European Journal of Control 74 (2023) 100865

Contents lists available at ScienceDirect

European Journal of Control

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

A real-time collision-free maneuver generation algorithm for autonomous driving

Francesco Laneve[a,b,*], Alessandro Rucco[b], Massimo Bertozzi[a]

[a] Dipartimento di Ingegneria e Architettura, Universitá di Parma, Parma, Italy
[b] VisLab srl, an Ambarella Inc. Company, Parma, Italy

ARTICLE INFO

Article history:
Received 15 May 2023
Accepted 8 June 2023
Available online 16 June 2023

Recommended by Prof. T Parisini

Keywords:
Collision avoidance
Trajectory optimization
Autonomous driving

ABSTRACT

In this paper we propose a real-time maneuver generation algorithm for Autonomous Vehicles (AVs). Given a planar road geometry with static and moving obstacles along it, we are interested in finding collision-free maneuvers that satisfied the AV's dynamics and subject to physical and comfort limits. Based on longitudinal and transverse coordinates, we propose a novel collision avoidance constraint and formulate a suitable maneuver regulation optimal control problem. Maneuver regulation has intrinsic robustness with respect to standard trajectory tracking that stems from the requirement of following a desired path with a desired velocity profile assigned on it. The optimization problem is solved by using a nonlinear optimal control technique that generates (local) optimal trajectories. We demonstrate the efficacy of the proposed algorithm by providing numerical computations on two different scenarios. Finally, experimental results are presented to demonstrate the efficiency of the proposed algorithm both in terms of computational effort and dynamic features captured.

1. Introduction

Autonomous Vehicles (AVs) have the potential to revolutionize the transportation sector by improving safety, reducing travel time, and lowering energy consumption. A critical, safe related, aspect is the AVs' ability to avoid dynamic obstacles, such as human-driven vehicles or pedestrians. To achieve this task, the AV needs to predict their future intentions (motion forecasting module), generate a feasible trajectory to avoid them (high-level planning module), and track or follow the planned trajectory (low-level controller module). In this paper, we focus on the generation of feasible trajectories and propose a novel algorithm based on nonlinear optimal control techniques.

Several approaches have been proposed in the literature to solve the collision-free trajectory generation problem, see [10] for a survey. In the last years, optimization-based methods have received increasing attention thanks to the availability of efficient numerical techniques, as, e.g., [1,8,9,18,26], and the possibility to systematically include vehicle dynamics and suitable constraints in the formulation. The design of optimal collision-free trajectories is challenging mainly because the collision avoidance constraint is non-convex and computationally difficult to handle, see, e.g., [7]. For these reasons, appropriate approximations are introduced. For example, in Turri et al. [24], a linear time-varying constraint on the lateral displacement is proposed. In [19], a non-convex (ellipse-shaped) constraint is taken into account. In [28], an exact reformulation of the collision avoidance constraint is proposed. However, such a reformulation implies the introduction of additional optimization variables and constraints. Once the avoidance constraint has been formulated, it can be embedded into (i) a constrained optimization problem, as in Laneve et al. [17], Spedicato and Notarstefano [23], Zanon et al. [27], or (ii) an appropriate unconstrained optimization problem in which the objective function is augmented with an artificial potential field representing the distance to the obstacle, see e.g., [15,25]. With the collision avoidance formulation in hand, a classical approach used in the literature to design a trajectory is trajectory tracking, aiming to force the vehicle to reach and track a time-parametrized path. However, the requirement of tracking a time-parametrized reference has the following main limitation: poor tracking of the reference path in presence of external disturbances and unmodeled dynamics, see, e.g., [21,22]. An alternative approach is maneuver-regulation, Hauser and Hindman [12], namely, converge and follow a desired path with the additional requirement to satisfy a velocity profile along it. We refer to Aguiar et al. [2] for a discussion on

* Corresponding author at: Dipartimento di Ingegneria e Architettura, Universitá di Parma, Parma, Italy.
E-mail address: francesco.laneve@unipr.it (F. Laneve).

https://doi.org/10.1016/j.ejcon.2023.100865
0947-3580/

trajectory tracking and maneuver regulation. In this paper, we take a maneuver regulation perspective to generate real-time collision-free trajectories.

The contributions of the paper are as follows. First, we propose a novel avoidance constraint formulation suitable for the maneuver regulation approach. Specifically, we rewrite the system dynamics in terms of longitudinal and transverse coordinates and use the longitudinal coordinate as the independent variable instead of the time. Then, we embed the time variable into the problem formulation and propose a new collision avoidance constraint with respect to the new set of variables. Second, we propose a real-time optimal control-based strategy to generate collision-free maneuvers. The strategy is based on the constrained PRojection Operator Newton method for Trajectory Optimization (PRONTO), Aguiar et al. [1]. PRONTO has the advantage of ensuring the recursive feasibility (in both dynamics and constraints) of intermediate solutions. This is a fundamental aspect for AVs when generating feasible trajectories. Indeed, by using PRONTO, one can reliably use sub-optimal intermediate solutions in cases where quick decisions are required for safe tasks (e.g., collision avoidance maneuvers). This capability is not available in other solvers, as, e.g., GPOPS II, Patterson and Rao [18], as both the dynamics and constraints are only loosely satisfied until the final solution is reached. We provide numerical computations to show the effectiveness of the proposed algorithm. Finally, we integrate our algorithm into an autonomous driving stack developed by Ambarella Inc.,[1] thus further highlighting that the proposed algorithm is effective in a real-world scenario.

The approaches most closely related to our problem setup and constraints formulation are the ones proposed in Rosolia et al. [19], Spedicato and Notarstefano [23], Zanon et al. [27]. Next, we highlight the main differences. In [19], a maneuver regulation perspective is adopted for generating collision-free trajectories. The independent variable of the optimization problem is the time and a suitable approximation (based on Bezier curves) is introduced in order to describe the road geometry (i.e., centerline of the road, road boundaries, target velocity). In contrast to the previous approach, we tackle the optimization problem using the longitudinal coordinate as the independent variable and thus we do not resort to approximations for the (reformulated) vehicle system. In [23,27], the model dynamics is described using the curvilinear coordinate. In order to generate obstacle collision-free trajectory, the road-boundaries (which are functions of the curvilinear coordinate) are re-shaped by taking into account the obstacles' position and size. However, such a constraint formulation becomes too conservative when dealing with moving obstacles, mainly because the time variable is neglected. On the contrary, we embed the time variable into vehicle dynamics and propose the collision avoidance constraint with respect to the new set of variables, thus handling both static and dynamic obstacles.

This paper is organized as follows. In Section 2, we describe the system dynamics with respect to the new set of coordinates and formulate the collision-free avoidance constraint. In Section 3, we describe our real-time strategy to compute collision-free maneuvers. Finally, in Section 4 and Section 5, we provide numerical computations and an experimental test, highlighting the main features of the proposed strategy.

2. Problem formulation

In this section we describe the AV model (from now on, referred as the ego-vehicle), re-write the system dynamics with respect to the longitudinal and transverse coordinates and formulate the collision avoidance constraint.

[1] https://www.ambarella.com/applications/automotive/

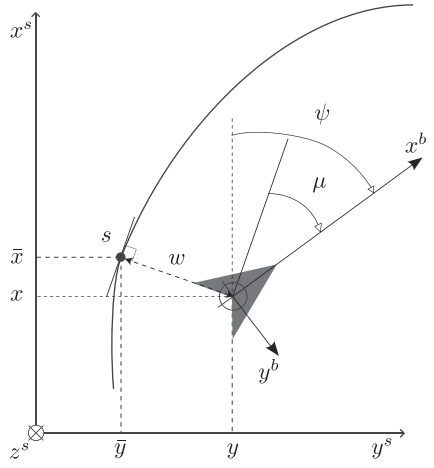

Fig. 1. Local coordinates around the geometry path. The bold triangle indicates the ego-vehicle. The solid line indicates the center-line of the lane.

2.1. Ego-vehicle model

In typical urban scenarios, where low acceleration values are applied, the 2D motion of the ego-vehicle can be described by

$$\begin{aligned}\dot{x} &= v\cos\psi\\ \dot{y} &= v\sin\psi\\ \dot{\psi} &= v\kappa,\\ \dot{v} &= a,\end{aligned} \tag{1}$$

where (x, y) are the longitudinal and lateral coordinates with respect to the inertial frame, ψ is the heading angle, and v is the velocity. The control inputs are the curvature κ and the acceleration a. For urban scenarios, such a (simple) vehicle model has comparable accuracy with a dynamic one, see, e.g., [16].

2.2. Longitudinal and transverse coordinates

We define a new set of coordinates: the longitudinal coordinate s represents the position along the lane center-line and the lateral coordinate w denotes the displacement transverse to the center-line, see Fig. 1. Given the road geometry, we assume that the path has a reasonably smooth (at least C^2) arc-length parametrized center-line, $(\bar{x}_{cl}(s), \bar{y}_{cl}(s))$. The course heading $\bar{\psi}_{cl}(s)$ and the curvature $\bar{\kappa}_{cl}(s)$ are related by differentiation:

$$\begin{aligned}\bar{x}'_{cl}(s) &= \cos\bar{\psi}_{cl}(s)\\ \bar{y}'_{cl}(s) &= \sin\bar{\psi}_{cl}(s)\\ \bar{\psi}'_{cl}(s) &= \bar{\kappa}_{cl}(s),\end{aligned} \tag{2}$$

where the bar symbol indicates that the variable is expressed as a function of the longitudinal coordinate s, and the prime symbol denotes differentiation with respect to s. Using the arc-length parametrization, the coordinates of the ego-vehicle can be defined as follows:

$$\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\bar{x}_{cl}(s)\\ \bar{y}_{cl}(s)\end{bmatrix} + R_z(\bar{\psi}_{cl}(s))\begin{bmatrix}0\\ w\end{bmatrix}, \tag{3}$$

where

$$R_z(\bar{\psi}_{cl}(s)) = \begin{bmatrix}\cos\bar{\psi}_{cl}(s) & -\sin\bar{\psi}_{cl}(s)\\ \sin\bar{\psi}_{cl}(s) & \cos\bar{\psi}_{cl}(s)\end{bmatrix}$$

is the rotation matrix transforming vectors from the velocity frame into the inertial frame.

Next, we describe the ego-vehicle position with respect to the (s, w) coordinates. Following the calculations in Bayer and Hauser

[4], see also [20], we differentiate (3) with respect to time and, by using Eqs. (1) and (2), we have

$$\begin{aligned}\dot{s} &= \frac{v\cos\mu}{1-w\bar{\kappa}_{cl}(s)}\\ \dot{w} &= v\sin\mu\\ \dot{\mu} &= v\kappa - \bar{\kappa}_{cl}(s)\dot{s},\end{aligned} \tag{4}$$

where $\mu = \psi - \bar{\psi}_{cl}(s)$ is the local heading error.

Remark II.1: The inverse of the map $(s, w) \mapsto (x, y)$ is well-defined if $1 - w\bar{\kappa}_{cl}(s) > 0$, i.e. when the ego-vehicle position is inside a tube around the center-line of the lane.

The nonlinear system (1) with respect to the new sets of coordinates is

$$\begin{aligned}\dot{s} &= \frac{v\cos\mu}{1-w\bar{\kappa}_{cl}(s)}\\ \dot{w} &= v\sin\mu\\ \dot{\mu} &= v\kappa - \bar{\kappa}_{cl}(s)\dot{s}\\ \dot{v} &= a\,.\end{aligned} \tag{5}$$

Finally, we rewrite (5) by using the longitudinal coordinates s as the independent variable. Let us denote with $\bar{t}(s)$ the inverse of $s(t)$, which satisfies $\bar{t}(s(t)) = t$. Given that $s(t)$ is invertible, a generic function of time $\alpha(t)$ can be expressed as a function of s, i.e. $\alpha(\bar{t}(s))$. By defining $\bar{\alpha} = \alpha(\bar{t})$, we have $\alpha(t) = \bar{\alpha}(s(t))$ and thus, in our case, it holds the following equivalences:

$$\begin{aligned}&t = \bar{t}(s), \quad w = \bar{w}(s), \quad \mu = \bar{\mu}(s),\\ &v = \bar{v}(s), \quad k = \bar{k}(s), \quad a = \bar{a}(s)\,.\end{aligned}$$

By differentiating with respect to time, for example, $w = \bar{w}(s)$, we see that $\dot{w} = \bar{w}'(s)\dot{s}$, so that now variables depend on time only through $s(t)$. In this way, we can obtain a description of the dynamics as function of the longitudinal coordinates. Formally, we get:

$$\begin{aligned}\bar{w}'(s) &= (1-\bar{\kappa}_{cl}(s)\bar{w}(s))\tan\bar{\mu}(s)\\ \bar{\mu}'(s) &= \frac{1-\bar{\kappa}_{cl}(s)\bar{w}(s)}{\cos\bar{\mu}(s)}\bar{\kappa}(s) - \bar{\kappa}_{cl}(s)\\ \bar{v}'(s) &= \frac{1-\bar{\kappa}_{cl}(s)\bar{w}(s)}{\bar{v}(s)\cos\bar{\mu}(s)}\bar{a}(s)\\ \bar{t}'(s) &= \frac{1-\bar{\kappa}_{cl}(s)\bar{w}(s)}{\bar{v}(s)\cos\bar{\mu}(s)},\end{aligned} \tag{6}$$

which requires that the ego-vehicle moves with $\bar{v} > 0$ and $|\bar{\mu}| < \pi/2$. For the sake of notational simplicity, from now on we will neglect the explicit dependence from s if it is clear from the context. In (6) the state and the control vector are $\bar{\mathbf{x}} = [\bar{w},\ \bar{\mu},\ \bar{v},\ \bar{t}\,]$ and $\bar{\mathbf{u}} = [\bar{\kappa},\ \bar{a}]$ respectively. We highlight that we include $\bar{t}$ into the dynamics. Such additional state variable will be exploited for the formulation of the avoidance constraint, as will be shown in the next subsection.

2.3. Obstacle avoidance formulation

A typical obstacle avoidance constraint formulation consists of imposing a security distance, $\tilde{d}$, between the ego-vehicle and the obstacle, see, e.g., [13]. Such a security distance takes into account the size of the ego-vehicle and the obstacle, as well as a safety margin. Specifically, given the coordinate center of the obstacle and its future predictions, $(x_{obs}(\cdot), y_{obs}(\cdot))$, we have that the following inequality constraint

$$\left(\frac{x - x_{obs}}{\tilde{d}}\right)^2 + \left(\frac{y - y_{obs}}{\tilde{d}}\right)^2 \geq 1 \tag{7}$$

must be satisfied for all times t. In order to include such a constraint in our maneuver regulation problem, we first re-write

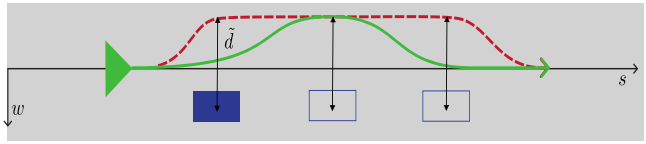

Fig. 2. Avoidance maneuver scenario. The ego-vehicle (green triangle), the obstacle (solid blue rectangle) and its predictions (empty blue rectangles) are shown. The avoidance maneuvers obtained by using (11) and (12) are depicted, respectively, in dashed red line and solid green line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(7) with respect to the (s, w) coordinates. Given the longitudinal coordinate s of the ego-vehicle, we can express the obstacle position as

$$\begin{bmatrix} x_{obs} \\ y_{obs} \end{bmatrix} = \begin{bmatrix} \bar{x}_{cl} \\ \bar{y}_{cl} \end{bmatrix} + R_z(\bar{\psi}_{cl}) \begin{bmatrix} s_{obs} - s \\ w_{obs} \end{bmatrix}, \tag{8}$$

where s_{obs} and $w_{obs}(s)$ are the longitudinal and lateral coordinates, respectively, of the obstacle. Next, by subtracting (8) from (3), we have

$$\begin{bmatrix} x - x_{obs} \\ y - y_{obs} \end{bmatrix} = R_z(\bar{\psi}_{cl}) \begin{bmatrix} s - s_{obs} \\ w - w_{obs} \end{bmatrix}. \tag{9}$$

Substituting (9) in (7), we get the following equivalent formulation

$$\left(\frac{s - s_{obs}}{\tilde{d}}\right)^2 + \left(\frac{w - w_{obs}}{\tilde{d}}\right)^2 \geq 1\,. \tag{10}$$

By transforming (10) from a t-dependent to s-dependent description, the constraint can be written as

$$\left(\frac{\bar{w} - \bar{w}_{obs}}{\tilde{d}}\right)^2 \geq 1\,. \tag{11}$$

The avoidance formulation as in (11) captures the case of static obstacles: at a given curvilinear coordinate s, the relative (lateral) distance between the ego-vehicle and the obstacle must be greater than $\tilde{d}$. However, it is very conservative (or, in some case, it fails) when dealing with moving obstacles. For example, let us consider the scenario illustrated in Fig. 2: the ego-vehicle is moving along a lane (with $w = 0$) and has to avoid a moving obstacle which is on the right-side of the center-line. By using the avoidance formulation (11), the ego-vehicle will satisfy (if possible) the security distance $\tilde{d}$ for all s, thus providing a too conservative collision-free maneuver. This is due to the fact that the formulation (11) does not take into account the time evolution of the ego-vehicle and the obstacle. In order to overcome this limitation, we propose a novel constraint formulation as follows. We start from the following (somehow) simple idea: in order to avoid collision with a moving obstacle, the ego-vehicle and the obstacle must not be at the same longitudinal coordinate s at the same time instant t. Based on this idea, we introduce the time variable into (11), as follows:

$$\left(\frac{\bar{t} - \bar{t}_{obs}}{\tilde{t}}\right)^2 + \left(\frac{\bar{w} - \bar{w}_{obs}}{\tilde{d}}\right)^2 \geq 1, \tag{12}$$

where $\bar{t}_{obs}$ is the time at which the obstacle is at the longitudinal coordinates s and $\tilde{t}$ is a temporal safety margin parameter. In Fig. 3 we provide a 3D representation of the proposed avoidance constraint. We point out that the inequality constraint (12) is based on the coordinate center of the obstacle. Such formulation can be easily extended in order to take into account the obstacle's size (i.e., by using the projection of the rear and front axes along the road geometry). *Remark II.2:* It is worth noting that the inequality constraint (12) is well-defined if unique values of $\bar{w}_{obs}$ and $\bar{t}_{obs}$ are determined for a given s. This is not the case for a static obstacle,

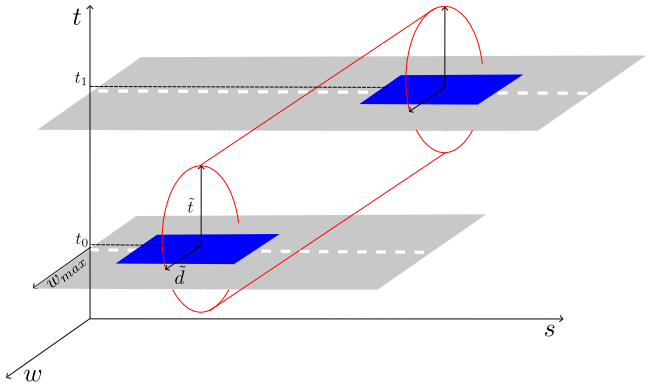

Fig. 3. Obstacle avoidance constraint representation. The lane, the obstacle and the ellipse constraint are depicted in gray, blue, and red, respectively. In order to satisfy the constraint, the AV must be outside the red boundaries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

because it has the same longitudinal coordinate s for all time values. For this reason, when dealing with static obstacles, we impose a sufficient large value for $\tilde{t}$.

3. Real-time maneuver generation algorithm

In this section, we describe the optimal control-based strategy used to compute real-time collision-free maneuvers.

First, we define additional constraints. Specifically, the ego-vehicle is required to satisfy the road boundaries. This constraint assumes a very simple form with respect to the new set of coordinates, that is,

$$|\bar{w}| \leq w_{\max}\,. \tag{13}$$

In order to take into account the operational limits of the kinematics model and the comfort of the passenger, we impose state and input constraints on (6) as follows. The velocity is bounded by two constants, i.e.,

$$v_{\min} \leq \bar{v} \leq v_{\max}\,. \tag{14}$$

while the longitudinal acceleration a and the lateral acceleration, $v^2\kappa$, are coupled by the ellipse constraint, [14],

$$\left(\frac{2\bar{a} - (a_{\max} + a_{\min})}{(a_{\max} - a_{\min})}\right)^2 + \left(\frac{\bar{v}^2\bar{\kappa}}{a_{lat_{\max}}}\right)^2 \leq 1\,. \tag{15}$$

Moreover, in order to take into account the limited wheel steer angle, the curvature is bounded in module as follows,

$$|\bar{\kappa}| \leq \kappa_{\max}\,. \tag{16}$$

Second, we need to define the cost function to be optimized. Specifically, we are interested to follow the center-line of the road with a given velocity profile, $\bar{v}_{cl}$, assigned on it, and, at the same time, minimize the control effort. Such a behavior can be captured by minimizing the following cost function,

$$J(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = (\bar{\mathbf{x}} - \bar{\mathbf{x}}^d)^T\, Q\, (\bar{\mathbf{x}} - \bar{\mathbf{x}}^d) + (\bar{\mathbf{u}} - \bar{\mathbf{u}}^d)^T\, R\, (\bar{\mathbf{u}} - \bar{\mathbf{u}}^d), \tag{17}$$

where $\bar{\mathbf{x}}^d = [0, 0, \bar{v}_{cl}, 0]$, $\bar{\mathbf{u}}^d = [\bar{\kappa}_{cl}, 0]$ is the desired maneuver, $Q = diag(q_1, q_2, q_3, q_4)$ are a positive-semidefinite matrix, while $R = diag(r_1, r_2)$ is positive-definite one. Now we are ready to formulate the optimal control problem:

$$\begin{aligned} \min_{\bar{\mathbf{x}}(\cdot)\,\bar{\mathbf{u}}(\cdot)} \quad & \int_0^{s_f} J(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau + m(\bar{\mathbf{x}}(s_f)) \\ \text{s.t.} \quad & \bar{\mathbf{x}}'(s) = f(\bar{\mathbf{x}}(s), \bar{\mathbf{u}}(s), s), \bar{\mathbf{x}}(0) = \mathbf{x}_0 \\ & h(\bar{\mathbf{x}}(s), \bar{\mathbf{u}}(s)) \leq 0, \end{aligned} \tag{18}$$

where $s_f > 0$ is a fixed horizon, $\bar{\mathbf{x}}' = f(\bar{\mathbf{x}}, \bar{\mathbf{u}}, s)$ describes the nonlinear Eq. (6), $h(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ are the state/input constraints (12)–(16), $J(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ is the stage cost as in (17) and $m(\bar{\mathbf{x}}(s_f))$ is the terminal cost which minimizes the L2 distance between the ego-vehicle state and the desired end state $\bar{\mathbf{x}}(s_f)$. We solve (18) by using PRONTO, [1]. PRONTO is a direct method for solving continuous-time optimal control problems. It exhibits a second-order convergence rate to a local minimizer satisfying second-order sufficient conditions of optimality. Constraints are handled using a barrier function approach. In particular, the state-input constraints are relaxed by adding them to the cost functional. A barrier function can be defined as

$$b_\delta(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \int_0^{s_f} \sum_j \beta_\delta(-h_j((\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))))d\tau,$$

where

$$\beta_\delta(z) = \begin{cases} -\log z & \text{for } z > \delta \\ \frac{1}{2}\left(\frac{z-2\delta}{\delta} - 1\right) - \log\delta & \text{for } z \leq \delta \end{cases}.$$

Using the barrier function β_δ, the problem (18) becomes:

$$\begin{aligned} \min_{\bar{\mathbf{x}}, \bar{\mathbf{u}}} \quad & \int_0^{s_f} J(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)) + \epsilon b_\delta(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau \\ \text{s.t.} \quad & \bar{\mathbf{x}}'(s) = f(\bar{\mathbf{x}}(s), \bar{\mathbf{u}}(s), s), \bar{\mathbf{x}}(0) = \mathbf{x}_0, \end{aligned} \tag{19}$$

for $\epsilon > 0$.

The strategy to find an approximated solution to (18) can be summarized as follows. Starting with a reasonable large ϵ and δ, Problem (19) is iteratively solved by reducing the parameters at each iteration and thus pushing the trajectory toward the constraint boundaries. PRONTO, being a Newton descent method, can only guarantee convergence to a local minimum. We choose the initial guess as follows. Given the reference state-input desired reference, $(\bar{\mathbf{x}}, \bar{\mathbf{u}})^d$, which is not a maneuver (it does not satisfy the dynamics), we use the projection operator [11] to project the state-input desired reference into the feasible maneuvers manifold to obtain a suitable initial guess. Now, with an initial maneuver for the initialization and a desired reference, the algorithm iterates the following step: i) compute the optimal collision-free maneuver by using PRONTO, ii) use the previous optimal maneuver as the initial guess for the next step, iii) update the constraints parameters, ϵ and β, iv) solve (19) with an updated barrier function. Algorithm 1

Algorithm 1 Real-time collision-free maneuver generation.

Input: road geometry $(\bar{x}_{cl}, \bar{y}_{cl}, \bar{\psi}_{cl}, \bar{\kappa}_{cl}, \bar{v}_{cl})$,
bounds $(w_{\max}, v_{\min}^{\max}, k_{\max}, a_{\min}^{\max}, a_{lat_{\max}})$,
obstacles data $(x_{obs}(t), y_{obs}(t))$
Initialization:

- compute $\mathbf{x}_0$ (i.e., project $(x, y, \psi)_0$ wrt the road geometry)
- setup dynamics $\bar{\mathbf{x}}' = f(\bar{\mathbf{x}}, \bar{\mathbf{u}}, s)$, $\mathbf{x}(0) = \mathbf{x}_0$
- compute $(\bar{w}_{obs}, \bar{t}_{obs})$ (i.e., project (x_{obs}, y_{obs}) w.r.t the road geometry)
- setup constraints $h(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ with bounds and $\bar{w}_{obs}$, $\bar{t}_{obs}$
- desired maneuver$\bar{\mathbf{x}}^d = [0, 0, \bar{v}_{cl}, 0]$, $\bar{\mathbf{u}}^d = [\bar{\kappa}_{cl}, 0]$

Set: $\epsilon = 1, \delta = 1$
for $k = 1, 2, \ldots$ **do**
compute: $(\bar{\mathbf{x}}, \bar{\mathbf{u}})_k = PRONTO((\bar{\mathbf{x}}, \bar{\mathbf{u}})_{k-1}, \epsilon, \delta)$
update ϵ, δ: $\epsilon \leftarrow \epsilon/6, \delta \leftarrow \delta/6$
end for
Output: $(\bar{\mathbf{x}}, \bar{\mathbf{u}})_{opt} = (\bar{\mathbf{x}}, \bar{\mathbf{u}})_k$

gives a pseudocode description of the real-time maneuver generation strategy. We want to stress that PRONTO ensures the recursive dynamics and constraints feasibility of intermediate solutions. As a

result, our algorithm can reliably use sub-optimal (intermediate) solutions in cases of overhead computations.

4. Numerical computations

In this section, we present numerical computations in order to show the effectiveness of both the proposed avoidance formulation and the proposed algorithm. We start our analysis with the following scenario: the AV is traveling along a straight path and an obstacle is moving with a low velocity on the right side of the lane. We refer to this scenario as the *lateral dynamic avoidance* case. Then, we consider the *longitudinal dynamic avoidance* case: a moving obstacle cuts-off the ego-vehicle's path. For both the scenarios, the dynamic obstacles are moving with constant velocity and their future positions are known (by simple integrating the velocity). We set the constraints parameters based on [16] and on driving experience, $w_{\max} = 1.25$ m, $v_{\min} = 0.1$ m/s, $v_{\max} = 19.4$ m/s, $a_{\min} = -1.5$ m/s^2, $a_{\max} = 1$ m/s^2, $\kappa_{\max} = 0.2$ m^{-1}, $a_{lat_{\max}} = 2.0$ m/s^2, $t_{safety} = 3$ s, $d_{safety} = 2.5$ m. We use a planning horizon of 100 m and a space discretization of 1 m.

4.1. Lateral dynamic avoidance maneuver

In the first scenario, the ego-vehicle is traveling with $v_0 = 50$ km/h (13.88 m/s) along a straight lane and has to avoid an obstacle moving on the same lane with a constant velocity of 20 km/h (about 5.55 m/s). After a trial and error process, we choose the following cost for the cost function: $q_1 = 0.1$, $q_2 = 0.1$, $q_3 = 1.0$, $q_4 = 0.0$, $r_1 = 100.0$ and, $r_2 = 0.1$. The optimal maneuver is shown in Figs. 4 and 5.

Next we highlight some important features of the computed maneuver. Specifically, we identify three phases. First, at the beginning, the ego-vehicle is following the reference path: the lateral displacement $\bar{w}$ is zero (Fig. 4(a)), and the velocity $\bar{v}$ is equal to the reference one (Fig. 4(a)). Second, the ego-vehicle executes a very smooth avoidance maneuver by applying first a negative curvature, Fig. 4(c), thus moving toward the left boundaries of the lane, Fig. 4(b), to avoid the obstacle. Note that the avoidance constraint, see Fig. 4(d), is active at about $s = 42$ m which is the curvilinear coordinate where the ego-vehicle comes alongside the obstacle, as depicted in Fig. 5. Third, once the ego-vehicle overtakes the obstacle, it goes back in following the center-line. It is interesting to note that the projection of the initial guess is infeasible for the avoidance constraint. Nevertheless, intermediate trajectories are all feasible, see dashed black lines in Fig. 4.

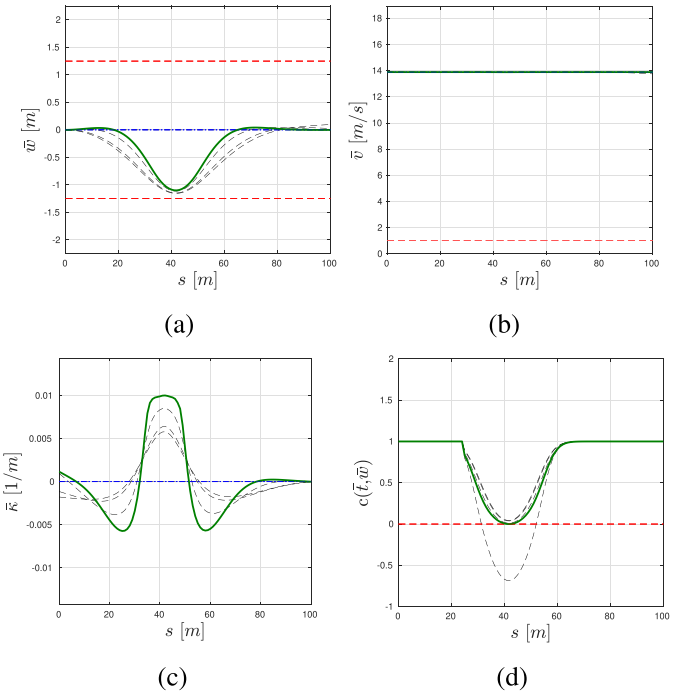

Fig. 4. Lateral avoidance scenario. The intermediate (dashed black lines) and optimal maneuvers (solid green line) are shown. The desired maneuver is depicted in dash-dotted blue line, while constraints are in dashed red line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Longitudinal dynamic avoidance maneuver

In the second scenario, a moving obstacle cuts off the ego-vehicle's path. Specifically, the ego-vehicle is traveling along a straight path with $v_0 = 13.9$ m/s (similarly in Fig. 5), the moving obstacle pulls out from the right-side of the path, at the longitudinal coordinate of $s = 40$ m and at $v_{obs} = 2.8$ m/s. The cost function weights (17) penalize deviation from center-line more than deviation to reference velocity: $q_1 = 10.0$, $q_2 = 10.0$, $q_3 = 0.1$, $q_4 = 0.0$, $r_1 = 100.0$ and, $r_2 = 0.1$. The optimal generated maneuver is depicted in Fig. 6.

At beginning of the maneuver, the ego-vehicle applies the minimum longitudinal acceleration, Fig. 6(c), in order to decrease its velocity, Fig. 6(b), and thus to let the obstacle pass. At about $s = 40$ m, the avoidance constraint becomes active, Fig. 6(d), and the

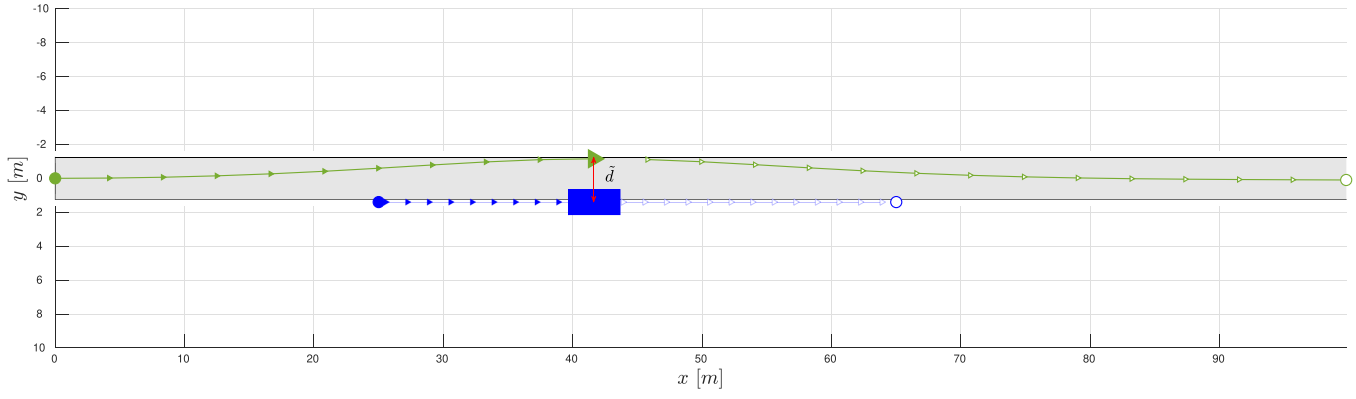

Fig. 5. Avoidance maneuver. The ego-vehicle (bold green triangle) and the obstacle (blue rectangle) are shown. The ego-vehicle and the obstacle trajectories are indicated with solid triangular green line and solid triangular blue line, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

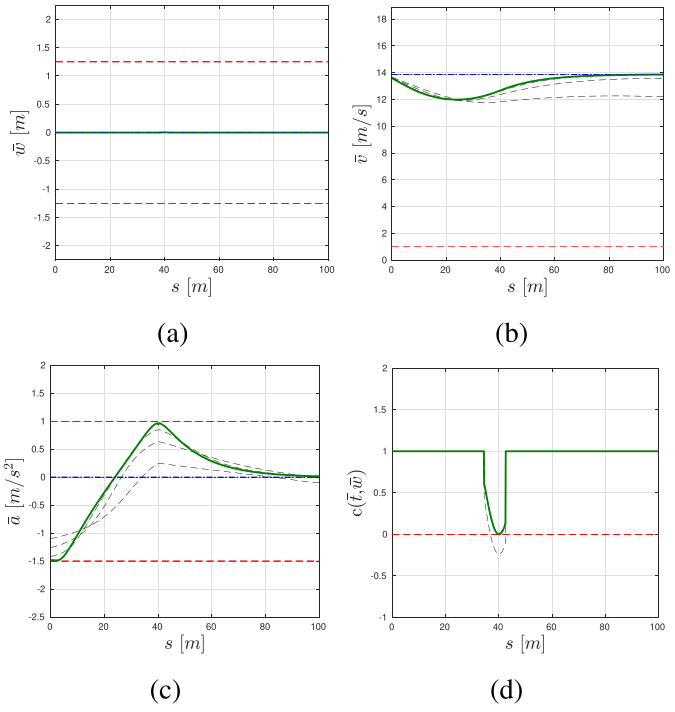

Fig. 6. Longitudinal avoidance scenario. The intermediate (dashed black lines) and optimal maneuvers (solid green line) are shown. The desired maneuver is depicted in dash-dotted blue line, while constraints are in dashed red line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ego-vehicle applies the maximum acceleration to reach the desired velocity. The lateral displacement, Fig. 6(a), is zero, as we expect for the longitudinal avoidance maneuver. Furthermore, we point out that only the initial trajectory is infeasible, see Fig. 6(d). Finally, we highlight that, by using the avoidance constraint (11) instead of (12), the optimal control problem has no (feasible) solution, thus confirming the importance of the proposed avoidance constraint formulation.

5. Experimental results

The proposed algorithm has been validated on a self-driving vehicle developed by VisLab. The experimental test has been carried out in Parma, Italy, on the campus area (urban roads) open to regular traffic. The main objective of the test is to demonstrate the efficacy of the proposed algorithm in generating real-time feasible maneuvers. We invite the reader to watch the video[2] attachment corresponding to the discussed test. Specifically, we embed our real-time maneuver generation algorithm into the planning module of the Autonomous Driving stack developed by Vislab. It is worth noting that the obstacles' predictions are provided by a motion forecasting module: at each time instant, the actual and the future obstacles' positions are given. For more details on the architecture of the autonomous vehicle system, we refer the interested reader to Broggi et al. [5,6]. The proposed algorithm has been implemented in c++ (in order to integrate the differential equations required by PRONTO, we use integrators based on *boost odeint*, Ahnert and Mulansky [3]), and it is applied in a receding horizon fashion during the experimental test. We use a fixed space horizon of 100 m, a space discretization of 1 m, and a replanning every 100 ms. The generated (optimal) collision-free maneuvers are used as reference trajectories for the low-level controller. In Fig. 7, we compare the actual lateral displacement, yaw-rate, and velocity of the actual ego-vehicle and the ones generated by our algorithm at the actual longitudinal coordinate s [m]. Next, we highlight three maneuvers generated by the proposed algorithm and successfully executed by the ego-vehicle.

First, at the beginning of the test, the ego-vehicle starts with zero velocity and accelerates smoothly (i.e., the acceleration constraint (15) is not active) in order to reach the desired velocity of 13.9 m/s. Along the straight lane, a bicycle (coming from a side road) crosses the road and starts riding on the right-side of the lane. As soon as the bicycle is detected, the ego-vehicle first decreases its velocity, then moves on the left side of the lane (thus avoiding the moving bicycle), and finally merge back to its own lane. It is worth noting that, during this dynamic avoidance maneuver, the constraint (12) is always satisfied, and the executed maneuver is very smooth as shown in the first highlighted section of Fig. 7. Second, after about 300 m, a static car parked along the right side of the lane is detected. As expected, the ego-vehicle avoids the static car in a smooth fashion as shown in the second highlighted section of Fig. 7. Again, the safety distance imposed by the avoidance constraints is satisfied and the ego-vehicle's velocity matches the one generated by the algorithm. Third, in the last highlighted section of Fig. 7, the ego-vehicle needs to avoid six parked cars and a pedestrian. Similarly to the previous case, the ego-vehicle successfully avoid the obstacles by tracking the generated maneuver.

Overall, we observe a good trajectory matching, even though some differences can be noticed. First, when the vehicle is driving in the roundabout (thus making a U-turn, refer to the video attachment), the ego-vehicle's position slightly deviates from the generated path, see Fig. 7(a) at about $s = 1100$ m, and the actual yaw-rate differs from the generate one, see Fig. 7(b). Since the low-level controller (which is out of the scope of this paper) is designed in order to penalize maneuvers close to the lane boundaries, the actual lateral displacement becomes lower than the reference one and the actual yaw rate becomes greater than the reference one (notice that the velocity is well-matched in this segment of the road). Similar behavior can be observed when the ego vehicle is executing a 90-degree turn (see Fig. 7 at about $s = 330$ m). Second, we observe that the actual velocity does not perfectly match the generated one in steady-state, see Fig. 7(c) for $v = 13.9$ m/s. Based on the generated velocity and acceleration profiles, the low-level controller generates the throttle (or braking) commands. However, it does not take into account aerodynamic and resistance forces and, consequently, the actual velocity becomes slightly lower than the generated one. In both cases, the actual trajectory satisfies the constraints and well-matches the generated one.

Finally, we provide the CPU time needed for the execution of the proposed algorithm. Specifically, the average computation time of the proposed algorithm in the case of no obstacles is 36 ms. For the avoidance scenarios (i.e., for the three highlighted zones in Fig. 7) the average computation time is 54 ms. These results confirm that the proposed trajectory generation algorithm allows one to compute feasible collision-free maneuvers with a computation times below 100 ms, thus enabling a real-time implementation at 10 Hz.

6. Conclusions

In this paper we address the problem of generating feasible trajectories for autonomous vehicles in presence of obstacles. We proposed a real-time strategy to compute optimal collision-free trajectories. In particular, we i) re-write the vehicle dynamics with respect to the transverse coordinates, ii) propose a novel avoidance constraint formulation and, iii) set up a maneuver

[2] https://youtu.be/x1glAcRP1TM

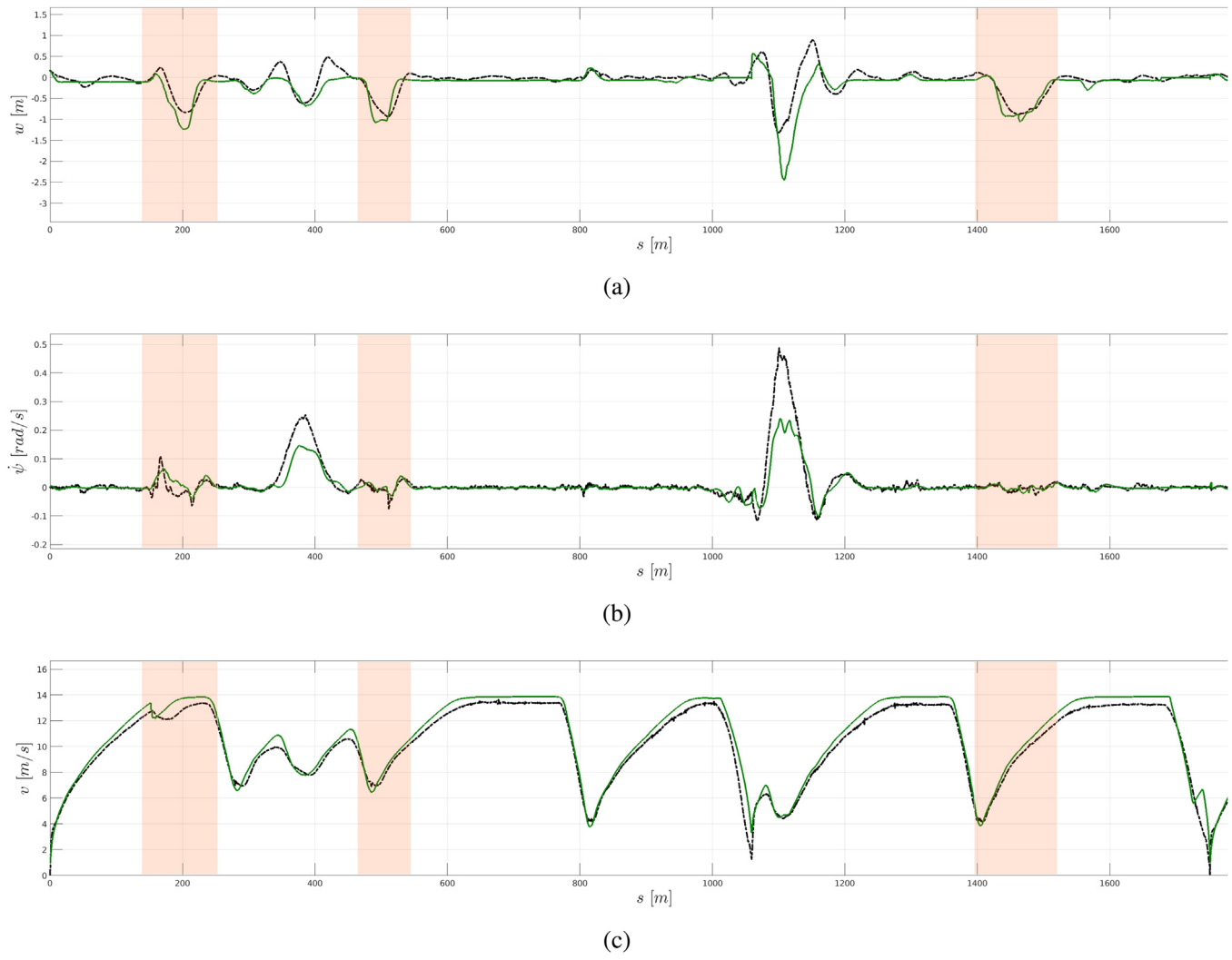

Fig. 7. Comparison between the reference maneuver (green solid line) generated by the proposed algorithm and actual maneuver of the autonomous vehicle (black dashed line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

regulation-based optimal control strategy in the transverse coordinates. We numerically solve the optimal control-based problem using PRONTO. We prove the effectiveness of the proposed approach in a simulated scenario and showed that the proposed constraint is able to avoid moving obstacles. Moreover, the optimal maneuvers are used as reference for a low-lever controller of a real vehicle. The integration of the proposed collision-free maneuver generation strategy on a real vehicle shows the feasibility of the computed maneuvers even in presence of unmodeled dynamic effects.

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.

References

[1] A.P. Aguiar, F.A. Bayer, J. Hauser, A.J. Häusler, G. Notarstefano, A.M. Pascoal, A. Rucco, A. Saccon, Constrained optimal motion planning for autonomous vehicles using PRONTO, in: Sensing and Control for Autonomous Vehicles, Springer, 2017, pp. 207–226.

[2] A.P. Aguiar, J.P. Hespanha, P.V. Kokotović, Performance limitations in reference tracking and path following for nonlinear systems, Automatica 44 (3) (2008) 598–610.

[3] K. Ahnert, M. Mulansky, Odeint–solving ordinary differential equations in C++, in: AIP Conference Proceedings, vol. 1389, American Institute of Physics, 2011, pp. 1586–1589.

[4] F. Bayer, J. Hauser, Trajectory optimization for vehicles in a constrained environment, in: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), IEEE, 2012, pp. 5625–5630.

[5] A. Broggi, S. Debattisti, P. Grisleri, M. Panciroli, The deeva autonomous vehicle platform, in: 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2015, pp. 692–699.

[6] A. Broggi, P. Medici, P. Zani, A. Coati, M. Panciroli, Autonomous vehicles control in the VisLab intercontinental autonomous challenge, Annu. Rev. Control 36 (1) (2012) 161–171.

[7] J. Canny, The Complexity of Robot Motion Planning, MIT Press, 1988.

[8] S. Diamond, S. Boyd, CVXPY: a python-embedded modeling language for convex optimization, J. Mach. Learn. Res. 17 (1) (2016) 2909–2913.

[9] M. Diehl, H.G. Bock, H. Diedam, P.-B. Wieber, Fast direct multiple shooting algorithms for optimal robot control, in: Fast Motions in Biomechanics and Robotics, Springer, 2006, pp. 65–93.

[10] D. González, J. Pérez, V. Milanés, F. Nashashibi, A review of motion planning techniques for automated vehicles, IEEE Trans. Intell. Transp. Syst. 17 (4) (2015) 1135–1145.

[11] J. Hauser, A projection operator approach to the optimization of trajectory functionals, IFAC Proc. Vol. 35 (1) (2002) 377–382.

[12] J. Hauser, R. Hindman, Maneuver regulation from trajectory tracking: feedback linearizable systems, IFAC Proc. Vol. 28 (14) (1995) 595–600.

[13] A.J. Häusler, A. Saccon, A.P. Aguiar, J. Hauser, A.M. Pascoal, Energy-optimal mo-

tion planning for multiple robotic vehicles with collision avoidance, IEEE Trans. Control Syst. Technol. 24 (3) (2015) 867–883.
[14] R. Hult, M. Zanon, S. Gros, P. Falcone, Optimal coordination of automated vehicles at intersections with turns, in: 2019 18th European Control Conference (ECC), IEEE, 2019, pp. 225–230.
[15] Y.K. Hwang, N. Ahuja, et al., A potential field approach to path planning, IEEE Trans. Robot. Autom. 8 (1) (1992) 23–32.
[16] 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), IEEE, 2015, pp. 1094–1099.
[17] F. Laneve, A. Rucco, M. Bertozzi, A trajectory optimization strategy for merging maneuvers of autonomous vehicles, in: APCA International Conference on Automatic Control and Soft Computing, Springer, 2022, pp. 3–14.
[18] M.A. Patterson, A.V. Rao, GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming, ACM Trans. Math. Softw. 41 (1) (2014) 1–37.
[19] U. Rosolia, S. De Bruyne, A.G. Alleyne, Autonomous vehicle control: a nonconvex approach for obstacle avoidance, IEEE Trans. Control Syst. Technol. 25 (2) (2016) 469–484.
[20] A. Rucco, G. Notarstefano, J. Hauser, An efficient minimum-time trajectory generation strategy for two-track car vehicles, IEEE Trans. Control Syst. Technol. 23 (4) (2015) 1505–1519.
[21] A. Saccon, J. Hauser, A. Beghi, A virtual rider for motorcycles: maneuver regulation of a multi-body vehicle model, IEEE Trans. Control Syst. Technol. 21 (2) (2012) 332–346.
[22] S. Spedicato, A. Franchi, G. Notarstefano, From tracking to robust maneuver regulation: an easy-to-design approach for VTOL aerial robots, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 2965–2970.
[23] S. Spedicato, G. Notarstefano, Minimum-time trajectory generation for quadrotors in constrained environments, IEEE Trans. Control Syst. Technol. 26 (4) (2017) 1335–1344.
[24] V. Turri, A. Carvalho, H.E. Tseng, K.H. Johansson, F. Borrelli, Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads, in: 16th International IEEE Conference on Intelligent Transportation Systems, IEEE, 2013, pp. 378–383.
[25] M.T. Wolf, J.W. Burdick, Artificial potential functions for highway driving with collision avoidance, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3731–3736.
[26] A. Zanelli, A. Domahidi, J. Jerez, M. Morari, FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs, Int. J. Control 93 (1) (2020) 13–29.
[27] M. Zanon, J.V. Frasch, M. Vukov, S. Sager, M. Diehl, Model predictive control of autonomous vehicles, in: Optimization and Optimal Control in Automotive Systems, Springer, 2014, pp. 41–57.
[28] X. Zhang, A. Liniger, F. Borrelli, Optimization-based collision avoidance, IEEE Trans. Control Syst. Technol. 29 (3) (2020) 972–983.