Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Challenges of Trajectory Planning with Integrator Models on Curved Roads ⋆

Jan Eilbrecht * **Olaf Stursberg** *

* *Institute of Control and System Theory, Department of Electrical Engineering and Computer Science, University of Kassel, 34121 Kassel, Germany, (e-mail: {jan.eilbrecht,stursberg}@uni-kassel.de).*

Abstract: In the context of trajectory planning for autonomous vehicles, a widely used vehicle model relies on linear integrator dynamics. We consider planning with this model type, with a focus on the requirement to account for curved road topologies. As our analysis reveals, this generally gives rise to non-convex, coupled constraints on the vehicle's states and inputs, which impedes computationally efficient planning. We propose a method to resolve this issue by modification of the non-convex constraints. This modification is based on inner approximations of sub-level sets of nonlinear functions, which are obtained by quantifier elimination. The efficacy of the method is demonstrated in two example scenarios.

Keywords: Autonomous driving, trajectory planning, vehicle dynamic systems, constrained control

1. INTRODUCTION

1.1 Motivation

A great challenge in the context of autonomous driving is to plan trajectories for a vehicle, i.e., to schedule its future positions and velocities without causing collisions with other vehicles or leaving the road. Due to its non-convex nature, this task constitutes a complex problem, which has found considerable attention in the fields of robotics, e.g. (LaValle, 2006), and autonomous driving, cf. (González et al., 2016). Many approaches formulate the problem as a discrete-time optimal control problem with the objective to minimize the value of a cost function J. For a given initial state $x_0 := x(t_0)$ of a dynamical system, the value of the cost function depends on the sequence of states $x(\cdot) := (x(t_k) \in \mathbb{R}^{n_x})_{k=0}^{H}$ resulting from a sequence of input signals $u(\cdot) := (u(t_k) \in \mathbb{R}^{n_u})_{k=0}^{H-1}$ as predicted over a horizon H. The optimal input sequence results from:

Problem 1. (Planning Problem).

$$\min_{u(\cdot)} J = \min_{u(\cdot)} \sum_{k=1}^{H} \|x(t_k) - x_{\text{ref}}\|_Q^2 + \|u(t_{k-1})\|_R^2,$$

subject to dynamical constraints on the state:

$$x(t_{k+1}) = f(x(t_k), u(t_k)) = Ax(t_k) + Bu(t_k). \quad (1)$$

Assume appropriate dimensions for A, B, Q, R, and C_z. Inputs and states are constrained to lie within sets $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ and $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, respectively. Also, coupling constraints:

$$\left[x(t_k)^\mathsf{T} \; u(t_k)^\mathsf{T}\right]^\mathsf{T} \in \mathcal{C} \subseteq \mathcal{X} \times \mathcal{U}, \quad (2)$$

may be imposed. Collision avoidance constraints require the system not to enter a set $\mathcal{F}(t_k) \subset \mathcal{X}$ of forbidden states:

$$x(t_k) \notin \mathcal{F}(t_k), \quad (3)$$

encoding, e.g., positions of other vehicles.

⋆ Financial support by the German Research Foundation (DFG) within priority program (SPP) 1835 is gratefully acknowledged.

Established methods exist for the implementation of (3), e.g. relying on mixed-integer programming (Williams, 1990). Since this is not the focus of this paper, we omit the details. Often (Qian et al., 2016; Eilbrecht and Stursberg, 2018; Hess et al., 2018; Burger and Lauer, 2018; Nilsson et al., 2015; Schürmann et al., 2017), the (discretized) dynamics (1) is based on the description of a vehicle's dynamics similar to:

$$\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \dot{v}_x\\ \dot{v}_y\end{bmatrix} = \begin{bmatrix}0&0&1&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}p_x\\p_y\\v_x\\v_y\end{bmatrix} + \begin{bmatrix}0&0\\0&0\\1&0\\0&1\end{bmatrix}\begin{bmatrix}u_x\\u_y\end{bmatrix}, \quad (4)$$

with p_x, p_y, v_x, v_y, u_x, u_y denoting longitudinal (x) and lateral (y) positions, velocities, and accelerations. Additional constraints, e.g. on the rate of change of the inputs, can readily be incorporated and lead to higher-order integrator models or additional constraints on states and inputs.

Clearly, we (and certainly all others who use integrator models for planning) are well aware of the fact that actual vehicle dynamics are more complicated than the integrator dynamics in (4). This per se does not constitute a problem since every model is only a simplified description of the actual system dynamics: Low-level control of vehicles usually relies on the so-called bicycle model (Pacejka, 2005), which considers nonlinear coupling of lateral and longitudinal dynamics as well as (potentially nonlinear) tire dynamics, but neglects vertical, roll, and pitch dynamics. These degrees of freedom, in contrast, are considered by multi-body vehicle models (Blundell and Harty, 2004) used for driving dynamics simulation. But even detailed multi-body vehicle models are often based on rigid body assumptions, neglecting complex elastic deformations of the car body, and so on.

Thus, the mere existence of more detailed vehicle models does not imply that a specific model is not suited for a

certain task. The following arguments promote the usefulness of integrator models for planning: At first, planning is understood as generation of reference trajectories, which are then fed to lower-layer controllers, which are based on more detailed vehicle models, such that vehicle operation is not directly based on the simple integrator models. Then, planning procedures can be combined with methods to verify the safety of a plan. This has been demonstrated by Schürmann et al. (2017) for plans generated based on integrator models. Finally, models with integrator structure result from feedback linearization of nonlinear systems, such that integrator models may actually reflect nonlinear dynamics. That being said, the ultimate purpose of this paper is not to investigate into the appropriateness of integrator models. Instead, we will assume its appropriateness and analyze which challenges arise in this context if curved roads have to be considered. To our knowledge, this has not been explicitly considered in approaches relying on integrator models.

1.2 Related Work

The question of how to represent constraints arising from the topology of a road has found different answers in the literature: A significant part of existing approaches to trajectory planning for autonomous vehicles plans in a Cartesian coordinate frame. These approaches either rely on local linearization, i.e., assume a straight road over the planning horizon, e.g. (Nilsson et al., 2015; Eilbrecht and Stursberg, 2018; Qian et al., 2016), or use approximations to the road boundaries, cf. (Ziegler et al., 2014). Clearly, the validity of the first kind of approach depends on the curvature of the road and the planning horizon. While the second type of approach is flexible and allows to extract information from maps easily, it leads to computationally demanding, non-convex optimization problems.

Other approaches do not use a global, Cartesian coordinate system, but plan in a Frenét frame (Perantoni and Limebeer, 2014; Hess et al., 2018), cf. Sec. 2. This allows to represent constraints arising from the road topology as convex constraints. However, it complicates the equations of motion of a vehicle and the imposed constraints, which generally requires to employ computationally very demanding algorithms (Perantoni and Limebeer, 2014). This problem can be circumvented by simplifying assumptions (Hess et al., 2018), however at the cost of reduced planning accuracy.

2. PRELIMINARIES

We define a planning model to be a tuple:

$$\mathcal{M} = (f, \mathcal{C}), \tag{5}$$

consisting of equations of motion f as in (1) and constraints $\mathcal{C}$ as in (2). Equations of motion are derived based on kinematic and kinetic considerations, in which the kinematics describe a motion, while the kinetics are concerned with the description of the forces causing the motion. In addition, a coordinate system is required to relate a motion to a global coordinate system. For planning, apart from the global coordinate system, two additional coordinate systems are widely used: a body-fixed one and a so-called Frenét frame. These are described here for the sake of completeness and for use in subsequent derivations. While different coordinate systems can give the same information, some may be more appropriate for certain tasks than others. For example, dynamical constraints can be best expressed in a body-fixed reference frame, while it is easier to enforce constraints resulting from the road topology in a Frenét frame.

2.1 Notations and Coordinate Systems

In the following, boldface letters denote vectors, e.g. $\mathbf{r}$. A vector can be expressed by a tuple of coefficients referred to in a certain basis, e.g.:

$$\mathbf{r} = r_1\mathbf{e}_1 + r_2\mathbf{e}_2 + r_3\mathbf{e}_3 =: \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}\mathbf{B}_{123}, \tag{6}$$

in which the basis is a tuple of basis vectors:

$$\mathbf{B}_{123} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix}^\mathsf{T}$$

and is assumed to be ordered and orthonormal. Three coordinate systems will be used, cf. Fig. 1:

Global Inertial Reference Frame The planar position of a vehicle in a global coordinate system with the basis $\mathbf{B}_\mathrm{g} = \begin{bmatrix} \mathbf{e}_{\mathrm{g},x} & \mathbf{e}_{\mathrm{g},y} & \mathbf{e}_{\mathrm{g},z} \end{bmatrix}^\mathsf{T}$ is given by:

$$\mathbf{r}(t) = \begin{bmatrix} p_x(t) & p_y(t) & 0 \end{bmatrix}\mathbf{B}_\mathrm{g}. \tag{7}$$

Frenét Frame In addition, a moving reference frame (Perantoni and Limebeer, 2014) with the basis $\mathbf{B}_\mathrm{tn} = \begin{bmatrix} \mathbf{e}_\mathrm{t} & \mathbf{e}_\mathrm{n} & \mathbf{e}_{\mathrm{g},z} \end{bmatrix}^\mathsf{T}$ will be used. Its basis moves along a path characterized by the path coordinate s and the angle $\theta(s)$ between the tangent at s and the basis vector $\mathbf{e}_{\mathrm{g},x}$ of the global reference frame. Positions are then given by the path coordinate s and an offset n normal to the path at s. A relation between $\mathbf{B}_\mathrm{tn}$ and $\mathbf{B}_\mathrm{g}$ can be established based on:

$$\mathrm{d}p_x = \mathrm{d}s\cos\theta(s) - n(s)\sin\theta(s), \tag{8}$$

$$\mathrm{d}p_y = \mathrm{d}s\sin\theta(s) + n(s)\cos\theta(s), \tag{9}$$

which leads to:

$$p_x(s) = \int_{s_\mathrm{min}}^{s} \cos\theta(\sigma)\,\mathrm{d}\sigma + p_x(s_\mathrm{min}) - n(s)\sin\theta(s), \tag{10}$$

$$p_y(s) = \int_{s_\mathrm{min}}^{s} \sin\theta(\sigma)\,\mathrm{d}\sigma + p_y(s_\mathrm{min}) + n(s)\cos\theta(s). \tag{11}$$

Defining the curvature $C = \frac{\mathrm{d}\theta}{\mathrm{d}s}$, the tangent angle is:

$$\theta(s) = \int_{s_\mathrm{min}}^{s} C(\sigma)\,\mathrm{d}\sigma + \theta(s_\mathrm{min}). \tag{12}$$

Thus, knowledge of the trajectories $C(s)$ and $n(s)$ enables one to transform every point given in the moving reference frame into the global reference frame, in which $n(s) = 0$ gives the shape of the path itself. A trajectory $s(t)$ then determines trajectories over time for all other quantities, e.g. $C = C(s(t))$. The benefit from using this reference frame is that constraints resulting from the road topology can be accounted for conveniently.

Body-fixed Coordinate System The third considered coordinate system has the basis $\mathbf{B}_\mathrm{b} = \begin{bmatrix} \mathbf{e}_{\mathrm{b},x} & \mathbf{e}_{\mathrm{b},y} & \mathbf{e}_{\mathrm{g},z} \end{bmatrix}^\mathsf{T}$, which is fixed at the center of gravity of a considered vehicle. The basis vectors $\mathbf{e}_{\mathrm{b},x}$ and $\mathbf{e}_{\mathrm{b},y}$ span the road plane and $\mathbf{e}_{\mathrm{b},x}$ is aligned with the longitudinal axis of the vehicle. A vector in this coordinate system can be expressed in the Frenét frame:

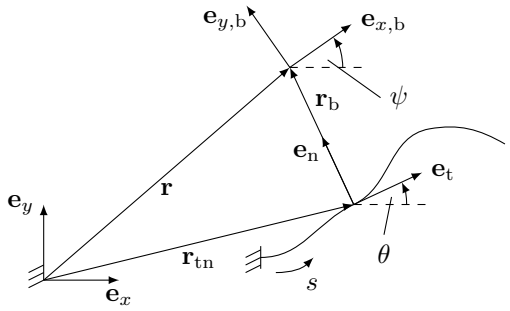


Fig. 1. Coordinate Systems.

$$v^\mathsf{T}\mathbf{B}_\mathrm{b} = v^\mathsf{T}\begin{bmatrix}\cos(\xi) & \sin(\xi) & 0\\ -\sin(\xi) & \cos(\xi) & 0\\ 0 & 0 & 1\end{bmatrix}\mathbf{B}_\mathrm{tn}. \tag{13}$$

The body-fixed frame makes it possible to easily express constraints resulting from the vehicle dynamics.

2.2 Kinematics

In order to describe the motion of a vehicle in one of the coordinate systems, information about its position, velocity, and acceleration is required – this is what kinematics is concerned with. A vector $\mathbf{r}$ can be expressed combining the global and the Frenét frame (cf. Fig. 1):

$$\mathbf{r} = \mathbf{r}_\mathrm{tn} + \mathbf{r}_\mathrm{b} = r_\mathrm{tn}^\mathsf{T}\mathbf{B}_\mathrm{g} + r_\mathrm{b}^\mathsf{T}\mathbf{B}_\mathrm{tn}, \tag{14}$$

in which $\mathbf{r}_\mathrm{tn}$ indicates the position of the origin of the basis $\mathbf{B}_\mathrm{tn}$ and $\mathbf{r}_\mathrm{b}$ the connection from there on to the position of the vehicle. If $\mathbf{r} = \mathbf{r}(t)$, differentiating (14) gives the velocity vector:

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_\mathrm{tn} + \dot{\mathbf{r}}_\mathrm{b}. \tag{15}$$

Alternatively, the velocity vector can be described conveniently in the body-fixed reference frame:

$$\dot{\mathbf{r}} = v^\mathsf{T}\mathbf{B}_\mathrm{b}, \tag{16}$$

which is rotated against the Frenét frame by the angle $\xi = \psi - \theta$. The latter indicates the heading of the vehicle as measured in the Frenét frame, and $v = \begin{bmatrix}v_x & v_y & 0\end{bmatrix}^\mathsf{T}$. Next, we have that:

$$\dot{\mathbf{r}}_\mathrm{tn} = \dot{r}_\mathrm{tn}^\mathsf{T}\mathbf{B}_\mathrm{g} = \begin{bmatrix}\dot{s} & 0 & 0\end{bmatrix}\mathbf{B}_\mathrm{g} \tag{17}$$

and, with $\dot{\theta}$ as angular velocity of $\mathbf{B}_\mathrm{tn}$:

$$\dot{\mathbf{r}}_\mathrm{b} = \dot{r}_\mathrm{b}^\mathsf{T}\mathbf{B}_\mathrm{tn} + \left(\dot{\theta}\times r_\mathrm{b}\right)^\mathsf{T}\mathbf{B}_\mathrm{tn} = \begin{bmatrix}0 & \dot{n} & 0\end{bmatrix}\mathbf{B}_\mathrm{tn} + \left(\begin{bmatrix}0\\0\\\dot{\theta}\end{bmatrix}\times\begin{bmatrix}0\\n\\0\end{bmatrix}\right)^\mathsf{T}\mathbf{B}_\mathrm{tn} = \begin{bmatrix}-n\dot{\theta} & \dot{n} & 0\end{bmatrix}\mathbf{B}_\mathrm{tn}. \tag{18}$$

From (15) and (16), it follows that:

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_\mathrm{tn} + \dot{\mathbf{r}}_\mathrm{b} = v^\mathsf{T}\mathbf{B}_\mathrm{b}, \tag{19}$$

and using $\dot{\theta} = \frac{\mathrm{d}\theta}{\mathrm{d}s}\dot{s} = C\dot{s}$ and (13), we have (cf. Perantoni and Limebeer (2014)):

$$\begin{bmatrix}\dot{s}(1-nC)\\ \dot{n}\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{tn} = \begin{bmatrix}v_x\cos(\xi) - v_y\sin(\xi)\\ v_x\sin(\xi) + v_y\cos(\xi)\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{tn}. \tag{20}$$

Differentiating (19) gives:

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_\mathrm{tn} + \ddot{\mathbf{r}}_\mathrm{b} = \dot{v}^\mathsf{T}\mathbf{B}_\mathrm{b} + \left(\dot{\psi}\times v_\mathrm{b}\right)^\mathsf{T}\mathbf{B}_\mathrm{b}, \tag{21}$$

in which $\dot{\psi}$ is the angular velocity of $\mathbf{B}_\mathrm{b}$, and:

$$\ddot{\mathbf{r}}_\mathrm{tn} = \left(\begin{bmatrix}\ddot{s}\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\theta}\end{bmatrix}\times\begin{bmatrix}\dot{s}\\0\\0\end{bmatrix}\right)^\mathsf{T}\mathbf{B}_\mathrm{tn} = \begin{bmatrix}\ddot{s} & \dot{\theta}\dot{s} & 0\end{bmatrix}\mathbf{B}_\mathrm{tn},$$

$$\ddot{\mathbf{r}}_\mathrm{b} = \left(\begin{bmatrix}-\dot{n}\dot{\theta} - n\ddot{\theta}\\ \ddot{n}\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\theta}\end{bmatrix}\times\begin{bmatrix}-n\dot{\theta}\\ \dot{n}\\ 0\end{bmatrix}\right)^\mathsf{T}\mathbf{B}_\mathrm{tn} = \begin{bmatrix}-2\dot{n}\dot{\theta} - n\ddot{\theta}\\ \ddot{n} - n\dot{\theta}^2\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{tn},$$

such that $\ddot{\mathbf{r}} = \begin{bmatrix}\ddot{s} - 2\dot{n}\dot{\theta} - n\ddot{\theta} & \dot{\theta}\dot{s} + \ddot{n} - n\dot{\theta}^2 & 0\end{bmatrix}\mathbf{B}_\mathrm{tn}$. With:

$$\dot{C}(s) = \frac{\mathrm{d}C(s)}{\mathrm{d}t} = \frac{\mathrm{d}C(s)}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\mathrm{d}C(s)}{\mathrm{d}s}\dot{s} =: C'(s)\dot{s},$$

it shows that $\ddot{\theta} = \frac{\mathrm{d}}{\mathrm{d}t}(C(s)\dot{s}) = C'(s)\dot{s}^2 + C(s)\ddot{s}$, such that

$$\ddot{\mathbf{r}} = \begin{bmatrix}\ddot{s}(1-nC(s)) - 2\dot{n}C(s)\dot{s} - nC'(s)\dot{s}^2\\ \ddot{n} + C(s)\dot{s}^2(1-nC(s))\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{tn} =: \begin{bmatrix}a_{x,\mathrm{tn}}\\ a_{y,\mathrm{tn}}\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{tn}, \tag{22}$$

with the external accelerations $a_{x,\mathrm{tn}}$ and $a_{y,\mathrm{tn}}$ in the Frenét frame. The right hand side of (21) reads:

$$\ddot{\mathbf{r}} = \left(\begin{bmatrix}\dot{v}_x\\ \dot{v}_y\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}\times\begin{bmatrix}v_x\\ v_y\\ 0\end{bmatrix}\right)^\mathsf{T}\mathbf{B}_\mathrm{b} = \begin{bmatrix}\dot{v}_x - v_y\dot{\psi}\\ \dot{v}_y + v_x\dot{\psi}\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{b} =: \begin{bmatrix}a_{x,\mathrm{b}}\\ a_{y,\mathrm{b}}\\ 0\end{bmatrix}^\mathsf{T}\mathbf{B}_\mathrm{b},$$

in which $a_{x,\mathrm{b}}$ and $a_{y,\mathrm{b}}$ are the accelerations in the body-fixed reference frame. For planning purposes, a state-space model is required. By introducing the yaw acceleration a_ψ, the state vector $x_\mathrm{b} = \begin{bmatrix}p_x & p_y & \psi & v_x & v_y & \dot{\psi}\end{bmatrix}^\mathsf{T}$, and $a_\mathrm{b} = \begin{bmatrix}a_{x,\mathrm{b}} & a_{y,\mathrm{b}}\end{bmatrix}^\mathsf{T}$, the model reads:

$$\dot{x}_\mathrm{b} = f_\mathrm{b}(x_\mathrm{b}, a_\mathrm{b}, a_\psi) = \begin{bmatrix}v_x\cos(\psi) - v_y\sin(\psi)\\ v_x\sin(\psi) + v_y\cos(\psi)\\ \dot{\psi}\\ a_{x,\mathrm{b}} + v_y\dot{\psi}\\ a_{y,\mathrm{b}} - v_x\dot{\psi}\\ a_\psi\end{bmatrix} \tag{23}$$

in the body-fixed reference frame. Note that the accelerations are independent of the position and the orientation. Alternatively, due to its ability to conveniently handle constraints arising from the road topology, a state-space model can be derived from the kinematics (22) in the Frenét frame:

$$\dot{x}_\mathrm{tn} = f_\mathrm{tn}(x_\mathrm{tn}, a_{x,\mathrm{tn}}, a_{y,\mathrm{tn}}, a_\psi) = \begin{bmatrix}\dot{s}\\ \dot{n}\\ \dot{\psi} - C(s)\dot{s}\\ \frac{a_{x,\mathrm{tn}} + 2\dot{n}C(s)\dot{s} + nC'(s)\dot{s}^2}{1-nC(s)}\\ a_{y,\mathrm{tn}} - C(s)\dot{s}^2(1-nC(s))\\ a_\psi\end{bmatrix} \tag{24}$$

with the state vector: $x_\mathrm{tn} = \begin{bmatrix}s & n & \xi & \dot{s} & \dot{n} & \dot{\psi}\end{bmatrix}^\mathsf{T}$ and $a_\mathrm{tn} = \begin{bmatrix}a_{x,\mathrm{tn}} & a_{y,\mathrm{tn}}\end{bmatrix}^\mathsf{T}$.

3. DERIVATION OF AN INTEGRATOR MODEL

3.1 Equations of Motion

The basis of planning in Frenét coordinates is the kinematic equation (24). In this section, we show that a simple model of the form (4) can be derived from (24) by

appropriate assumptions. Kinematic equations describe a vehicle's motion without considering the forces causing it and generally hold for all vehicles. Kinetics, in contrast, provide a model of these forces or external accelerations, typically expressed in the body-fixed reference frame.

Usually, the acceleration depends on both the vehicle's velocities (e.g. due to wind resistance) and its inputs (e.g. steering and throttle commands), while we assume independence of the position and orientation. Clearly, this relation may differ from vehicle to vehicle. For the model class at hand, it is assumed that the external accelerations a_b acting on the vehicle in the body-fixed reference frame can be directly controlled, i.e.:

$$u = \begin{bmatrix} a_{x,\mathrm{b}} & a_{y,\mathrm{b}} & a_\psi \end{bmatrix}^\mathsf{T}. \tag{25}$$

This assumes that questions concerning tire dynamics or air resistance have been accounted for, e.g. by lower-level feedback controllers and proper choice of constraints on a_b.

Another crucial assumption concerns the orientation ξ relative to the road. In the case of integrator models, this is assumed to be always aligned with the orientation of the road, i.e.:

$$\xi = \psi - \theta = 0. \tag{26}$$

This makes the considered model less a vehicle model, but a model of a road-aligned box which may be occupied by a vehicle at a certain point in time. According to (13), this equates the accelerations in the body-fixed frame and the Frenét frame, i.e.:

$$a_\mathrm{b} = a_\mathrm{tn}, \tag{27}$$

and fixes the rotational velocity: $\dot\psi = \dot\theta = C(s)\dot s$ as well as rotational acceleration:

$$\ddot\psi = a_\psi = C'(s)\dot s^2 + C(s)\ddot s, \tag{28}$$

eliminating ξ and $\dot\psi$ as a states from (24), leaving $x_1 := \begin{bmatrix} s & n & \dot s & \dot n \end{bmatrix}^\mathsf{T}$. Despite these simplifying assumptions, the resulting equations of motion are still nonlinear:

$$\dot x_1 := f_1(x_1, a_\mathrm{tn}) = \begin{bmatrix} \dot s \\ \dot n \\ \frac{a_{x,\mathrm{tn}} + 2\dot n C(s)\dot s + nC'(s)\dot s^2}{1 - nC(s)} \\ a_{y,\mathrm{tn}} - C(s)\dot s^2(1 - nC(s)) \end{bmatrix}. \tag{29}$$

Further simplifications can be obtained by noting that (24) is affine in the accelerations $a_{x,\mathrm{tn}}$ and $a_{y,\mathrm{tn}}$, while the states are arranged in an integrator chain. Therefore, we introduce artificial system inputs similar to feedback linearization of input affine systems (Khalil, 1996):

$$\tilde u := \begin{bmatrix} u_\mathrm{t} \\ u_\mathrm{n} \end{bmatrix} = \begin{bmatrix} \frac{a_{x,\mathrm{tn}} + 2\dot n C(s)\dot s + nC'(s)\dot s^2}{1 - nC(s)} \\ a_{y,\mathrm{tn}} - C(s)\dot s^2(1 - nC(s)) \end{bmatrix}, \tag{30}$$

such that (29) becomes:

$$\dot x_2 := f_2(x_2, \tilde u) = \begin{bmatrix} \dot s & \dot n & u_\mathrm{t} & u_\mathrm{n} \end{bmatrix}^\mathsf{T}, \tag{31}$$

where $x_2 \equiv x_1$. The similarity to (4) is obvious.

3.2 Constraints

A model often not only consists of equations of motion, but also of constraints. Even though planning requires all constraints to be defined in the Frenét frame, some may initially be defined in different coordinate systems and have to be transformed. This will be addressed in this section.

Constraints on a vehicle's velocity are typically imposed in the body-fixed coordinate system as interval bounds:

$$\begin{bmatrix} v_{x\min} \\ v_{y\min} \end{bmatrix} \le \begin{bmatrix} v_x \\ v_y \end{bmatrix} \le \begin{bmatrix} v_{x\max} \\ v_{y\max} \end{bmatrix}. \tag{32}$$

The transformation of (32) from the body-fixed frame to the Frenét frame results according to (13), which, considering that $\xi = 0$ as per (26), leads to:

$$\begin{bmatrix} v_{x\min} \\ v_{y\min} \end{bmatrix} \le \begin{bmatrix} \dot s(1 + nC(s)) \\ \dot n \end{bmatrix} \le \begin{bmatrix} v_{x\max} \\ v_{y\max} \end{bmatrix}. \tag{33}$$

State constraints resulting from the road topology are typically expressed in the Frenét frame. Combining these with the transformed velocity constraints (33) yields the state constraint set:

$$\begin{aligned} \mathcal{X}_\mathrm{tn} := \Big\{ x_\mathrm{tn} | & C_\mathrm{min} \le C(s) \le C_\mathrm{max},\ 0 \le \dot s \le \dot s_\mathrm{max}(s), \\ & n_\mathrm{min}(s) \le n \le n_\mathrm{max}(s),\ \xi_\mathrm{min} \le \xi \le \xi_\mathrm{max}, \\ & \begin{bmatrix} v_{x\min} \\ v_{y\min} \end{bmatrix} \le \begin{bmatrix} \dot s(1 + nC(s)) \\ \dot n \end{bmatrix} \le \begin{bmatrix} v_{x\max} \\ v_{y\max} \end{bmatrix} \Big\}. \end{aligned} \tag{34}$$

The interval bounds on n and $\dot s$ depend on s in order to account for changes in the road width or speed limits. Note that requirements from collision avoidance with other vehicles are not in the focus of this paper. They can be formulated in either the Frenét frame or the global coordinate system without affecting any of the subsequent derivations.

The accelerations are coupled and constrained by the so-called friction ellipsoid (Pacejka, 2005):

$$a_{x,\mathrm{b}}^2 + a_{y,\mathrm{b}}^2 \le \text{const.} \tag{35}$$

In addition, each acceleration is confined to an interval:

$$a_\mathrm{b,min} \le a_\mathrm{b} \le a_\mathrm{b,max}. \tag{36}$$

By combining (35) and (36) and by transforming the accelerations from the body-fixed to the Frenét frame according to (27), the following set of constraints on the accelerations results:

$$\mathcal{U}_\mathrm{tn} := \left\{ a_\mathrm{tn} | a_{x,\mathrm{tn}}^2 + a_{y,\mathrm{tn}}^2 \le \text{const.},\ a_\mathrm{b,min} \le a_\mathrm{tn} \le a_\mathrm{b,max} \right\}. \tag{37}$$

The planning model (31) is not based on the actual accelerations, but on artificial inputs $\tilde u$, for which a constraint set must be specified. According to (30), $\tilde u$ depends on both a_tn and x_tn. Therefore, it must be chosen such that both (34) and (37) are not violated. Solving (30) for a_tn yields:

$$g(x_\mathrm{tn}, \tilde u) := \begin{bmatrix} (1 - nC(s))u_\mathrm{t} - (2\dot n C(s)\dot s + nC'\dot s^2) \\ u_\mathrm{n} + C(s)\dot s^2(1 - nC(s)) \end{bmatrix}. \tag{38}$$

Note that inserting (38) into (37) leads to nonlinear, mixed state-input constraints. Collecting (34) and (37) (combined with (38)) then yields the constraints:

$$\begin{aligned} \mathcal{Z} = \Bigg\{ & \begin{bmatrix} x_\mathrm{tn} \\ \tilde u \end{bmatrix} | C_\mathrm{min} \le C(s) \le C_\mathrm{max},\ n_\mathrm{min} \le n \le n_\mathrm{max}, \\ & \begin{bmatrix} v_{x\min} \\ v_{y\min} \end{bmatrix} \le \begin{bmatrix} \dot s(1 + nC(s)) \\ \dot n \end{bmatrix} \le \begin{bmatrix} v_{x\max} \\ v_{y\max} \end{bmatrix}, \\ & \dot\psi_\mathrm{min} \le C(s)\dot s \le \dot\psi_\mathrm{max}, \\ & a_{\psi,\mathrm{b,min}} \le C'\dot s^2 + C(s)u_\mathrm{t} \le a_{\psi,\mathrm{b,max}}, \\ & a_\mathrm{b,min} \le g(x_\mathrm{tn}, \tilde u) \le a_\mathrm{b,max},\ \ \|g(x_\mathrm{tn}, \tilde u)\|^2 \le \text{const.} \Bigg\}. \end{aligned} \tag{39}$$

3.3 Constraint Simplification

Clearly, while (24) has been substantially simplified by introduction of the artificial inputs $\tilde{u}$, the new constraint (39) is much more complex than the original constraints on the accelerations (37). To be precise, (39) is defined by level sets of nonlinear functions, such that it generally can be non-convex or even non-connected. In general, these properties cannot even be checked efficiently (Weber and Reissig, 2014). All the more, (39) depends on the longitudinal position s, which introduces further nonlinearities into the planning. For computationally efficient planning, in contrast, simple, if possible polytopic constraints are required. Thus, instead of using (39) directly for planning, this section is concerned with the following problem:

Problem 2. Find a polyoptic inner approximation to (39):

$$\underline{\mathcal{Z}} = \left\{ \begin{bmatrix} x_{\mathrm{tn}} \\ \tilde{u} \end{bmatrix} | N \begin{bmatrix} x_{\mathrm{tn}} \\ \tilde{u} \end{bmatrix} \leq b \right\} \subseteq \mathcal{Z} \tag{40}$$

by appropriate choice of a matrix N and a vector b.

In a first step, we obtain a polytopic inner approximation for the convex, but nonlinear constraint (35). For simplicity, we employ a rectangle, which decouples the interdependence of $a_{x,\mathrm{b}}$ and $a_{y,\mathrm{b}}$. Thus, the input constraints essentially reduce to the type (36) with appropriately tightened bounds $a_{\mathrm{b,min}}$ and $a_{\mathrm{b,max}}$.

In a next step, we aim to resolve the position dependence of (39). This is achieved by elimination of s, $C(s)$, n (and $\dot{n}$ likewise in order to further reduce the dimensionality of the resulting set) from (39) through computation of a set:

$$\underline{\tilde{\mathcal{Z}}} = \left\{ \begin{bmatrix} \dot{s} \\ u_{\mathrm{t}} \\ u_{\mathrm{n}} \end{bmatrix} | \begin{bmatrix} x_{\mathrm{tn}} \\ \tilde{u} \end{bmatrix} \in \mathcal{Z} \ \forall \begin{bmatrix} s \\ C(s) \\ n \\ \dot{n} \end{bmatrix} \in \begin{bmatrix} s_{\min},\ s_{\max} \\ C_{\min},\ C_{\max} \\ n_{\min},\ n_{\max} \\ v_{y\min},\ v_{y\max} \end{bmatrix} \right\}. \tag{41}$$

Here, a typical road topology allows to assume $C(s)$ to be linear or constant, such that C' is constant or zero, i.e., not a variable anymore. The set $\underline{\tilde{\mathcal{Z}}}$ is an inner approximation of the projection of $\mathcal{Z}$ and can be determined by elimination of the quantifier $\forall$ by using algorithms as implemented in Mathematica (Wolfram, 1991) or Maple (Heck, 1993). Note that this is in general a computationally very demanding procedure. In the given case, however, it turns out to be tractable.

The resulting set has been reduced to three dimensions, namely $\dot{s}$, u_{t}, and u_{n}. This allows to gain insight into its shape, for example to assess connectivity by visual inspection. If the set is convex, but not polytopic, standard procedures for computation of inner approximations of convex sets (Lotov et al., 2013, Chpt. 8), (Le Guernic, 2009, Sec. 4.2.2) can be readily employed. Otherwise, for arbitrarily complex, but simply connected sets, the procedure from Xue et al. (2016) can be used if an automated approach is required. However, as we demonstrate in Sec. 4, the sets resulting for typical scenarios are relatively simple even in the non-convex case, such that human intuition quickly leads to sufficiently accurate approximations. The set $\underline{\mathcal{Z}}$ can then be obtained as follows:

$$\underline{\mathcal{Z}} = \underline{\tilde{\mathcal{Z}}} \times [s_{\min},\ s_{\max}] \times [n_{\min},\ n_{\max}] \times [v_{y\min},\ v_{y\max}]. \tag{42}$$

3.4 Resulting Models

In the previous sections, we have derived several equations of motion and corresponding sets of (mixed) states and input constraints. Different combinations of these lead to different planning models:

Combining the nonlinear equations of motion (29) with the input and state constraints (34) and (37) gives:

$$\mathcal{M}_1 = (f_1, \mathcal{X}_{\mathrm{tn}} \times \mathcal{U}_{\mathrm{tn}}). \tag{43}$$

The second model is obtained by combination of the simplified, linear dynamics (31) with the nonlinear, mixed state-input constraints (39):

$$\mathcal{M}_2 = (f_2, \mathcal{Z}). \tag{44}$$

Combining the same dynamics with the linearized constraints (40) gives the third model:

$$\mathcal{M}_3 = (f_2, \underline{\mathcal{Z}}). \tag{45}$$

For comparison purposes, we also introduce a fourth model, combining the linear dynamics (31) with the original input and state constraints (34) and (37):

$$\mathcal{M}_4 = (f_2, \mathcal{X}_{\mathrm{tn}} \times \mathcal{U}_{\mathrm{tn}}). \tag{46}$$

This model clearly is naive in so far as that it ignores the constraint coupling resulting from (38). Based on these models, our main result is summarized as follows:

Proposition 1. Planning with the linear model $\mathcal{M}_3$ generates trajectories which are feasible for the nonlinear model $\mathcal{M}_1$. More formally, given the constraint set $\underline{\mathcal{Z}}$ from (45) and the original constraint sets $\mathcal{X}_{\mathrm{tn}}$ and $\mathcal{U}_{\mathrm{tn}}$ from (43),

$$[x_{\mathrm{tn}}\ \tilde{u}]^{\mathsf{T}} \in \underline{\mathcal{Z}} \Rightarrow x_{\mathrm{tn}} \in \mathcal{X}_{\mathrm{tn}} \wedge \exists a_{\mathrm{tn}} \in \mathcal{U}_{\mathrm{tn}} : a_{\mathrm{tn}} \text{ and } \tilde{u} \text{ fulfill (30)}.$$

Proof. Due to (39), $[x_{\mathrm{tn}}\ \tilde{u}]^{\mathsf{T}} \in \underline{\mathcal{Z}} \Rightarrow [x_{\mathrm{tn}}\ \tilde{u}]^{\mathsf{T}} \in \mathcal{Z}$. Since $\mathcal{Z}$ is defined as the conjunction of (34) and (37) (mapped to $\tilde{u}$), the result is immediate. □

3.5 Strategies to Reduce Conservatism

Clearly, the procedure described in Sec. 3.3 introduces conservatism: first by imposing conditions *for all* possible values of certain variables, and second by inner approximating the resulting set. While finer inner approximations obviously lead to less conservatism, a remedy to the first issue is less obvious. We propose to partition roads into segments of a certain length, in which the ranges of C and n are small enough to introduce only little conservatism when eliminating these variables. Also, a constant value for C' is used. This introduces again position dependence, but on a coarser scale, constituting a good compromise between computational tractability and conservatism. In addition, maneuver-dependent bounds on $\dot{n}$ can be used, e.g. when the vehicle simply has to follow a lane or has to overtake.

4. EXAMPLE

We demonstrate the efficacy of our approach by planning trajectories for a vehicle driving on the road depicted in Fig. 2. The track consists of two circles connected by so-called Euler spirals (road segments of linearly varying curvature) – two common shapes in road design. Fig. 3a shows the curvature values, which represent a relatively

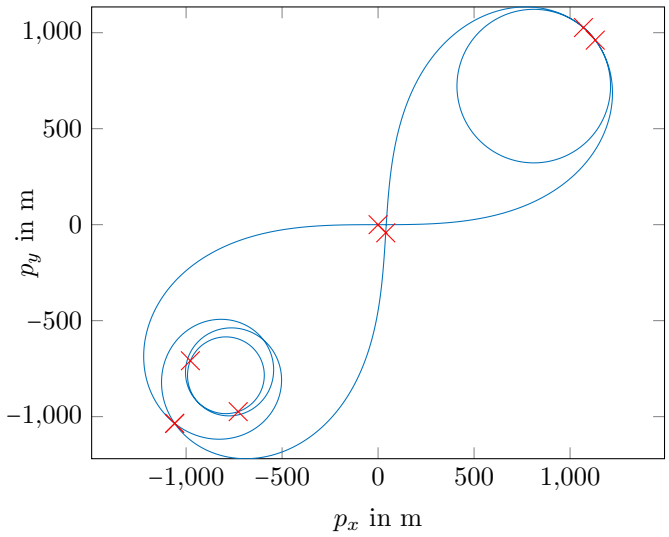
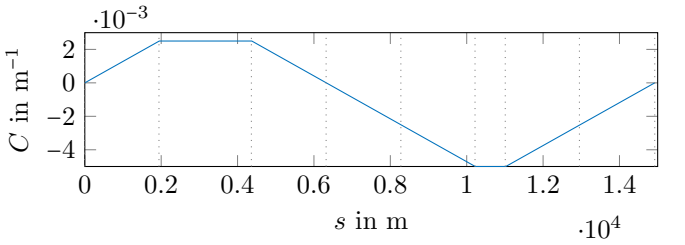
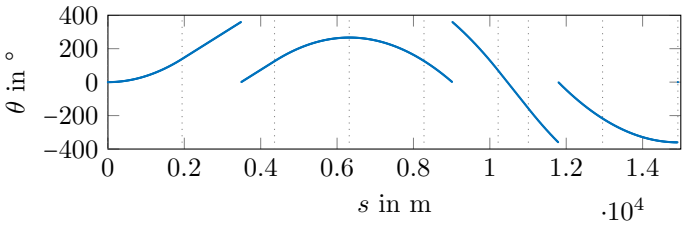


Fig. 2. Test Track (crosses indicate segment ends; $s = 0$ at $(0, 0)$, increasing eastwards).



(a) Curvature C along the track.



(b) Track angle θ along the track.

Fig. 3. Properties of the test track.

demanding, but still comfortably drivable track. In order to reduce conservatism, we divide the track into eight segments with boundaries as indicated by red crosses in Fig. 2 and dotted vertical lines in Fig. 3a, respectively. For all sections, if not mentioned otherwise, the reference speed as well as the maximum allowable speed are set to $v_{x,\max} = 100\,\mathrm{km\,h^{-1}}$, while $v_{x,\min} = 0\,\mathrm{km\,h^{-1}}$ and $-18\,\mathrm{km\,h^{-1}} \le v_y \le 18\,\mathrm{km\,h^{-1}}$. For reasons of comfort and safety, longitudinal and lateral accelerations are limited to $\sqrt{a_{x,\mathrm{b}}^2 + a_{y,\mathrm{b}}^2} \le \sqrt{2} \cdot 3\,\mathrm{m\,s^{-2}}$, which is inner-approximated by requiring that $-3\,\mathrm{m\,s^{-2}} \le a_{i,\mathrm{b}} \le 3\,\mathrm{m\,s^{-2}}$, $i \in \{x, y\}$.

Based on these constraints, for each segment, a set $\underline{\mathcal{Z}}$ is computed according to Sec. 3.3. Projections of the resulting sets on the $\dot{s}$ - u_t and $\dot{s}$ - u_n - space are shown in Fig. 4, in which dotted lines denote the inner approximations (where none are visible, these coincide with the bounds of the original set). Note that the inner approximation can be obtained by inner-approximating the projections of the set because the accelerations are independent of each other. The obtained constraints $\underline{\mathcal{Z}}$ are then used to define model $\mathcal{M}_3$ in each segment. Depending on the segment in which the vehicle is predicted to be located in, different constraints are activated, which is implemented using binary variables and the Big-M-method, cf. (Williams, 1990). In the following, the equations of motion and constraints of two different models from Sec. 3.4 will be used when solving Problem 1: Model $\mathcal{M}_3$, which is the result of the proposed method, and model $\mathcal{M}_4$, which represents a naive approach to the planning problem.

4.1 Scenario 1

We first demonstrate that solving Problem 1 based on the proposed method, i.e., model $\mathcal{M}_3$, allows excessive lateral motion on curved roads without violating constraints, while the naive model $\mathcal{M}_4$ leads to immediate failure. To that end, we impose a sinusoidal reference trajectory for the lateral motion of the vehicle while traversing the northern circle. In order to follow that motion, lateral inputs as shown in Fig. 5 are applied, which correspond to lateral accelerations as given in Fig. 6. Clearly, the inputs as determined by both methods are admissible. However, in the naive approach, the resulting accelerations violate the imposed constraints as indicated by the gray lines. The proposed method, in contrast, yields admissible accelerations due to the employed state-dependent input constraints. Certainly, admissible accelerations could also be obtained based on the naive model $\mathcal{M}_4$ by introducing speed limits. However, as the next scenario shows, constant speed limits may not be sufficient to guarantee constraint compliance.

4.2 Scenario 2

Secondly, we focus on the aspect of varying curvature, while the vehicle keeps a constant lateral position during its journey along the track in Fig. 2. As the previous section has demonstrated, failure of the naive approach on roads with high curvature is foreseeable if no speed limits are present. Therefore, in order to give the naive approach a chance, the maximal admissible longitudinal velocity is reduced to $70\,\mathrm{km\,h^{-1}}$ in the smaller circle of the track and in the first segments preceding and following it. For the proposed approach, it is not necessary to enforce this constraint, because one can rely on its ability to automatically choose admissible speeds.

The longitudinal input trajectories resulting from both planning procedures are given in Fig. 7. Use of both models leads to feasible inputs, i.e., the input constraints are not violated, which are constant in the case of the naive planner and vary with the speed in the proposed method. The longitudinal and lateral accelerations resulting for the vehicle from these inputs are given in Fig. 8. Despite overall similar appearance, subtle but crucial differences demonstrate the value of our approach: Some longitudinal accelerations are not admissible in the naive case (cf. markings). The proposed method, in contrast, automatically limits the longitudinal inputs as shown in Fig. 7 at the critical instances, leading to admissible accelerations.

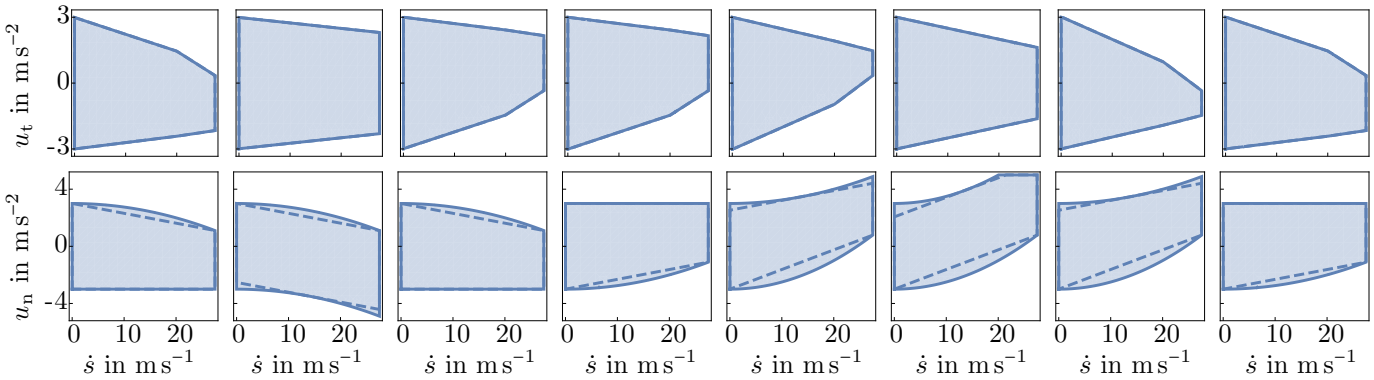
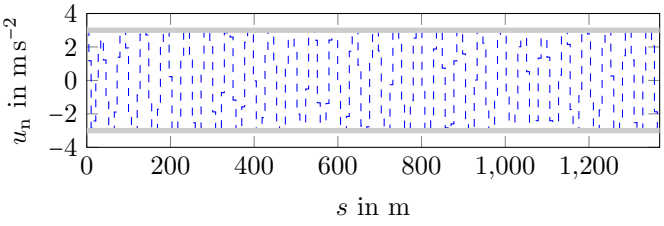
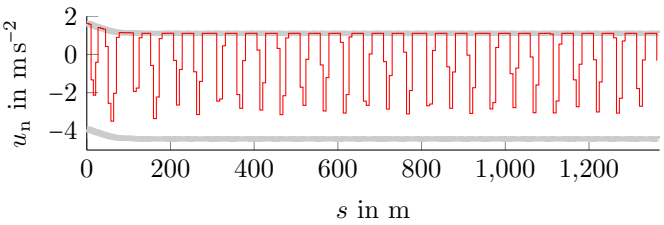


Fig. 4. Projections of $\tilde{\underline{\mathcal{Z}}}$ and $\underline{\mathcal{Z}}$ on the $\dot{s}$-u_t-plane (top) and on the $\dot{s}$-u_n-plane (bottom).



(a) Based on model $\mathcal{M}_4$ (Naive approach).



(b) Based on model $\mathcal{M}_3$ (Proposed method).

Fig. 5. Scenario 1: Input u_n.

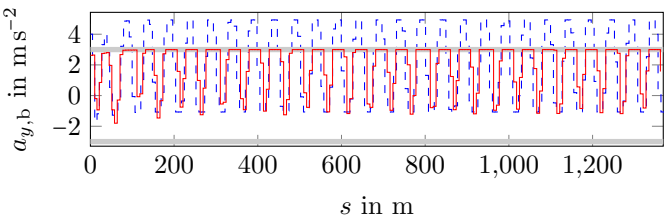
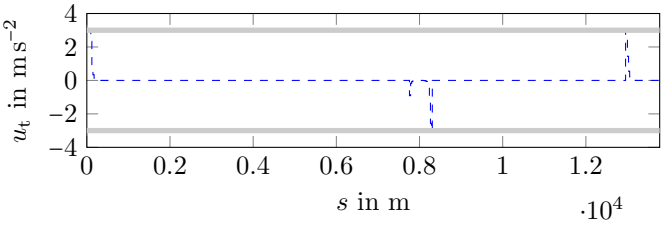


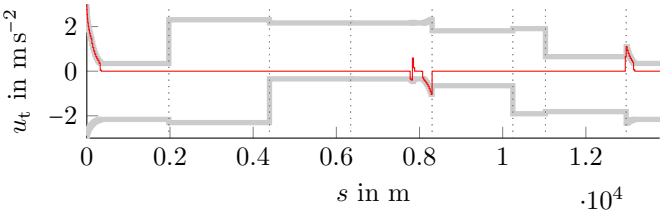
Fig. 6. Scenario 1: accelerations $a_{y,b}$ (dashed blue: naive approach; solid red: proposed method; thick grey: constraints).

5. CONCLUSION

While planning based on the naive model $\mathcal{M}_4$ violates the acceleration constraints in both scenarios, it can be noted that the degree of violation is relatively small. Also, one could argue that $\mathcal{M}_4$ could certainly return admissible accelerations by tuning of its cost function or by introducing additional constraints (such as further reducing the maximal speed). While in human driving, speed limits are the standard mean to prevent dynamically infeasible situations, this is not sufficient for autonomous vehicles, because traveling at a prescribed speed does not *imply* compliance with dynamical constraints, which also depends on the other states of the vehicle and the

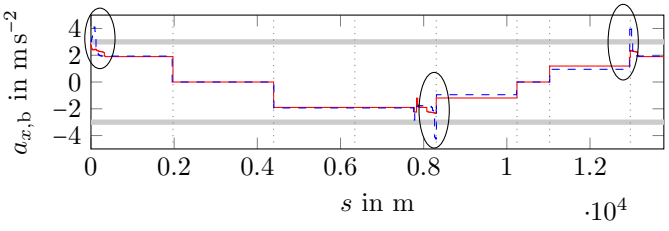


(a) Based on model $\mathcal{M}_4$ (Naive approach).

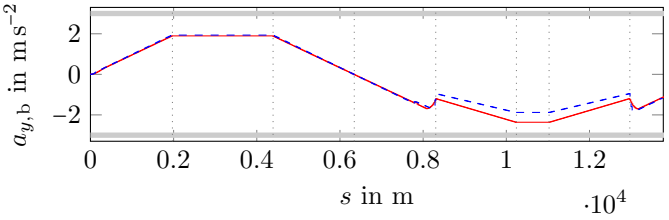


(b) Based on model $\mathcal{M}_3$ (Proposed method).

Fig. 7. Scenario 2: Input u_t.



(a) Longitudinal acceleration $a_{x,b}$.



(b) Lateral acceleration $a_{y,b}$.

Fig. 8. Scenario 2: accelerations in the body-fixed frame (dashed blue: Naive approach; solid red: proposed method).

chosen inputs. In order to select these inputs properly, a human driver relies on experience and typically tries to not operate the vehicle at the bounds of the admissible region. This is not viable for autonomous vehicles, whose

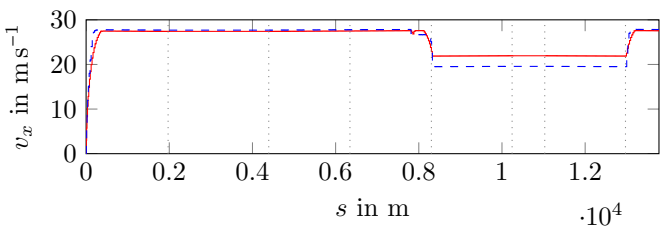


Fig. 9. Resulting velocities in the body-fixed frame (dashed blue: naive approach; solid red: proposed method).

planning algorithms do not have human experience. Also, an optimization-based planner will often find an optimum on the boundary of a given admissible set, not leading to cautious driving. Therefore, the tuning effort may be significant.

In addition, tuning goes along with significant testing effort in order to reduce the likelihood of failure, without being able to ever completely exclude it. Our method, in contrast, is able to *guarantee* compliance with the original constraints based on a theoretically sound approach. While it is conservative, constraints tailored to a certain vehicle can still lead to less conservative behavior than simple speed limits imposed on a road segment, which do not account for the dynamics of a certain vehicle. This becomes clear when considering the velocity trajectories resulting from both planning procedures as given in Fig. 9: while the naive planner reduces the vehicle's longitudinal velocity according to the speed limits, the planner based on the proposed method does not have to account for a global speed limit. Thus, it only reduces the vehicle's speed based on its speed-dependent input constraints, leading to overall less reduction, while still complying with the acceleration constraints (unlike the more conservative naive approach).

REFERENCES

Blundell, M. and Harty, D. (2004). *Multibody systems approach to vehicle dynamics*. Elsevier.

Burger, C. and Lauer, M. (2018). Cooperative multiple vehicle trajectory planning using miqp. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 602–607. IEEE.

Eilbrecht, J. and Stursberg, O. (2018). Optimization-based maneuver automata for cooperative trajectory planning of autonomous vehicles. In *European Control Conf.*, 82–88.

González, D., Pérez, J., Milanés, V., and Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4), 1135–1145.

Heck, A. (1993). *Introduction to MAPLE*. Springer.

Hess, D., Lattarulo, R., Pérez Rastelli, J., Schindler, J., Hesse, T.W., and Köster, F. (2018). Fast maneuver planning for cooperative automated vehicles. In *Intelligent Transportation Systems Conference*. IEEE.

Khalil, H. (1996). Noninear systems. *Prentice-Hall*.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press.

Le Guernic, C. (2009). *Reachability analysis of hybrid systems with linear continuous dynamics*. Ph.D. thesis, Université Joseph-Fourier-Grenoble I.

Lotov, A.V., Bushenkov, V.A., and Kamenev, G.K. (2013). *Interactive decision maps: Approximation and visualization of Pareto frontier*. Springer.

Nilsson, J., Falcone, P., Ali, M., and Sjöberg, J. (2015). Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice*, 41, 124–133.

Pacejka, H. (2005). *Tire and vehicle dynamics*. Elsevier.

Perantoni, G. and Limebeer, D.J. (2014). Optimal control for a formula one car with variable parameters. *Vehicle System Dynamics*, 52(5), 653–678.

Qian, X., Altché, F., Bender, P., Stiller, C., and de La Fortelle, A. (2016). Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective. In *Intelligent Transportation Systems Conf.*, 205–210. IEEE.

Schürmann, B., Heß, D., Eilbrecht, J., Stursberg, O., Köster, F., and Althoff, M. (2017). Ensuring drivability of planned motions using formal methods. In *Intelligent Transportation Systems Conf.*, 1–8. IEEE.

Weber, A. and Reissig, G. (2014). Classical and strong convexity of sublevel sets and application to attainable sets of nonlinear systems. *J. on Control and Optimization*, 52(5), 2857–2876.

Williams, H.P. (1990). *Model building in mathematical programming*. Wiley.

Wolfram, S. (1991). *Mathematica: a system for doing mathematics by computer*. Addison-Wesley.

Xue, B., She, Z., and Easwaran, A. (2016). Under-approximating backward reachable sets by polytopes. In *Int. Conf. on Computer Aided Verification*, 457–476. Springer.

Ziegler, J., Bender, P., Dang, T., and Stiller, C. (2014). Trajectory planning for bertha—a local, continuous method. In *Intelligent Vehicles Symp.*, 450–457. IEEE.