Optimization-Based Collision Avoidance

Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli

***Abstract*— This article presents a novel method for exactly reformulating nondifferentiable collision avoidance constraints into smooth, differentiable constraints using strong duality of convex optimization. We focus on a controlled object whose goal is to avoid obstacles while moving in an n-dimensional space. The proposed reformulation is exact, does not introduce any approximations, and applies to general obstacles and controlled objects that can be represented as the union of convex sets. We connect our results with the notion of signed distance, which is widely used in traditional trajectory generation algorithms. Our method can be applied to generic navigation and trajectory planning tasks, and the smoothness property allows the use of general-purpose gradient- and Hessian-based optimization algorithms. Finally, in case a collision cannot be avoided, our framework allows us to find "least-intrusive" trajectories, measured in terms of penetration. We demonstrate the efficacy of our framework on an automated parking problem, where our numerical experiments suggest that the proposed method is robust and enables real-time optimization-based trajectory planning in tight environments. Sample code of our example is provided at https://github.com/XiaojingGeorgeZhang/OBCA.**

***Index Terms*— Autonomous driving, collision avoidance, model predictive control (MPC), navigation in tight environments, nonlinear optimization, obstacle avoidance, path planning, trajectory optimization.**

I. Introduction

MANEUVERING autonomous systems in an environment with obstacles is a challenging problem that arises in a number of practical applications, including robotic manipulators and trajectory planning for autonomous systems, such as self-driving cars and quadcopters. In almost all of those applications, a fundamental feature is the system's ability to avoid collision with obstacles, which are, for example, humans operating in the same area, other autonomous systems, or static objects such as walls.

Optimization-based trajectory planning algorithms, such as model predictive control (MPC), have received significant attention recently, ranging from (unmanned) aircraft to robots to autonomous cars [1]–[15]. This can be attributed to the increase in computational resources, the availability of robust numerical algorithms for solving optimization problems, as well as MPC's ability to systematically encode system dynamics and constraints inside its formulation.

One fundamental challenge in optimization-based trajectory planning is the appropriate formulation of collision avoidance constraints, which are known to be nonconvex and computationally difficult to handle in general. While a number of formulations have been proposed in the literature for dealing with collision avoidance constraints, they are typically limited by one of the following features.

1) The collision avoidance constraints are approximated through linear constraint, and it is difficult to establish the approximation error [10].
2) Existing formulations focus on point-mass controlled objects and are not applicable to full-dimensional objects.
3) If the obstacles are polyhedral, then the collision avoidance constraints are often reformulated using integer variables [16].

While this third reformulation is attractive for linear systems with convex constraints (since, in this case, a mixed-integer convex optimization problem can be solved), integer variables should generally be avoided when dealing with nonlinear systems when designing real-time controllers for robotic systems.

In this article, we focus on a controlled object that moves in a general n-dimensional space while avoiding obstacles, and propose a novel approach for modeling obstacle avoidance constraints that overcomes the aforementioned limitations. Specifically, the contributions of this article can be summarized as follows.

1) We show that if the controlled object and the obstacles are described by convex sets, such as polytopes or ellipsoids (or can be decomposed into a finite union of such convex sets), then the collision avoidance constraints can be exactly and nonconservatively reformulated as a set of smooth nonconvex constraints. This is achieved by appropriately reformulating the distance function between two convex sets using strong duality of convex optimization.
2) We provide a second, also exact and smooth, formulation for collision avoidance based on the notion of signed distance, which characterizes not only the distance between two objects but also their penetration. This reformulation allows us to compute "least-intrusive" trajectories in case collisions cannot be avoided.
3) We demonstrate the efficacy of the proposed obstacle avoidance reformulations on an autonomous parking application, where the controlled vehicles must navigate in tight environments. We further show that, if the

Manuscript received May 10, 2018; revised February 3, 2019 and July 23, 2019; accepted September 22, 2019. Manuscript received in final form October 19, 2019. This work was supported in part by the Hyundai Center of Excellence, UC Berkeley. Recommended by Associate Editor S. Di Cairano. *(Corresponding author: Xiaojing Zhang.)*

X. Zhang and F. Borrelli are with the Model Predictive Control Laboratory, Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: xiaojing.zhang@berkeley.edu; fborrelli@berkeley.edu).

A. Liniger is with the Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland (e-mail: liniger@control.ee.ethz.ch).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2019.2949540

numerical solvers are appropriately initialized using methods such as Hybrid A*, then both the distance reformulation and the signed distance reformulation are able to find dynamically feasible trajectories even in challenging circumstances.

Since our framework allows the incorporation of system dynamics and state/input constraints, we can generate safe and kinodynamically feasible trajectories that can be tracked by simple low-level controllers.

This article is organized as follows. Section II introduces the problem setup. Section III presents the collision avoidance and minimum-penetration formulations for the case when the controlled object is a point mass. These results are then extended to full-dimensional controlled objects in Section IV. Numerical experiments demonstrating the efficacy of the proposed method are given in Section V, and conclusions are drawn in Section VI. The Appendix contains auxiliary results needed to prove the main results of this article. The source code of the autonomous parking example described in Section V is provided at https://github.com/XiaojingGeorgeZhang/OBCA.

A. Related Work

A large body of work exists on the topic of obstacle avoidance. In this article, we do not review, or compare, optimization-based collision avoidance methods with alternative approaches, such as those based on dynamic programming [17], reachability analysis [18]–[20], graph search methods, such as A* and Hybrid A* [21]–[23], (random) sampling, such as rapidly exploring random tree (RRT) and RRT* [24]–[27], or interpolating curves [28]. Indeed, collision avoidance problems are known to be NP-hard in general [29], and all of the practical methods constitute some sort of "heuristics," whose performance depends on the specific problem and configuration at hand. In the following, we briefly review optimization-based approaches and refer the interested reader to [30]–[34] for a comprehensive review on the existing trajectory planning and obstacle avoidance algorithms.

The basic idea in optimization-based methods is to express the collision avoidance problem as an optimal control problem and then solve it using numerical optimization techniques. One way of dealing with obstacle avoidance is to use unconstrained optimization, in which case the objective function is augmented with "artificial potential fields" that represent the obstacles [35]–[39]. More recently, methods based on constrained optimization has attracted attention in the control community due to their ability to explicitly formulate collision avoidance through constraints [1], [4], [6], [8], [10], [15], [40]. Broadly speaking, constrained optimization-based collision-avoidance algorithms can be divided into two cases, based on the modeling of the controlled object: point-mass models and full-dimensional objects. Due to its conceptual simplicity, the vast majority of the literature focuses on collision avoidance for point-mass models and considers the shape of the controlled object by inflating the obstacles. The obstacles are generally assumed to be either polytopes or ellipsoids. For polyhedral obstacles, disjunctive programming can be used to ensure collision avoidance, which is often reformulated as a mixed-integer optimization problem [1], [4], [16]. In the case of ellipsoidal obstacles, the collision avoidance constraints can be formulated as a smooth nonconvex constraint [41], [42], and the resulting optimization problem can be solved using generic nonlinear programming solvers.

The case of full-dimensional controlled objects has, to the best of our knowledge, not been widely studied in the context of optimization-based methods, with the exception of [8], [10], and [43]. Li and Shao [43] model the controlled object through its vertices and, under the assumption that all involved objects are rectangles, ensure collision avoidance by keeping all vertices of the controlled object outside the obstacle. A more general way of handling collision avoidance for full-dimensional controlled objects has been proposed in [10] using the notion of signed distance, where the authors also propose a sequential linearization technique to deal with the nonconvexity of the signed distance function. Gerdts *et al.* [8] study the obstacle avoidance formulation for time-optimal trajectories of robotic manipulators using separating hyperplanes.

The approaches most closely related to our formulation are the work of [6] and [8], where the authors propose smooth and exact reformulations of the collision avoidance constraint for point-mass controlled objects and polyhedral obstacles [6] and full-dimensional controlled objects and polyhedral obstacles [8]. Our approach differs from [6] in that we consider the general case of full-dimensional controlled objects. Furthermore, our approach differs from [8] in that we consider general (not necessarily polyhedral) obstacles and are also able to compute least-intrusive trajectories in case collisions cannot be avoided.

Cones:
1. https://en.wikipedia.org/wiki/Convex_cone
2. T. Rockafellar, Convex Analysis. 1970. pp.10-15.

B. Notation

Given a proper cone $\mathcal{K} \subset \mathbb{R}^l$ and two vectors $a, b \in \mathbb{R}^l$, then $a \preceq_{\mathcal{K}} b$ is equivalent to $(b-a) \in \mathcal{K}$. If $\mathcal{K} = \mathbb{R}^l_+$ is the standard cone, then $\preceq_{\mathbb{R}^l_+}$ is equivalent to the standard (elementwise) inequality $\leq$. Moreover, $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$, and $\mathcal{K}^* \subset \mathbb{R}^l$ is the dual cone of $\mathcal{K}$. Note that if $\|\cdot\| = \|\cdot\|_2$ is the Euclidean distance, then $\|\cdot\|_* = \|\cdot\|_2$. The "space" occupied by the controlled object (e.g., a drone, vehicle, or robot in general) is denoted as $\mathbb{E} \subset \mathbb{R}^n$; similarly, the space occupied by the obstacles is denoted as $\mathbb{O} \subset \mathbb{R}^n$.

dual norms:
1. https://math.stackexchange.com/questions/903484/dual-norm-intuition

II. Problem Description

A. Dynamics, Objective, and Constraints

We assume that the dynamics of the controlled object takes the form

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

where $x_k \in \mathbb{R}^{n_x}$ is the state of the controlled object at time step k given an initial state $x_0 = x_S$, $u_k \in \mathbb{R}^{n_u}$ is the control input, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ describes the dynamics of the system. In most of the cases, the state x_k contains information, such as the position $p_k \in \mathbb{R}^n$ and angles $\theta_k \in \mathbb{R}^n$ of the controlled object, as well the velocities $\dot{p}_k$ and angular rates $\dot{\theta}_k$. In this article, we assume that no disturbance is present, and that the system is subject to input and state constraints of the form

$$h(x_k, u_k) \leq 0 \tag{2}$$

where $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$, n_h is the number of constraints, and the inequality in (2) is interpreted elementwise. Our goal is to find a control sequence, over a horizon N, which allows the controlled object to navigate from the initial state x_S to its final state $x_F \in \mathbb{R}^{n_x}$ while optimizing some objective function $J = \sum_{k=0}^{N} \ell(x_k, u_k)$, where $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is a stage cost, and avoiding $M \geq 1$ obstacles $\mathbb{O}^{(1)}, \mathbb{O}^{(2)}, \ldots, \mathbb{O}^{(M)} \subset \mathbb{R}^n$. Throughout this article, we assume that the functions $f(\cdot,\cdot)$, $h(\cdot,\cdot)$, and $\ell(\cdot,\cdot)$ are smooth. Smoothness is assumed for simplicity, although all forthcoming statements apply equally to cases when those functions are twice continuously differentiable.

B. Obstacle and Controlled Object Modeling

Given the state x_k, we denote by $\mathbb{E}(x_k) \subset \mathbb{R}^n$ the "space" occupied by the controlled object at time k, which we assume is a subset of $\mathbb{R}^n$. The collision avoidance constraint at time k is now given by[1]

$$\mathbb{E}(x_k) \cap \mathbb{O}^{(m)} = \emptyset \quad \forall m = 1, \ldots, M. \tag{3}$$

Constraint (3) is nondifferentiable in general, e.g., when the obstacles are polytopic [6], [10]. In this article, we will remodel (3) in such a way that both continuity and differentiability are preserved. To this end, we assume that the obstacles $\mathbb{O}^{(m)}$ are convex compact sets with nonempty relative interior,[2] and can be represented as

(A, b) represents the coefficients and constants of the linear constraints, where l is the number of constraint equations

$$\mathbb{O}^{(m)} = \{y \in \mathbb{R}^n : A^{(m)} y \preceq_{\mathcal{K}} b^{(m)}\} \tag{4}$$

where $A^{(m)} \in \mathbb{R}^{l \times n}$, $b^{(m)} \in \mathbb{R}^l$, and $\mathcal{K} \subset \mathbb{R}^l$ is a closed convex pointed cone with nonempty interior. Representation (4) is entirely generic since any compact convex set admits a conic representation of the form (4) [44, p.15]. In particular, polyhedral obstacles can be represented as (4) by choosing $\mathcal{K} = \mathbb{R}^l_+$; in this case, $\preceq_{\mathcal{K}}$ corresponds to the well-known elementwise inequality $\leq$. Likewise, ellipsoidal obstacles can be represented by letting $\mathcal{K}$ be the second-order cone; see [45] for details. To simplify the upcoming exposition, the same cone $\mathcal{K}$ is assumed for all obstacles; the extension to obstacle-specific cones $\mathcal{K}^{(m)}$ is straightforward. Similarly, we assume for simplicity that the obstacles $\mathbb{O}^{(m)}$ are static. Nevertheless, the forthcoming reformulations can be adapted to time-varying obstacles $\mathbb{O}^{(m)}_k = \{y \in \mathbb{R}^n : A^{(m)}_k y \preceq_{\mathcal{K}} b^{(m)}_k\}$ by replacing the time-invariant matrices $A^{(m)}$ and $b^{(m)}$ in (4) with time-varying matrices $A^{(m)}_k$ and $b^{(m)}_k$. We also note that, in the case of time-varying obstacles, issues related to recursive feasibility might arise. We will not dwell on these issues in this article and refer the interested reader to [46] for results along that direction.

move everything to the positive sector of the coordinate system, then both K and K* can be ignored

In this article, we will consider controlled objects $\mathbb{E}(x_k)$ that are modeled as point-masses as well as full-dimensional objects. In the former case, $\mathbb{E}(x_k)$ simply extracts the position p_k from the state x_k, that is

point mass representation

$$\mathbb{E}(x_k) = p_k. \tag{5a}$$

In the latter case, we will model the controlled object $\mathbb{E}(x_k)$ as the rotation and translation of an "initial" convex set $\mathbb{B} \subset \mathbb{R}^n$, that is

full-dimension representation

$$\mathbb{E}(x_k) = R(x_k)\mathbb{B} + t(x_k), \quad \mathbb{B} := \{y : Gy \preceq_{\bar{\mathcal{K}}} g\} \tag{5b}$$

where $R : \mathbb{R}^{n_x} \to \mathbb{R}^{n \times n}$ is an (orthogonal) rotation matrix and $t : \mathbb{R}^{n_x} \to \mathbb{R}^n$ is the translation vector. The matrices $(G, g) \in \mathbb{R}^{h \times n} \times \mathbb{R}^h$ and the cone $\bar{\mathcal{K}} \subset \mathbb{R}^h$, which we assume is closed, convex, and pointed, define the shape of our initial (compact) set $\mathbb{B}$ and are assumed to be known. Often, the rotation matrix $R(\cdot)$ depends on the angles θ_k of the controlled object, while the translation vector $t(\cdot)$ depends on the position p_k of the controlled object. We assume throughout that the functions $R(\cdot)$ and $t(\cdot)$ are smooth.

C. Optimal Control Problem With Collision Avoidance

By combining (1)–(3), the constrained finite-horizon optimal control problem with collision avoidance constraint is given by

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \quad & \sum_{k=0}^{N} \ell(x_k, u_k) \\
\text{s.t.} \quad & x_0 = x_S, \quad x_{N+1} = x_F \\
& \left.\begin{aligned} & x_{k+1} = f(x_k, u_k), \\ & h(x_k, u_k) \leq 0, \\ & \mathbb{E}(x_k) \cap \mathbb{O}^{(m)} = \emptyset, \end{aligned}\right\} \begin{aligned} & k = 0, \ldots, N \\ & m = 1, \ldots, M \end{aligned}
\end{aligned} \tag{6}$$

where $\mathbb{E}(x_k)$ is either given by (5a) (point-mass model) or (5b) (full-dimensional set), $\mathbf{x} := [x_0, x_1, \ldots, x_{N+1}]$ is the collection of all states, and $\mathbf{u} := [u_0, u_1, \ldots, u_N]$ is the collection of all inputs. A key difficulty in solving problem (6), even for linear systems with convex objective function and convex state/input constraints, is the presence of the collision-avoidance constraints $\mathbb{E}(x_k) \cap \mathbb{O}^{(m)} = \emptyset$, which, in general, are nonconvex and nondifferentiable [6], [10]. In the following, we present two novel approaches for modeling collision avoidance constraints that preserve continuity and differentiability and are amendable for use with existing off-the-shelf gradient- and Hessian-based optimization algorithms.

D. Collision Avoidance

A popular way of formulating collision avoidance is based on the notion of signed distance [10]

$$\mathrm{sd}(\mathbb{E}(x), \mathbb{O}) := \mathrm{dist}(\mathbb{E}(x), \mathbb{O}) - \mathrm{pen}(\mathbb{E}(x), \mathbb{O}) \tag{7}$$

where $\mathrm{dist}(\cdot,\cdot)$ and $\mathrm{pen}(\cdot,\cdot)$ are the distance and penetration function and are defined as

$$\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) := \min_t\{\|t\| : (\mathbb{E}(x) + t) \cap \mathbb{O} \neq \emptyset\} \tag{8a}$$

$$\mathrm{pen}(\mathbb{E}(x), \mathbb{O}) := \min_t\{\|t\| : (\mathbb{E}(x) + t) \cap \mathbb{O} = \emptyset\}. \tag{8b}$$

Roughly speaking, the signed distance is positive if $\mathbb{E}(x)$ and $\mathbb{O}$ do not intersect, and negative if they overlap. Therefore, collision avoidance can be ensured by requiring $\mathrm{sd}(\mathbb{E}(x), \mathbb{O}) > 0$. Unfortunately, directly enforcing $\mathrm{sd}(\mathbb{E}(x), \mathbb{O}) > 0$ inside the

[1]In this article, we only consider collision avoidance constraints that are associated with the position and geometric shape of the controlled object, which are typically defined by its position p_k and angles θ_k. This is not a restriction of the theory, as the forthcoming approaches can be easily generalized to collision avoidance involving other states, but done to simplify exposition of the material.

[2]Nonconvex obstacles can often be approximated/decomposed as the union of convex obstacles.

optimization problem (6) is generally difficult since it is nonconvex and nondifferentiable in general [10]. Furthermore, for optimization algorithms to be numerically efficient, they require an explicit representation of the functions they are dealing with, in this case $\mathrm{sd}(\cdot,\cdot)$. This, however, is difficult to obtain in practice since $\mathrm{sd}(\cdot,\cdot)$ itself is the solution of the optimization problems (8a) and (8b). As a result, existing algorithms in the literature often approximate (7) through local linearization [10], for which it is difficult to establish bounds on approximation errors.

In the following, we propose two reformulation techniques for obstacles avoidance that overcome the issues of nondifferentiability and do not require an explicit representation of the signed distance. We begin with the point-mass models in Section III and treat the general case of the full-dimensional controlled objects in Section IV.

III. Collision Avoidance for Point-Mass Models

In this section, we first present a smooth reformulation of (6) when $\mathbb{E}(x_k) = p_k$ in Section III-A and then extend the approach in Section III-B to generate the minimum-penetration trajectories in case collisions cannot be avoided. To simplify notation, the time indices k are omitted in the remainder of this section.

A. Collision-Free Trajectory Generation

if a lambda exists, where any dimension is larger than 0, and the rest are exactly 0, then (Ap - b)^T \lambda > d_{dim} means point p is outside of one of the obstacle boundaries.

Proposition 1: Assume that the obstacle $\mathbb{O}$ and the controlled object are given as in (4) and (5a), respectively, and let $\mathsf{d}_{\min} \geq 0$ be a desired safety margin between the controlled object and the obstacle. Then, we have

inner point x in obstacle: Ax < b

p = p_{k} = E(x_{k}) is the point to avoid collision

$$\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}_{\min}$$
$$\iff \exists \lambda \succeq_{\mathcal{K}^*} 0 \colon (A\,p - b)^\top \lambda > \mathsf{d}_{\min}, \quad \|A^\top \lambda\|_* \leq 1. \tag{9}$$

Proof: It follows from (4) and (8a) that $\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) = \min_t\{\|t\| \colon A(\mathbb{E}(x) + t) \preceq_{\mathcal{K}} b\}$. Following [45, p.401], its dual problem is given by $\max_\lambda\{(A\mathbb{E}(x) - b)^\top \lambda \colon \|A^\top \lambda\|_* \leq 1,\ \lambda \succeq_{\mathcal{K}^*} 0\}$, where $\|\cdot\|_*$ is the dual norm associated with $\|\cdot\|$ and $\mathcal{K}^*$ is the dual cone of $\mathcal{K}$. Since $\mathbb{O}$ is assumed to have nonempty relative interior, strong duality holds, and $\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) = \max_\lambda\{(A\,\mathbb{E}(x) - b)^\top \lambda \colon \|A^\top \lambda\|_* \leq 1,\ \lambda \succeq_{\mathcal{K}^*} 0\}$. Hence, for any nonnegative scalar $\mathsf{d}_{\min}$, $\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}_{\min}$ is satisfied if and only if, there exists $\lambda \succeq_{\mathcal{K}^*} 0 \colon (A\,\mathbb{E}(x) - b)^\top \lambda > \mathsf{d}_{\min}$, $\|A^\top \lambda\|_* \leq 1$. The desired result follows from identity (5a). ∎

Intuitively speaking, any variable λ satisfying the right-hand side of (9) is a certificate verifying the condition $\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}_{\min}$. Since $\mathbb{E}(x) \cap \mathbb{O} = \emptyset$ is equivalent to $\mathrm{dist}(\mathbb{E}(x), \mathbb{O}) > 0$, the optimal control problem (6) for the point-mass model (5a) is given by

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u},\boldsymbol{\lambda}} \quad & \sum_{k=0}^{N} \ell(x_k, u_k) \\
\text{s.t.} \quad & x_0 = x_S, \quad x_{N+1} = x_F \\
& x_{k+1} = f(x_k, u_k), \quad h(x_k, u_k) \leq 0 \\
& (A^{(m)} p_k - b^{(m)})^\top \lambda_k^{(m)} > 0 \\
& \left\|{A^{(m)}}^\top \lambda_k^{(m)}\right\|_* \leq 1, \quad \lambda_k^{(m)} \succeq_{\mathcal{K}^*} 0 \\
& \text{for } k = 0, \dots, N, \quad m = 1, \dots, M
\end{aligned} \tag{10}$$

where p_k is the position of the controlled object at time k, $\lambda_k^{(m)}$ is the dual variable associated with obstacle $\mathbb{O}^{(m)}$ at time step k, and the optimization is performed over the states $\mathbf{x}$, the inputs $\mathbf{u}$, and the dual variables $\boldsymbol{\lambda} = [\lambda_0^{(1)}, \dots, \lambda_0^{(m)}, \lambda_1^{(1)}, \dots, \lambda_N^{(m)}]$. We emphasize that (10) is an exact reformulation of (6) and that the optimal trajectory $\mathbf{x}^* = [x_0^*, x_1^*, \dots, x_{N+1}^*]$ obtained by solving (6) is kinodynamically feasible.

Remark 1: Without further assumptions on the norm $\|\cdot\|$ and the cone $\mathcal{K}$, the last two constraints in (10) are not guaranteed to be smooth, a property that many general-purpose nonlinear optimization algorithms require.[3] Fortunately, it turns out that these constraints are smooth for the practically relevant cases of $\|\cdot\|$ being the Euclidean distance and $\mathcal{K}$ either the standard cone or the second-order cone, which allows us to model polyhedral and ellipsoidal obstacles. In these cases, and under the assumption that the functions $f(\cdot,\cdot)$, $h(\cdot,\cdot)$, and $\ell(\cdot,\cdot)$ are smooth, (10) is a smooth nonlinear optimization problem that is amendable to general-purpose nonlinear optimization algorithms, such as Interior Point OPTimizer (IPOPT) [47]. Without going into details, we point out that the smoothness is retained when $\|\cdot\| = \|\cdot\|_p$ is a general p-norm, with $p \in (1, \infty)$, and $\mathcal{K}$ is the Cartesian product of p-order cones $\mathcal{K}_p := \{(s, z) \colon \|z\|_p \leq s\}$, with $p \in (1, \infty)$. In this case, the dual norm is given by $\|\cdot\|_* = \|\cdot\|_q$ and the dual cone is $(\mathcal{K}_p)^* = \mathcal{K}_q$, where q satisfies $1/p + 1/q = 1$; see [45] for details on dual norms and dual cones.

While reformulation (10) can be used for obstacle avoidance, it is limited to finding collision-free trajectories. Indeed, in case collisions cannot be avoided, the aforementioned formulation is not able to find "least-intrusive" trajectories by softening the constraints. Intuitively speaking, this is because (10) is based on the notion of distance, and the distance between two overlapping objects (as is in the case of collision) is always zero, regardless of the penetration. From a practical point of view, this implies that slack variables cannot be included in the constraints of (9) because the optimal control problem is not able to distinguish between "severe" and "less severe" colliding trajectories. Furthermore, in practice, it is often desirable to soften constraints and includes slack variables to ensure the feasibility of the (nonconvex) optimization problem since (local) infeasibilities in a nonconvex optimization problem are known to cause numerical difficulties. In the following, we show how the above-mentioned limitations can be overcome by considering the notion of penetration and softening the collision avoidance constraints.

B. Minimum-Penetration Trajectory Generation

In this section, we consider the design of minimum-penetration trajectories for cases when a collision cannot be avoided and the goal is to find a "least-intrusive" trajectory. Following the literature [48], [49], we measure "intrusion" in terms of penetration as defined in (8b).

[3] Strictly speaking, these solvers often require the cost function and constraints to be twice continuously differentiable only. Smoothness is assumed in this article for the sake of simplicity.

Proposition 2: Assume that the obstacle $\mathbb{O}$ and the controlled object are given as in (4) and (5a), respectively, and let $\mathsf{p}_{\max} \geq 0$ be a desired maximum penetration of the controlled object and the obstacle. Then, we have

$$\begin{aligned}&\text{pen}(\mathbb{E}(x), \mathbb{O}) < \mathsf{p}_{\max}\\ &\quad\iff \exists \lambda \succeq_{\mathcal{K}^*} 0\colon (b - A\,p)^\top \lambda < \mathsf{p}_{\max}, \quad \|A^\top \lambda\|_* = 1. \end{aligned} \tag{11}$$

Proof: The proof, along with auxiliary lemmas, is given in the Appendix. ■

Proposition 2 resembles Proposition 1 with the difference that the convex inequality constraint $\|A^\top \lambda\|_* \leq 1$ is replaced with the nonconvex equality constraint $\|A^\top \lambda\|_* = 1$. In the following, we will see that Propositions 1 and 2 can be combined to represent the signed distance as defined in (7).

Theorem 1: Assume that the obstacle $\mathbb{O}$ and the controlled object are given as in (4) and (5a), respectively. Then, for any value $\mathsf{d} \in \mathbb{R}$, we have

Generalization of both distance and penetration

$$\begin{aligned}&\text{sd}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}\\ &\quad\iff \exists \lambda \succeq_{\mathcal{K}^*} 0\colon (A\,p - b)^\top \lambda > \mathsf{d}, \quad \|A^\top \lambda\|_* = 1. \end{aligned} \tag{12}$$

Proof: By definition, $\text{sd}(\mathbb{E}(x), \mathbb{O}) = \text{dist}(\mathbb{E}(x), \mathbb{O})$ if $\mathbb{E}(x) \cap \mathbb{O} = \emptyset$, and $\text{sd}(\mathbb{E}(x), \mathbb{O}) = -\text{pen}(\mathbb{E}(x), \mathbb{O})$ if $\mathbb{E}(x) \cap \mathbb{O} \neq \emptyset$. Let $\mathbb{E}(x) \cap \mathbb{O} \neq \emptyset$, in which case (12) follows directly from (11). If $\mathbb{E}(x) \cap \mathbb{O} = \emptyset$, then we have from (9) that $\text{sd}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}$ is equivalent to $\exists \lambda \succeq_{\mathcal{K}^*} 0\colon (A\,p - b)^\top \lambda > \mathsf{d},\ \|A^\top \lambda\|_* \leq 1$. Due to homogeneity with respect to λ, if the previous condition is satisfied, then there always exists a (scaled) dual-multiplier $\lambda' \succeq_{\mathcal{K}^*} 0$, such that $(A\,p - b)^\top \lambda' > \mathsf{d},\ \|A^\top \lambda'\|_* = 1$. This concludes the proof. ■

Reformulation (12) is similar to reformulation (9) with the difference that (12) holds for all $\mathsf{d} \in \mathbb{R}$, while (9) only holds for $\mathsf{d} \geq 0$. The "price" we pay for this generalization is that the convex constraint $\|A^\top \lambda\|_* \leq 1$ is turned into the nonconvex equality constraint $\|A^\top \lambda\|_* = 1$, which, as we will see later on, generally results in longer computation times. Nevertheless, Theorem 1 allows us to compute trajectories of least penetration whenever collision cannot be avoided by solving the following soft-constrained *minimum-penetration* problem:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{u},\mathbf{s},\boldsymbol{\lambda}} \quad & \sum_{k=0}^{N} \left[\ell(x_k, u_k) + \kappa \cdot \sum_{m=1}^{M} s_k^{(m)} \right]\\ \text{s.t.} \quad & x_0 = x(0), \quad x_{N+1} = x_F\\ & x_{k+1} = f(x_k, u_k), \quad h(x_k, u_k) \leq 0\\ & (A^{(m)} p_k - b^{(m)})^\top \lambda_k^{(m)} > -s_k^{(m)}\\ & \|A^{(m)\top} \lambda_k^{(m)}\|_* = 1,\\ & s_k^{(m)} \geq 0, \quad \lambda_k^{(m)} \succeq_{\mathcal{K}^*} 0\\ & \text{for } k = 0, \ldots, N, \quad m = 1, \ldots, M \end{aligned} \tag{13}$$

where p_k is the position of the controlled object at time k, $s_k^{(m)} \in \mathbb{R}_+$ is the slack variable associated with the object $\mathbb{O}^{(m)}$ at time step k, and $\kappa \geq 0$ is a weight factor that keeps the slack variable as close to zero as possible. Without going into details, we point out that the weight κ should be chosen "big enough," such that the slack variables only become active when the original problem is infeasible, i.e., when the obstacle avoidance is not possible [50]. Note that a positive slack variable implies a colliding trajectory, where the penetration depth is given by $s_k^{(m)}$. We close this section by pointing out that if, *a priori*, it is known that a collision-free trajectory can be generated, then formulation (10) should be given preference over formulation (13) because the former has fewer decision variables, and because the constraint $\|A^{(m)\top} \lambda_k^{(m)}\|_* \leq 1$ is convex, which generally leads to improved solution times. The smoothness of (13) is ensured if $\|\cdot\|$ is the Euclidean distance, and $\mathcal{K}$ is either the standard cone or the second-order cone; see Remark 1 for details.

IV. Collision Avoidance for Full-Dimensional Controlled Objects

Section III provided a framework for computing collision-free and minimum-penetration trajectories for controlled objects that are described by the point-mass model. While such models can be used to generate trajectories for "ball-shaped" controlled objects, done by setting the minimum distance $d_{\min}$ equal to the radius of the controlled object, it can be restrictive in other cases. For example, modeling a car in a parking lot as a Euclidean ball can be very conservative and prevent the car from finding a parking spot. To alleviate this issue, we show in this section how the results of Section III can be extended to full-dimensional controlled objects.

A. Collision-Free Trajectory Generation

Similar to Section III, we begin by first reformulating the distance function, which will allow us to generate collision-free trajectories.

Proposition 3: Assume that the controlled object and the obstacle are given as in (5b) and (4), respectively, and let $\mathsf{d}_{\min} \geq 0$ be a desired safety margin. Then, we have

$$\begin{aligned}&\text{dist}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}_{\min}\\ &\Leftrightarrow \exists \lambda \succeq_{\mathcal{K}^*} 0, \quad \mu \succeq_{\bar{\mathcal{K}}^*} 0\colon -g^\top \mu + (At(x) - b)^\top \lambda > \mathsf{d}_{\min},\\ &\quad G^\top \mu + R(x)^\top A^\top \lambda = 0, \quad \|A^\top \lambda\|_* \leq 1. \end{aligned} \tag{14}$$

Proof: Recall that $\text{dist}(\mathbb{E}(x), \mathbb{O}) = \min_{e,o}\{\|e - o\|\colon Ao \preceq_{\mathcal{K}} b, e \in \mathbb{E}(x)\} = \min_{e',o}\{\|R(x)e' + t(x) - o\|\colon Ao \preceq_{\mathcal{K}} b, Ge' \preceq_{\bar{\mathcal{K}}} g\}$, where the last equality follows from (5b). The dual of this minimization problem is given by $\max_{\lambda,\mu}\{-g^\top \mu + (At(x) - b)^\top \lambda\colon G^\top \mu + R(x)^\top A^\top \lambda = 0,\ \|A^\top \lambda\|_* \leq 1,\ \lambda \succeq_{\mathcal{K}^*} 0,\ \mu \succeq_{\bar{\mathcal{K}}^*} 0\}$ (see [45, Sec. 8.2] for the derivation), where $\|\cdot\|_*$ is the dual norm and $\mathcal{K}^*$ and $\bar{\mathcal{K}}^*$ are the dual cones of $\mathcal{K}$ and $\bar{\mathcal{K}}$, respectively. Since $\mathbb{O}$ and $\mathbb{B}$ are assumed to have nonempty relative interior, strong duality holds, and $\text{dist}(\mathbb{E}(x), \mathbb{O}) > \mathsf{d}_{\min} \Leftrightarrow \max_{\lambda,\mu}\{-g^\top \mu + (At(x) - b)^\top \lambda\colon G^\top \mu + R(x)^\top A^\top \lambda = 0,\ \|A^\top \lambda\|_* \leq 1,\ \lambda \succeq_{\mathcal{K}^*} 0,\ \mu \succeq_{\bar{\mathcal{K}}^*} 0\} > \mathsf{d}_{\min} \Leftrightarrow \exists \lambda \succeq_{\mathcal{K}^*} 0, \mu \succeq_{\bar{\mathcal{K}}^*} 0\colon -g^\top \mu + (At(x) - b)^\top \lambda > \mathsf{d}_{\min},\ G^\top \mu + R(x)^\top A^\top \lambda = 0,\ \|A^\top \lambda\|_* \leq 1$. ■

Compared with Proposition 1, we see that the full-dimensional controlled objects require the introduction of the additional dual variables $\mu^{(m)}$, one for each obstacle $\mathbb{O}^{(m)}$. By setting $\mathsf{d}_{\min} = 0$, we now obtain the following reformulation of (6) for the case of full-dimensional objects:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}} \quad & \sum_{k=0}^{N} \ell(x_k,u_k) \\
\text{s.t.} \quad & x_0 = x(0), \quad x_{N+1} = x_F \\
& x_{k+1} = f(x_k,u_k), \quad h(x_k,u_k) \le 0 \\
& -g^\top \mu_k^{(m)} + (A^{(m)} t(x_k) - b^{(m)})^\top \lambda_k^{(m)} > 0 \\
& G^\top \mu_k^{(m)} + R(x_k)^\top A^{(m)\top} \lambda_k^{(m)} = 0 \\
& \|A^{(m)\top} \lambda_k^{(m)}\|_* \le 1, \quad \lambda_k^{(m)} \succeq_{\mathcal{K}^*} 0, \quad \mu_k^{(m)} \succeq_{\bar{\mathcal{K}}^*} 0 \\
& \text{for } k = 0,\dots,N, \quad m = 1,\dots,M
\end{aligned} \tag{15}$$

where $\lambda_k^{(m)}$ and $\mu_k^{(m)}$ are the dual variables associated with the obstacle $\mathbb{O}^{(m)}$ at step k, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the collection of all $\lambda_k^{(m)}$ and $\mu_k^{(m)}$, respectively, and the optimization is performed over $(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu})$. Note that (15) is an exact reformulation of (6). The smoothness of (15) is ensured if $\|\cdot\|$ is the Euclidean distance, and $\mathcal{K}$ and $\bar{\mathcal{K}}$ are either the standard cone or the second-order cone; see also Remark 1 for details.

Similar to the point-mass case in Section III-A, the optimal control problem (15) is able to generate collision-free trajectories, but it is unable to find "least-intrusive" trajectories in case collision-free trajectories do not exist. This limitation is addressed next.

B. Minimum-Penetration Trajectory Generation

We overcome the above-mentioned limitation by considering again the notion of penetration. We begin with the following result.

Proposition 4: Assume that the obstacles and the controlled object are given as in (4) and (5b), respectively, and let $\mathsf{p}_{\max} \ge 0$ be a maximal penetration depth. Then, we have

$$\begin{aligned}
&\mathrm{pen}(\mathbb{E}(x),\mathbb{O}) < \mathsf{p}_{\max} \\
&\iff \exists \lambda \succeq_{\mathcal{K}^*} 0, \quad \mu \succeq_{\bar{\mathcal{K}}^*} 0 \colon g^\top \mu + (b - A\,t(x))^\top \lambda < \mathsf{p}_{\max}, \\
&\qquad G^\top \mu + R(x)^\top A^\top \lambda = 0, \quad \|A^\top \lambda\|_* = 1.
\end{aligned} \tag{16}$$

Proof: It follows from [48] that $\mathrm{pen}(\mathbb{E}(x),\mathbb{O}) = \mathrm{pen}(0, \mathbb{O} - \mathbb{E}(x))$, where $\mathbb{O} - \mathbb{E}(x) := \{o - e \colon o \in \mathbb{O},\ e \in \mathbb{E}(x)\}$ is the Minkowski difference. Furthermore, we have from the proof of Proposition 2 that $\mathrm{pen}(0, \mathbb{O} - \mathbb{E}(x)) = \inf_{\{z\colon \|z\|_*=1\}}\{\max_{y\in\mathbb{O}-\mathbb{E}(x)}\{y^\top z\}\}$. Using strong duality of convex optimization, we can dualize the inner maximization problem as $\max_{o\in\mathbb{O},e\in\mathbb{E}(x)}\{z^\top(o-e)\} = \max_{o\in\mathbb{O},e'\in\mathbb{B}}\{z^\top(o - R(x)e' - t(x))\} = \min_{\lambda,\mu}\{b^\top\lambda + g^\top\mu - z^\top t(x)\colon A^\top\lambda = z,\ G^\top\mu = -R^\top z \colon \lambda \succeq_{K^*} 0,\ \mu \succeq_{\bar{\mathcal{K}}^*} 0\}$. Hence, $\mathrm{pen}(0, \mathbb{O} - \mathbb{E}(x)) = \inf_{z,\lambda,\mu}\{b^\top\lambda + g^\top\mu - z^\top t(x) \colon A^\top\lambda = z,\ G^\top\mu = -R^\top z,\ \|z\|_* = 1\}$. Eliminating the z-variable using the first equality constraint and following the steps of the proof of Proposition 2 give the desired result. ■

We notice that an alternative way of deriving the results of Propositions 3 and 4, which, however, requires further investigation, could be to model the controlled object as a point mass, "add" its shape to the obstacle's shape, and then apply the results of Section III.

Theorem 2 shows that Propositions 3 and 4 can be combined to represent the signed distance function.

Theorem 2: Assume that the obstacles and the controlled object are given as in (4) and (5b), respectively. Then, for any $\mathsf{d} \in \mathbb{R}$, we have

$$\begin{aligned}
&\mathrm{sd}(\mathbb{E}(x),\mathbb{O}) > \mathsf{d} \\
&\iff \exists \lambda \succeq_{\mathcal{K}^*} 0, \quad \mu \succeq_{\bar{\mathcal{K}}^*} 0 \colon -g^\top \mu + (At(x) - b)^\top \lambda > \mathsf{d}, \\
&\qquad G^\top \mu + R(x)^\top A^\top \lambda = 0, \quad \|A^\top \lambda\|_* = 1.
\end{aligned} \tag{17}$$

Proof: By definition, $\mathrm{sd}(\mathbb{E}(x),\mathbb{O}) = \mathrm{dist}(\mathbb{E}(x),\mathbb{O})$ if $\mathbb{E}(x) \cap \mathbb{O} = \emptyset$, and $\mathrm{sd}(\mathbb{E}(x),\mathbb{O}) = -\mathrm{pen}(\mathbb{E}(x),\mathbb{O})$ if $\mathbb{E}(x) \cap \mathbb{O} \ne \emptyset$. Consider now $\mathbb{E}(x) \cap \mathbb{O} \ne \emptyset$, in which case (17) follows directly from (16). Assume now that $\mathbb{E}(x) \cap \mathbb{O} = \emptyset$; then, we have from (14) that $\mathrm{sd}(\mathbb{E}(x),\mathbb{O}) > \mathsf{d}$ is equivalent to $\exists \lambda \succeq_{\mathcal{K}^*} 0, \mu \succeq_{\bar{\mathcal{K}}^*} 0 \colon -g^\top\mu + (At(x) - b)^\top\lambda > \mathsf{d},\ G^\top\mu + R(x)^\top A^\top \lambda = 0,\ \|A^\top\lambda\|_* \le 1$. Due to homogeneity with respect to λ and μ, if the previous condition is satisfied, then there also exist $\lambda' \succeq_{\mathcal{K}^*} 0$ and $\mu' \succeq_{\bar{\mathcal{K}}^*} 0$, such that $-g^\top\mu + (At(x) - b)^\top\lambda > \mathsf{d}$, $G^\top\mu + R(x)^\top A^\top\lambda = 0$, and $\|A^\top\lambda\|_* = 1$. This concludes the proof. ■

Theorem 2 allows us to formulate the following soft-constrained *minimum-penetration* optimal control problem:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u},\mathbf{s},\boldsymbol{\lambda},\boldsymbol{\mu}} \quad & \sum_{k=0}^{N} \left[\ell(x_k,u_k) + \kappa \cdot \sum_{m=1}^{M} s_k^{(m)} \right] \\
\text{s.t.} \quad & x_0 = x_S, \quad x_{N+1} = x_F \\
& x_{k+1} = f(x_k,u_k), \quad h(x_k,u_k) \le 0 \\
& -g^\top \mu_k^{(m)} + (A^{(m)} t(x_k) - b^{(m)})^\top \lambda_k^{(m)} > -s_k^{(m)} \\
& G^\top \mu_k^{(m)} + R(x_k)^\top A^{(m)\top} \lambda_k^{(m)} = 0 \\
& \|A^{(m)\top} \lambda_k^{(m)}\|_* = 1 \\
& s_k^{(m)} \ge 0, \quad \lambda_k^{(m)} \succeq_{\mathcal{K}^*} 0, \quad \mu_k^{(m)} \succeq_{\bar{\mathcal{K}}^*} 0 \\
& \text{for } k = 0,\dots,N, \quad m = 1,\dots,M
\end{aligned} \tag{18}$$

where $s_k^{(m)} \in \mathbb{R}_+$ is the slack variable associated with obstacle $\mathbb{O}^{(m)}$ at time step k and $\kappa \ge 0$ is a weight factor that keeps the slack variable as small as possible. The smoothness of (18) is ensured if $\|\cdot\|$ is the Euclidean distance, and $\mathcal{K}$ and $\bar{\mathcal{K}}$ are either the standard cone or the second-order cone; see Remark 1 for details.

In Section V, we illustrate our obstacle avoidance formulations on an automated parking problem, where the full-dimensional obstacle avoidance problem formulation is used. We refer the interested reader to [51], an extended version of this article, for a quadrotor navigation example that uses the point-mass model.

V. EXAMPLE: AUTONOMOUS PARKING

As an application for our collision avoidance formulation, we consider the autonomous parking problem for self-driving cars. Specifically, we consider the case where the maneuverable space is limited and where the vehicle has to move in a tight environment (see Figs. 1 and 2). Intuitively, in order for a vehicle to find a collision-free path in tight environments, it is important to model the vehicle's shape as accurately and least conservatively as possible. Indeed, modeling a vehicle's shape too conservatively, such as approximating it through a ball or ellipse, maybe prevents the vehicle from finding a feasible

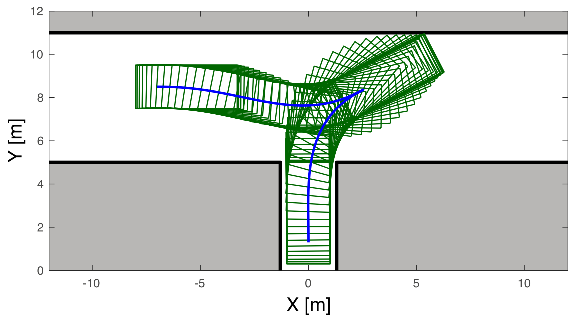


Fig. 1. Reverse parking maneuver. The controlled vehicle is shown in green at every time step. Vehicle starts on the left facing to the right and ends facing upward; see https://youtu.be/V7IUPW2qDFc for an animation.

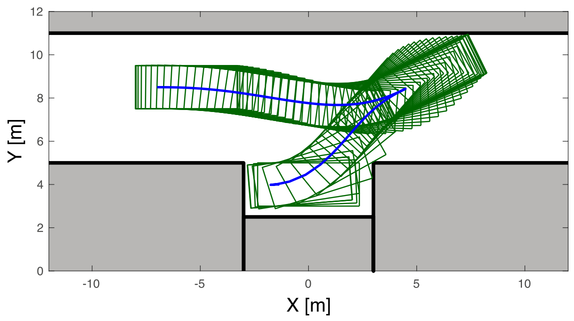


Fig. 2. Parallel parking maneuver. The controlled vehicle is shown in green at every time step. Vehicle starts facing to the right and ends facing to the right; see https://youtu.be/FST7li4M6lU for an animation.

trajectory. In this section, we model the car as a rectangle and then employ the full-dimensional formulation described in Section IV. We show that our modeling framework allows us to find obstacle-free parking trajectories even in tight environments. Two scenarios are considered: reverse parking (see Fig. 1) and parallel parking (see Fig. 2).

A. Environment and Obstacle Modeling

For the reverse parking scenario, the parking spot is assumed 2.6 m wide and 5.2 m long. The width of the road, where the car can maneuver in, is 6 m (see Fig. 1). For the parallel parking scenario, the parking spot is 2.5 m deep and 6 m long, and the space to maneuver is 6 m wide (see Fig. 2). Note that the obstacles in the reverse parking scenario can be described by three axis-aligned rectangles, while the obstacles in the parallel parking scenario can be described by four axis-aligned rectangles. In both cases, the controlled vehicle is modeled as a rectangle of size 4.7×2 m, whose orientation is determined by the car's yaw angle.

B. System Dynamics and Cost Function

The car is described by the classical kinematic bicycle model, which is well-suited for velocities used in typical parking scenarios. The states (X, Y) correspond to the center of the rear axis, while φ is the yaw angle with respect to the x-axis and υ is the velocity with respect to the rear axis. The inputs are the steering angle δ and the acceleration a. Hence, the continuous-time dynamics of the car is given by

$$\begin{aligned} \dot{X} &= \upsilon \cos(\varphi) \\ \dot{Y} &= \upsilon \sin(\varphi) \\ \dot{\varphi} &= \frac{\upsilon \tan(\delta)}{L} \\ \dot{\upsilon} &= a \end{aligned} \tag{19}$$

where $L = 2.7$ m is the wheel base of the car. The steering angle is limited between ± 0.6 rad (approximately 34°), with rate constraints $\dot{\delta} \in [-0.6, 0.6]$ rad/s; acceleration is limited to be between ± 1 m/s^2. We limit the car's velocity to lie between -1 and 2 m/s. The dynamics can be brought into the form (1) using a (forward) Euler discretization, such that $x_{k+1} = x_k + T_{\text{opt}} \tilde{f}(x_k, u_k)$, where T_{opt} is the sampling time, $u_k := (\delta_k, a_k)$ is the control input, and $\tilde{f}(\cdot, \cdot)$ is the continuous-time dynamics that can be obtained from (19).

Our control objective is to navigate the vehicle as fast as possible while avoiding excessive control inputs and rate of changes in the control inputs. We combine these competing goals as a weighted sum of the form $J = q\tau_F + \sum_{k=0}^{N-1} u_k^\top R u_k + \Delta u_k^\top R_\Delta \Delta u_k$, where τ_F is the final time, $\Delta u_k := (u_k - u_{k-1})/T_{\text{opt}}$ is the change in control inputs, and $R = R^\top \succeq 0$, $R_\Delta = R_\Delta^\top \succeq 0$ and $q \geq 0$ are the weighting factors. Motivated by [52], we do not directly minimize τ_F; instead, observing that $\tau_F = NT_{\text{opt}}$, we will treat the discretization time T_{opt} as a decision variable, giving rise to the following cost function:

$$J(\mathbf{u}, T_{\text{opt}}) = qNT_{\text{opt}} + \sum_{k=0}^{N-1} u_k^\top R u_k + \Delta u_k^\top R_\Delta \Delta u_k. \tag{20}$$

Note that treating T_{opt} as an optimization variable has the additional benefit that the duration of the maneuvers does not need to be fixed *a priori*, allowing us to avoid feasibility issues caused by maneuvers that are too short. However, having T_{opt} as a decision variable comes at the cost of introducing an additional decision variable T_{opt}, which renders the dynamics "more nonlinear," which can be seen when looking at the Euler discretization.

C. Choice of Initial Guess

Recall that (15) and (18) are nonconvex optimization problems and hence computationally challenging to solve in general. In practice, one has to content oneself with a locally optimal solution that, for instance, satisfies the Karush–Kuhn–Tucker (KKT) conditions [47] since most of the numerical solvers operate locally. Furthermore, it is well-known that the solution quality critically depends on the initial guess ("warm starting point") that is provided to the solvers, and that different initial guesses can lead to different (local) optima. Unfortunately, computing a good initial guess is often difficult and highly problem-dependent; ideally, the initial guess should be collision-free and approximately satisfy the system dynamics.

For our parking example, we have observed that the Hybrid A* algorithm from [22] is able to provide good initial guesses. Hybrid A* is an extension of the A* algorithm [53], [54],

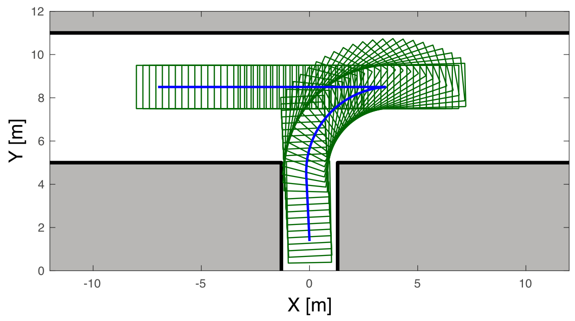


Fig. 3. Initial guess provided by Hybrid A* for reverse parking.

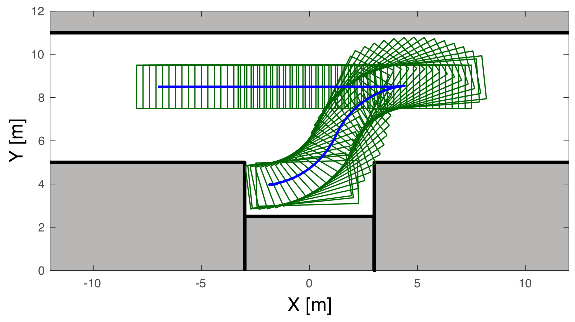
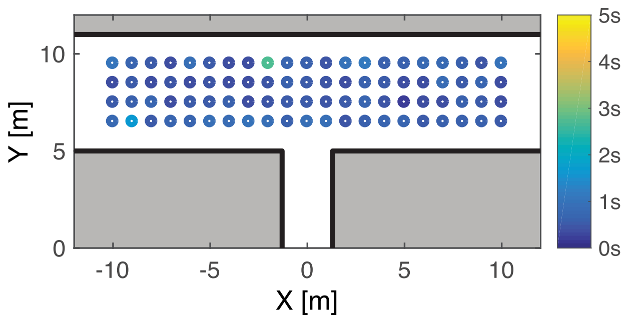


Fig. 4. Initial guess provided by Hybrid A* for parallel parking.



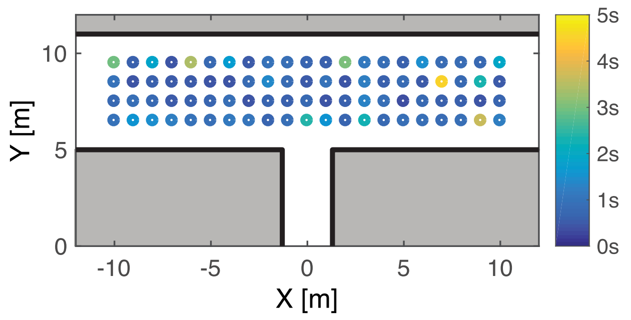


Fig. 5. Solution time for reverse parking with distance formulation (15) (top) and signed-distance formulation (18) (bottom).

and the main idea behind Hybrid A* is to use a simplified vehicle model with states (X, Y, φ) and a finite number of steering inputs to generate a coarse parking trajectory. Like A*, Hybrid A* grids the state space and performs a tree search, where the nodes are expanded using the simplified vehicle model. We refer the interested reader to [22] for details on Hybrid A*. Figs. 3 and 4 show two trajectories obtained from the Hybrid A* algorithm. We observe that, due to the discretization of state and input, the paths generated by Hybrid A* seem more "bang-bang" and less "smooth" than those shown in Figs. 1 and 2.

D. Simulation Results

To evaluate the performance of formulations (15) and (18), we study the reverse and parallel trajectory planning problems. For both cases, we consider different starting positions but one fixed end position at $X = 0$ m and investigate the computation time of each method. The starting positions are generated by gridding the maneuvering space within $X \in [-10, 10]$ m and $Y \in [6.5, 9.5]$ m, with 21 grid points in the x-direction and 4 grid points in the y-direction (see Fig. 5). The orientation for all the starting points is $\varphi = 0$, resulting in a total of 84 starting points. The horizon length N is given by the Hybrid A* algorithm. The optimization problems are implemented with the modeling toolbox JuMP in the programming language Julia [55], and IPOPT [47] is used as the numerical solver. The problems are solved on a 2013 MacBook Pro with an i7 processor clocked at 2.6 GHz. A Julia-based example code can be found at https://github.com/XiaojingGeorgeZhang/OBCA.

We begin by considering the reverse parking case, where one specific maneuver is shown in Fig. 1. The computation times for the distance and the signed distance formulation are listed in Table I (upper half) and shown in Fig. 5, for all 84 initial conditions. Table I indicates that the distance formulation is generally faster than the signed-distance formulation, with a mean computation time of 0.60 s compared with 1.03 s. This is not surprising since the signed distance formulation has more decision variables due to the presence of the slack variables $s_k^{(m)}$, see (18). Furthermore, we see from Fig. 5 that both approaches are able to find feasible parking trajectories, for all 84 considered initial conditions. Interestingly, we see that there are no obvious relations between the starting positions and the solution times.

The computation times of the parallel parking case are shown in Fig. 6 and Table I (lower half). Similar as in the reverse parking case, we see that both approaches have a 100% success rate and that, again due to the presence of the slack variables, the signed distance formulation requires longer computation time (1.67 s on average) than the distance formulation (0.87 s on average). Compared with reverse parking, we see that parallel parking is computationally more demanding. We believe that this is due to the fact that the paths in parallel parking are generally longer than in reverse parking since the car first needs to drive to the right before it can back into the parking lot (see also Fig. 2).

We close this section with the following remarks. First, we notice from Table I that the computation time of Hybrid A* is comparable to solving the (signed) distance optimization problem. Furthermore, the maximum overall computation time of Hybrid A* and signed distance reformulation is 7.7 s (reserve parking) and 9.2 s (parallel parking). This implies that, when initialized with Hybrid A*, the proposed collision avoidance framework enables practical real-time autonomous parking in tight environments. This is because stopping a

TABLE I
COMPUTATION TIME OF HYBRID A*, DISTANCE FORMULATION (15), AND SIGNED DISTANCE FORMULATION (18)

	min	max	mean
Reverse Parking			
warm start (Hybrid A*)	0.0315 s	3.2230 s	0.5491 s
distance formulation (15)	0.2111 s	2.7166 s	0.6046 s
signed distance formulation (18)	0.3200 s	4.4840 s	1.0344 s
Parallel Parking			
warm start (Hybrid A*)	0.0421 s	2.4766 s	0.3012 s
distance formulation (15)	0.2561 s	3.9885 s	0.8682 s
signed distance reformulation (18)	0.3850 s	6.7266 s	1.6703 s

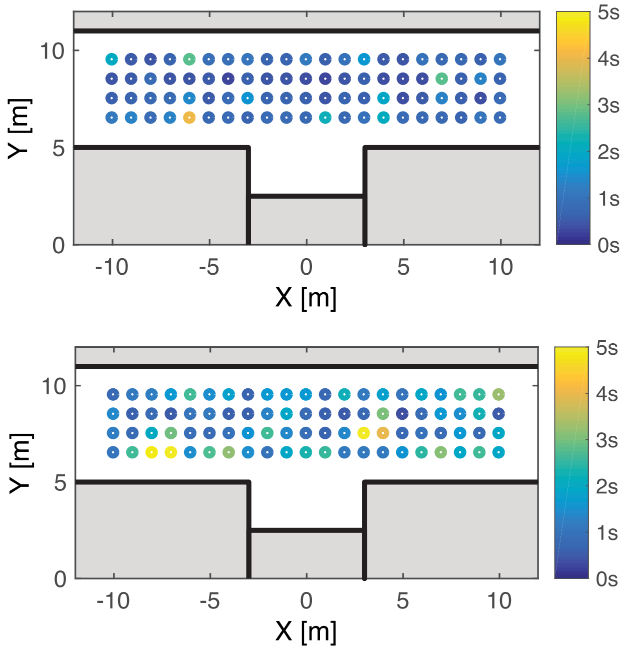


Fig. 6. Solution time for parallel parking with distance formulation (15) (top) and signed-distance formulation (18) (bottom).

vehicle and allowing it to plan for a few seconds is a viable approach in practice.[4] Second, we point out that, due to the time discretization, collisions in between the sampled times are not considered and are not ensured automatically. Similarly, the proposed framework currently does not consider other issues, such as sensor noise, disturbances, and/or model uncertainties. The interested reader is referred to [10] and [56] for techniques dealing with those two challenges.

E. Driveability and Comparison to Hybrid A* Trajectory

In this section, we compare the paths generated by (15) with the paths generated by Hybrid A*. More specifically, we compare the maneuver time, the tracking error, and the input cost (21) below when the respective paths are tracked with a simple path-following controller, which consists of a P-controller in the longitudinal direction and

[4]We also point out that the numerics were carried out on a 2013 MacBook Pro, and better computation times can be expected on an (future) autonomous vehicle that is generally equipped with multiple state-of-the-art high-performance processors.

TABLE II
COMPARISON OF MANEUVERING TIME, MAXIMUM TRACKING ERROR, AND INPUT COST

	min	max	mean
Reverse Parking			
maneuver time (15)	14.0 s	36.6 s	24.4 s
maneuver time (Hybrid A*)	30.3 s	63.6 s	44.2 s
max tracking error (15)	0.02 m	0.10 m	0.06 m
max tracking error (Hybrid A*)	0.04 m	0.14 m	0.07 m
input cost J_u (15)	0.41	1.25	0.71
input cost J_u (Hybrid A*)	1.77	2.98	2.15
Parallel Parking			
maneuver time (15)	17.8 s	60.9 s	38.3 s
maneuver time (Hybrid A*)	41.0 s	103.3 s	67.7 s
max tracking error (15)	0.04 m	0.12 m	0.08 m
max tracking error (Hybrid A*)	0.02 m	0.13 m	0.08 m
input cost J_u (15)	0.38	1.49	0.80
input cost J_u (Hybrid A*)	2.50	5.55	3.82

an linear quadratic regulator (LQR)-controller in the lateral direction. This is a common approach in vehicle path-following [57], and the implementation details are provided in the Appendix. Table II reports the maneuvering time, the maximum tracking error along each trajectory, and the averaged input cost

$$J_u = \frac{1}{N_{\text{PF}}}\left(\sum_{k=0}^{N_{\text{PF}}} u_k^\top R u_k + \Delta u_k^\top R_\Delta \Delta u_k\right) \tag{21}$$

where N_{PF} is the number of time steps, with respect to a sampling time of $T_{\text{PF}} = 0.05$ s, required by the path follower to finish the parking maneuvers, and $R = \text{diag}(0.01, 0.5)$, and $R_\Delta = \text{diag}(0.1, 0.1)$. With $u = (\delta, a)$, we see that steering is penalized significantly less than acceleration since constant steering is not problematic in parking. Changes in both inputs are penalized equally. Note that we consider such an average input cost to ensure that the cost is independent of the maneuver duration.

We see from Table II that the paths generated by (15) can be tracked more accurately than those generated by Hybrid A*, albeit not by a big margin. However, we can observe that the paths generated by Hybrid A* require, on average, twice as much time to track than those generated by (15). This is mainly due to the fact that Hybrid A* relies on a simplified model, which does not generate a velocity profile, and also because, due to the finite number of steering inputs, Hybrid A* allows for jumps in the steering command. Therefore, the path-following controller, which has to consider the steering rate constraints, must be more "cautious" when tracking these paths. In contrast, (15) is able to take into account the full system dynamics, which includes a velocity profile, as well as steering rate constraints. Moreover, we see from Table II that the input efforts required to follow the Hybrid A* paths are, on average, 3–4.75 times higher compared with (15). This is due to the fact that Hybrid A* does not take into account input rate cost and uses a finite number of inputs.

Summarizing Table II, we see that, while the paths generated by Hybrid A* are collision-free and kinematically feasible, they are challenging to track with the low-level path-following controllers and result in higher input cost because they do not incorporate information on the velocity and do not take into account the rate constraints in both steering and acceleration, requiring the vehicle to go slow in order to take “aggressive” maneuvers.

We conclude this section by pointing out that it is, in principle, possible to include a velocity profile and input rate constraints into Hybrid A* by augmenting the state (X, Y, φ) with (v, δ, a). While this is theoretically possible, it is generally not done in practice since gridding in higher dimensions will result in significantly longer computation times (the curse of dimensionality).

F. Discussion and Comparison to Sampling Methods

Sampling-based algorithms, such as RRT and RRT*, are popular path generation methods for autonomous parking problems [24]–[27]. In the following, we briefly review the basic idea behind RRT* for parking purposes and then highlight the conceptual differences between it and our optimization-based formulations (15) and (18).

The main idea of RRT* is to construct a search tree in the reduced state space (X, Y, φ) by randomly sampling the state space and connecting those samples to the existing search tree by means of the so-called steering function.[5] In general, finding the optimal steering function is very difficult and requires solving a (constrained) optimal control problem. However, under the simplifying assumptions that: 1) the vehicle travels at a constant velocity; 2) the objective is to find the shortest path; and 3) the environment is obstacle-free, the optimal steering function admits an analytic solution and is given by one of the so-called Reeds–Shepp curves [58], [59]. Indeed, the availability of those Reeds–Shepp curves allows RRT* to find obstacle-free paths [27]. Furthermore, it has been shown that, as the number of samples goes to infinity, then RRT* is indeed able to find the shortest-path solution [26].

While RRT* is often able to quickly find obstacle-free paths, its classical implementations, which rely on the Reeds–Shepp curves as steering functions are limited by two factors. First, because the Reeds–Shepp curves are derived from a vehicle model that assumes constant velocity, such RRT* methods generally do not provide a velocity profile. Second, without further modifications and for the same reason as above, RRT* is not able to take into account the input rate constraints. Recall that Hybrid A* also suffers from these two limitations, and interpolating the results of Section V-E, we may conclude that tracking a path generated by RRT* will be slower and result in higher cost when compared with following a path obtained by solving (15) and (18). We point out that, similar to Hybrid A*, it is in principle possible to include both a velocity profile and input/state rate constraints into RRT* by augmenting the (reduced) state (X, Y, φ) with (v, δ, a). This will, however, require sampling in higher dimensional state space and finding new steering functions, both of which can be computationally nontrivial. Indeed, the optimal steering functions may not admit analytic formulations anymore and require solving an optimal control problem instead to grow the search tree [60]. As a result, the RRT* algorithms that provide velocity profiles and consider input/state rate constraints may become considerably slower, and these modifications are generally not made in practice.

[5]Roughly speaking, the steering function solves the boundary value problem between the new sample and a node from the existing search tree, by taking into account the nonholonomic vehicle dynamics.

We conclude this section by pointing out that the proposed framework presents a unified approach toward motion planning for autonomous parking that, in a disciplined fashion, is able to consider generic objective functions, vehicle models, system constraints, and obstacle avoidance constraints. Numerical simulations indicate that the proposed methods are able to generate high-quality paths, which can easily be tracked by simple path-following controllers. Finally, we notice that since our approach takes into account the vehicle’s velocity, it is able to explicitly plan a path as a function of time, allowing *time-varying obstacles* to be naturally integrated into the path planning process.

VI. Conclusion

In this article, we presented smooth and exact reformulations for collision avoidance constraints for problems where the controlled object and the obstacle can be represented as the finite union of convex sets. We have shown that nondifferentiable polytopic obstacle constraints can be dealt with via dualization techniques to preserve differentiability, allowing the use of gradient- and Hessian-based optimization methods. The presented reformulation techniques are exact and nonconservative and apply equally to point-mass and full-dimensional controlled objects. Furthermore, in case collision-free trajectories cannot be generated, our framework allows us to find least-intrusive trajectories, measured in terms of penetration.

Our numerical studies, performed on an autonomous car parking example, indicate that, when appropriately initialized, the proposed framework is robust, real-time feasible, and able to generate dynamically feasible trajectories that satisfy the system constraints. We note that the exact method to initialize the numerical optimization algorithms is problem-dependent and should be chosen depending on the system at hand. Ongoing research focuses on appropriately warm starting the discretization time T_{opt}, as well as on the methods for further speeding up computation times.

Appendix A: Proof of Proposition 2

We need the following standard results from convex analysis.

Lemma 1: Let $\mathbb{C} \subset \mathbb{R}^n$ be a compact convex set.

1) Then, $\mathcal{H}_{\mathbb{C}}(z) := \{x \in \mathbb{R}^n : z^\top x \leq \max_{y\in\mathbb{C}} y^\top z\}$ is a supporting half-space with normal vector z, and

$$\mathbb{C} = \bigcap_{z:\, \|z\|=1} \mathcal{H}_{\mathbb{C}}(z) \tag{22}$$

for any norm $\|\cdot\|$.

2) Let $\partial\mathcal{H}_{\mathbb{C}}(z) := \{x \in \mathbb{R}^n : z^\top x = \max_{y\in\mathbb{C}} y^\top z\}$ be the supporting hyperplane with normal vector z. Then, for any $\bar{x} \in \mathbb{C}$, it holds that

$$\text{dist}(\bar{x}, \partial\mathcal{H}_{\mathbb{C}}(z)) = \frac{\max_{y\in\mathbb{C}}\{y^\top z\} - z^\top \bar{x}}{\|z\|_*} \tag{23}$$

where $\text{dist}(\cdot,\cdot)$ is defined as in (8a).

Proof of Proposition 2: First observe that $\text{pen}(\mathbb{E}(x), \mathbb{O}) = \text{dist}(\mathbb{E}(x), \mathbb{O}^{\complement})$, where $\mathbb{O}^{\complement} \subset \mathbb{R}^n$ denotes the complement of the set $\mathbb{O}$. Using this relationship and recalling the definition of $\text{dist}(\cdot,\cdot)$, it follows from (22) that $\text{pen}(\mathbb{E}(x), \mathbb{O}) = \inf_{\{z:\, \|z\|_*=1\}} \text{dist}(\mathbb{E}(x), \partial\mathcal{H}_{\mathbb{O}}(z))$, where we exploited the fact that (22) holds for any norm[6] and hence also the dual norm $\|\cdot\|_*$. Furthermore, it follows from (23) that, since $\|z\|_* = 1$, $\text{dist}(\mathbb{E}(x), \partial\mathcal{H}_{\mathbb{O}}(z)) = \max_{y\in\mathbb{O}}\{y^\top z\} - z^\top \mathbb{E}(x)$, which allows us to rewrite the penetration function as $\text{pen}(\mathbb{E}(x), \mathbb{O}) = \inf_{\{z:\, \|z\|_*=1\}}\{\max_{y\in\mathbb{O}}\{y^\top z\} - z^\top \mathbb{E}(x)\}$. To see that the min-max problem is equivalent to (11), we use strong duality of convex optimization to reformulate the inner maximization problem as $\max_{y\in\mathbb{O}}\{y^\top z\} = \min_\lambda\{b^\top\lambda : A^\top\lambda = z,\ \lambda \succeq_{\mathcal{K}^*} 0\}$. Hence, $\text{pen}(\mathbb{E}(x), \mathbb{O}) = \inf_{z,\lambda}\{b^\top\lambda - z^\top\mathbb{E}(x) : \|z\|_* = 1,\ A^\top\lambda = z,\ \lambda \succeq_{\mathcal{K}^*} 0\} = \inf_\lambda\{(b - A\mathbb{E}(x))^\top\lambda : \|A^\top\lambda\|_* = 1,\ \lambda \succeq_{\mathcal{K}^*} 0\}$. Finally, we have $\text{pen}(\mathbb{E}(x), \mathbb{O}) < \mathsf{p}_{\max} \Leftrightarrow \inf_\lambda\{(b - A\mathbb{E}(x))^\top\lambda : \|A^\top\lambda\|_* = 1,\ \lambda \succeq_{\mathcal{K}^*} 0\} < \mathsf{p}_{\max} \Leftrightarrow \lambda \succeq_{\mathcal{K}^*} 0 : (b - A\,\mathbb{E}(x))^\top\lambda < \mathsf{p}_{\max}$, which concludes the proof.

Appendix B: Sketch of Path-Following Controller

Following the literature, the path follower consists of a longitudinal controller and a lateral controller [57]. The longitudinal controller is a simple proportional controller of the form $a = P(v - v_{\text{ref}}(s)) + a_{\text{ff}}(s)$, where s is the progress along a given "reference" path, v_{ref} is the reference velocity, and $a_{\text{ff}}(s)$ is a feedforward term provided by the reference trajectory. To design the lateral controller, a change of coordinate from the global to the local error state, with respect to the reference path, is performed ("curvilinear abscissa"). Let $e = (n, \alpha)$ be the resulting error state, where n is the orthogonal distance to the reference path and α is the angular error; see [57, Ch. 2.5] for details. Similar to the longitudinal controller, the lateral controller consists of a feedback term and a feedforward term $\delta_{\text{ff}}(s)$, which depends on the curvature of the path. Upon linearization, the error dynamics takes the form

$$\dot{e} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \frac{v}{L} \end{bmatrix} \delta, \quad \delta = -Ke + \delta_{\text{ff}}(s).$$

The above-mentioned system is discretized with a sampling time of 50 ms, and a standard LQR-controller is used to stabilize the system. A sequential linearization approach is taken to deal with the velocity term v in the above-mentioned system matrix.

References

[1] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. Amer. Control Conf.*, May 2002, pp. 1936–1941.

[2] F. Borrelli, T. Keviczky, and G. J. Balas, "Collision-free UAV formation flight using decentralized optimization and invariant sets," in *Proc. 43rd IEEE Conf. Decis. Control*, vol. 1, Dec. 2004, pp. 1099–1104.

[6]Geometrically speaking, $\text{pen}(\mathbb{E}(x), \mathbb{O})$ is the minimum distance between $\mathbb{E}(x)$ and any supporting hyperplane $\partial\mathcal{H}_{\mathbb{O}}(z)$ of $\mathbb{O}$.

[3] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Berlin, Germany: Springer, 2006, pp. 65–93.

[4] L. Blackmore and B. Williams, "Optimal manipulator path planning with obstacles using disjunctive programming," in *Proc. Amer. Control Conf.*, Jun. 2006, p. 3.

[5] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proc. ASME Dyn. Syst. Control Conf.*, 2010, pp. 265–272.

[6] R. B. Patel and P. J. Goulart, "Trajectory generation for aircraft avoidance maneuvers using online optimization," *J. Guid., Control, Dyn.*, vol. 34, no. 1, pp. 218–230, Jan./Feb. 2011.

[7] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.

[8] M. Gerdts, R. Henrion, D. Hömberg, and C. Landry, "Path planning and collision avoidance for robots," *Numer. Algebra, Control Optim.*, vol. 2, no. 3, pp. 437–463, 2012.

[9] S. Sutrisno, E. Joelianto, A. Budiyono, I. E. Wijayanti, and N. Y. Megawati, "Model predictive control for obstacle avoidance as hybrid systems of small scale helicopter," in *Proc. 3rd Int. Conf. Instrum. Control Automat.*, Aug. 2013, pp. 127–132.

[10] J. Schulman *et al.*, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.

[11] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 Scale RC cars," *Optim. Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[12] J. V. Carrau, A. Liniger, X. Zhang, and J. Lygeros, "Efficient implementation of randomized MPC for miniature race cars," in *Proc. Eur. Control Conf.*, Jun. 2016, pp. 957–962.

[13] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *Proc. Amer. Control Conf.*, May 2017, pp. 5115–5120.

[14] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.

[15] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *Proc. IEEE Conf. Decis. Control*, Dec. 2018, pp. 4327–4332.

[16] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optim. Eng.*, vol. 3, no. 3, pp. 227–252, 2002.

[17] G. Schildbach and F. Borrelli, "A dynamic programming approach for nonholonomic vehicle maneuvering in tight environments," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2016, pp. 151–156.

[18] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998.

[19] K. Margellos and J. Lygeros, "Hamilton–Jacobi formulation for reach–Avoid differential games," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1849–1861, Aug. 2011.

[20] K. Margellos and J. Lygeros, "Toward 4-D trajectory management in air traffic control: A study based on Monte Carlo simulation and reachability analysis," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1820–1833, Sep. 2013.

[21] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *Int. J. Robot. Res.*, vol. 28, no. 8, pp. 933–945, 2009.

[22] D. Dmitri *et al*, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, 2010.

[23] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-heuristic a," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 224–243, 2016.

[24] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[25] J. Ziegler, J. Schroder, and M. Werling, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 787–791.

[26] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[27] H. Banzhaf, L. Palmieri, D. Nienhüser, T. Schamm, S. Knoop, and J. M. Zöllner, "Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, Oct. 2017, pp. 2239–2246.

[28] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 465–471.

[29] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA, USA: MIT Press, 1988.

[30] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[31] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[32] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[33] B. Paden, M. Čáp, S. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[34] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[35] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[36] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4569–4574.

[37] C. Park, J. Pan, and D. Manocha, "ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments," in *Proc. 22nd Int. Conf. Int. Conf. Automated Planning Scheduling*, 2012, pp. 207–215.

[38] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian optimization for motion planning," *Int. J. Robot. Res.*, vol. 32, nos. 9–10, pp. 1164–1193, 2013.

[39] L. Li, X. Long, and M. A. Gennert, "BiRRTOpt: A combined sampling and optimizing motion planner for humanoid robots," in *Proc. 16th Int. Conf. Hum. Robots*, Nov. 2016, pp. 469–476.

[40] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 450–457.

[41] U. Rosolia, S. de Bruyne, and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 469–484, Mar. 2017.

[42] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1696–1703, Feb. 2017.

[43] B. Li and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowl.-Based Syst.*, vol. 86, pp. 11–20, Sep. 2015.

[44] T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.

[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[46] R. Soloperto, J. Köhler, M. A. Müller, and F. Allgöwer, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control," in *Proc. 18th Eur. Control Conf.*, Jun. 2019, pp. 811–817.

[47] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[48] S. Cameron and R. Culley, "Determining the minimum translational distance between two convex polyhedra," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Apr. 1986, pp. 591–596.

[49] D. Dobkin, J. Hershberger, D. Kirkpatrick, and S. Suri, "Computing the intersection-depth of polyhedra," *Algorithmica*, vol. 9, no. 6, pp. 518–533, Jun. 1993.

[50] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[51] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," 2017, *arXiv:1711.03449*. [Online]. Available: https://arxiv.org/abs/1711.03449

[52] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," in *Proc. Amer. Control Conf.*, vol. 6, Jun. 2000, pp. 3860–3864.

[53] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[54] P. E. Hart, N. J. Nilsson, and B. Raphael, "Correction to 'A formal basis for the heuristic determination of minimum cost paths,'" in *Proc. ACM SIGART Bull.*, New York, NY, USA, no. 37, Dec. 1972, pp. 28–29.

[55] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Rev.*, vol. 59, no. 2, pp. 295–320, 2017.

[56] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, Aug. 2014.

[57] R. Rajamani, *Vehicle Dynamics and Control*. Boston, MA, USA: Springer, 2011.

[58] J.-D. Boissonnat, A. Cérézo, and J. Leblond, "Shortest paths of bounded curvature in the plane," *J. Intell. Robot. Syst.*, vol. 11, nos. 1–2, pp. 5–20, 1994.

[59] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, Oct. 1990.

[60] B. A. Paden, "A generalized label correcting method for optimal kinodynamic motion planning," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2017.

Xiaojing (George) Zhang received the B.Sc. and M.Sc. degrees in electrical engineering and information technology from ETH Zurich, Zürich, Switzerland, in 2010 and 2012, respectively, and the Ph.D. degree from the Automatic Control Laboratory, ETH Zurich, in January 2017.

He is currently a Post-Doctoral Fellow with the Model Predictive Control Laboratory and the Associate Director of the Hyundai Center of Excellence, University of California at Berkeley, Berkeley, CA, USA. His current research interests include the modeling, analysis, and control of stochastic uncertain systems, machine learning, randomized algorithms, and robust and stochastic optimization. Applications include autonomous driving, robotics, energy-efficient building climate control, and smart grids.

Alexander Liniger received the B.Sc. and M.Sc. degrees in mechanical engineering from the Department of Mechanical and Process Engineering, ETH Zurich, Zürich, Switzerland, in 2010 and 2013, respectively, and the Ph.D. degree from the Automatic Control Laboratory, ETH Zurich, in 2018.

He is currently a Post-Doctoral Fellow with the Computer Vision Laboratory, ETH Zurich. His current research interests include safe predictive control algorithms, end-to-end policy learning, and data-efficient reinforcement and imitation learning and their application to autonomous driving and racing.

Francesco Borrelli received the Laurea degree in computer science engineering from the University of Naples Federico II, Naples, Italy, in 1998, and the Ph.D. degree from ETH Zurich, Zürich, Switzerland, in 2002.

He is currently a Professor with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA, USA. He has authored more than 100 publications in the field of predictive control and has authored the book *Constrained Optimal Control of Linear and Hybrid Systems* (Springer-Verlag). His current research interests include constrained optimal control, model predictive control, and its application to advanced automotive control and energy-efficient building operation.

Dr. Borrelli was a recipient of the 2009 National Science Foundation CAREER Award and the 2012 IEEE Control System Technology Award. In 2008, he was appointed as the Chair of the IEEE Technical Committee on Automotive Control.