CHALMERS
UNIVERSITY OF TECHNOLOGY

Efficient Optimization-Based Trajectory Planning for Unmanned Systems in Confined Environments

Downloaded from: https://research.chalmers.se, 2026-09-07 18:54 UTC

Citation for the original published paper (version of record):
Fan, J., Murgovski, N., Liang, J. (2024). Efficient Optimization-Based Trajectory Planning for Unmanned Systems in Confined Environments. IEEE Transactions on Intelligent Transportation Systems, 25(11): 18547-18560. http://dx.doi.org/10.1109/TITS.2024.3436045

N.B. When citing this work, cite the original published paper.

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Efficient Optimization-Based Trajectory Planning for Unmanned Systems in Confined Environments

Jiayu Fan, Nikolce Murgovski, and Jun Liang

***Abstract*— This research addresses the increasing demand for advanced navigation systems capable of operating within confined surroundings. A significant challenge in this field is developing an efficient planning framework that can generalize across various types of collision avoidance missions. Utilizing numerical optimal control techniques, this study proposes a unified optimization-based planning framework to meet these demands. We focus on handling two collision avoidance problems, i.e., the object not colliding with obstacles and not colliding with boundaries of the constrained region. The object or obstacle is denoted as a union of convex polytopes and ellipsoids, and the constrained region is denoted as an intersection of such convex sets. Using these representations, collision avoidance can be approached by formulating explicit constraints that separate two convex sets, or ensure that a convex set is contained in another convex set, referred to as separating constraints and containing constraints, respectively. We propose to use the hyperplane separation theorem to formulate differentiable separating constraints, and utilize the S-procedure and geometrical methods to formulate smooth containing constraints. We state that compared to the state of the art, the proposed formulations allow a considerable reduction in nonlinear program size and geometry-based initialization in auxiliary variables used to formulate collision avoidance constraints. Finally, the efficacy of the proposed unified planning framework is evaluated in two contexts, autonomous parking in tractor-trailer vehicles and overtaking on curved lanes. The results in both cases exhibit an improved computational performance compared to existing methods.**

***Index Terms*— Optimization-based planning, trajectory planning, efficient collision avoidance, nonlinear optimization, hyperplane separation theorem, S-procedure.**

I. Introduction

A. Literature Review

THE optimization-based planning method holds immense potential in planning motions for unmanned systems since it allows to naturally include system dynamics as well as further constraints in formulations. It is known that the optimal control problem (OCP) serves as the basis for model predictive control (MPC). Since MPC requires repeated solutions of the OCP, the computational efficiency strongly depends on the capability of solving the OCP. Thus, for readability, the planning problem is often formulated as an OCP [1], [2], [3]. This is often followed by transcription of the OCP to a nonlinear program (NLP), which is then solved by robust numerical algorithms using, e.g., off-the-shelf solvers.

When utilizing an optimization-based method to plan trajectories for a controlled object, the most crucial concern revolves around implementing effective collision avoidance, requiring the object not to collide with obstacles and environment boundaries. Typically, obstacles take the form of nonconvex shapes and controlled objects themselves can be either convex or nonconvex shapes [4], [5], [6]. Furthermore, given that most controlled objects operate within constrained environments, e.g., vessel docking in limited regions [7] or car parking in tight parking lots [8], planning problems involving exact collision avoidance become inherently complex. In general, the planning problems are recognized as NP-hard [4], [10], [11], [12], [13], and the challenges intensify when controlled objects are expected to accurately and efficiently navigate nonconvex obstacles within confined surroundings. Since central to the optimization method is obtaining a computationally efficient NLP solution, the construction of smooth (at least twice differentiable) cost function and constraints [3], [4], [7] play a pivotal role. This research focuses on the investigation of explicit and differentiable collision avoidance constraints in fully dimensional domains.

When it comes to constructing collision avoidance constraints for controlled objects, several typical formulations have been developed. A natural choice is the disjunctive programming method [14], [15]. This method can explicitly model the non-convexity in the OCP and reformulate the OCP as a mixed-integer optimization problem. Earlier implementations have typically used the faces of the obstacles themselves to define convex safe regions. The requirement that a point is outside all obstacles is transformed to the requirement that, for each obstacle, the point must be on the outside of at least one face of that obstacle [16], [17]. For convex obstacles, these conditions are equivalent and have been successfully used to encode obstacle avoidance as a mixed-integer linear program [18]. Researchers have also investigated a nonconvex feasible region the vehicle resides in.

Manuscript received 16 January 2024; revised 19 June 2024 and 18 July 2024; accepted 28 July 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1600500; in part by the Chalmers Area of Advance Transport, Project EFFECT; and in part by China Scholarship Council under Grant 202206320304. The Associate Editor for this article was B. Du. *(Corresponding author: Jun Liang.)*

Jiayu Fan and Jun Liang are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: jiayu.fanhsz@gmail.com; jliang@zju.edu.cn).

Nikolce Murgovski is with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: nikolce.murgovski@chalmers.se).

Digital Object Identifier 10.1109/TITS.2024.3436045

1558-0016 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

An active research topic in this area is the efficient description of the nonconvex regions. Preliminary results show that these regions are often characterized by exploiting hyperplane arrangements (interested readers can refer to complementary materials in [19] and [20]). For example, the authors in [21] and [22] present a partitioning approach through equally-sized square cells, and then using this approach, they show that the collision avoidance problem becomes a resource allocation problem. Apart from these, in papers [23], [24] binary variables combined with the Big-M method [25] are employed to avoid collisions in autonomous overtaking scenarios. However, the mixed-integer formulation requires allocating multiple binary variables to collision constraints, which is computationally expensive [26], [27]. Various measures have been proposed to mitigate this issue. The authors in [14] propose to use hyperplane arrangements associated with binary variables for constraint description in a multi-obstacle environment. An over-approximation region is characterized by reducing to strictly binary formulations at the price of increasing conservatism. To accelerate the computation, the authors in [24], [28], and [29] put forward several approaches to reduce the number of binary variables in their respective case studies. However, finding a general approach has proven difficult. Another approach relies on the relaxation of mixed-integer constraints [30], [31], but these relaxation strategies may sacrifice the solution's accuracy.

In many applications, approximation methods typically resort to approximating nonconvex or convex shapes with simplistic shapes like ellipsoids and spheres [32], [33], [34], [35]. For instance, a controlled object and an obstacle with polyhedral shapes can be over-approximated by two ellipsoids. Since the distance functions between two ellipsoids have analytical expressions, collision avoidance can be easily achieved by requiring nonnegative distance. However, it can be noted that the movable region where a controlled object resides is reduced by this type of approximate method, so this way often leads to conservative solutions and may result in deadlock maneuvers [36], thus hindering controlled objects from finding feasible and collision-free trajectories in confined environments. Specifically, checking vertices of polygonal objects is also a practical way to formulate collision constraints. The idea is that a pair of convex polygons are not intersecting when their vertices are kept outside each other. Following this idea, the authors in [8], [9], and [37] impose collision constraints on vertices of the vehicle through triangle-area-based criterion in tight parking maneuver. It is noted that the constraints formulated on the triangle-area-based criterion are generally non-smooth and non-differentiable [8], leading to solutions at such points with inferior numerical properties.

Additionally, the research on collision avoidance between two convex shapes is emphasized since nonconvex shapes can be decomposed into a union of convex shapes [3], [4], [7], [38]. However, the decomposition is highly problem-independent and should be carefully selected to optimize computational performance in whatever algorithms that model is used. The effective decomposition ways refer to the reported methods in [16]. Following the decomposition idea, the authors in [3] first investigate the nonnegative distance between two convex sets, and then impose differentiable collision constraints by reformulating the distance via dualization techniques. Furthermore, to approach exact collision avoidance between two polytopes, the authors in [7] introduce two indicator constraints by exploiting Farkas' Lemma. Since, for many applications, a measure of the severity of collisions is of interest, the previous distance can be replaced with a signed distance between two convex sets, which can provide a measure of penetration depth in case of collision [3], [4], [7], [39], [40]. Through the strong duality of convex optimization, the requirement of a nonnegative signed distance can be exactly reformulated as smooth collision avoidance constraints. This approach has been applied to various kinds of applications spanning from robot avoidance maneuver [41], vessels docking [42], to autonomous car parking [26]. However, this type of dual reformulation between a pair of convex sets, whether using strong duality or Farkas' Lemma, needs adding quite a few dual variables and constraints, depending on the complexity of convex sets [3], [7]. The resulting growth in problem size can be computationally expensive.

B. Research Objectives and Contributions

In this study, we propose a comprehensive optimization framework for accurately and efficiently navigating a controlled object in a confined high-dimensional region with obstacles. We model the object or obstacle as a union of convex sets and the constrained region is denoted as an intersection of convex sets. Utilizing the defined representations, the challenges of collision avoidance in the context of ensuring that the controlled object avoids collisions with both obstacles and the boundaries of the constrained region can be reformulated as problems related to the separation of two convex sets and the containment of one convex set within another. In light of the requirement for smooth collision avoidance constraints within the optimization methods, we employ the hyperplane separation theorem to address the first separation problem, while the second containment problem is approached through the application of the S-procedure and geometrical methods. Furthermore, we detailed the delineation of explicit and differentiable collision avoidance constraints, which can be effectively resolved using readily available solvers through gradient- and Hessian-based algorithms. In summary, the contributions of this research can be summarized as follows:

1) We define a unified optimization-based planning framework for collision-free planning tasks regarding the controlled objects and obstacles represented by a union of convex polytopes and ellipsoids, and the constrained region represented by an intersection of such convex sets.
2) We propose smooth methods that ensure a convex polytope or ellipsoid is contained inside another convex polytope or ellipsoid using S-procedure and geometrical methods. The smoothness property allows the use of generic solvers with gradient- and Hessian-based algorithms.
3) We introduce efficient collision avoidance formulations by utilizing the hyperplane separation theorem to separate two convex sets, which admits a considerable

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

reduction in the OCP problem size compared to the existing distance reformulations using dual methods.

4) We propose novel initialization of auxiliary variables used to formulate collision avoidance constraints by primal methods. The primal methods can provide a direct geometric interpretation of the auxiliary variables, which is not obvious when using the existing dual methods.

The methods most closely related to ours are the work of [1] and [39], where the authors represent convex shapes using a vertex representation [1] and a support function representation [39]. Their collision avoidance formulations also show an improved computational performance compared to the dual reformulations. Our formulations differ from [1] and [39] in that we present a more considerable reduction in problem size, i.e., adding fewer auxiliary variables and fewer constraints. Furthermore, rather than utilizing the distance between objects as [1] and [39] do, our formulations using the hyperplane separation theorem directly find hyperplanes separating the objects, and allow finding good initial guesses in auxiliary variables. Additionally, we also study the problem regarding how to formulate smooth constraints under the condition of a convex set being a subset of a convex set.

The remainder of this paper is organized as follows. Section II formulates a unified OCP problem to describe the optimization-based planning framework. In Section III, notations to be utilized are defined. Then in Section IV separation constraints between a pair of convex sets are introduced using the hyperplane separation theorem. In Section V, containing constraints are present to ensure a convex set inside another convex set. Section VI illustrates the advantages in reducing problem size and well-initializing auxiliary variables compared to the state of the art. Section VII introduces two applications of the proposed optimization-based planning framework and states the efficacy of the proposed methods compared to existing methods in each application. Finally, Section VIII provides the concluding remarks and future plans.

II. Problem Statement

In this paper, trajectory planning problems are studied for a controlled object that navigates obstacles within a constrained region. The system motion is modeled as

$$\dot{\boldsymbol{\xi}} = f\left(\boldsymbol{\xi}, \boldsymbol{u}\right), \ \boldsymbol{\xi}(0) = \boldsymbol{\xi}_{\text{init}}, \tag{1}$$

where $\boldsymbol{\xi}$ is the state vector, $\boldsymbol{\xi}_{\text{init}}$ is the initial state, $\boldsymbol{u}$ is the control input and f is the system dynamics.

The constrained region (canvas) is modeled as an intersection of convex sets, denoted as $\mathcal{W}$, and the controlled object and obstacles are represented by a union of convex sets,[1] denoted as $\mathcal{B}$ and $\mathcal{O}$, respectively. The constrained OCP under study is planning the motion of the object, represented by the set $\mathcal{B}$, from its initial state $\boldsymbol{\xi}_{\textbf{init}}$ to a terminal state $\boldsymbol{\xi}_{\textbf{final}}$ while always residing within the constrained region $\mathcal{W}$, and not colliding with obstacles $\mathcal{O}$. The continuous-time constrained OCP is formulated as

$$\min_{\boldsymbol{\xi}, \boldsymbol{u}} \int_0^{t_\text{f}} \ell\left(\boldsymbol{\xi}(t), \boldsymbol{u}(t)\right) \mathrm{d}t \tag{2a}$$

$$\text{s.t. } \boldsymbol{\xi}(0) = \boldsymbol{\xi}_{\textbf{init}}, \ \boldsymbol{\xi}(t_\text{f}) = \boldsymbol{\xi}_{\textbf{final}}, \tag{2b}$$

$$\dot{\boldsymbol{\xi}}(t) = f\left(\boldsymbol{\xi}(t), \boldsymbol{u}(t)\right), \tag{2c}$$

$$\boldsymbol{\xi}(t) \in \mathcal{X}, \ \boldsymbol{u}(t) \in \mathcal{U}, \tag{2d}$$

$$\mathcal{B}(\boldsymbol{\xi}) \subset \mathcal{W}, \tag{2e}$$

$$\mathcal{B}(\boldsymbol{\xi}) \cap \mathcal{O} = \emptyset, \tag{2f}$$

where t_f is the final time, and $\mathcal{X}$ and $\mathcal{U}$ are the admissible sets of state and control, respectively. The function ℓ represents the time-dependent cost. Constraints (2c)-(2f) are imposed for $\forall t \in [0, t_\text{f}]$. For the sake of clarity, the time dependence of $\mathcal{O}$ is omitted, but we remark that the proposed methods can directly be applied to problems with moving obstacles. Constraints (2e) are imposed to prevent the controlled object from colliding with the constrained region boundaries (faces), referred to as containing constraints, and constraints (2f) enforces collision avoidance between the object and obstacles, referred to separating constraints. Note that if the problem (2) is to be solved by generic solvers using gradient-based algorithms, constraints (2e) and (2f) should be transformed into smooth constraints in an explicit form. In the following, we outline general propositions about containing constraints (2e) and separating constraints (2f) and then exactly reformulate (2e) and (2f) as explicit and differentiable constraints.

III. Notations

A convex polytope can be represented as a linear matrix inequality

$$\mathcal{P}(P, \boldsymbol{p}, \boldsymbol{s}) = \{\boldsymbol{s} \in \mathbb{R}^N \mid P\boldsymbol{s} \leq \boldsymbol{p}\}, \tag{3}$$

where $P = \left[\boldsymbol{p}_{1,P}, \ldots, \boldsymbol{p}_{N_P,P}\right]^\top \in \mathbb{R}^{N_P \times N}$, $\boldsymbol{p} \in \mathbb{R}^{N_P}$. Here, $\boldsymbol{p}_{k,P}^\top$ and p_k denote the k-th row vector in P and the k-th entry in $\boldsymbol{p}$, respectively.

The symbol $V_P = \left[\boldsymbol{v}_{1,P}, \ldots, \boldsymbol{v}_{N_P,P}\right] \in \mathbb{R}^{N \times N_P}$ represents a matrix of vertices in polytope $\mathcal{P}$. Symbol $\boldsymbol{v}_{i,P}$ describes the position of the i-th vertex of polytope $\mathcal{P}$.

An ellipsoid can be denoted as a quadratic inequality

$$\mathcal{E}(E, \boldsymbol{e}, \boldsymbol{s}) = \left\{\boldsymbol{s} \in \mathbb{R}^N \mid (\boldsymbol{s} - \boldsymbol{e})^\top E\,(\boldsymbol{s} - \boldsymbol{e}) \leq 1\right\}, \tag{4}$$

where $\boldsymbol{e} \in \mathbb{R}^N$ denotes the center of the ellipsoid and $E \in \mathbb{R}^{N \times N}$ is a positive definite, symmetric matrix defining the shape.

A hyperplane with normal vector $\boldsymbol{\lambda} \in \mathbb{R}^N$ can be defined as

$$\mathcal{H}(\boldsymbol{\lambda}, \mu) = \left\{\boldsymbol{s} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{s} = \mu\right\}. \tag{5}$$

The hyperplane $\mathcal{H}(\boldsymbol{\lambda}, \mu)$ divides the whole space $\mathbb{R}^N$ into two closed half-spaces, denoted as

$$\mathcal{H}^-(\boldsymbol{\lambda}, \mu) = \left\{\boldsymbol{s} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{s} \leq \mu\right\}, \tag{6}$$

and

$$\mathcal{H}^+(\boldsymbol{\lambda}, \mu) = \left\{\boldsymbol{s} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{s} \geq \mu\right\}. \tag{7}$$

[1] In the commonly used scenarios, such as car parking [3], [36] and vessel docking [42] cases, most nonconvex object can be decomposed into a union of convex polytopes and ellipsoids, so in this study, we focus on convex polytopes and ellipsoids.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

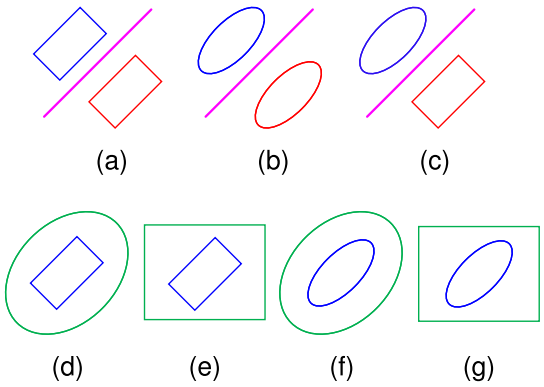


Fig. 1. For readability, this figure illustrates the studied problems in two dimensions as examples. In subplots (a)–(g) the controlled object $\mathcal{B}_m$ is depicted in blue. In (a)–(c) the obstacle $\mathcal{O}_n$ is depicted in red. In (d)–(g) the constrained region $\mathcal{W}_l$ is depicted in green. (a)–(c) illustrate the problem of $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$, i.e., (a) Separation between two polytopes; (b) Separation between two ellipsoids; (c) Separation between a polytope and an ellipsoid. (d)–(g) illustrate the problem of $\mathcal{B}_m \subset \mathcal{W}_l$, i.e., (d) Polytope contained in ellipsoid; (e) Polytope contained in polytope; (f) Ellipsoid contained in ellipsoid; (g) Ellipsoid contained in polytope.

In the following sections, the constrained region is described by an intersection of convex polytopes and ellipsoids, denoted as $\mathcal{W} = \bigcap_l \mathcal{W}_l$. The controlled object or obstacle is described using a union of convex polytopes and ellipsoids, denoted as $\mathcal{B} = \bigcup_m \mathcal{B}_m$, $\mathcal{O} = \bigcup_n \mathcal{O}_n$. Sets $\mathcal{W}_l, \mathcal{B}_m, \mathcal{O}_n$ are either polytopes or ellisposids, described using representations (3) and (4).

IV. Separating Constraints

In this section, we aim at formulating effective collision avoidance constraints between the controlled object and obstacles, denoted as $\mathcal{B}(\boldsymbol{\xi}) \cap \mathcal{O} = \emptyset$ in (2). Since $\mathcal{B} = \bigcup_m \mathcal{B}_m$, $\mathcal{O} = \bigcup_m \mathcal{O}_n$, separating constraints $\mathcal{B}(\boldsymbol{\xi}) \cap \mathcal{O} = \emptyset$ can be represented as

$$\mathcal{B} \cap \mathcal{O} = \emptyset \iff \mathcal{B}_m \cap \mathcal{O}_n = \emptyset, \ \forall m, n. \tag{8}$$

As mentioned, set $\mathcal{B}_m$ or $\mathcal{O}_n$ can either be a convex polytope or an ellipsoid. To approach (8), the conditions between two polytopes, between ellipsoids, and between a polytope and an ellipsoid need to be considered, respectively (see Fig. 1). In the following, we first outline the hyperplane separation theorem and then two general propositions are stated for a polytope contained inside a convex set and an ellipsoid contained inside a half-space. Finally the requirement of $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ is reformulated as explicit constraints by exploiting the propositions.

Theorem 1 (Hyperplane Separation Theorem [43]): Let $\mathcal{C}^1$ and $\mathcal{C}^2$ represent two compact convex sets in $\mathbb{R}^N$. Then,

$$\mathcal{C}^1 \cap \mathcal{C}^2 = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N \backslash \mathbf{0}, \mu \in \mathbb{R}: \\ \forall \boldsymbol{q} \in \mathcal{C}^1, \forall \boldsymbol{p} \in \mathcal{C}^2, \boldsymbol{\lambda}^\top \boldsymbol{q} \geq \mu, \boldsymbol{\lambda}^\top \boldsymbol{p} \leq \mu. \tag{9}$$

Proposition 1 (Polytope within a convex set): Let a convex set be denoted as $\mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K}) = \{\boldsymbol{s} \in \mathbb{R}^N \mid \Omega \boldsymbol{s} \preceq_{\mathcal{K}} \boldsymbol{\mu}\}$[2] where $\Omega \in \mathbb{R}^{M \times N}$, $\boldsymbol{\mu} \in \mathbb{R}^M$, and $\mathcal{K} \subset \mathbb{R}^M$ is a closed convex pointed cone with nonempty interior. Let $\mathcal{V}_\mathcal{P}$ represent the set of vertices of polytope $\mathcal{P}$ denoted as (3). Then,

$$V_\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K}) \iff \mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K}). \tag{10}$$

Proof: Since any vertex in $\mathcal{V}_\mathcal{P}$ belongs to set $\mathcal{P}$, we have $\mathcal{V}_\mathcal{P} \subset \mathcal{P}$. Then, if $\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$ it follows that $\mathcal{V}_\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$. This proves the right-to-left implication. The left-to-right implication, given $\mathcal{V}_\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$ is proven by contradiction. Assume $\exists \boldsymbol{s} \in \mathcal{P}$, $\boldsymbol{s} \notin \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$, such that $\Omega \boldsymbol{s} \succ_{\mathcal{K}} \boldsymbol{\mu}$. It is known that any interior point of a convex polytope can be expressed as a convex combination of its vertices, where all coefficients are non-negative [46]. Let $\boldsymbol{s} = \sum_{i=1}^{N_m} \alpha_i \boldsymbol{v}_{i,P}$, where $\alpha_i \geq 0$. Then, $\sum_{i=1}^{N_m} \alpha_i \Omega \boldsymbol{v}_{i,P} \succ_{\mathcal{K}} \boldsymbol{\mu}$, indicates that $\exists i$ for which $\Omega \boldsymbol{v}_{i,P} \succ_{\mathcal{K}} \boldsymbol{\mu}$, i.e., $\boldsymbol{v}_{i,P} \notin \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$, which contradicts with $\mathcal{V}_\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$. So, it follows that $\forall \boldsymbol{s} \in \mathcal{P}$, $\boldsymbol{s} \in \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$, i.e., $\mathcal{P} \subset \mathcal{C}(\Omega, \boldsymbol{\mu}, \mathcal{K})$, which completes the proof. □

Proposition 2 (Ellipsoid Within a Half-Space): Let an ellipsoid $\mathcal{E}$ be denoted as (4). Let half-space $\mathcal{H}^-(\boldsymbol{\lambda}, \mu)$ be denoted as (6). Then,

$$\mathcal{E} \subset \mathcal{H}^-(\boldsymbol{\lambda}, \mu) = \emptyset \iff \sqrt{\boldsymbol{\lambda}^\top E^{-1} \boldsymbol{\lambda}} + \boldsymbol{\lambda}^\top \boldsymbol{e} \leq \mu. \tag{11}$$

Proof: It is known that an ellipsoid can be described as an affine transformation of the unit sphere. Using the representation in (4), $\mathcal{E}$ can be re-denoted as $\mathcal{E} := \{M\boldsymbol{s} + \boldsymbol{e} \mid \boldsymbol{s} \in \mathbb{R}^N, \|\boldsymbol{s}\| \leq 1\}$, where M is a positive definite matrix obtained by $M^\top M = E^{-1}$ [48]. Thus, the requirement of $\mathcal{E} \subset \mathcal{H}^-(\boldsymbol{\lambda}, \mu) = \emptyset$ is identical to stating an optimization problem

$$\max_{\boldsymbol{s}} \boldsymbol{\lambda}^\top (M\boldsymbol{s} + \boldsymbol{e}) \leq \mu, \|\boldsymbol{s}\| \leq 1.$$

According to the well-known Cauchy-Schwartz inequality [49], $\boldsymbol{\lambda}^\top \boldsymbol{s} \leq \|\boldsymbol{\lambda}\| \|\boldsymbol{s}\|$. Since $\|\boldsymbol{s}\| \leq 1$, $\boldsymbol{\lambda}^\top \boldsymbol{s} \leq \|\boldsymbol{\lambda}\|$, where the right part is obtained by the vector $\boldsymbol{s} = \boldsymbol{\lambda}/\|\boldsymbol{\lambda}\|$, so $\max_{\boldsymbol{s}} \boldsymbol{\lambda}^\top (M\boldsymbol{s} + \boldsymbol{e}) = \|\boldsymbol{\lambda} M^\top\| + \boldsymbol{\lambda}^\top \boldsymbol{e} = \sqrt{\boldsymbol{\lambda}^\top E^{-1} \boldsymbol{\lambda}} + \boldsymbol{\lambda}^\top \boldsymbol{e} \leq \mu$, which completes the proof. □

In the following, we discuss three separation questions, i.e., how to formulate $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ when set $\mathcal{B}_m$ is a polytope and $\mathcal{O}_n$ is a polytope, set $\mathcal{B}_m$ is an ellipsoid and $\mathcal{O}_n$ is an ellipsoid, set $\mathcal{B}_m$ is a polytope and $\mathcal{O}_n$ is an ellipsoid.

A. Separation Between Two Polytopes

According to Theorem 1, when compact sets $\mathcal{C}^1$ and set $\mathcal{C}^2$ are disjoint, there exists a separating hyperplane between them, i.e., points belonging to $\mathcal{C}^1$ are on one side of the hyperplane while points of $\mathcal{C}^2$ are on the other side. However, checking all points of $\mathcal{C}^1$ and $\mathcal{C}^2$ is computationally expensive. One efficient way is only checking vertices of $\mathcal{C}^1$ and $\mathcal{C}^2$ when these sets are polytopes.

Proposition 3: Let sets $\mathcal{B}_m$, $\mathcal{O}_n$ be denoted using polytopes $\mathcal{P}(P_1, \boldsymbol{p}_1, \boldsymbol{s})$ and $\mathcal{P}(P_2, \boldsymbol{p}_2, \boldsymbol{s})$ based on (3). Let V_{P_1} and V_{P_2} be matrices of vertices as stated earlier. Then,

$$\mathcal{B}_m \cap \mathcal{O}_n = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N, \mu \in \mathbb{R}: \\ \boldsymbol{\lambda}^\top V_{P_1} \geq \mu \mathbf{1}^\top, \boldsymbol{\lambda}^\top V_{P_2} \leq \mu \mathbf{1}^\top, \|\boldsymbol{\lambda}\| > 0. \tag{12}$$

Proof: According to (3), sets $\mathcal{B}_m$ and $\mathcal{O}_n$ are closed and bounded. Since a closed and bounded set of R^N is

[2]This form is entirely generic since any compact convex set admits a conic representation. A polytope can be represented by choosing $\mathcal{K} = \mathbb{R}^N_+$. An ellipsoid is represented by letting $\mathcal{K}$ be the second-order cone [3], [44], [45].

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

compact [47], $\mathcal{B}_m$ and $\mathcal{O}_n$ are compact sets. Thus, based on the hyperplane separation theorem $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ is identical to stating that there exists a separating hyperplane between them. Let $\mathcal{H}^+(\boldsymbol{\lambda}, \mu)$ and $\mathcal{H}^-(\boldsymbol{\lambda}, \mu)$ be defined as stated earlier, i.e., $\mathcal{H}^-(\boldsymbol{\lambda}, \mu) = \left\{\boldsymbol{q} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{q} \leq \mu\right\}$ and $\mathcal{H}^+(\boldsymbol{\lambda}, \mu) = \left\{\boldsymbol{q} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{q} \geq \mu\right\}$. Assume $\mathcal{B}_m \subset \mathcal{H}^+(\boldsymbol{\lambda}, \mu)$, $\mathcal{O}_n \subset \mathcal{H}^-(\boldsymbol{\lambda}, \mu)$. Using Proposition 1, formulations $\mathcal{B}_m \subset \mathcal{H}^+(\boldsymbol{\lambda}, \mu)$, $\mathcal{O}_n \subset \mathcal{H}^-(\boldsymbol{\lambda}, \mu)$ can be reformulated by ensuring that the vertices of $\mathcal{B}_m$ are in half-space $\mathcal{H}^+(\boldsymbol{\lambda}, \mu)$ and vertices of $\mathcal{O}_n$ are in the other half-space $\mathcal{H}^-(\boldsymbol{\lambda}, \mu)$, i.e., $\boldsymbol{\lambda}^\top V_{P_1} \geq \mu \mathbf{1}^\top, \boldsymbol{\lambda}^\top V_{P_2} \leq \mu \mathbf{1}^\top$. Additionally, $\|\boldsymbol{\lambda}\| > 0$ is the sufficient and necessary condition of $\boldsymbol{\lambda} \neq \mathbf{0}$, ensuring the existence of such a separating hyperplane, which completes the proof. □

B. Separation Between Two Ellipsoids

Proposition 4: Let sets $\mathcal{B}_m$, $\mathcal{O}_n$ be denoted using ellipsoids $\mathcal{E}(E_1, \boldsymbol{e}_1, \boldsymbol{s})$ and $\mathcal{E}(E_2, \boldsymbol{e}_2, \boldsymbol{s})$ based on (4). Then,

$$\mathcal{B}_m \cap \mathcal{O}_n = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N, \mu \in \mathbb{R} : \|\boldsymbol{\lambda}\| > 0, \sqrt{\boldsymbol{\lambda}^\top E_1^{-1} \boldsymbol{\lambda}} + \boldsymbol{\lambda}^\top \boldsymbol{e_1} \leq \mu, -\sqrt{\boldsymbol{\lambda}^\top E_2^{-1} \boldsymbol{\lambda}} + \boldsymbol{\lambda}^\top \boldsymbol{e_2} \geq \mu. \tag{13}$$

Proof: For the same argumentation as the Proof of Proposition 3, sets $\mathcal{B}_m$ and $\mathcal{O}_n$ are compact, and then using hyperplane separation theorem $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ is identical to stating they can be separated by a hyperplane. And then Proposition 4 can be approached by using Proposition 2. □

C. Separation Between a Polytope and an Ellipsoid

Proposition 5: Let set $\mathcal{B}_m$ be denoted using polytope $\mathcal{P}(P, \boldsymbol{p}, \boldsymbol{s})$ as (3). let V_P be stated earlier. Let set $\mathcal{O}_n$ be denoted using ellipsoid $\mathcal{E}(E, \boldsymbol{e}, \boldsymbol{s})$ as (4). Then,[3]

$$\mathcal{B}_m \cap \mathcal{O}_n = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N, \mu \in \mathbb{R} : \boldsymbol{\lambda}^\top V_P \geq \mu \mathbf{1}^\top, \sqrt{\boldsymbol{\lambda}^\top E_1^{-1} \boldsymbol{\lambda}} + \boldsymbol{\lambda}^\top \boldsymbol{e_1} \leq \mu, \|\boldsymbol{\lambda}\| > 0. \tag{14}$$

Proof: Following the proof of Proposition 3 and Proposition 4, the hyperplane separation theorem is utilized to formulate Proposition 5. □

Remark 1: Notably, collision formulations related to the ellipsoid representation, e.g., in Proposition 4 and Proposition 5, involve the square root, which is not actually differentiable at the origin. However, such a singularity in its gradient at the origin is excluded because $\|\boldsymbol{\lambda}\| > 0$ [1], [39]. In practice, it can be approached by $\|\boldsymbol{\lambda}\| \geq \epsilon$, where ϵ is set as a small value, to ensure $\|\boldsymbol{\lambda}\|$ strictly positive. Additionally, the proposed geometry-based guesses, covered in Section IV-B, guarantee that $\boldsymbol{\lambda}$ is initialized sufficiently away from the origin, so numerical solvers do not run into problems. This is also supported by the numerical results in Section VII.

V. Containing Constraints

In this part, we continue to discuss the explicit form of containing constraints (2e) in problem (2). Since the controlled object is denoted as a union and the constrained region is represented as an intersection, i.e., $\mathcal{B} = \bigcup_m \mathcal{B}_m$, $\mathcal{W} = \bigcap_l \mathcal{W}_l$, containing constraints (2e) can be substituted with

$$\mathcal{B} \subset \mathcal{W} \iff \mathcal{B}_m \subset \mathcal{W}_l, \ \forall m, l, \tag{15}$$

where (15) means that set B_m is a subset of set $\mathcal{W}_l$. The sole criterion for $\mathcal{B}_m \subset \mathcal{W}_l$ is identical to stating that points of set $\mathcal{B}_m$ belong to set $\mathcal{W}_l$. However, checking all points of $\mathcal{B}_m$ is impossible in reality. Next, by using the previous propositions and S-procedure, we explicitly formulate $\mathcal{B}_m \subset \mathcal{W}_l$ when B_m and $\mathcal{W}_l$ are represented as convex polytopes and ellipsoids, respectively (see Fig. 1).

A. A Polytope Contained in a Polytope or an Ellipsoid

When set $\mathcal{B}_m$ is a polytope, $\mathcal{B}_m \subset \mathcal{W}_l$ is identical to stating that the vertices of $\mathcal{B}_m$ are inside $\mathcal{W}_l$ by using Proposition 1. Since $\mathcal{W}_l$ is denoted as either a convex polytope or ellipsoid, the containing constraint $\mathcal{B}_m \subset \mathcal{W}_l$ is formulated as a linear inequality or quadratic inequality.

Proposition 6: Let set $\mathcal{B}_m$ be denoted using polytope $\mathcal{P}(P_1, \boldsymbol{p}_1, \boldsymbol{s})$ as (3). Let set $\mathcal{W}_l$ be denoted using polytope $\mathcal{P}(P_2, \boldsymbol{p}_2, \boldsymbol{s})$ as (3) or ellipsoid $\mathcal{E}(E, \boldsymbol{e}, \boldsymbol{s})$ as (4). Then, $\mathcal{B}_m \subset \mathcal{W}_l \iff$

$$\begin{cases} P_2 \boldsymbol{v}_{i,P_1} \leq \boldsymbol{p}_2, \ \forall i & \text{if } \mathcal{W}_l := (3) \\ \left(\boldsymbol{v}_{i,P_1} - \boldsymbol{e}\right)^\top E \left(\boldsymbol{v}_{i,P_1} - \boldsymbol{e}\right) \leq 1, \ \forall i & \text{if } \mathcal{W}_l := (4) \end{cases}. \tag{16}$$

Proposition 3 shows that $\mathcal{B}_m \subset \mathcal{W}_l$, in the condition of a polytope $\mathcal{B}_m$, can be formulated by imposing constraints on finite vertices of set $\mathcal{B}_m$. However, when set $\mathcal{B}_m$ is described by an ellipsoid as (4), $\mathcal{B}_m \subset \mathcal{W}_l$ cannot be ensured by checking vertices of $\mathcal{B}_m$. To handle this, we propose to formulate $\mathcal{B}_m \subset \mathcal{W}_l$ by S-procedure and geometrical methods, respectively. Next, we discuss two questions, i.e., how to formulate $\mathcal{B}_m \subset \mathcal{W}_l$ when an ellipsoid $\mathcal{B}_m$ is a subset of an ellipsoid $\mathcal{W}_l$ and an ellipsoid $\mathcal{B}_m$ is a subset of a polytope $\mathcal{W}_l$, respectively.

B. An Ellipsoid Contained in an Ellipsoid

Lemma 1 (S-Procedure [50]): Let $F_1, F_2 \in \mathbb{R}^{N \times N}$, $\boldsymbol{g_1}, \boldsymbol{g_2} \in \mathbb{R}^N$, and $h_1, h_2 \in \mathbb{R}$. If $F_1 = F_1^\top, F_2 = F_2^\top$ and $\exists \boldsymbol{x} \in \mathbb{R}^N$ for which the strict inequality $\boldsymbol{x}^\top F_1 \boldsymbol{x} + 2\boldsymbol{g_1}^\top \boldsymbol{x} + h_1 < 0$, then

$$\boldsymbol{x}^\top F_1 \boldsymbol{x} + 2\boldsymbol{g_1}^\top \boldsymbol{x} + h_1 \leq 0 \Longrightarrow \boldsymbol{x}^\top F_2 \boldsymbol{x} + 2\boldsymbol{g_2}^\top \boldsymbol{x} + h_2 \leq 0, \tag{17}$$

$$\iff \exists \lambda \geq 0, \ \lambda \begin{bmatrix} F_1 & \boldsymbol{g_1} \\ \boldsymbol{g_1}^\top & h_1 \end{bmatrix} - \begin{bmatrix} F_2 & \boldsymbol{g_2} \\ \boldsymbol{g_2}^\top & h_2 \end{bmatrix} \succeq 0. \tag{18}$$

Proposition 7: Let set $\mathcal{B}_m$ be denoted using ellipsoid $\mathcal{E}(E_1, \boldsymbol{e}_1, \boldsymbol{s})$ as (4) and let set $\mathcal{W}_l$ be denoted using ellipsoid $\mathcal{E}(E_2, \boldsymbol{e}_2, \boldsymbol{s})$ as (4). Then, $\mathcal{B}_m \subset \mathcal{W}_l \iff$

$$\exists \Upsilon \in \mathbb{R}^{(N+1)\times(N+1)}, \lambda \geq 0 : \begin{bmatrix} \lambda E_1 - E_2 & E_2^\top \boldsymbol{e}_2 - \lambda E_1^\top \boldsymbol{e}_1 \\ \boldsymbol{e}_2^\top E_2 - \lambda \boldsymbol{e}_1^\top E_1 & \boldsymbol{e}_1^\top \lambda E_1 \boldsymbol{e}_1 - \boldsymbol{e}_2^\top E_2 \boldsymbol{e}_2 - \lambda + 1 \end{bmatrix} = \Upsilon \Upsilon^\top, \Upsilon(i,i) \geq 0, \ i = 1 \cdots N+1. \tag{19}$$

Proof: In representations (6) and (4), E_1 and E_2 are positive definite, symmetric matrices, i.e., $E_1 = E_1^\top$,

[3]The other way around, i.e., set $\mathcal{B}_m$ is an ellipsoid and set $\mathcal{O}_n$ is a polytope, works as well for formulation (14).

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$E_2 = E_2^\top$. Set $\mathcal{B}_m$ is a convex set with nonempty interior, so $(\boldsymbol{s}-\boldsymbol{e}_1)^\top E_1 (\boldsymbol{s}-\boldsymbol{e}_1) < 1$ holds. Thus, constraint $\mathcal{B}_m \subset \mathcal{W}_l$ is identical to stating that $\forall \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{x}^\top E_1 \boldsymbol{x} - 2\boldsymbol{e}_1^\top E_1 \boldsymbol{x} + \boldsymbol{e}_1^\top E_1 \boldsymbol{e}_1 - 1 \le 0 \Longrightarrow \boldsymbol{x}^\top E_2 \boldsymbol{x} - 2\boldsymbol{e}_2^\top E_2 \boldsymbol{x} + \boldsymbol{e}_2^\top E_2 \boldsymbol{e}_2 - 1 \le 0$. Based on Lemma 1, $\exists \lambda \ge 0$, for which the matrix

$$\begin{bmatrix} \lambda E_1 - E_2 & E_2^\top \boldsymbol{e}_2 - \lambda E_1^\top \boldsymbol{e}_1 \\ \boldsymbol{e}_2^\top E_2 - \lambda \boldsymbol{e}_1^\top E_1 & \boldsymbol{e}_1^\top \lambda E_1 \boldsymbol{e}_1 - \boldsymbol{e}_2^\top E_2 \boldsymbol{e}_2 - \lambda + 1 \end{bmatrix},$$

denoted as symbol G for clarity, is a positive semidefinite matrix, i.e., $G \succeq 0$. Next, we prove $G \succeq 0 \iff \exists \Upsilon \in \mathbb{R}^{(N+1)\times(N+1)}$, such that $G = \Upsilon\Upsilon^\top, \Upsilon(i,i) \ge 0, i = 1 \cdots N+1$.

According to the Cholesky theory [51], $G \succeq 0$ has a decomposition of the form $G = LL^\top$ where $L \in \mathbb{R}^{(N+1)\times(N+1)}$ is a lower triangular matrix with nonnegative diagonal entries, so L is a special case of Υ. This proves the left-to-right implication. Based on the definition of positive semidefinite matrix, $\forall \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{x}^\top \Upsilon\Upsilon^\top \boldsymbol{x} = \|\boldsymbol{x}^\top \Upsilon\|^2 \ge 0$, let $G = \Upsilon\Upsilon^\top$, thus $G \succeq 0$, which completes the right-to-left statement. □

C. An Ellipsoid Contained in a Polytope

Proposition 8: Let set $\mathcal{B}_m$ be denoted using ellipsoid $\mathcal{E}(E, \boldsymbol{e}, \boldsymbol{s})$ as (4). Let set $\mathcal{W}_l$ be denoted using polytope $\mathcal{P}(P, \boldsymbol{p}, \boldsymbol{s})$ as (3). Then, $\mathcal{B}_m \subset \mathcal{W}_l \iff$

$$\sqrt{\boldsymbol{p}_{i,P}^\top E^{-1} \boldsymbol{p}_{i,P}} + \boldsymbol{p}_{i,P}^\top \boldsymbol{e} \le p_i, \ i = 1 \cdots N_P. \tag{20}$$

Proof: Let $\mathcal{H}^-(\boldsymbol{\lambda}, \mu)$ be defined as stated earlier, i.e., $\mathcal{H}^-(\boldsymbol{\lambda}, \mu) = \{\boldsymbol{s} \in \mathbb{R}^N \mid \boldsymbol{\lambda}^\top \boldsymbol{s} \le \mu\}$. From (3), polytope $\mathcal{W}_l$ is represented as $\{\boldsymbol{s} \in \mathbb{R}^N \mid P\boldsymbol{s} \le \boldsymbol{p}\}$, i.e., the intersection of N_P half-spaces, denoted as $\mathcal{W}_l = \mathcal{H}^-(\boldsymbol{p}_{1,P}, p_1) \cap \mathcal{H}^-(\boldsymbol{p}_{2,P}, p_2) \cdots \cap \mathcal{H}^-(\boldsymbol{p}_{N_P,P}, p_{N_P})$. Thus, constraint $\mathcal{B}_m \subset \mathcal{W}_l$ is identical to stating $\mathcal{B}_m \subset \mathcal{H}^-(\boldsymbol{p}_{1,P}, p_1)$, $\mathcal{B}_m \subset \mathcal{H}^-(\boldsymbol{p}_{2,P}, p_2)$, $\cdots, \mathcal{B}_m \subset \mathcal{H}^-(\boldsymbol{p}_{N_P,P}, p_{N_P})$ hold. Then, based on Proposition 2, $\mathcal{B}_m \subset \mathcal{W}_l$ can be formulated by checking collisions between N_P half-spaces and $\mathcal{B}_m$. □

VI. Analytical Evaluation of the Collision Avoidance Formulations

To illustrate the advantages of the proposed collision avoidance formulations in efficiency, a few prominent methods in literature to model collision constraints (2f) are used as a benchmark to compare. We highlight the advantages of saving computational resources in aspects of problem size and setting initial guesses.

A. Problem Size

When considering the differentiable formulations of separating constraints (2f), i.e., $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$, between two convex sets, one prominent way is reformulating the distance between them using dual techniques, such as the proposed collision formulations by Zhang et al. [3], where strong duality theory of convex optimization is utilized to achieve exact collision avoidance. Additionally, Helling and Meurer [7] leverage Farkas' Lemma to arrive at conditions ensuring collision avoidance between two polytopes. However, these methods require introducing many dual variables to be optimized in the NLP formulation. The resulting growth in problem size can be computationally expensive. On the contrary, our formulations using the hyperplane separation theorem need fewer variables, admitting an improved computational performance compared to existing approaches.

TABLE I
The Number of Additional Variables Based on Different Formulations

Methods	Hyperplane separation theorem	Dual formulations	Farkas' Lemma
Two polytopes	$N+1$	$N_m + N_n$ [1]	$N_m + N_n$
Two ellipsoids	$N+1$	$2(N+1)$	- [2]
Polytope, ellipsoid	$N+1$	$N_m + N$	-

[1] The least number of faces to construct a bounded polytope in N dimension is $N+1$, i.e., $N_m \ge N+1, N_n \ge N+1$, indicating the number using Dual formulations is always larger than that using Hyperplane separation theorem in each row.
[2] - means Farkas' Lemma cannot achieve it.

Table I summarizes the needed number of auxiliary variables to formulate $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ when they are polyhedral or ellipsoidal sets. It can be seen that generally the proposed formulations need $N+1$ variables, e.g., $2+1$ variables in two dimensions and $3+1$ variables in three dimensions, to construct the separating hyperplane between $\mathcal{B}_m$ and $\mathcal{O}_n$, not depending on the complexity of set $\mathcal{B}_m$ or $\mathcal{O}_n$. As a contrast, the existing formulations in [3] and [7] identify additional variables proportional to the complexity of $\mathcal{B}_m$ and $\mathcal{O}_n$ being represented, always needing more variables than the proposed formulations. It also states that in the presence of polytopes when the number of edges increases, more variables are needed to formulate using formulations in [3] and [7], while the number of needed variables remains unchanged using the proposed formulations. Typically, obstacles are various nonconvex polytopes composed of many convex polytopes, so this advantage of the proposed formulations is highly beneficial for efficient collision avoidance in real-world scenarios.

B. Geometry-Based Initial Guess

Since the planning OCP (2) is a non-convex optimization problem, good initialization in state variables, control variables, and auxiliary variables used to formulate collision avoidance constraints can improve the computational performance and the resulting solution quality. Much existing work has designed efficient initial guesses in state variables and control variables, e.g., Hybrid A* algorithm [1], [26]. Some researchers choose to initialize auxiliary variables by solving a convex optimization problem [26], [36]. However, it is noted that the number of auxiliary variables is very large, especially needed using the aforementioned dual formulations and Farkas' Lemma, so this numerical optimization way is not efficient. Other researchers [1], [39] prefer to initialize auxiliary variables with constants, e.g., zero or a small value to avoid a singularity, but this way may affect solution quality on account of not being problem-dependent.

Owing to the proposed formulations using the hyperplane separation theorem, we propose a geometry-based method to initialize auxiliary variables. According to Theorem 1, when convex sets $\mathcal{B}_m$ and $\mathcal{O}_n$ are disjoint, i.e., $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$, there

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

exists a separating hyperplane, which is described by auxiliary variables, e.g., $\boldsymbol{\lambda}$, μ in formulation (13). Rather than initializing these auxiliary variables, as an alternative, we initialize the separating hyperplane from a view of geometry. It is done by finding a middle hyperplane between $\mathcal{B}_m$ and $\mathcal{O}_n$, i.e., let auxiliary variables be initialized with a coefficient vector describing the separating hyperplane which is orthogonal to the line crossing the sample on the initial path and the centroid[4] of an obstacle, and crosses certain point[5] between the sample and the centroid. An example of this initial separating hyperplane is illustrated in Fig 2. Because the initial guesses in auxiliary variables using the proposed geometry-based method can be obtained by calculating analytical solutions directly, it is simple and computationally efficient.

In the previous paragraph, we introduce a general method of how to initialize the separating hyperplanes, i.e., using an orthogonal hyperplane, but the geometry-based method is not limited to this. Because the initialization of the separating hyperplanes is highly problem-dependent, the initialization method should be carefully selected to optimize computational performance in various planning cases.

VII. Applications

In this section, we show two applications of the proposed unified planning framework involving containing constraints and separating constraints. One case is autonomous parking in tractor-trailer vehicles in a limited moveable environment. The other case is car overtaking within narrow curved lanes. Additionally, in each case, comparisons with the state of the art on the same premise are done to illustrate the advantages of the proposed formulations.

A. Autonomous Parking for Tractor-Trailer Vehicles

1) Problem Description: Tractor-trailer vehicles are longer than standard cars, resulting in larger turning curvatures and a wider swept area, which makes their control difficult. The difficulty of parking is also significantly influenced by the parking geometry, whether it be parallel, angular, or perpendicular. Besides, in a limited nonconvex parking region, maintaining an acceptable safety distance from obstacles to prevent collisions poses a demanding task even for skilled drivers [8], [56], so it is an intractable issue for tractor-trailer vehicles parking within confined surroundings. Since the execution of reverse parking presents greater complexity in typical scenarios, autonomous reverse parking in tractor-trailer vehicles is studied by exploiting separating constraints (12) and containing constraints (16).

2) Case Description: In the example of Fig. 3(a) and Fig. 3(b), the driving environment $\mathcal{W}$ is modeled by a convex polygon, depicted in green, denoted as $\mathcal{W} := \{\boldsymbol{s} \in \mathbb{R}^2 \mid P_1\boldsymbol{s} \leq \boldsymbol{p_1}\}$, where $P_1 = [0, 1, 0, -1; 1, 0, -1, 0]^\top$, $\boldsymbol{p_1} = [10, 10, 10, 6]^\top$. The obstacle $\mathcal{O}$ is modeled by a convex polygon, depicted in red solid. Let $V_O \in \mathbb{R}^{2\times 4}$ represent a matrix of vertices in $\mathcal{O}$, denoted as $V_O = [7, 7, -6, -6; -3, -10, -10, -3]$. The tractor-trailer vehicle $\mathcal{B}$ is modeled using a union of two convex polygons $\mathcal{B}_1$ and $\mathcal{B}_2$, depicted in magenta and blue. Polygons $\mathcal{B}_1$ and $\mathcal{B}_2$ are connected at the common articulated joint point. Because the position and orientation of the vehicle change, its set $\mathcal{B}_i(\boldsymbol{\xi})$ should be written to depend on the vehicle state $\boldsymbol{\xi}$, $i = 1, 2$. The matrix of vertices in $\mathcal{B}_i(\boldsymbol{\xi})$ is represented by $V_i(\boldsymbol{\xi}) \in \mathbb{R}^{2\times 4}$. Matrix $V_i(\boldsymbol{\xi})$ can be obtained by the rotation and translation of the initial matrix as

$$V_i(\boldsymbol{\xi}) = R_i(\boldsymbol{\xi})V_i(0) + \boldsymbol{T_i}(\boldsymbol{\xi})\mathbf{1}^\top, \tag{21}$$

where $V_i(0)$ is denoted as the initial matrix, denoted as $V_1(0) = [1.5, 1.5, -0.5, -0.5; 1, -1, -1, 1]$, $V_2(0) = [0.5, 0.5, -5, -5; 1, -1, -1, 1]$. The rotation matrix is represented

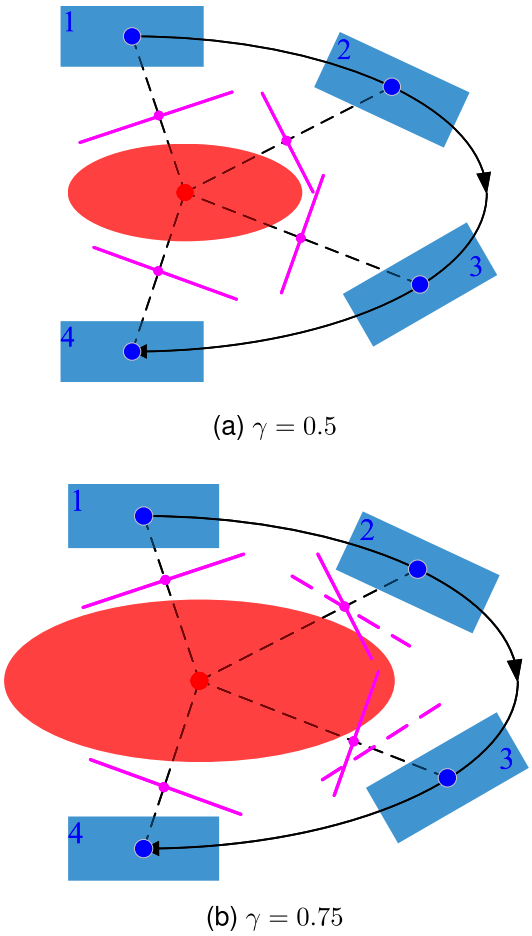


Fig. 2. Positions of the proposed initial separating hyperplanes when weight factor γ is set as 0.5 and 0.75, depending on the size of the obstacle. The controlled object $\mathcal{B}$ is denoted as a polygon, depicted in blue. The obstacle $\mathcal{O}$ is denoted as an ellipse, depicted in red. Centers of $\mathcal{B}$ and $\mathcal{O}$ are depicted in blue point and red point, respectively. The initial path is depicted in black solid line. The magenta lines in samples 1–4 are the initial separating hyperplanes. In (a), the size of $\mathcal{O}$ is smaller, we choose $\gamma = 0.5$ to keep the hyperplane separating $\mathcal{B}$ from $\mathcal{O}$ in each sample, while in (b), $\gamma = 0.75$ is set to keep the separating hyperplanes as far away from $\mathcal{O}$ as possible. Although the hyperplanes in samples 2 and 4 are not fully separating $\mathcal{B}$ from $\mathcal{O}$, such initialization may still help the solver quickly find the resulting separating hyperplanes in the neighborhood, i.e., starting from the solid lines, the solver may more easily find the magenta dotted lines.

[4]The centroid of a polytope is the arithmetic average of the coordinates of its vertices and the centroid of an ellipsoid is its center.

[5]Let (x_1, y_1) and (x_2, y_2) represent the coordinates of the sample and the centroid. The position of a certain point is denoted as $\gamma(x_1, y_1) + (1-\gamma)(x_2, y_2)$, where γ is a problem-dependent weight factor.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

as $R_i(\boldsymbol{\xi})$. The translation vector is $\boldsymbol{T_i}(\boldsymbol{\xi})$. The collision avoidance requirements, i.e., $\mathcal{B} \cap \mathcal{O} = \emptyset$ and $\mathcal{B} \subset \mathcal{W}$ are approached by using the proposed (12) and (16), respectively.

3) Vehicle Model and Objective Function: The vehicle motion is described using a kinematic model, extended with a trailer [57], [58] as

$$\begin{aligned} \dot{x} &= v\cos(\theta_1), \\ \dot{y} &= v\sin(\theta_1), \\ \dot{\theta}_1 &= v\tan(\delta)/L_1, \\ \dot{\theta}_2 &= v\sin(\theta_1-\theta_2)/L_2, \\ \dot{v} &= a, \\ \dot{\delta} &= \omega, \end{aligned} \tag{22}$$

where $L_1 = 1\,\mathrm{m}$ and $L_2 = 4.5\,\mathrm{m}$ are wheelbases of the tractor and the trailer, respectively. The position of the tractor-trailer joint is identical to the vector (x, y), and θ_1, θ_2 represent yaw angles of the tractor and the trailer with respect to the horizontal axle. The velocity and acceleration of the tractor-trailer joint are v and a. The steering angle and the gradient of the steering angle of the tractor are denoted by δ and ω, respectively. The state and control vectors can be summarized as

$$\boldsymbol{\xi} = \begin{bmatrix} x & y & \theta_1 & \theta_2 & v & \delta \end{bmatrix}^\top, \boldsymbol{u} = \begin{bmatrix} a & \omega \end{bmatrix}^\top, \tag{23}$$

so the vehicle dynamics can be denoted as $\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}, \boldsymbol{u})$. The feasible values of $\theta_1, \theta_2, v, \delta$ are limited to $|\theta_1| \leq 180°$, $|\theta_2| \leq 180°$, $|v| \leq \frac{5}{3.6}\mathrm{m/s^2}$, $|\delta| \leq 40°$. The feasible inputs are given by $|a| \leq 1\,\mathrm{m/s^2}$ and $|\omega| \leq 5\,°/\mathrm{s^2}$. Specifically, the articulated joint angle at the tractor-trailer joint is limited using the constraint

$$|\theta_1 - \theta_2| \leq 60°. \tag{24}$$

In parking cases, we expect the autonomous vehicle to complete the parking maneuvers subjected to minimum traveling time. However, the parking duration, denoted as t_f, is not known in advance. To address this problem, We introduce a new independent variable $\tau \in [0, 1]$, and express the time as $t = t_f\tau$, where $t_f \geq 0$ is a scalar optimization variable that needs to be added to the problem. Thus, the vehicle dynamics in (22) become a function of τ as

$$\frac{\mathrm{d}\boldsymbol{\xi}(\tau)}{\mathrm{d}\tau} = t_f f(\boldsymbol{\xi}(\tau), \boldsymbol{u}(\tau)). \tag{25}$$

Apart from the penalty on traveling time, the control input a and ω should be small to yield smooth trajectories. Thus, a typical time-energy cost function in OCP (2) is formulated as

$$\min_{\boldsymbol{\xi}, \boldsymbol{u}, t_f, (\cdot)} t_f \left(r + \int_0^1 \|\boldsymbol{u}(\tau)\|_Q^2 \,\mathrm{d}\tau \right), \tag{26}$$

where r is a penalty factor that trades the parking time with the rest of the cost. The notation $\|\cdot\|_Q^2$ represents squared Euclidean norm weighted by matrix Q. For the studied cases we choose $r = 1$, $Q = \mathrm{diag}(100, 200)$.

The OCP (2) is generally reformulated as a discrete NLP problem that can be solved using off-the-shelf solvers. The whole region $[0, 1]$ is discretized into $k_f + 1$ parts, i.e., $1 = k_f\Delta\tau$, where $\Delta\tau$ is the sampling interval. Variable τ is replaced with $\tau = k\Delta\tau$, where $k = 0, \ldots, k_f$. Thus, the OCP (2) involving collision avoidance can be formulated as a discrete NLP problem

$$\min_{\boldsymbol{\xi}, \boldsymbol{u}, t_f, (\cdot)} rt_f + \sum_{k=0}^{k_f-1} \tilde{\ell}(\boldsymbol{u}(k), t_f) \tag{27a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}(0) = \boldsymbol{\xi}_{\mathbf{init}}, \boldsymbol{\xi}(k_\mathrm{f}) = \boldsymbol{\xi}_{\mathbf{final}}, \tag{27b}$$

$$\boldsymbol{\xi}(k+1) = \tilde{f}\left(\boldsymbol{\xi}(k), \boldsymbol{u}(k), t_f\right), \tag{27c}$$

$$\boldsymbol{\xi}(k) \in \mathcal{X}, \ \boldsymbol{u}(k) \in \mathcal{U}, \tag{27d}$$

$$(12), (16), \tag{27e}$$

where symbol $(\cdot)$ is a shorthand notation for decision variables that need to be added to the vector of existing optimization variables, e.g., μ in formulation (12). Symbol $\tilde{f}$ denotes the discretized model dynamics of (25) and symbol $\tilde{\ell}$ is the discretized form of the state cost $t_f \|\boldsymbol{u}(\tau)\|_Q^2$ in function (26). In the simulation, the 4-th order Runge-Kutta method [39] is chosen to express the discretized $\tilde{\ell}$ and $\tilde{f}$.

Since the resulting NLP (27) is nonconvex, multiple local solutions exist depending on different initial guesses. An initial guess in the neighborhood of a local solution is more likely to cause the NLP solver to return that local solution. In a reverse parking scenario, we use piecewise polynomials to initialize states x, y. In the example of Fig 3(a) and Fig 3(b), we expect that the vehicle parks into the garage by reversing only once, so the initial guess in positions is defined by using a second-order polynomial from the starting position to an interim position and a linear function from the interim position and ending position (see Fig. 3(a)). The yaw angle is initialized as the angle between a line tangent to the polynomial trajectory at each sample and the horizontal axle. Other states and inputs are initialized as zero. As mentioned in Section VI-B, let the initial separating hyperplane, which is used to initialize auxiliary variables $\boldsymbol{\lambda}$, μ in formulation (12), be initialized with the hyperplane orthogonal to the line crossing the sample on the polynomial trajectory and the centroid of an obstacle. In the example, we set $\gamma = 0.75$ to position the initial hyperplane near the vehicle. Finding an initial guess of parking time t_f is challenging and highly problem-dependent, so for this reverse parking case, we use an estimated value of 50 s to warmly start t_f [3], [26].

Simulations are conducted in MATLAB R2020a and executed on a laptop with AMD R7-5800H CPU at 3.20 GHz and 16GB RAM. Since we use a multiple-shooting approach, the problem is formulated as a collection of k_f phases. In this case, $k_\mathrm{f} = 30$ phases are chosen. The problems are then implemented using CasADi [54]. The resulting discrete NLP (27) is solved using solver IPOPT [55].

4) Solution: Simulation results are visually interpreted in Fig. 3(b). It can be seen that the planned optimal path is smooth and collision-free. Vertices of polygons $\mathcal{B}_1$ and $\mathcal{B}_2$ do not enter into obstacle $\mathcal{O}$, and vertices of $\mathcal{O}$ are prevented from passing the edges of $\mathcal{B}_1$ and $\mathcal{B}_2$. It can be noticed that since the parking environment $\mathcal{W}$ is limited, the vehicle keeps close to the environment boundaries to allow larger turning curvatures,

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

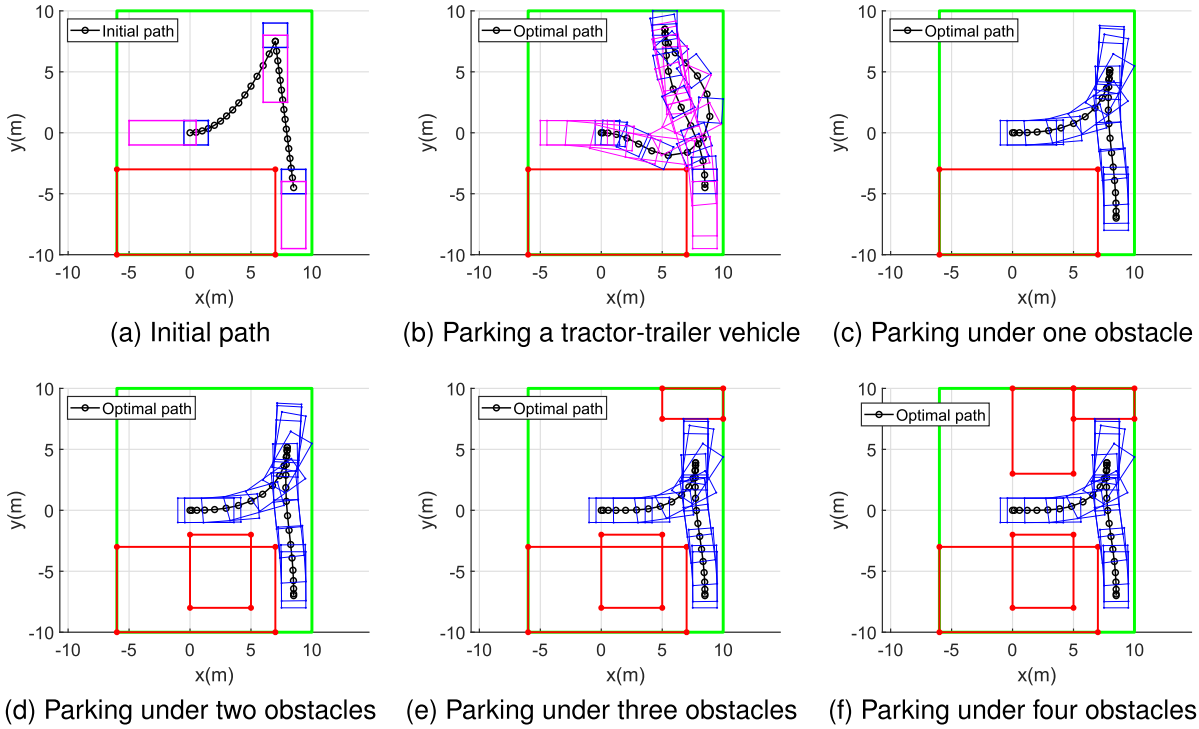


Fig. 3. Subplot (a) illustrates the initial path using piecewise polynomials. The interim position used to obtain an initial path is denoted as $[7\,\mathrm{m}, 7.5\,\mathrm{m}]$. Subplot (b) shows the planned optimal path of a tractor-trailer vehicle. In (b), the starting state $\boldsymbol{\xi}_{\mathbf{init}}$ is $[0\,\mathrm{m}, 0\,\mathrm{m}, 0°, 0°, 0\,\mathrm{m/s}, 0°]$. The ending state $\boldsymbol{\xi}_{\mathbf{final}}$ is $[8.5\,\mathrm{m}, -4.5\,\mathrm{m}, 90°, 90°, 0\,\mathrm{m/s}, 0°]$. Subplots (c)–(f) are the optimal paths of a single car under different obstacles. In (c)–(f), the starting state $\boldsymbol{\xi}_{\mathbf{init}}$ is $[0\,\mathrm{m}, 0\,\mathrm{m}, 0°, 0\,\mathrm{m/s}, 0°]$. The ending state $\boldsymbol{\xi}_{\mathbf{final}}$ is $[8.5\,\mathrm{m}, -7\,\mathrm{m}, 90°, 0\,\mathrm{m/s}, 0°]$.

but it is contained within $\mathcal{W}$ all the time. In the resulting solution, the vehicle conducts a reverse behavior to adjust its pose to enter into the confined parking place. Additionally, the implementing time to solve problem (27) using IPOPT is 2.264 s and the resulting parking duration is 69.89 s.

5) Comparisons in Problem Size: This part is used to support the advantage of improving computational performance on account of admitting a reduced problem size compared to existing approaches. It is noted that the nonconvex parking problem (27) may converge to multiple local solutions even if based on the same initial guess. Indeed, a vehicle may take infinitely many paths to reach the goal, especially for a tractor-trailer vehicle with more states. However, we expect to compare the computational performance of the proposed algorithms with the state of the art under the condition of converging to the same local solution, so we design a new case of a single-car parking problem. Here, we use a simplified vehicle model of (22), with $L_1 = 2.6$. The vehicle dynamics is modelled with the reduced state vector $\boldsymbol{\xi} = \left[x, y, \theta_1, v, \delta\right]^\top$ and control vector $\boldsymbol{u}\,(s) = [a, \omega]^\top$. Let the initial matrix $V_1(0)$ be denoted as $V_1(0) = \left[3.6, 3.6, -1, -1; 1, -1, -1, 1\right]$.

To keep the same premise of the NLP configuration, the initial guess in states x, y, θ_1 is a linear function (starting from the beginning and ending at the final values). The rest of the states and inputs are set as zero. Let $\boldsymbol{\lambda}$ in formulation (12) be initialized with a positive value vector to prevent a singularity and μ be set to zero. We conduct numerical experiments by comparing formulation (12) with the formulations in Proposition 3 of [3] and in Section III-A-2 of [7]. Others remain unchanged in NLP (27). We design four scenarios to test collision avoidance with different placement of obstacles. The resulting optimal trajectories are illustrated in Fig. 3(c)–(f). The computational results are shown in Table II. It can be seen that these planned trajectories are collision-free and smooth. Apart from the scenario of three obstacles, the solver converges to the same local solution using different methods. Alternatively, we can verify the comparison results in the other three scenarios. The primal observation in Table II is the least number of auxiliary variables by exploiting the proposed formulation (12) in each scenario, which illustrates that using

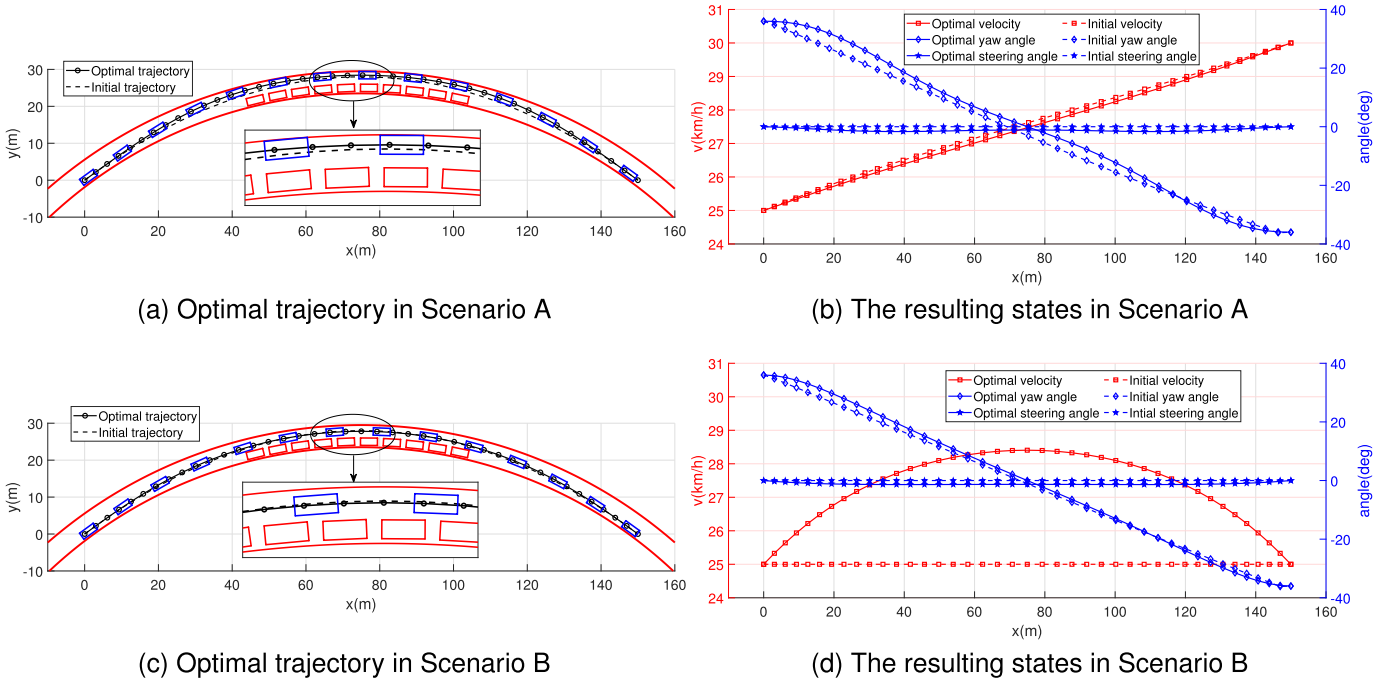
TABLE II
THE RESULT OF COMPUTATIONAL COST BY USING DIFFERENT METHODS

Performance	Algorithms	The number of placed obstacles			
		One	Two	Three	Four
Number of auxiliary variables to be optimized	proposed	**306**	**396**	**486**	**576**
	[3]	456	696	936	1176
	[7]	456	696	936	1176
Solving time (s)	proposed	**1.92**	**1.336**	**2.239**	**3.048**
	[3]	2.03	3.008	20.086	19.977
	[7]	4.098	4.66	18.449	18.701
Objective value	proposed	73.45	73.45	75.13	75.13
	[3]	73.45	73.45	77.702[1]	75.13
	[7]	73.45	73.45	75.13	75.13
t_f (s)	proposed	52.56	52.56	51.55	51.55
	[3]	52.56	52.56	54.17	51.55
	[7]	52.56	52.56	51.55	51.55

[1] The red values represent the obtained other local solutions based on the same initial guess.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Optimal trajectory in Scenario A

(b) The resulting states in Scenario A

(c) Optimal trajectory in Scenario B

(d) The resulting states in Scenario B

Fig. 4. The resulting trajectory and states in different scenarios. In subplot (a) and (c), the curved lane is surrounded by two curves depicted in red solid. The boundary of obstacle $\mathcal{O}_1$ is denoted as the inner red curve. Positions of moving obstacle $\mathcal{O}_2$ are illustrated using red polygons. Positions of the overtaking process of the ego vehicle $\mathcal{B}$ are illustrated using blue polygons. In Scenario A and B, the ego vehicle $\mathcal{B}$ plans to overtake the leading obstacle $\mathcal{O}_2$ in the constrained curved lane. In Scenario A, the starting state $\boldsymbol{\xi}_{\mathbf{init}}$ is $[0\,\mathrm{m}, 0\,\mathrm{m}, 36°, 25\,\mathrm{km/h}, 0°]$ and the ending state $\boldsymbol{\xi}_{\mathbf{final}}$ is $[150\,\mathrm{m}, 0\,\mathrm{m}, -36°, 30\,\mathrm{km/h}, 0°]$. In Scenario B, the starting state $\boldsymbol{\xi}_{\mathbf{init}}$ is $[0\,\mathrm{m}, 0\,\mathrm{m}, 36°, 25\,\mathrm{km/h}, 0°]$ and the ending state $\boldsymbol{\xi}_{\mathbf{final}}$ is $[150\,\mathrm{m}, 0\,\mathrm{m}, -36°, 25\,\mathrm{km/h}, 0°]$. The initial paths and the planned optimal paths are depicted in subplot (a) and (c). The designed initial states and the resulting states are depicted in subplot (b) and (d).

the proposed formulation (12) shows a considerable reduction in the problem size of NLP formulations compared to the other two existing formulations. From Fig. 3(c) to Fig. 3(f), we increase another three polygonal obstacles orderly, and it can be seen that the number of variables experiences a relatively low increase using the proposed formulation (12) but a sharp increase using formulations in [3] and [7]. In Fig. 3(c)–(f), our method needs $3 \times k_f = 90$ auxiliary variables ($\boldsymbol{\lambda} \in \mathbb{R}^2, \mu \in \mathbb{R}$ in formulation (12)) per obstacle. On the contrary, existing formulations in [3] and [7] identify variables associated with the edges of a vehicle and an obstacle, depending on the number of the total edges of the vehicle and all obstacles, i.e., adding $(4+4) \times k_f = 240$ auxiliary variables per obstacle. The comparison of needed additional variables states that when the number of edges of each obstacle increases, more variables are needed to formulate constraints of [3] and [7], while the number of variables by the proposed formulation (12) remains unchanged. Besides, it can be seen that as more obstacles are considered, it takes more time to solve an NLP. In each scenario, since the problem size using the proposed algorithm is smaller, solvers can converge to the same local optimums faster, which shows that the proposed method is more efficient compared with other algorithms.

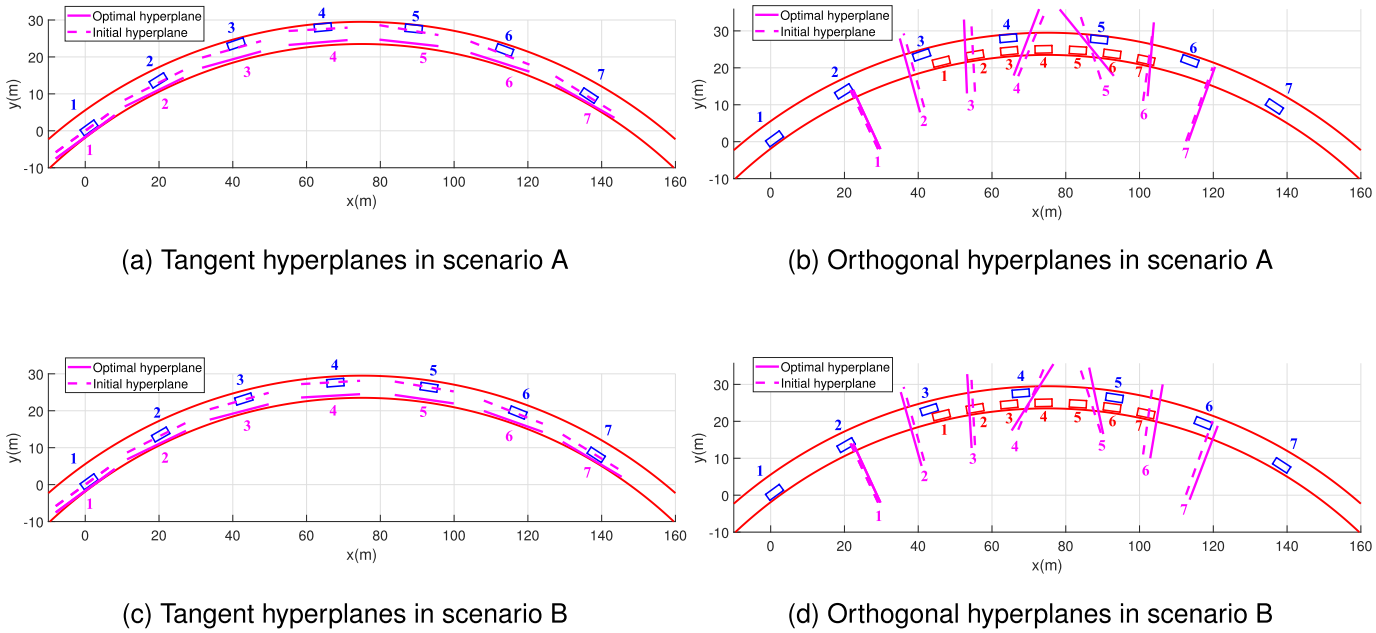
B. Overtaking in Curved Lanes

1) Problem Description: It is common for a vehicle to drive in curved lanes, e.g., trucks safely traveling on mountain roads and cars racing in closed-loop tracks [35], [52], [53]. These vehicles are expected to reside within a narrow curved lane, not colliding with moving obstacles. Typically, it is difficult to overtake a moving obstacle for intelligent vehicles in a narrow curved environment, especially for a long-size vehicle. To approach this issue, we use separating constraints (12), (14) and containing constraints (16) in the proposed planning framework.

2) Case Description: In the examples of Fig. 4(a) and Fig. 4(c), the driving environment $\mathcal{W}$ is modeled by an ellipse, depicted in outer red solid line, denoted as $\mathcal{W} := \left\{\boldsymbol{s} \in \mathbb{R}^2 \mid (\boldsymbol{s}-\boldsymbol{e_1})^\top E_1 (\boldsymbol{s}-\boldsymbol{e_1}) \leq 1\right\}$, where $E_1 = \mathrm{diag}\,(1/129.5^2, 1/129.5^2)$, $\boldsymbol{e_1} = [75, -100]^\top$. The obstacle $\mathcal{O}_1$ is modeled by an ellipse denoted as $\mathcal{O}_1 := \left\{\boldsymbol{s} \in \mathbb{R}^2 \mid (\boldsymbol{s}-\boldsymbol{e_2})^\top E_2 (\boldsymbol{s}-\boldsymbol{e_2}) \leq 1\right\}$, where $E_2 = \mathrm{diag}(1/123.5^2, 1/123.5^2)$, $\boldsymbol{e_2} = [75, -100]^\top$. It is illustrated by the inner red solid line in the figure. It can be noticed that the curved lane is constructed with an intersection of ellipse $\mathcal{W}$ and ellipse $\mathcal{O}_1$, i.e., $\mathcal{W} \cap \mathcal{O}_1$. Another moving obstacle is denoted as a polygon $\mathcal{O}_2$ of size $4.6\,\mathrm{m} \times 2\,\mathrm{m}$. We assume obstacle $\mathcal{O}_2$ moves slowly in a predefined path, which is defined as a uniform circular motion with $1.32\,°/\mathrm{s}$. It is illustrated by a red solid polygon in Fig. 4(a) and Fig. 4(c). The ego vehicle $\mathcal{B}$ is modeled as a polygon of the same size as $\mathcal{O}_2$, depicted in blue. The matrix of vertices in $\mathcal{B}(\boldsymbol{\xi})$ is represented by $V(\boldsymbol{\xi}) \in \mathbb{R}^{2\times 4}$. In this overtaking planning problem, the ego vehicle should reside in the curved lane and not collide with obstacles, represented as $\mathcal{B} \cap \mathcal{O}_1 = \emptyset$, $\mathcal{B} \cap \mathcal{O}_2 = \emptyset$, and $\mathcal{B} \subset \mathcal{W}$, which are approached by using the proposed formulations (14), (12), and (16). To support the efficacy of the proposed methods, we design two different overtaking scenarios, i.e., the vehicle

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Tangent hyperplanes in scenario A

(b) Orthogonal hyperplanes in scenario A

(c) Tangent hyperplanes in scenario B

(d) Orthogonal hyperplanes in scenario B

Fig. 5. The hyperplanes to initialize auxiliary variables in constraints (14) and (12) are depicted in magenta dotted lines. The resulting hyperplanes are depicted in magenta solid lines. In (b) and (d) positions of moving obstacle $\mathcal{O}_2$ are depicted in red polygons. Subplots (a)-(b) and subplots (c)-(d) illustrate positions of the initial hyperplanes and resulting hyperplanes in seven samples in Scenario A and Scenario B, respectively. It can be seen that in each sample the initial hyperplane is close to the resulting hyperplane and the resulting hyperplane can separate the vehicle from obstacle $\mathcal{O}_1$ in (a) and (c) and separate the vehicle from moving obstacle $\mathcal{O}_2$ in (b) and (d).

keeps accelerating in Scenario A while in Scenario B the vehicle first speeds up and then slows down to its original speed.

3) Vehicle Model and Cost Function: As Section VII-A.5 does, we use a simplified vehicle model of (22), i.e., with the reduced state vector $\boldsymbol{\xi} = \left[x, y, \theta, v, \delta\right]^\top$ and control vector $\boldsymbol{u}(s) = [a, \omega]^\top$, where θ represents the yaw angle with respect to the horizontal axle. The feasible values of θ, v, δ are limited to $|\theta| \leq 85°$, $\frac{5}{3.6}\mathrm{m/s^2} \leq v \leq \frac{60}{3.6}\mathrm{m/s^2}$, $|\delta| \leq 45°$. Additionally, the feasible inputs are given by $-1\,\mathrm{m/s^2} \leq a \leq 2\,\mathrm{m/s^2}$ and $|\omega| \leq 5\,°/\mathrm{s^2}$.

In this overtaking case, we expect the vehicle to drive smoothly to a target position subject to a given time t_f. A discretization procedure can then similarly be applied as proposed in Section VII-A.3, i.e., the given time t_f is divided into k_f+1 parts. In the objective function, the control inputs a and ω are penalized as $\sum_{k=0}^{k_f-1} \|\boldsymbol{u}(k)\|_Q^2$, with $Q = \mathrm{diag}(10, 20)$. Thus, the OCP (2) is formulated as the NLP

$$\min_{\boldsymbol{\xi},\boldsymbol{u},(\cdot)} \quad \sum_{k=0}^{k_f-1} \|\boldsymbol{u}(k)\|_Q^2 \tag{28a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}(0) = \boldsymbol{\xi}_{\mathbf{init}},\ \boldsymbol{\xi}(k_\mathrm{f}) = \boldsymbol{\xi}_{\mathbf{final}}, \tag{28b}$$

$$\boldsymbol{\xi}(k+1) = \tilde{f}\left(\boldsymbol{\xi}(k), \boldsymbol{u}(k)\right), \tag{28c}$$

$$\boldsymbol{\xi}(k) \in \mathcal{X},\ \boldsymbol{u}(k) \in \mathcal{U}, \tag{28d}$$

$$(12), (14), (16), \tag{28e}$$

where symbol $\tilde{f}$ denotes the discretized symbol of vehicle model (22) with removing state θ_2 related to the trailer. The 4-th order Runge-Kutta method is used to express the state evolution of (28c).

The 4-th order polynomial function is applied to generate a trajectory as the initial guesses [59], [60] in states (x, y, θ). The initial guess for v is a linear function from the initial to the final value. The rest of the states and inputs are set as zero. Let the initial separating hyperplane, which is used to initialize auxiliary variables $\boldsymbol{\lambda}$, μ in formulation (14), be initialized with the hyperplane tangent to the polynomial trajectory at each sample. Likewise in the parking case, let the hyperplane, which is to initialize $\boldsymbol{\lambda}$, μ in formulation (12), be orthogonal to the line crossing the sample on the polynomial trajectory and the centroid of an obstacle, and $\gamma = 0.5$ is set. In this overtaking case, the horizon time is set to 21 s and $k_\mathrm{f} = 40$. Other NLP configurations are the same as that of the parking case.

4) Solution: The resulting optimal trajectory and states are visually interpreted in Fig. 4. The primary observation in Fig. 4(a) and Fig. 4(c) is that the resulting optimal paths are smooth and collision-free. In Fig. 4(a) derived from Scenario A, since a higher speed is acquired, a larger turning radius of the car is needed. Because this curved lane is narrow, it can be seen that the vehicle keeps on close to the environment boundaries when driving during [40 m,100 m], while each vertex of the vehicle does not cross the boundaries. The proposed containing constraint (16) contributes to this. On the contrary, in Fig. 4(c) from Scenario B, the vehicle mostly drives along the centerline, not keeping close to the environment boundaries. Additionally, in these two scenarios, the vehicle completes the overtaking task subject to a given time, not colliding with the moving obstacle $\mathcal{O}_2$, and also during this whole process the vehicle does not enter into obstacle $\mathcal{O}_2$. The proposed separating constraint (12) contributes to this.

Another observation in Fig. 4(a)–Fig. 4(d) is that the resulting optimal paths and states are close to the designed initial guesses, indicating that the given initial guesses can improve the efficiency and solution quality in solving the nonconvex NLP (28). Additionally, from the state profiles of

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
RESULTS IN COMPUTATIONAL EFFORT USING DIFFERENT INITIAL GUESSES

Scenario	Ways to initialize auxiliary variables in (12) and (21)	Objective value	TS[1]	Solving time(s)
Scenario A	Proposed to (12) & Proposed to (21)	0.907	0.168	**1.362**[2]
	Existing to (12) & Proposed to (21)	0.907	0.168	8.314
	Proposed to (12) & Existing to (21)	0.907	0.168	2.132
Scenario B	Proposed to (12) & Proposed to (21)	0.169	0.410	**0.962**
	Existing to (12) & Proposed to (21)	0.169	0.410	4.717
	Proposed to (12) & Existing to (21)	0.169	0.410	1.763

[1] The value TS $= \sum_{k=0}^{k_f-1} \left(a(k)^2 + \omega(k)^2\right)$ is calculated to double check if the NLP (28) converges to the same local solution under a same objective value.
[2] The bold value denotes the least solving time.

Fig. 4(b) and Fig. 4(d), the steering angle, the velocity, and the yaw angle of the vehicle vary steadily during the traveling process and achieve the expected target states. Specifically, in Scenario A the speed increases steadily from the beginning 25 km/h to the expected 30 km/h while in Scenario B the speed first increases from 25 km/h to 28.4 km/h and returns to the expected 25 km/h in the end.

5) Comparisons in Initializing Auxiliary Variables: To investigate the computational cost of the proposed formulations (12) and (14) when initialized differently, comparative experiments are conducted under existing initialization methods in [1] and the proposed geometry-based methods, respectively. The existing method initializes $\boldsymbol{\lambda}$, in formulations (12) and (14) of NLP (28), with a positive value to prevent a singularity and μ is set to zero. The proposed method uses an orthogonal hyperplane to initialize the auxiliary variables in formulation (12) and a tangent hyperplane to initialize the auxiliary variables in formulation (14). Other configurations of NLP (28) remain unchanged.

Fig. 5 illustrates the proposed geometry-based initial guesses in Scenario A and Scenario B. It can be seen that the designed tangent or orthogonal hyperplane in each sample is in the neighborhood of the resulting separating hyperplane, showing that the auxiliary variables related to constraints (12) and (14) have been well initialized. This is beneficial to solve the nonconvex NLP (28), which is supported in Table III showing the computational results using different initialization methods. From Table III, though using different initial guesses the resulting optimal solutions are exactly the same in each scenario, i.e., the objective value is 0.907 and 0.169, respectively. However, the result of solving time shows that the proposed geometry-based guesses lead to the less computation time than the existing guesses, especially when initializing the variables in constraints (12). The reason behind this is that the proposed geometry-based method can well initialize variables to be optimized in NLP (28). The results in Table III illustrates the computational advantages of our methods, which are highly beneficial for efficient collision avoidance.

VIII. CONCLUSION AND FUTURE WORK

This article investigates optimization-based methods aiming to achieve exact collision avoidance of controlled objects. In the proposed unified planning framework, we formulate smooth collision avoidance constraints by utilizing the hyperplane separation theorem to separate a pair of convex sets, and S-procedure and geometrical methods to contain a convex set inside another convex set. The proposed collision avoidance formulations are highly beneficial for reducing an NLP problem size and efficient initialization of auxiliary variables, which admits an improved computational performance compared to the state of the art.

Based on the proposed planning framework, the next work will first focus on the research of the uncertainty from localization errors, system delay, etc. Additionally, since collision formulations using dual methods in [3] (with higher complexity than the proposed collision formulations using the hyperplane separation theorem) have successfully been implemented in real-time [36] and even in real-world scenarios [61], [62], it is possible to execute the proposed methods in real-time by sequential quadratic programming implementations [63], [64]. Real-time control implementation of the proposed method will also be considered in future studies.

APPENDIX A
EXTENSIONS OF SEPARATING CONSTRAINTS

A. Hyperplane With Bounded Slope or Displacement

In Proposition 3, Proposition 4, and Proposition 5, we propose using $N+1$ auxiliary variables to describe a separating hyperplane. The three propositions are sufficient and necessary conditions of $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$. In principle, it is possible to use fewer variables to obtain sufficient conditions of $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$. For instance, consider Proposition 3 under $N = 2$. According to the proposition, 3 auxiliary variables are needed to construct a hyperplane as

$$\begin{aligned} \mathcal{B}_m \cap \mathcal{O}_n = \emptyset \iff & \exists \boldsymbol{\lambda} \in \mathbb{R}^2, \mu \in \mathbb{R}: \\ & \boldsymbol{\lambda}^\top V_{P_1} \geq \mu \mathbf{1}^\top, \boldsymbol{\lambda}^\top V_{P_2} \leq \mu \mathbf{1}^\top, \|\boldsymbol{\lambda}\| > 0. \end{aligned} \tag{29}$$

In certain cases, it is possible to use only 2 auxiliary variables, e.g., if it is known that $\mu \neq 0$, the inequalities can be normalized by μ and a sufficient condition of $\mathcal{B}_m \cap \mathcal{O}_n = \emptyset$ can be formulated as

$$\begin{aligned} \mathcal{B}_m \cap \mathcal{O}_n = \emptyset \impliedby & \exists \boldsymbol{\lambda} \in \mathbb{R}^2: \\ & \boldsymbol{\lambda}^\top V_{P_1} \geq \mathbf{1}^\top, \boldsymbol{\lambda}^\top V_{P_2} \leq \mathbf{1}^\top, \|\boldsymbol{\lambda}\| > 0. \end{aligned} \tag{30}$$

Alternatively, if it is known that both horizontal and vertical hyperplane slopes are not occurring in the same problem, then one of the $\boldsymbol{\lambda}$ is not zero and the inequality can be normalized with that value. For e.g., such a condition can be written as

$$\begin{aligned} \mathcal{B}_m \cap \mathcal{O}_n = \emptyset \impliedby & \exists \lambda \in \mathbb{R}, \mu \in \mathbb{R}: \\ & \begin{bmatrix} \lambda & 1 \end{bmatrix} V_{P_1} \geq \mu \mathbf{1}^\top, \begin{bmatrix} \lambda & 1 \end{bmatrix} V_{P_2} \leq \mu \mathbf{1}^\top, \lambda^2 > 0. \end{aligned} \tag{31}$$

Thus, using fewer variables can simplify the Proposition 3, Proposition 4, and Proposition 5, which can further reduce problem size and computational load.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

B. A Point-Mass Controlled Object

Apart from the investigation of collision avoidance between polytopes and ellipsoids in this article, the hyperplane separation theorem can readily be applied to other compact sets. For instance, sometimes the controlled object $\mathcal{B}$ is regarded as a point-mass system. Let the position of a point p be denoted as $\boldsymbol{p} \in \mathbb{R}^N$. Let set $\mathcal{O}_n$ be denoted using polytope $\mathcal{P}(P, \boldsymbol{p}, \boldsymbol{s})$ as (3). Then,

$$\begin{aligned} p \cap \mathcal{O}_n = \emptyset \iff \exists \boldsymbol{\lambda} \in \mathbb{R}^N, \mu \in \mathbb{R}: \\ \boldsymbol{\lambda}^\top \boldsymbol{p} \geq \mu, \boldsymbol{\lambda}^\top V_P \leq \mu \mathbf{1}^\top, \|\boldsymbol{\lambda}\| > 0. \end{aligned} \tag{32}$$

In formulation (32), the number of additional variables i.e., $\boldsymbol{\lambda}, \mu$, does not depend on the complexity of polytope $\mathcal{O}_n$, so formulation (32) also admits a reduced problem size compared to formulations in Proposition 1 of [3] and Section III-A-1 of [7].

It is possible to even further reduce the number of variables to formulate (32). The hyperplane for separation can be attached to the vehicle point p, i.e., $\boldsymbol{\lambda}^\top \boldsymbol{p} = \mu$, and a sufficient condition of $p \cap \mathcal{O}_n = \emptyset$ can be formulated as

$$\begin{aligned} p \cap \mathcal{O}_n = \emptyset \impliedby \exists \boldsymbol{\lambda} \in \mathbb{R}^N: \\ \boldsymbol{\lambda}^\top V_P \leq \boldsymbol{\lambda}^\top \boldsymbol{p} \mathbf{1}^\top, \|\boldsymbol{\lambda}\| > 0. \end{aligned} \tag{33}$$

ACKNOWLEDGMENT

The authors thank Dr. Jian Zhou from Linköping University for valuable suggestions.

REFERENCES

[1] C. Dietz, S. Albrecht, A. Nurkanovic, and M. Diehl, "Efficient collision modelling for numerical optimal control," in *Proc. Eur. Control Conf. (ECC)*, Bucharest, Romania, Jun. 2023, pp. 1–7.

[2] J. Karlsson, N. Murgovski, and J. Sjöberg, "Computationally efficient autonomous overtaking on highways," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3169–3183, Aug. 2020.

[3] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 972–983, May 2021.

[4] 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, Aug. 2014.

[5] V. Sunkara, A. Chakravarthy, and D. Ghose, "Collision avoidance of arbitrarily shaped deforming objects using collision cones," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2156–2163, Apr. 2019.

[6] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.

[7] S. Helling and T. Meurer, "Dual collision detection in model predictive control including culling techniques," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2449–2464, Nov. 2023, doi: 10.1109/TCST.2023.3259822.

[8] B. Li, T. Acarman, Y. Zhang, L. Zhang, C. Yaman, and Q. Kong, "Tractor-trailer vehicle trajectory planning in narrow environments with a progressively constrained optimal control approach," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 414–425, Sep. 2020.

[9] B. Li et al., "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11970–11981, Aug. 2022.

[10] P. F. Lima, "Optimization-based motion planning and model predictive control for autonomous driving: With experimental evaluation on a heavy-duty construction truck," Ph.D. dissertation, Dept. Autom. Control, KTH Royal Inst. Technol., Stockholm, Sweden, 2018.

[11] J. Zhou, B. Olofsson, and E. Frisk, "Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1305–1319, Jan. 2024, doi: 10.1109/TIV.2023.3314709.

[12] J. Zhou, B. Olofsson, and E. Frisk, "Interaction-aware moving target model predictive control for autonomous vehicles motion planning," in *Proc. Eur. Control Conf. (ECC)*, 2022, pp. 154–161.

[13] J. Hu, H. Zhang, L. Liu, X. Zhu, C. Zhao, and Q. Pan, "Convergent multiagent formation control with collision avoidance," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1805–1818, Dec. 2020.

[14] D. Ioan, I. Prodan, S. Olaru, F. Stoican, and S.-I. Niculescu, "Mixed-integer programming in motion planning," *Annu. Rev. Control*, vol. 51, pp. 65–87, Feb. 2021.

[15] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optim. Eng.*, vol. 3, pp. 227–252, Sep. 2002.

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

[17] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 42–49.

[18] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 477–483.

[19] I. Prodan, F. Stoican, S. Olaru, and S. I. Niculescu, *Mixed-Integer Representations in Control Design Mathematical Foundations and Applications*. Cham, Switzerland: Springer. 2015.

[20] J. P. Vielma, "Mixed integer linear programming formulation techniques," *SIAM Rev.*, vol. 57, no. 1, pp. 3–57, Jan. 2015.

[21] X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, "Collision avoidance of mobile robots by using initial time delays," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 324–329.

[22] I. Haghighi, S. Sadraddini, and C. Belta, "Robotic swarm control from spatio-temporal specifications," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Las Vegas, NV, USA, Dec. 2016, pp. 5708–5713.

[23] J. Karlsson, N. Murgovski, and J. Sjoberg, "Comparison between mixed-integer and second order cone programming for autonomous overtaking," in *Proc. Eur. Control Conf. (ECC)*, Limassol, Cyprus, 2018, pp. 386–391.

[24] F. Molinari, N. N. Anh, and L. Del Re, "Efficient mixed integer programming for autonomous overtaking," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 2303–2308.

[25] H. P. Williams, *Model Building in Mathematical Programming*. Hoboken, NJ, USA: Wiley, 2013.

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

[27] I. Ballesteros-Tolosana, S. Olaru, P. Rodriguez-Ayerbe, G. Pita-Gil, and R. Deborne, "Collision-free trajectory planning for overtaking on highways," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 2551–2556.

[28] R. J. M. Afonso, R. K. H. Galvão, and K. H. Kienitz, "Reduction in the number of binary variables for inter-sample avoidance in trajectory optimizers using mixed-integer linear programming," *Int. J. Robust Nonlinear Control*, vol. 26, no. 16, pp. 3662–3669, Nov. 2016.

[29] F. Stoican, I. Prodan, and S. Olaru, "Hyperplane arrangements in mixed-integer programming techniques. Collision avoidance application with zonotopic sets," in *Proc. Eur. Control Conf. (ECC)*, Zurich, Switzerland, Jul. 2013, pp. 3155–3160.

[30] A. Ganesan, S. Gros, and N. Murgovski, "Numerical strategies for mixed-integer optimization of power-split and gear selection in hybrid electric vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3194–3210, Mar. 2023.

[31] C. Kirches, *Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control*. Cham, Switzerland: Springer, 2010.

[32] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 5568–5573.

[33] 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.

[34] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 2958–2964.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[35] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. IEEE Amer. Control Conf. (ACC)*, 2021, pp. 3882–3889.

[36] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, May 2022, pp. 286–292.

[37] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1400–1413, Apr. 2022.

[38] J. Fan, N. Murgovski, and J. Liang, "Efficient collision avoidance for autonomous vehicles in polygonal domains," 2023, *arXiv:2308.09103*.

[39] J. Guthrie, "A differentiable signed distance representation for continuous collision avoidance in optimization-based motion planning," in *Proc. IEEE Conf. Decision Control (CDC)*, Sep. 2022, pp. 7214–7221.

[40] Y. Zheng and K. Yamane, "Generalized distance between compact convex sets: Algorithms and applications," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 988–1003, Aug. 2015.

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

[42] M. Lutz and T. Meurer, "Efficient formulation of collision avoidance constraints in optimization based trajectory planning and control," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2021, pp. 228–233.

[43] V. Soltan, "Support and separation properties of convex sets in finite dimension," *Extracta Mathematicae*, vol. 36, no. 2, pp. 241–278, Dec. 2021.

[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, 2009.

[46] J. Tao, S. Scott, W. Joe, and D. Mathieu, "A geometric construction of coordinates for convex polyhedra using polar duals," in *Proc. Symp. Geom. Process.*, Vienna, Austria, pp. 181–186, 2005.

[47] C. A. Hayes, "The Heine–Borel theorem," *Amer. Math. Monthly*, vol. 63, no. 3, p. 180, Mar. 1956.

[48] M. Kesäniemi and K. Virtanen, "Direct least square fitting of hyperellipsoids," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 63–76, Jan. 2018.

[49] J. M. Aldaz, S. Barza, M. Fujii, and M. S. Moslehian, "Advances in operator cauchy-schwarz inequalities and their reverses," *Ann. Funct. Anal.*, vol. 6, no. 3, pp. 275–295, 2015.

[50] I. Pólik and T. Terlaky, "A survey of the S-lemma," *SIAM Rev.*, vol. 49, no. 3, pp. 371–418, Jan. 2007.

[51] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[52] 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.

[53] J. Wang, Y. Yan, K. Zhang, Y. Chen, M. Cao, and G. Yin, "Path planning on large curvature roads using driver-vehicle-road system based on the kinematic vehicle model," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 311–325, Jan. 2022.

[54] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

[55] A. Wächter and L. T. 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.

[56] A. C. Manav and I. Lazoglu, "A novel cascade path planning algorithm for autonomous truck-trailer parking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6821–6835, Jul. 2022.

[57] E. Börve, N. Murgovski, and L. Laine, "Interaction-aware trajectory prediction and planning in dense highway traffic using distributed model predictive control," 2023, *arXiv:2308.13053*.

[58] X. Ruan, Y. Huang, Y. Wang, Z. Fu, L. Xiong, and Z. Yu, "An efficient trajectory planning method with a reconfigurable model for any tractor-trailer vehicle," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 2, pp. 3360–3374, Sep. 2022.

[59] A. Elawad, N. Murgovski, M. Jonasson, and J. Sjoberg, "Road boundary modeling for autonomous bus docking subject to rectangular geometry constraints," in *Proc. Eur. Control Conf. (ECC)*, Delft, The Netherlands, Jun. 2021, pp. 1745–1750.

[60] Z. Zhang, L. Zhang, J. Deng, M. Wang, Z. Wang, and D. Cao, "An enabling trajectory planning scheme for lane change collision avoidance on highways," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 147–158, Jan. 2023.

[61] S. Gilroy et al., "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Lyon, France, Aug. 2021, pp. 2132–2139.

[62] X. Shen, E. L. Zhu, Y. R. Stürz, and F. Borrelli, "Collision avoidance in tightly-constrained environments without coordination: A hierarchical control approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2674–2680.

[63] R. Verschueren et al., "Acados—A modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, Mar. 2022.

[64] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

Jiayu Fan received the M.S. degree from the College of Mechanical Engineering, Southeast University, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His current research interests include motion planning and collision avoidance.

Nikolce Murgovski received the M.S. degree in software engineering from University West, Trollhättan, Sweden, in 2007, and the M.S. degree in applied physics and the Ph.D. degree in systems and control from the Chalmers University of Technology, Gothenburg, Sweden, in 2007 and 2012, respectively.

He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology. His typical research projects are in electromobility, autonomous driving, and active safety in automobiles. His research interests include modeling and simulation of dynamic systems and their use in problems of optimal plant design and control.

Jun Liang received the B.S. degree from the Department of Automation, Tsinghua University, in 1984, and the master's and Ph.D. degrees from the College of Control Science and Engineering, Zhejiang University.

He is currently a Professor with the College of Control Science and Engineering, Zhejiang University. His research interests include intelligent unmanned systems and industrial intelligence.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on August 28,2024 at 07:39:32 UTC from IEEE Xplore. Restrictions apply.