between the center of gravity (CG) to the rear wheel axis $l_r$ or front wheel axis $l_f$ by $l = l_f + l_r$.

### 2.1. CCF vehicle model

Using simple kinematic relations, the dynamics of the Cartesian states can be written as

$$\dot{x}^{c,C} = f^{c,C}(x^C, u) = \begin{bmatrix} v\cos(\varphi) \\ v\sin(\varphi) \\ \frac{v}{l}\tan(\delta) \end{bmatrix}. \tag{3}$$

The full five-state Cartesian vehicle model is given by

$$\dot{x}^C = \begin{bmatrix} f^{c,C}(x^C, u) \\ f^{\neg c}(x^{\neg c}, u, \varphi) \end{bmatrix}. \tag{4}$$

### 2.2. FCF vehicle model

Since in usual vehicle motion control tasks the vehicle moves mainly close to a reference curve $\gamma : \mathbb{R} \to \mathbb{R}^2$, i.e., the street center line, the transformation into a curvilinear CF is a natural choice. The reference curve $\gamma(\sigma) = [\gamma_x(\sigma) \quad \gamma_y(\sigma)]^\top$ is parameterized by its path length $\sigma$ and can be fully described by one initial transformation offset $\gamma(\sigma_0)$, an initial orientation $\varphi_0$ and the curvature $\kappa(\sigma) = \frac{d\varphi^\gamma}{d\sigma}$ along its path. We use $\varphi^\gamma(\sigma)$ for the tangent angle in each point of the curve. As part of the Frenet transformation, we project the Cartesian vehicle reference point $p^{\text{veh}} \in \mathbb{R}^2$ on the closest point along the reference curve with

$$s^*(p^{\text{veh}}) = \arg\min_{\sigma} \left\| p^{\text{veh}} - \gamma(\sigma) \right\|_2^2. \tag{5}$$

W.l.o.g., we always set the initial reference point of the transformation to zero. The position $s$ along the curve is used as longitudinal FCF state. The vector $(p^{\text{veh}} - \gamma(s^*))$ is the difference of the closest point on the curve to the vehicle. By using the 90 degree rotation matrix $R^{90}$ and projection to the normal unit vector of the curve $e_n = R^{90}\gamma'(s^*)$, we obtain the Frenet state $n = (p^{\text{veh}} - \gamma(s^*))^\top e_n$. The third Frenet state $\alpha$ relates the tangent angle of the curve to the heading of the vehicle with $\alpha = \varphi - \varphi^\gamma(s^*)$. The transformation relations are shown in Fig. 3. We write the transformation of a Cartesian state $x^{c,C} = [x \quad y \quad \varphi]^\top$ to a Frenet state $x^{c,F} = [s \quad n \quad \alpha]^\top$ by means of the transformation

$$x^{c,F} = \mathcal{F}_\gamma(x^{c,C}) = \begin{bmatrix} s^* \\ (p^{\text{veh}} - \gamma(s^*))^\top e_n \\ \varphi^\gamma(s^*) - \varphi \end{bmatrix}, \tag{6}$$

and its inverse by

$$x^{c,C} = \mathcal{F}_\gamma{}^{-1}(x^{c,F}) = \begin{bmatrix} \gamma_x(s) - n\sin(\varphi^\gamma(s)) \\ \gamma_y(s) + n\cos(\varphi^\gamma(s)) \\ \varphi^\gamma(s) - \alpha \end{bmatrix}. \tag{7}$$

The existence and uniqueness of the transformation are guaranteed under mild assumptions, which are discussed in detail in Reiter and Diehl [19]. As shown in Kloeser et al. [9], we obtain the ODE for the kinematic motion in the FCF as

$$\dot{x}^{c,F} = f^{c,F}(x^F, u) = \begin{bmatrix} \frac{v\cos(\alpha)}{1-n\kappa(s)} \\ v\sin(\alpha) \\ \frac{v}{l}\tan(\delta) - \frac{\kappa(s)v\cos(\alpha)}{1-n\kappa(s)} \end{bmatrix}. \tag{8}$$

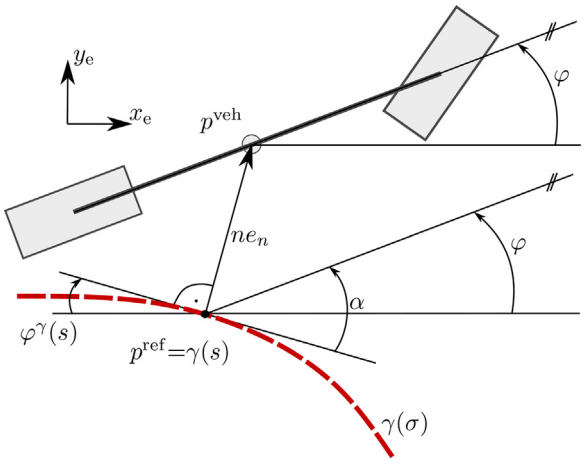


**Fig. 3.** State relations between the CFC and FCF.

**Table 1**
Comparison of the two model representations .

| Feature | CCF | FCF |
|---|---|---|
| Reference definition | ✗ | √ |
| Boundary constraints | ✗ | √ |
| Obstacle specification | √ | ✗ |
| Disturbance specification | √ | ✗ |

The Cartesian state $\varphi$ is needed in order to formulate the wind disturbance. It is not available in the FCF, consequently it needs to be computed by evaluating the tangent angle $\varphi^\gamma(s)$ of the current position $s$ on the reference curve $\gamma(\sigma)$. This can be approximated by a spline function $\hat{\varphi}^\gamma(s)$ that is computed for the road layout and yields an approximation $\hat{\varphi}(s, \alpha) = \hat{\varphi}^\gamma(s) + \alpha$ of the heading angle. The full FCF vehicle model is consequently given by the five-state model

$$\dot{x}^F = \begin{bmatrix} f^{c,F}(x^F, u) \\ f^{\neg c}(x^{\neg c}, u, \hat{\varphi}) \end{bmatrix}. \tag{9}$$

### 2.3. Model comparison

As indicated in Section 1 and Table 1, the CF models have different advantages when used in a NMPC formulation. The definition of road boundaries and the reference curve, which are often lane-aligned curves, are straightforward to define in the FCF, but hard to define in the CCF. However, the obstacle definition in the FCF is cumbersome for several reasons. Despite nonconvex obstacle shapes in the FCF, safety cannot be guaranteed when using sequential quadratic programming (SQP) to solve the NMPC problem with the Frenet model. Convex obstacle shapes cannot be guaranteed to be convex, if transformed into the FCF. This fact can be seen from the following counterexample. Consider a straight line, which is a convex set, and a circular road. Let the line intersect the road at coordinates $\gamma(\sigma_1) = [x_1 \quad y_1]^\top$ and $\gamma(\sigma_2) = [x_1 \quad y_1]^\top$. The transformed Frenet states $n_1, n_2$ are zero in either point. At $\sigma_3 \in (\sigma_1, \sigma_2)$, the transformed state $n_3 \neq 0$, thus the transformed set is not convex. Considering Lemma 2.1, it can be shown that convex obstacles are guaranteed to be a subset of the linearized constraints within an SQP iteration, thus safely over-approximated.

**Lemma 2.1.** *Regard the set* $\mathcal{C} = \{x \in \mathbb{R}^n \mid g(x) \geq 0\}$ *and* $\mathcal{C}^{\text{lin}}(x^*) = \{x \in \mathbb{R}^n \mid g(x^*) + \nabla g(x^*)^\top (x - x^*) \geq 0\}$. *Suppose that the function* $g : \mathbb{R}^n \to \mathbb{R}$ *is convex, then* $\mathcal{C} \subseteq \mathcal{C}^{\text{lin}}(x^*)$ *for any* $x^*$.

**Proof.** Due to convexity, $g(x^*) + \nabla g(x^*)^\top (x - x^*) \leq g(x)$ and therefore, it follows that $\mathcal{C} \subseteq \mathcal{C}^{\text{lin}}$. □

**Table 2**
Comparison of CF Formulations in NMPC. Formulations are compared among their CF specifications, the required number of differential/algebraic states $n_x/n_z$ and their relevance. Bold-typed formulations are compared within this paper.

| Formulation | ODE CF | Obstacle CF | Cost CF | $n_x$ | $n_z$ | Practical relevance | Issues |
|---|---|---|---|---|---|---|---|
| **Conventional Frenet** | Frenet | Frenet | Frenet | 5 | 0 | yes | nonconvex state-dependent obstacle shapes usually over-approximated [13,17,22,26] |
| **Direct Elimination Frenet** | Frenet | Cartesian | Frenet | 5 | 0 | yes | additional nonlinearities (objective, constraints) |
| **Lifted ODE Frenet** | Frenet | Cartesian | Frenet | 8 | 0 | yes | redundant states |
| DAE Frenet | Frenet | Cartesian | Frenet | 5 | 3 | no | bad convergence in our experiments |
| Conventional Cartesian | Cartesian | Cartesian | Cartesian | 5 | 0 | no | nonconvex boundary constraints [12] |
| Cartesian with Frenet States | Cartesian | Cartesian | Frenet | {5, 8} | {0, 3} | yes | Difficult bi-level problem. Approximations, e.g. [12] |

Nonconvex obstacles, which result of the transformation of convex shapes into the FCF, are not safely over-approximated within SQP algorithms. Another problem that arises with objects in the FCF is the dependence of the shape on the state. Consequently, if the obstacle constraints are defined along a discretized time horizon, at each time step $i = 0, \dots, N$, the shape has to be transformed separately, cf. Fig. 1. In typical applications, this could lead to $N$ transformations for each obstacle in every NMPC iteration, followed by a convexification (e.g., bounding boxes, convex polygons, covering circles) to guarantee safety. Alternatively, an over-approximation could be used to capture all possible transformed shapes. However, this would lead to a striking conservatism, especially for long vehicles and small curve radii.

### 2.4. Overview of CF lifting formulations

As outlined in Section 2.3, having states of both CFs in the NLP formulation is beneficial to simplify the constraints. Several different ways of including both CFs are possible, and a summary is given in Table 2.

First, one can choose the primary CF ODE and introduce the states related to the other CF as algebraic variables that are determined by the primary CF and obtain a DAE of index 1. One could either have a CCF based DAE

$$\begin{aligned} \dot{x}^{\mathrm{C}} &= f^{\mathrm{C}}(x^{\mathrm{C}}, u) \\ 0 &= x^{\mathrm{c,F}} - \mathcal{F}_\gamma(x^{\mathrm{c,C}}) \end{aligned} \tag{10}$$

or a FCF based DAE

$$\begin{aligned} \dot{x}^{\mathrm{F}} &= f^{\mathrm{F}}(x^{\mathrm{F}}, x^{\mathrm{c,C}}, u) \\ 0 &= x^{\mathrm{c,C}} - \mathcal{F}_\gamma{}^{-1}(x^{\mathrm{c,F}}). \end{aligned} \tag{11}$$

The inverse transformation $\mathcal{F}_\gamma{}^{-1}$ is computationally cheap since it just needs explicit function evaluations, whereas the forward transformation $\mathcal{F}_\gamma$ requires solving an NLP as in (5), resulting in a computationally expensive bi-level problem in the final NMPC formulation. Therefore, we choose the FCF of (11) as a basis and exclude CCF formulations (10) from further comparisons. The DAE of index 1 (*Lifted DAE Frenet*) is one possible way to formulate the problem and was similarly used in Xing et al. [29] for a linearized model. Another possible formulation (*Direct Elimination Frenet*) is to directly eliminate the algebraic variables in (11) by using the inverse Frenet transformation in the nonlinear constraint formulation. If the objective includes Cartesian states with quadratic costs and lifted constraints, the direct elimination would lead to a nonlinear objective and constraints. Alternatively, we can perform an index reduction of (11), which is obtained by differentiation of the algebraic constraint, leading to

$$\dot{x}^{\mathrm{c,F}} = f^{\mathrm{c,F}}(x^{\mathrm{c,F}}, u) \tag{12a}$$

$$0 = \dot{x}^{\mathrm{c,C}} - \frac{\partial \mathcal{F}_\gamma{}^{-1}(x^{\mathrm{c,F}})}{\partial x^{\mathrm{c,F}}} f^{\mathrm{c,F}}(x^{\mathrm{c,F}}, u), \tag{12b}$$

$$\text{and} \quad x^{\mathrm{c,C}}(0) := \mathcal{F}_\gamma{}^{-1}(x^{\mathrm{c,F}}). \tag{12c}$$

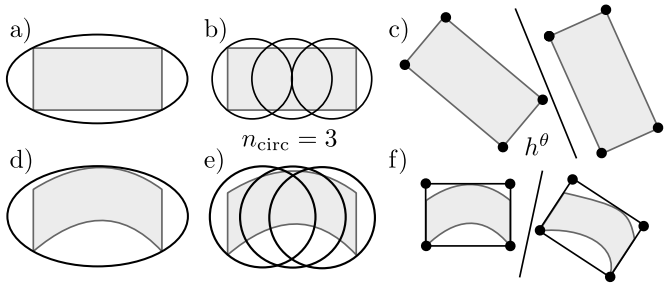


**Fig. 4.** Schematic drawing of obstacle constraints. (a: ellipse CCF, b: covering circles CCF, c: separating hyperplanes CCF, d: ellipse FCF, e: covering circles FCF, f: separating hyperplanes FCF).

Detailed computation (not presented here) shows the equivalence of (12b) to

$$\dot{x}^{\mathrm{c,C}} = f^{\mathrm{c,C}}(x^{\mathrm{c,C}}, u) \tag{13}$$

The approach in (12) or (13) (*Lifted ODE Frenet*) results in redundant states in both CFs, which are coupled through the inputs and the initial state.

## 3. Obstacle avoidance formulations

Different formulations for obstacle avoidance constraints are used in NMPC and visualized in Fig. 4. We assume that rectangles represent the real vehicle shapes. Often simple geometric covering shapes (circles [8] or ellipses [3]) and related distance functions are used. Alternatively, covering polygons and restrictions on edges or vertices (hyperplanes) are formulated in Brossette and Wieber [4], Evens et al. [6], Sathya et al. [23]. Furthermore, the road boundaries can be deflected in order to cover the obstacle by the boundary constraints [9]. The latter approach is not within the scope of this work due to the generally different formulations that for instance, include a combinatorial planner for choosing the passing side [20]. We compare the formulation of obstacle avoidance constraints by an *ellipse* [3], *covering circles* [8,27] and *separating hyperplanes* [4]. We also implemented a formulation introduced in Sathya et al. [23], which we refer to as *set-vertices-exclusion*, but which poorly converged in our experiments. We assume a rectangular shape of the vehicles with the rear/front chassis length $l_{\mathrm{ch}} = l_{\mathrm{r,ch}} + l_{\mathrm{f,ch}}$ related to the vehicle CG and chassis width $w_{\mathrm{ch}}$. The separating hyperplane formulation does not require increased obstacle sizes beyond their actual rectangular shape, whereas circle and ellipse formulations require over-approximations.

### 3.1. Obstacle approximation by an ellipse

Constraining the distance between a circle and an ellipse is less complex than constraining the distance between two ellipses. This can be easily argued by the rotational invariance of a circle which allows for a simple shape inflation of the ellipse

by the radius of the circle, followed by a level set constraint. The distance between two ellipses depends on the orientation of both, thus shape inflation of the one ellipse and a point reduction of the other one is inhibited. Thus, we cover the ego car with a circle. The main axes $a, b$ of an ellipse covering a rectangle are computed by $a = \frac{1}{\sqrt{2}}(l_{\mathrm{f,ch}} + l_{\mathrm{r,ch}})$ and $b = \frac{1}{\sqrt{2}} w_{\mathrm{ch}}$. Increased by the ego radius $r^{\mathrm{ego}}$, this leads to the extended ellipse matrix $D = \mathrm{diag}([a + r^{\mathrm{ego}}, b + r^{\mathrm{ego}}])$. With the rotation matrix $R(x^{\mathrm{c,opp}}) \in \mathbb{R}^{2\times 2}$ and a translation vector $t(x^{\mathrm{c,opp}}) \in \mathbb{R}^2$ related to the orientation and position of the obstacle vehicle, we can formulate the collision avoidance constraint with the matrix $\Sigma(x^{\mathrm{c,opp}}) = R(x^{\mathrm{c,opp}}) D R(x^{\mathrm{c,opp}})^\top$ via the feasible set

$$\mathcal{P}^{\mathrm{ell}}(x^{\mathrm{c,opp}}) = \left\{ x^{\mathrm{c}} \in \mathbb{R}^3 \,\middle|\, \| x^{\mathrm{c}} - t(x^{\mathrm{c,opp}}) \|^2_{\Sigma^{-1}(x^{\mathrm{c,opp}})} \geq 1 \right\}. \tag{14}$$

### 3.2. Obstacle approximation by covering circles

We use the union of a *set of circles* to cover the vehicle shape as shown in Khorkov and Galiev [8], Wang et al. [27], Ziegler et al. [30]. For $l_{\mathrm{ch}} \geq w_{\mathrm{ch}}$, the number of covering circles $n_{\mathrm{circ}}$ must be larger than $\lceil \frac{l_{\mathrm{ch}}}{w_{\mathrm{ch}}} \rceil$ and have a radius $r^{\{\mathrm{ego,opp}\}}$ of $w_{\mathrm{ch}} \frac{1}{\sqrt{2}}$. For each combination of the $n_{\mathrm{circ,ego}}$ and $n_{\mathrm{circ,opp}}$ circles a distance constraint must be satisfied, leading to $n_{\mathrm{circ,ego}} n_{\mathrm{circ,opp}}$ inequality constraints. The covering circle center points are computed according to Ziegler et al. [30], which gives us a function $p^i : \mathbb{R}^3 \to \mathbb{R}^2$ for the circle center $i$ that computes the center positions $p_i = p^i(x^{\mathrm{c,C}})$ from the states $x^{\mathrm{c,C}}$. With $\Delta r = r^{\mathrm{ego}} + r^{\mathrm{opp}}$ and $x := x^{\mathrm{c,C}}$, we can denote the free set as

$$\mathcal{P}^{\mathrm{circ}}(x^{\mathrm{opp}}) = \left\{ x \in \mathbb{R}^3 \,\middle|\, \left\| p^i(x) - p^j(x^{\mathrm{opp}}) \right\|_2 \geq \Delta r, \right.$$
$$\left. \text{for } 1 \leq i \leq n_{\mathrm{circ,ego}},\ 1 \leq j \leq n_{\mathrm{circ,opp}} \right\} \tag{15}$$

### 3.3. Obstacle approximation by separating hyperplanes

When formulating collision avoidance with separating hyperplanes, we optimize for a feasible solution of the parameters $\theta \in \mathbb{R}^3$ of a hyperplane $h^\theta(p)$. The parameterized hyperplane needs to separate all (four) vertices $p_i^{\{\mathrm{ego,opp}\}}(x^{\mathrm{c}\{\mathrm{ego,opp}\}})$ of either vehicle's bounding box. We write the feasible region using the related hyperplane existence problem with points $\tilde{p}^{\{.\}\top} = [p^{\{.\}\top} \quad 1]$ as

$$\mathcal{P}^{\mathrm{hp}}(x^{\mathrm{opp}}) = \left\{ x \in \mathbb{R}^3, \theta \in \mathbb{R}^3 \,\middle|\, \theta_1^2 + \theta_2^2 = 1, \right.$$
$$\left. \theta^\top \tilde{p}_i^{\mathrm{ego}}(x) \leq 0,\ \theta^\top \tilde{p}_i^{\mathrm{opp}}(x^{\mathrm{opp}}) \geq 0,\ \forall i = 0, \ldots, 3 \right\}. \tag{16}$$

With the constraint $\theta_1^2 + \theta_2^2 = 1$ for the hyperplane parameters we avoid a degenerate solution.

## 4. NMPC formulation

The NMPC aims to plan a feasible trajectory for a vehicle to drive on a road with bounded curvature on a reference lane parallel to the center line and with a desired reference speed. Furthermore, the NMPC must avoid static and dynamic obstacles. As motivated in Section 2.4, we use two variants of an FCF-based ODE to obtain Cartesian states, i.e., the *direct elimination* and *lifted ODE* formulation and compare it to the *conventional* formulation with over-approximation, such as shown in Raji et al. [17], Rosolia et al. [22]. First, we define the costs and constraints.

### 4.1. General costs & constraints

Some constraints are unrelated to the CF, such as the lower and upper bounds for states $x^{\neg \mathrm{c}}$ and inputs $u$. For control costs $u^\top R u$, we use the positive semi-definite weight matrix $R \in \mathbb{R}^{2\times 2}$.

### 4.2. FCF related costs & constraints

State costs are related to FCF states since there is no practical advantage of including CCF state costs. A cost related to a desired reference path parallel to the road center line is accounted for by a square penalty of the deviation of the Frenet lateral coordinate $n$ to its reference $n_{\mathrm{ref}}$. For a reference speed $v_{\mathrm{ref}}$, a square penalty with positive weight $w_s$, as well as a penalty on precomputed longitudinal reference positions $s_{\mathrm{ref},i} = \hat{s}_0 + i \Delta t v_{\mathrm{ref}}$ is used, with the measured state $\hat{s}_0$ and sampling time $\Delta t$. Since we assume a road with constant width, boundary constraints simplify in the FCF to box constraints for an upper bound $\overline{n}$ and a lower bound $\underline{n}$.

### 4.3. CCF related costs & constraints

We use the collision avoidance formulations, which could be one out of $\mathcal{O} = \{\mathrm{ell}, \mathrm{circ}, \mathrm{hp}\}$ in the CCF, thus have the constraint $x^{\mathrm{c,C}} \in \mathcal{P}^{\{\mathrm{ell,circ,hp}\}}$. FCF costs are defined via the positive weight matrix $Q = \mathrm{diag}(q)$ with the weight vector $q \in \mathbb{R}^5$ and the reference states $x^{\mathrm{F}}_{\mathrm{ref}}$. We use a terminal cost $Q_N = \mathrm{diag}(q_N)$ with the weight vector $q_N \in \mathbb{R}^5$ after $N$ discrete time steps. We can therefore write the objective function as

$$J(x_0^{\mathrm{F}}, \ldots, x_N^{\mathrm{F}}, u_0, \ldots, u_{N-1}) = \sum_{k=0}^{N-1} \|u_k\|_R^2 + \left\| x_k^{\mathrm{F}} - x_{\mathrm{ref},k}^{\mathrm{F}} \right\|_Q^2 + \left\| x_N^{\mathrm{F}} - x_{\mathrm{ref},N}^{\mathrm{F}} \right\|_{Q_N}^2. \tag{17}$$

### 4.4. Direct elimination NMPC formulation

With the direct formulation, we can directly use the inverse transformation $x^{\mathrm{c,C}} = {\mathcal{F}_\gamma}^{-1}(x^{\mathrm{c,F}})$ to eliminate the Cartesian states in the constraint formulation. Consequently, we obtain fewer states but "more" nonlinear constraints. We discretize the continuous trajectory with $N-1$ control intervals and use direct multiple shooting [2] with one step of an RK4 integration function $\Phi^{\mathrm{F}}(x^{\mathrm{F}}, u, \Delta t)$ for the ODE in (9) and the NLP formulation

$$\min_{\substack{x_0^{\mathrm{F}}, \ldots, x_N^{\mathrm{F}}, \\ u_0, \ldots, u_{N-1} \\ \theta_1, \ldots, \theta_{n_{\mathrm{opp}}}}} \quad J(x_0^{\mathrm{F}}, \ldots, x_N^{\mathrm{F}}, u_0, \ldots, u_{N-1}) \tag{18a}$$

s.t.

$$x_0^{\mathrm{F}} = \hat{x}_0^{\mathrm{F}}, \tag{18b}$$
$$x_{i+1}^{\mathrm{F}} = \Phi^{\mathrm{F}}(x_i^{\mathrm{F}}, u_i, \Delta t), \qquad i = 0, \ldots, N-1, \tag{18c}$$
$$\underline{u} \leq u_i \leq \overline{u}, \qquad i = 0, \ldots, N-1, \tag{18d}$$
$$\underline{x}^{\mathrm{F}} \leq x_i^{\mathrm{F}} \leq \overline{x}^{\mathrm{F}}, \qquad i = 0, \ldots, N, \tag{18e}$$
$$\underline{x}^{\mathrm{c,C}} \leq {\mathcal{F}_\gamma}^{-1}(x^{\mathrm{c,F}}) \leq \overline{x}^{\mathrm{c,C}}, \qquad i = 0, \ldots, N, \tag{18f}$$
$$\underline{a}^{\mathrm{lat}} \leq a_{\mathrm{lat}}^{\mathrm{F}}(x_i) \leq \overline{a}^{\mathrm{lat}}, \qquad i = 0, \ldots, N, \tag{18g}$$
$$v_N \leq \overline{v}_N, \tag{18h}$$
$${\mathcal{F}_\gamma}^{-1}(x^{\mathrm{c,F}}) \in \mathcal{P}(x_i^{\mathrm{c,opp,j}}, \theta_j), \quad i = 0, \ldots, N-1, \quad j = 1, \ldots, n_{\mathrm{opp}}. \tag{18i}$$

Decision variables $\theta_1, \ldots, \theta_{n_{\mathrm{opp}}}$, where $\theta_j = [\theta_j^0, \ldots, \theta_j^N] \in \mathbb{R}^{3\times N}$ are only used for the separating hyperplanes formulation.

### 4.5. Lifted ODE NMPC formulation

In this formulation we use the extended state $x^{\mathrm{d}} = [x^{\mathrm{F}\top} \quad x^{\mathrm{c,C}\top}]^\top$ and the extended ODE (13). The additional states increase the size of the state-space to eight states in our case, where three states stem from either CF and additional two states are CF independent states. In this formulation, we use the RK4 integration function $\Phi^{\mathrm{d}}(x^{\mathrm{d}}, u, \Delta t)$ of dynamics (13). We can write the final NLP for the *lifted ODE* formulation as

$$\min_{\substack{x_0^{\mathrm{d}},\dots,x_N^{\mathrm{d}},\\ u_0,\dots,u_{N-1}\\ \theta_1,\dots,\theta_{n_{\mathrm{opp}}}}} \quad J(x_0^{\mathrm{F}},\dots,x_N^{\mathrm{F}},u_0,\dots,u_{N-1}) \tag{19a}$$

$$s.t. \tag{19a}$$

$$x_0^{\mathrm{d}} = \hat{x}_0^{\mathrm{d}}, \tag{19b}$$

$$x_{i+1}^{\mathrm{d}} = \Phi^{\mathrm{d}}(x_i^{\mathrm{d}}, u_i, \Delta t), i = 0,\dots,N-1, \tag{19c}$$

$$\underline{u} \le u_i \le \overline{u}, \qquad i = 0,\dots,N-1, \tag{19d}$$

$$\underline{x}^{\mathrm{d}} \le x_i^{\mathrm{d}} \le \overline{x}^{\mathrm{d}}, \qquad i = 0,\dots,N, \tag{19e}$$

$$\underline{a}^{\mathrm{lat}} \le a_{\mathrm{lat}}(x_i^{\mathrm{d}}) \le \overline{a}^{\mathrm{lat}}, \quad i = 0,\dots,N, \tag{19f}$$

$$v_N \le \overline{v}_N, \tag{19g}$$

$$x_i^{\mathrm{c,C}} \in \mathcal{P}(x_i^{\mathrm{c,opp,j}}, \theta_j), \quad i = 0,\dots,N-1, \quad j = 1,\dots,n_{\mathrm{opp}}. \tag{19h}$$

## 5. Numerical experiments

In order to evaluate the performance of the proposed approach, we simulate two randomized scenarios that constitute three non-ego vehicles in front of the ego vehicle with a lower cruise speed. The scenario is simulated for 20 s, where usually three overtakes are possible. In total, 500 full simulation runs are evaluated for each scenario type. We record the solution times of the NMPC and the final driven distance after the simulation ends, which we take as a performance indicator. We use different types of obstacles, particularly long ones in the dimensions of a truck (truck-sized), as well as short ones resembling normal cars (car-sized). We make several simplifications in order to avoid performance influences of sources unrelated to our formulation. Firstly, there is no model-plant mismatch, i.e., the simulation framework and the NLP use the same kinematic vehicle model and discretization. Secondly, the ego NMPC has complete knowledge of the other vehicles' planned trajectories to avoid the influence of prediction errors. Finally, we model non-ego participants to be non-interactive. They aim at driving along a reference line parallel to the center line. The simulations were run on an Alienware m-15 Notebook with an Intel Core i7-8550 CPU (1.8 GHz). The parameters for the environment and the NMPC are shown in Tables 3 and 4, respectively.

We use the NLP solver `acados` [25] with `HPIPM` [7], RTI iterations and a partial condensing horizon of $\frac{N}{2}$.

We use obstacle constraint formulations of Section 3. Besides the different obstacle dimensions, the proposed NMPC formulations *conventional*, *direct elimination* and *lifted ODE* were evaluated with the different obstacle formulations of Section 3. We use the ellipse ("EL"), the $n$ covering circles ("C$n$") and the separating hyperplane ("HP") formulation. In Fig. 5, the computation times and the maximum achieved progress of the randomized scenarios are shown for truck- and car-sized vehicles. Clearly, the final progress after overtaking in the truck-sized scenario is increased by the proposed formulation significantly due to the more accurate representation of the obstacle shape. For car-sized vehicles, the extended states do not yield a prominent advantage since in this case, the Frenet transformation does not deform the obstacles vastly. The maximum progress is nearly equal for both proposed approaches since the obstacle constraint formulations based on Cartesian states are equal. A striking difference between the two proposed formulations can be seen in the computation times, shown detailed in Table 5. While the *lifted ODE* formulation even decreases the average computation time for nearly all obstacle formulations, the *direct elimination* formulation increases the computation time by around 30%. Remarkably, the ellipsoidal obstacle formulation in the proposed lifted ODE formulation outperforms all other obstacle formulations in both, the computation time as well

**Table 3**
Environment parameters.[1] Randomized with uniform distribution.[2] Parameters only differ in long vehicle scenario. The parameters are equal for all vehicles, if not noted explicitly. We use SI units, if not stated explicitly.

| Module | Name | Variable | Value |
|---|---|---|---|
| Road | road bounds[2] | $\underline{n}, \overline{n}$ | $\pm 10, \pm 8.5$ |
| | curvature[1] | $\kappa$ | $[-0.05, 0.05]$ |
| | wind speed | $v_{\mathrm{wind}}$ | 20 |
| | wind direction | $\varphi_{\mathrm{wind}}$ | 0 |
| Ego vehicle | length wheelbase | $l_{\mathrm{r}}, l_{\mathrm{f}}$ | 1.7 |
| | length chassis | $l_{\mathrm{r,ch}}, l_{\mathrm{f,ch}}$ | 2 |
| | width chassis | $w_{\mathrm{ch}}$ | 1.9 |
| | mass | m | 1160 |
| | lateral acc. bound | $\underline{a}_{\mathrm{lat}}, \overline{a}_{\mathrm{lat}}$ | $\pm 5$ |
| | input bounds | $\underline{u}, \overline{u}$ | $\pm$[10 kN, 0.39] |
| | velocity bound | $\overline{v}$ | 40 |
| | steering angle bound | $\underline{\delta}, \overline{\delta}$ | $\pm 0.3$ |
| | starting position[1] | $x_0^{\mathrm{c,F}}$ | $[0,-5,0] - [0,5,0]$ |
| | reference speed | $v_{\mathrm{ref}}$ | 40 |
| Opp. vehicles | length wheelbase[2] | $l_{\mathrm{r}}, l_{\mathrm{f}}$ | 10 |
| | length chassis[2] | $l_{\mathrm{r,ch}}, l_{\mathrm{f,ch}}$ | 13 |
| | width chassis[2] | $w_{\mathrm{ch}}$ | 4 |
| | mass[2] | m | 3000 |
| | input bounds[2] | $\overline{u}$ | [30 kN, 0.39] |
| | input bounds[2] | $\underline{u}$ | [−45 kN, −0.39] |
| | starting position[1] $i$ | $s_0^i$ | $50(i+1)$ |
| | | $n_0^i$ | $[-5, 5]$ |
| | reference speed | $v_{\mathrm{ref}}$ | 15 |

**Table 4**
Parameters for NMPC in SI units.

| Name | Variable | Value |
|---|---|---|
| nodes / disc. time | $N$/ $\Delta t$ | 40/ 0.1 |
| terminal velocity | $\overline{v}_N$ | 15 |
| state weights | $q$ | $[1, 500, 10^3, 10^3, 10^4]\Delta t$ |
| terminal state weights | $q_N$ | [10, 90, 100, 10, 10] |
| control weights | $R$ | $\mathrm{diag}([10^{-3}, 2 \times 10^6])\Delta t$ |

**Table 5**
Mean and standard deviation of computation times for different scenarios, obstacle formulations and lifting formulations. Additionally, the difference in percent to the conventional formulation is given.

| | Computation times (ms) for truck-sized obstacles | | | | |
|---|---|---|---|---|---|
| | Conventional | Direct elimination | | Lifted ODE | |
| EL | $1.5 \pm 0.4$ | $1.9 \pm 0.2$ | 28.9% | $1.4 \pm 0.3$ | −6.6% |
| C5 | $7.2 \pm 1.9$ | $7.6 \pm 1.7$ | 5.5% | $7.2 \pm 1.8$ | −0.0% |
| C7 | $14.0 \pm 3.2$ | $14.0 \pm 2.8$ | −0.1% | $13.9 \pm 2.9$ | −0.4% |
| HP | $7.5 \pm 1.5$ | $7.5 \pm 1.5$ | −0.1% | $7.4 \pm 1.7$ | −1.6% |
| Car-sized obstacles | | | | | |
| EL | $1.5 \pm 0.5$ | $2.0 \pm 0.4$ | 29.6% | $1.4 \pm 0.4$ | −5.7% |
| C1 | $1.4 \pm 0.4$ | $1.9 \pm 0.4$ | 34.0% | $1.4 \pm 0.4$ | −3.5% |
| C3 | $3.6 \pm 1.1$ | $4.0 \pm 1.0$ | 12.4% | $3.6 \pm 1.1$ | 0.6% |
| HP | $8.0 \pm 2.3$ | $7.9 \pm 1.9$ | −0.6% | $7.7 \pm 2.0$ | −4.0% |

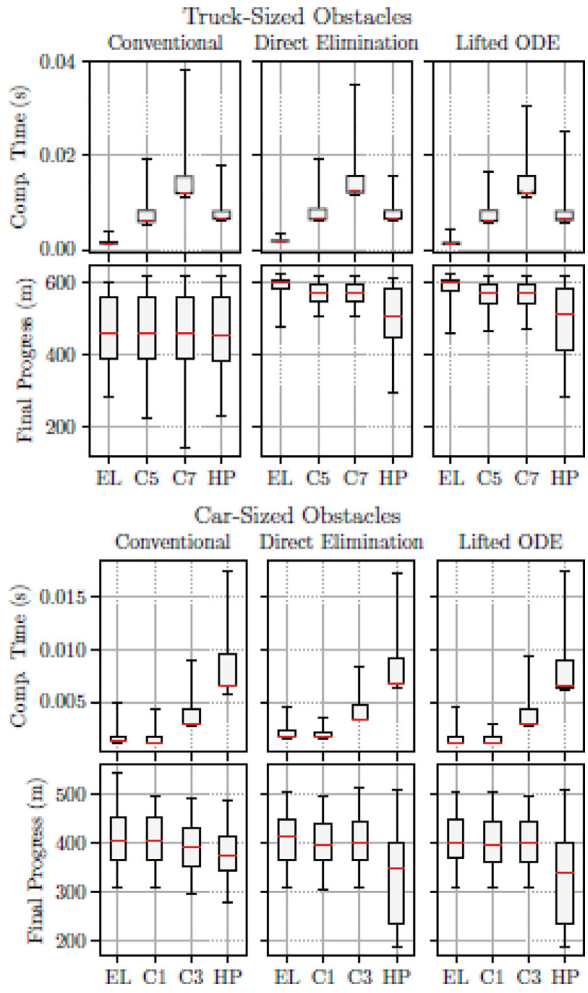


**Fig. 5.** Box-plot comparison of the NMPC solution timings for each real-time iteration and the final progress after 20 s for different obstacle formulations for truck- and car-sized vehicles.

as the performance measured in the average progress after overtaking, which highlights the advantage of the formulation. Contrary to our expectations, the separating hyperplane formulation shows weaker performance in computation time and average progress. In theory, separating hyperplanes should be more accurate in capturing the obstacle shape. Nevertheless, due to the disadvantageous linearizations within the SQP iterations, the shape is not captured well. This might be mainly due to the nonconvex and nonlinear constraint in (16). Note that the proposed lifting approach is not limited to kinematic vehicle models. It extends to higher fidelity models, since the lifting is limited to the six coordinate related states that appear equally in high fidelity models [24], namely CCF positions $x$ and $y$, the CCF heading angle $\theta$, the FCF position states $s$ and $n$, as well as the FCF angle $\alpha$.

## 6. Conclusions

We have presented two novel FCF-based formulations of NMPC for vehicles that include states of the CCF in order to gain numerical advantages. Simulated evaluations and the comparison of several wide-spread obstacle constraint formulations show that the proposed approaches are capable of representing the obstacle shapes more suitably and that with the *lifted ODE* formulation, even the computation time was decreased. Furthermore, our evaluations show that an ellipsoidal obstacle representation outperforms all other obstacle formulations in computation time. In conclusion, the combination of the ellipsoidal obstacle constraint formulation with the *lifted ODE* formulation yields superior results in all categories.

## 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.

## Acknowledgment

This research was supported by DFG via Research Unit FOR 2401 and project 424107692 and by the EU via ELO-X 953348.

## References

[1] J. Albersmeyer, M. Diehl, The lifted Newton method and its application in optimization, SIAM J. Optim. 20 (3) (2010) 1655–1684.
[2] H.G. Bock, K.J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems, in: Proceedings of the IFAC World Congress, Pergamon Press, 1984, pp. 242–247.
[3] B. Brito, B. Floor, L. Ferranti, J. Alonso-Mora, Model predictive contouring control for collision avoidance in unstructured dynamic environments, IEEE Robot. Autom. Lett. 4 (4) (2019) 4459–4466.
[4] S. Brossette, P.-B. Wieber, Collision avoidance based on separating planes for feet trajectory generation, in: 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), IEEE, 2017, pp. 509–514.
[5] A. Buyval, A. Gabdulin, R. Mustafin, I. Shimchik, Deriving overtaking strategy from nonlinear model predictive control for a race car, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 2623–2628.
[6] B. Evens, M. Schuurmans, P. Patrinos, Learning MPC for interaction-aware autonomous driving: a game-theoretic approach, in: 2022 European Control Conference (ECC), IEEE, 2022, pp. 34–39.
[7] G. Frison, M. Diehl, HPIPM: a high-performance quadratic programming framework for model predictive control, in: Proceedings of the IFAC World Congress, Berlin, Germany, 2020.
[8] A.V. Khorkov, S.I. Galiev, Optimization of a $k$-covering of a bounded set with circles of two given radii, Open Comput. Sci. 11 (1) (2021) 232–240.
[9] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Frison, M. Diehl, NMPC for racing using a singularity-free path-parametric model with obstacle avoidance, Proceedings of the IFAC World Congress, 2020.
[10] J. Kong, M. Pfeiffer, G. Schildbach, F. Borrelli, Kinematic and dynamic vehicle models for autonomous driving control design, in: 2015 IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 1094–1099, doi:10.1109/IVS.2015.7225830.
[11] D. Lam, C. Manzie, M. Good, Model predictive contouring control, in: Proceedings of the IEEE Conference on Decision and Control (CDC), 2010.
[12] A. Liniger, A. Domahidi, M. Morari, Optimization-based autonomous racing of 1:43 scale RC cars, Optim. Control Appl. Methods 36 (5) (2015) 628–647.
[13] N. Li, E. Goubault, L. Pautet, S. Putot, Autonomous racecar control in head-to-head competition using mixed-integer quadratic programming, in: Opportunities and Challenges with Autonomous Racing, 2021 ICRA Workshop, Online, United States, 2021. https://hal.archives-ouvertes.fr/hal-03749355
[14] A. Meshginqalam, J. Bauman, Two-level MPC speed profile optimization of autonomous electric vehicles considering detailed internal and external losses, IEEE Access 8 (2020) 206559–206570, doi:10.1109/ACCESS.2020.3038050.
[15] A. Meshginqalam, J. Bauman, Integrated convex speed planning and energy management for autonomous fuel cell hybrid electric vehicles, IEEE Trans. Transp. Electrif. 9 (1) (2023) 1072–1086, doi:10.1109/TTE.2022.3200013.
[16] S.H. Nair, E.H. Tseng, F. Borrelli, Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control, in: 2022 IEEE 61st Conference on Decision and Control (CDC), 2022, pp. 5267–5272, doi:10.1109/CDC51059.2022.9993319.
[17] A. Raji, A. Liniger, A. Giove, A. Toschi, N. Musiu, D. Morra, M. Verucchi, D. Caporale, M. Bertogna, Motion planning and control for multi vehicle autonomous racing at high speeds, in: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 2022, pp. 2775–2782, doi:10.1109/ITSC55140.2022.9922239.
[18] Y. Rasekhipour, A. Khajepour, S.-K. Chen, B. Litkouhi, A potential field-based model predictive path-planning controller for autonomous road vehicles, IEEE Trans. Intell. Transp. Syst. 18 (5) (2016) 1255–1267.
[19] R. Reiter, M. Diehl, Parameterization approach of the Frenet transformation for model predictive control of autonomous vehicles, in: Proceedings of the European Control Conference (ECC), 2021.
[20] R. Reiter, M. Kirchengast, D. Watzenig, M. Diehl, Mixed-integer optimization-based planning for autonomous racing with obstacles and rewards, in: Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC), 2021.

[21] R. Reiter, F. Messerer, M. Schratter, D. Watzenig, M. Diehl, An inverse optimal control approach for trajectory prediction of autonomous race cars, Proceedings of the European Control Conference (ECC), 2022.
[22] U. Rosolia, S. De Bruyne, A.G. Alleyne, Autonomous vehicle control: a nonconvex approach for obstacle avoidance, IEEE Trans. Control Syst. Technol. 25 (2) (2017) 469–484, doi:10.1109/TCST.2016.2569468.
[23] A. Sathya, P. Sopasakis, A. Themelis, R.V. Parys, G. Pipeleers, P. Patrinos, Embedded nonlinear model predictive control for obstacle avoidance using PANOC, Proceedings of the European Control Conference (ECC), 2018.
[24] J.L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, J. Lygeros, Optimization-based hierarchical motion planning for autonomous racing, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 2397–2403, doi:10.1109/IROS45743.2020.9341731.
[25] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, M. Diehl, Acados – a modular open-source framework for fast embedded optimal control, Math. Program. Comput. (2021), doi:10.1007/s12532-021-00208-8.
[26] Q. Wang, T. Weiskircher, B. Ayalew, Hierarchical hybrid predictive control of an autonomous road vehicle, ASME 2015 Dynamic Systems and Control Conference, 2015, doi:10.1115/DSCC2015-9773.
[27] J. Wang, Y. Yan, K. Zhang, Y. Chen, M. Cao, G. Yin, Path planning on large curvature roads using driver-vehicle-road system based on the kinematic vehicle model, IEEE Trans. Veh. Technol. 71 (1) (2021) 311–325.
[28] M. Werling, J. Ziegler, S. Kammel, S. Thrun, Optimal trajectory generation for dynamic street scenarios in a Frenet frame, in: Proceedings - IEEE International Conference on Robotics and Automation, 2010, pp. 987– 993, doi:10.1109/ROBOT.2010.5509799.
[29] X. Xing, B. Zhao, C. Han, D. Ren, H. Xia, Vehicle motion planning with joint Cartesian-Frenét MPC, IEEE Robot. Autom. Lett. 7 (4) (2022) 10738–10745, doi:10.1109/LRA.2022.3194330.
[30] J. Ziegler, P. Bender, T. Dang, C. Stiller, Trajectory planning for bertha - a local, continuous method, in: IEEE Intelligent Vehicles Symposium, Proceedings, 2014, pp. 450–457, doi:10.1109/IVS.2014.6856581.