Efficient Formulation of Collision Avoidance Constraints in Optimization Based Trajectory Planning and Control

Max Lutz and Thomas Meurer

Abstract— To be applicable to real world scenarios trajectory planning schemes for mobile autonomous systems must be able to efficiently deal with obstacles in the area of operation. In the context of optimization based trajectory planning and control a number of different approaches to formulate collision avoidance constraints can be found in the literature. This work presents a novel formulation building on constructive solid geometry (CSG) to describe collision avoidance constraints. It is highly efficient due to a very low number of nonlinear inequality constraints required for a given number of obstacles and sample points and in contrast to the original CSG formulation allows to consider the controlled system's shape. To allow for a comparison, popular methods to represent obstacles from the literature are summarized and characterized, namely the simple ellipsoidal representation, the original CSG method as well as a direct and an indirect implementation of a signed distance based approach. A benchmark example shows the good performance of the proposed formulation. Here, optimal trajectory planning for marine surface vessels formulated as a nonlinear programming problem is used, where the scenario is designed based on the maritime test field in Kiel, Germany.

I. INTRODUCTION

Obstacle avoidance is an important element of optimization based trajectory planning for all autonomous systems that are not track bound such as vehicles, vessels, aircraft, robots and alike. Solution techniques that are based on the formulation of an optimal control problem (OCP), subsequent discretization and iterative solution using a nonlinear programming problem (NLP) solver require an explicit mathematical model of the obstacles, i.e., occupied or forbidden subsets of the controlled system's operating space. A natural choice due to its simplicity is the use of ellipsoids or even spheres, see, e.g., [1], [2], [3]. There are several important drawbacks associated with this simple modeling technique when realistic scenarios are considered, as discussed, e.g., in [4] for the task of parallel parking a car, or for the motion planning for a humanoid robot in [5]. It is thus not surprising that the interest in numerically efficient representations of objects in 2D and 3D goes back to the early research on numerical implementations of optimal control problems, e.g., aiming to solve optimal trajectory planning tasks for robots [6]. In the quest for an efficient mathematical formulation numerous different methods were developed. Some, such as the Gilbert-Johnson-Keerthi (GJK) algorithm [7], a simplex based iterative approach to calculate the distance between two convex sets, have become highly popular in other applications, namely computer graphics and physics simulators

Max Lutz and Thomas Meurer are with the Chair of Automatic Control, Faculty of Engineering, Kiel University, 24143 Kiel, Germany {mlut, tm}@tf.uni-kiel.de

(physics engines). On the other hand a constructive solid geometry (CSG) method with smooth approximation of the intersection and union operations has originally been developed for computer graphics applications in [8] but allows to represent obstacles in the OCP context in a compact way. Super ellipsoids, i.e., sets described by different p-norms, as used in the optimal trajectory planning for vehicles by, e.g., [9] and [10] are essentially a subset of the shapes attainable by this CSG method. Examples for the application of this CSG method itself include the formulation of constraint equations for the trajectory planning for underwater robots in [11] which also discusses the drawbacks of using a potential field for obstacle avoidance as proposed by, e.g., [12], [13] in combination with CSG. More recently, in the context of autonomous docking of marine surface vessels harbor structures are avoided using CSG based collision avoidance constraints in [14]. An interesting discussion on the connections between different approaches to CSG and some applications in various fields are given by [15] and the references therein. An example for the use of the GJK algorithm is given by the before mentioned humanoid robot motion planning [5], while collision avoidance for the car parking problem [4] is realized using dual variables. Due to the vast number of different approaches that have been proposed this short summary is by no means complete and only focuses on some popular examples chosen as a comparison for the novel formulation introduced in this paper. It is organized as follows. First the general OCP formulation is outlined in Section II. Then, Section III recapitulates popular obstacle modeling approaches. In Section IV the novel formulation of collision avoidance constraints is introduced. The benchmark scenario with simulation results follows in Section V. The paper concludes with some summarizing remarks.

II. PROBLEM FORMULATION

Often the infinite dimensional OCP expressing the trajectory planning task for a dynamic system is discretized by, e.g., a direct method to yield a NLP of the form

$$\min_{\boldsymbol{\xi}} c\left(\boldsymbol{\xi}\right) = \sum_{k=0}^{N} l_k\left(\boldsymbol{\xi}\right) \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{g}\left(\boldsymbol{\xi}\right) = \boldsymbol{0}, \tag{1b}$$

$$\boldsymbol{h}\left(\boldsymbol{\xi}\right) \leq \boldsymbol{0}, \tag{1c}$$

using $(N+1)$ equally spaced sample points in time. The cost function (1a) with the running cost $l_k(\boldsymbol{\xi})$, the equality (1b) and inequality constraints (1c) are written as functions of the vector of decision variables $\boldsymbol{\xi} \in \mathbb{R}^M$. In this context

obstacles in the operating space $\mathbb{R}^n$ are considered by formulating inequality constraints and including them in (1c). To allow for an efficient solution of the NLP with state of the art solvers based on iterative techniques, such as sequential quadratic programming or interior point methods, availability of an analytical expression for the gradient of the cost function as well as the nonlinear equality and inequality constraints w.r.t. the decision variables $\boldsymbol{\xi}$ is important.

Due to a lack of space subsequently only static obstacles are considered to facilitate the notation and also the illustration of simulation results. All approaches presented can easily be extended to accommodate for dynamic obstacle positions by introducing a time dependency in the obstacle modeling equations, the increase in computational effort of this can be expected to be similar for all methods. Other issues that may arise with dynamic obstacles such as, e.g. the question of recursive feasibility when repeatedly solving the optimization problem in the context of model predictive control, are generally not linked to the choice of obstacle modeling approach and thus also do not affect the latter.

In the following the controlled object's position is denoted by the point $\boldsymbol{p}(\boldsymbol{\xi}) \in \mathbb{R}^n$ if collision avoidance is enforced for a reference point only, where the dependency on the decision variables is stated for clarity. If the system's shape is considered the occupied area is denoted by $\mathcal{V}(\boldsymbol{\xi}) \subset \mathbb{R}^n$.

III. Existing Obstacle Formulations

A. Spherical and Ellipsoidal Obstacles

Modeling obstacles as regions with ellipsoidal or as a special case spherical shape is attractive due to the straightforward mathematical representation of the obstacle region

$$\mathcal{O} = \left\{ \boldsymbol{p} \in \mathbb{R}^n \,|\, (\boldsymbol{p} - \boldsymbol{p}_{\mathcal{O}})^{\mathsf{T}} A (\boldsymbol{p} - \boldsymbol{p}_{\mathcal{O}}) \leq 1 \right\}, \quad (2)$$

where $\boldsymbol{p}_{\mathcal{O}} \in \mathbb{R}^n$ denotes the obstacle center and A is a positive definite, symmetric matrix defining the shape. For a number of Q obstacle regions a total number of $Q(N + 1)$ nonlinear inequalities have to be added to the OCP, the calculation of the corresponding gradient is straightforward. Clearly the assumption that all obstacles or obstacle sub-regions can be approximated well by an ellipsoid may not hold in a realistic scenario. Further complications arise if a safety distance to the obstacle is to be respected, as a surface with a uniform distance to an ellipsoid is not an ellipsoid itself [16]. Also, if the controlled object's shape is to be taken into account it is not trivial to efficiently calculate a distance between two general ellipsoids, even in the 2D case.

B. Constructive Solid Geometry

Some of the shortcomings of the spherical/ellipsoidal obstacle representation can be overcome by the CSG method proposed by [8]. Central to the method is the non-negative function f_{o} called the (characteristic) defining function, a type of real-valued implicit point membership classification (PMC) function [15]. It describes geometric shapes in space, i.e., $f_{\mathrm{o}} : \boldsymbol{p} \in \mathbb{R}^n \mapsto \mathbb{R}^+$, where $f_{\mathrm{o}} = 1$ defines the boundary, $f_{\mathrm{o}} < 1$ the interior and $f_{\mathrm{o}} > 1$ the exterior. Arbitrarily shaped convex or non-convex regions may be constructed

using the intersection and union operations. In [8] the smooth approximations $I_p(f_{\mathrm{o},1}, \ldots, f_{\mathrm{o},K}) = (f_{\mathrm{o},1}^p + \cdots + f_{\mathrm{o},K}^p)^{\frac{1}{p}}$ and $U_p(f_{\mathrm{o},1}, \ldots, f_{\mathrm{o},K}) = (f_{\mathrm{o},1}^{-p} + \cdots + f_{\mathrm{o},K}^{-p})^{-\frac{1}{p}}$ of the intersection and union operation with $p \in \mathbb{N}$ are proposed, for details refer to [8]. Equation (2) gives an example for the defining function of an ellipsoid, if A is allowed to be indefinite it can also express elliptic cylinders and slabs.

In addition to the improved flexibility w.r.t. the possible obstacle shapes the concept also allows to drastically reduce the number of nonlinear inequality constraints, as irrespective of the number of individual obstacles the application of the approximated union operation allows to construct $f_{\mathrm{o},\Sigma}(\boldsymbol{\xi}) = U_p(f_{\mathrm{o},1}, \ldots, f_{\mathrm{o},Q}) \in \mathbb{R}$, thus only $(N + 1)$ constraints of the form $1 - f_{\mathrm{o},\Sigma}(x_k) \leq 0$ for $k = 0, \ldots, N$ need to be added to the OCP. While the complexity of the constraint equations increases in comparison to simple ellipsoidal shapes, the analytic gradient can still be easily computed. However, the value of a PMC function does not yield a distance measure that can directly be mapped to the euclidean distance, complicating the introduction of a uniform safety distance. Another drawback of the method is that it is difficult to consider the controlled object's shape. A solution by sampling its boundary is given in, e.g., [14].

C. Dual Signed Distance Obstacle Representation

The distance between two sets $\mathcal{V}(\boldsymbol{\xi}), \mathcal{O} \in \mathbb{R}^n$ is generalized by the signed distance function

$$\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \mathrm{dist}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) - \mathrm{pen}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}), \quad (3)$$

combining the euclidean distance and the penetration,

$$\mathrm{dist}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \inf_{\boldsymbol{p} \in \mathbb{R}^n} \{ \|\boldsymbol{p}\| : (\mathcal{V}(\boldsymbol{\xi}) + \boldsymbol{p}) \cap \mathcal{O} \neq \emptyset \}, \quad (4a)$$

$$\mathrm{pen}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \inf_{\boldsymbol{p} \in \mathbb{R}^n} \{ \|\boldsymbol{p}\| : (\mathcal{V}(\boldsymbol{\xi}) + \boldsymbol{p}) \cap \mathcal{O} = \emptyset \}. \quad (4b)$$

Simply speaking the signed distance extends the notion of distance to the case where objects penetrate, in which case it is equal to the negative euclidean distance either object needs to be moved for them to be out of contact. To calculate the signed distance (3) generally the optimization problems (4a) and (4b) have to be solved, as such there is no explicit representation of the signed distance in terms of the decision variables $\boldsymbol{\xi}$. For compact convex sets however, strong duality holds and an implementation using dual variables is proposed by [4] in the context of optimal control. To simplify notation in the following shapes are limited to convex polytopes, a limitation that results in a negligible restriction regarding the ability to express various practically relevant obstacle shapes.

A polyhedral set in $\mathbb{R}^n$ can elegantly be described by

$$\mathcal{O} = \left\{ \boldsymbol{p} \in \mathbb{R}^n \,|\, A\boldsymbol{p} - \boldsymbol{b} \leq \boldsymbol{0} \right\}, \quad (5)$$

using the K face unit normal vectors $\boldsymbol{n}_i$, $i = 1, \ldots, K$ to build the matrix $A = [\boldsymbol{n}_1, \ldots, \boldsymbol{n}_K]^{\mathsf{T}} \in \mathbb{R}^{K \times n}$ and arbitrary points $\boldsymbol{o}_i$, $i = 1, \ldots, K$ on the hyperplanes containing the faces to form the vector $\boldsymbol{b} = [\boldsymbol{n}_1^{\mathsf{T}} \boldsymbol{o}_1, \ldots, \boldsymbol{n}_K^{\mathsf{T}} \boldsymbol{o}_K]^{\mathsf{T}} \in \mathbb{R}^K$. As the formulation is based on the signed distance it is straightforward to introduce a minimum distance $d_{\min}$ to

229

each obstacle. For the point system case with position $\boldsymbol{p}(\boldsymbol{\xi})$ the collision-avoidance constraints may then be expressed as

$$\mathrm{sd}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O}) \geq d_{\min} \;\Leftrightarrow\; \exists \boldsymbol{\mu} \geq \mathbf{0}:$$
$$\boldsymbol{\mu}^{\mathsf{T}}(A\boldsymbol{p}(\boldsymbol{\xi}) - \boldsymbol{b}) \geq d_{\min}, \qquad (6)$$
$$\|A^{\mathsf{T}}\boldsymbol{\mu}\| = 1$$

using the dual variables $\boldsymbol{\mu} \in \mathbb{R}^K$. If the shape of the controlled system is to be considered, additional dual variables $\boldsymbol{\lambda} \in \mathbb{R}^L$ have to be used, where L is the number of faces of the polyhedron representing the controlled system. The matrix with the corresponding face unit normal vectors is denoted by $V(\boldsymbol{\xi}) = [\boldsymbol{n}_1(\boldsymbol{\xi}), \ldots, \boldsymbol{n}_L(\boldsymbol{\xi})]^{\mathsf{T}} \in \mathbb{R}^{L \times n}$. The components of the vector $\boldsymbol{q}(\boldsymbol{\xi}) = [\boldsymbol{n}_1^{\mathsf{T}}(\boldsymbol{\xi})\boldsymbol{v}_1(\boldsymbol{\xi}), \ldots, \boldsymbol{n}_L^{\mathsf{T}}(\boldsymbol{\xi})\boldsymbol{v}_L(\boldsymbol{\xi})]^{\mathsf{T}} \in \mathbb{R}^L$ are given by the inner product of the face unit normal vector and a point $\boldsymbol{v}_k(\boldsymbol{\xi})$, $k = 1, \ldots, L$ that lies on the hyperplane containing the corresponding face. The collision avoidance constraints

$$\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) \geq d_{\min} \;\Leftrightarrow\; \exists \boldsymbol{\mu} \geq \mathbf{0},\; \boldsymbol{\lambda} \geq \mathbf{0}:$$
$$-\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{q}(\boldsymbol{\xi}) - \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{b} \geq d_{\min},$$
$$V^{\mathsf{T}}(\boldsymbol{\xi})\boldsymbol{\lambda} + A^{\mathsf{T}}\boldsymbol{\mu} = \mathbf{0}, \qquad (7)$$
$$\|A^{\mathsf{T}}\boldsymbol{\mu}\| = 1$$

result. For the derivation of these dual implementations and the formulation for the case of general convex sets the interested reader is referred to [4] and the references therein.

Obviously the dual signed distance based obstacle representation comes at the cost of a large number of equality and inequality constraints that have to be added to the problem and an additional increase of the problem dimension due to the introduction of the dual variables. For the point system case with $(N+1)$ sample points and Q obstacles $2(N+1)Q$ nonlinear equality and inequality constraints and $(N+1)\sum_{i=1}^{Q}K_i$ additional dual decision variables result. If the shape of the controlled system is to be considered the increase in dimension and number of constraints is even more drastic, with further $n(N+1)Q$ linear equality constraints, where n is the dimension of the operating space, and $(N+1)QL$ additional dual decision variables.

D. Direct Signed Distance Obstacle Representation

The dual signed distance obstacle representation uses a reformulation of the signed distance problem (4) in terms of dual variables and subsequently solves the problem with the NLP solver used to tackle the OCP itself, yet there are dedicated algorithms specifically developed to solve the signed distance problem. As mentioned above a direct implementation of a signed distance based obstacle formulation has to deal with the lack of an explicit equation of the signed distance in terms of the decision variables. When the GJK algorithm [7] and the expanding polytope algorithm (EPA) [17] are used to calculate the distance and the penalty, respectively, they also provide the separation/penetration vector, i.e., $\boldsymbol{p}$ in (4a) and (4b). This does allow the calculation of the proper gradient in the current point in non-degenerate situations, as also used by [5]. Degenerate situations are encountered when the closest features of two shapes are

not unique points but parallel faces, compare [5, Fig. 4]. The approach is dismissed as a local linearization with uncertain error bounds by [4], but this is not inherent to the method itself but rather caused by the way it is implemented in the context of a sequential convex optimization scheme by [5]. Essentially, except for degenerate situations, from a numerical point of view the calculation of the obstacle constraints with the GJK and EPA algorithms can be handled in the same way as any nonlinear constraint in NLP, where the gradient in the current point, i.e., the local linearization, is provided to the solver. In line with the findings of [5] degenerate situations do not appear to be a problem in practical application and, in any case, do not occur if the controlled system's shape is neglected. To include the collision avoidance constraints in the OCP for Q arbitrarily shaped but convex obstacles only $(N+1)Q$ inequalities of the form $d_{i,\min} - \mathrm{sd}(\cdot, \mathcal{O}_i) \leq 0$ for $i = 1, \ldots, Q$ have to be considered, where $\cdot$ stands for $\{\boldsymbol{p}(\boldsymbol{\xi})\}$ or $\mathcal{V}(\boldsymbol{\xi})$. However, the computational cost of running the GJK and, in case of penetration, the EPA algorithm has to be taken into account.

IV. NOVEL CSG-BASED OBSTACLE AVOIDANCE CONSTRAINT FORMULATION

In this section a novel formulation based on the CSG concept for obstacle constraints in an NLP describing an optimization based trajectory planning and control problem is outlined. First the general idea is presented for the case in which the collision avoidance is enforced for the controlled system's reference point, then it is discussed how the concept can be extended to consider the controlled system's shape.

Assumption 1: An obstacle can be represented as a set $\mathcal{O} \in \mathbb{R}^n$ that results from the intersection of a finite number K of primitives $\mathcal{P}_i \in \mathbb{R}^n$, $i = 1, \ldots, K$, i.e., $\mathcal{O} = \bigcap_{i=1}^{K} \mathcal{P}_i$.

Lemma 1 ([15]): The signed distance $\mathrm{sd}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O})$ is a valid PMC function to model objects with CSG.

Proposition 1: When Assumption 1 holds the signed distance $\mathrm{sd}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O})$ is bound from below by

$$\max_{\mathcal{P}_i}\{\mathrm{sd}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{P}_i)\} \leq \mathrm{sd}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O}), \qquad (8)$$

where the controlled system's position is $\boldsymbol{p}(\boldsymbol{\xi})$.

Proof: As $\forall i \in \{1, \ldots, K\} : \mathcal{O} \subseteq \mathcal{P}_i$ obviously $\mathrm{dist}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{P}_i) \leq \mathrm{dist}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O})$ and $\mathrm{pen}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{P}_i) \geq \mathrm{pen}(\{\boldsymbol{p}(\boldsymbol{\xi})\}, \mathcal{O})$ follow. With definition of the signed distance (3) Proposition 1 holds. ∎

Clearly this is advantageous whenever it is easy to determine the signed distance to the individual primitives, as, e.g., if they are half-spaces, balls and alike.

Assumption 2: Let the controlled system's shape be described by the set $\mathcal{V}(\boldsymbol{\xi}) \in \mathbb{R}^n$, resulting from the intersection of a finite number L of primitives $\mathcal{P}_{\mathcal{V},j}(\boldsymbol{\xi}) \in \mathbb{R}^n$, $j = 1, \ldots, L$, i.e., $\mathcal{V}(\boldsymbol{\xi}) = \bigcap_{j=1}^{L} \mathcal{P}_{\mathcal{V},j}(\boldsymbol{\xi})$.

To be able to consider the controlled system's shape the Minkowski difference, defined for two sets $\mathcal{V}(\boldsymbol{\xi}), \mathcal{O} \in \mathbb{R}^n$

$$\mathcal{O} - \mathcal{V}(\boldsymbol{\xi}) := \{(\boldsymbol{o} - \boldsymbol{v}) \in \mathbb{R}^n | \boldsymbol{o} \in \mathcal{O} \wedge \boldsymbol{v} \in \mathcal{V}(\boldsymbol{\xi})\} \quad (9)$$

is adopted, a concept that the signed distance based methods outlined above also make use of internally.

230

Lemma 2 ([18]): $\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \mathrm{sd}(\{\mathbf{0}\}, \mathcal{O} - \mathcal{V}(\boldsymbol{\xi}))$
This reduces the problem to again finding the signed distance between a point and a set instead of between two sets.

Lemma 3: The Minkowski difference distributes over the set intersection, i.e., for sets $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbb{R}^n$ it holds $(\mathcal{A} \cap \mathcal{B}) - \mathcal{C} = (\mathcal{A} - \mathcal{C}) \cap (\mathcal{B} - \mathcal{C})$.

Proof: Consider $\boldsymbol{x} \in (\mathcal{A} - \mathcal{C}) \cap (\mathcal{B} - \mathcal{C})$ and $\boldsymbol{c} \in \mathcal{C}$. Clearly $\boldsymbol{x} \in (\mathcal{A} - \mathcal{C}) \wedge \boldsymbol{x} \in (\mathcal{B} - \mathcal{C})$. With (9) $\exists \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B} : \boldsymbol{x} = \boldsymbol{a} - \boldsymbol{c} \wedge \boldsymbol{x} = \boldsymbol{b} - \boldsymbol{c} \Rightarrow \boldsymbol{a} = \boldsymbol{b} \in \mathcal{A} \cap \mathcal{B}$. Thus applying (9) again $(\mathcal{A} \cap \mathcal{B}) - \mathcal{C} \supseteq (\mathcal{A} - \mathcal{C}) \cap (\mathcal{B} - \mathcal{C})$ follows.

Now take $\boldsymbol{y} \in (\mathcal{A} \cap \mathcal{B}) - \mathcal{C}$. Again using (9) $\exists \boldsymbol{z} \in \mathcal{A} \cap \mathcal{B}, \boldsymbol{c} \in \mathcal{C} : \boldsymbol{y} = \boldsymbol{z} - \boldsymbol{c}$. Now with $\boldsymbol{z} \in \mathcal{A} \cap \mathcal{B} \Rightarrow \exists \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B} : \boldsymbol{a} = \boldsymbol{b} = \boldsymbol{z}$. Thus with (9) $\boldsymbol{y} \in \mathcal{A} - \mathcal{C} \wedge \boldsymbol{y} \in \mathcal{B} - \mathcal{C}$ and $(\mathcal{A} - \mathcal{C}) \cap (\mathcal{B} - \mathcal{C}) \supseteq (\mathcal{A} \cap \mathcal{B}) - \mathcal{C}$. ∎

Proposition 2: If Assumption 1 and Assumption 2 hold the signed distance $\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O})$ is bound from below by

$$\max_{\mathcal{P}_i, \mathcal{P}_{\mathcal{V}, j}} \{\mathrm{sd}(\{\mathbf{0}\}, \mathcal{O} - \mathcal{P}_{\mathcal{V}, j}(\boldsymbol{\xi})), \mathrm{sd}(\{\mathbf{0}\}, \mathcal{P}_i - \mathcal{V}(\boldsymbol{\xi}))\}$$
$$\leq \mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}), \quad (10)$$

with $i \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, L\}$.

Proof: Under Assumptions 1 and 2, using the Minkowski difference (9) and Lemma 3 the equalities $\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \mathrm{sd}(\{\mathbf{0}\}, \mathcal{O} - \mathcal{V}(\boldsymbol{\xi})) = \mathrm{sd}(\{\mathbf{0}\}, \mathcal{O} - \bigcap_{j=1}^{L} \mathcal{P}_{\mathcal{V}, j}) = \mathrm{sd}(\{\mathbf{0}\}, \bigcap_{j=1}^{L}(\mathcal{O} - \mathcal{P}_{\mathcal{V}, j}))$ hold. In the same manner $\mathrm{sd}(\mathcal{V}(\boldsymbol{\xi}), \mathcal{O}) = \mathrm{sd}(\{\mathbf{0}\}, \bigcap_{i=1}^{K}(\mathcal{P}_i - \mathcal{V}(\boldsymbol{\xi})))$ results. Following the proof of Proposition 1 $\max_{\mathcal{P}_{\mathcal{V}, j}} \mathrm{sd}(\{\mathbf{0}\}, \mathcal{O} - \mathcal{P}_{\mathcal{V}, j}) \leq \mathrm{sd}(\{\mathbf{0}\}, \bigcap_{j=1}^{L}(\mathcal{O} - \mathcal{P}_{\mathcal{V}, j}))$ and $\max_{\mathcal{P}_i} \mathrm{sd}(\{\mathbf{0}\}, \mathcal{P}_i - \mathcal{V}(\boldsymbol{\xi})) \leq \mathrm{sd}(\{\mathbf{0}\}, \bigcap_{i=1}^{K}(\mathcal{P}_i - \mathcal{V}(\boldsymbol{\xi})))$ follow. ∎

Informally speaking, similar to the point system case the lower bound on the signed distance between the set occupied by the system to an obstacle is given by the maximum of the signed distance between the origin and the geometric primitives forming the Minkowski difference of the system and the obstacle. For the gradient the same arguments as made in subsection III-D hold regarding degenerate situations. Also, the maximum function yields a non-smooth gradient elsewhere. However, if a smooth gradient is required for all non-degenerate situations the maximum function can be implemented using a smooth approximation. The LogSumExp (LSE) function $\mathrm{LSE} : \mathbb{R}^m \mapsto \mathbb{R}$,

$$\mathrm{LSE}(d_1, \ldots, d_m) = \frac{1}{\alpha} \log \left(\sum_{i=1}^{m} \alpha(d_i - d_0) \right) + d_0, \quad (11)$$

is a stable solution and well established in the machine learning and artificial intelligence community. Here $\alpha \in \{\mathbb{R} \backslash 0\}$ determines the quality of the approximation, with $\mathrm{LSE}(d_1, \ldots, d_m) \to \max(d_1, \ldots, d_m)$ for $\alpha \to \infty$ and $\mathrm{LSE}(d_1, \ldots, d_m) \to \min(d_1, \ldots, d_m)$ for $\alpha \to -\infty$. The parameter d_0, typically chosen to be $d_0 = \max(d_1, \ldots, d_m)$ for $\alpha > 0$ or $d_0 \min(d_1, \ldots, d_m)$ for $\alpha < 0$, is introduced for reasons of numerical stability. For a given $\alpha > 0$ the approximation error is bound according to $\max(d_1, \ldots, d_m) \leq \mathrm{LSE}(d_1, \ldots, d_m) \leq \max(d_1, \ldots, d_m) + \frac{1}{\alpha} \log(m)$, the equivalent holds for $\alpha < 0$. The availability of an upper bound on the approximation error allows to introduce a small

safety distance $d_{\min} = 1/\alpha \log m$ to ensure that despite the use of the approximation no collision occurs in any point.

As a CSG formulation it is possible to consider the union over all obstacles, possibly using the smooth approximation of the minimum (11), depending on the requirements of the solver. This reduces the required number of collision avoidance constraints needed to $(N + 1)$, the number of sample points, irrespective of the number of (sub-)obstacles Q. The same is true for the simple ellipsoidal obstacle representation and the direct signed distance obstacle representation from subsection III-D, as both are valid PMC functions.

V. COMPARATIVE SIMULATION STUDY

As a benchmark problem the energy optimal point to point trajectory planning for underactuated autonomous surface vessels as presented in [19] is considered. There it is demonstrated how using a flatness based direct method this task may be expressed as a finite dimensional NLP. In the following first key elements of the approach are outlined, then the layout of the comparative study of the formulations discussed above is described. The former is intentionally kept brief as the focus of the present work is on the latter.

The OCP is formulated using the nonlinear dynamics

$$\dot{\boldsymbol{\eta}} = R(\psi)\boldsymbol{\nu} \quad (12a)$$
$$M\dot{\boldsymbol{\nu}} = -(C(\boldsymbol{\nu}) + D(\boldsymbol{\nu}))\boldsymbol{\nu} + \boldsymbol{\tau}, \quad (12b)$$

where $\boldsymbol{\eta}$ denotes the vessel pose in the north-east-down (NED) frame. The body fixed velocities $\boldsymbol{\nu}$ are calculated based on the effect of external forces and moments $\boldsymbol{\tau} = [\tau_u, \tau_v, \tau_r]^\mathsf{T}$. Here M is the inertia, $C(\boldsymbol{\nu})$ the coriolis and $D(\boldsymbol{\nu})$ the damping matrix including nonlinear terms. Actuator and actuator rate limits are considered as box constraints.

To realize energy optimal trajectory planning $l(t) = (\tau_u / \tau_{u, \max})^2 + (\tau_r / \tau_{r, \max})^2$ is used as the running cost, assuming that the quadratic form adequately approximates instantaneous energy consumption of the actuators.

For the fully actuated case the dynamics (12) are differentially flat, with a flat output $\boldsymbol{z}$ given by the pose of the system in the NED frame, i.e., $\boldsymbol{z} = \boldsymbol{\nu}$. Thus exact flat parametrizations $\boldsymbol{\theta_x}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}})$ and $\boldsymbol{\theta_\tau}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}})$ of the system states and inputs as functions of the flat output and its time derivatives exist [20]. This allows to write all constraints in terms of the flat output and its derivatives, as well as the cost function, which then implicitly encodes the system dynamics. Considering a parametrized ansatz function for the flat output yields a finite dimensional NLP of the form (1), where the decision variables are the parameters of the ansatz function. The control input is given by the surge force τ_u and the yaw moment τ_r, to apply to the underactuated case the sway force τ_v is constrained to zero. For details and parameter values see [19] and the references therein.

The benchmark scenario is built to replicate the setting of the real world maritime test field in Kiel, Germany at $54°19'40''$ N, $10°09'54''$ E. The setting is re-scaled as tests using a small unmanned surface vessel (USV) are planned. For further information on the USV model the reader is referred to [21]. A local NED coordinate system
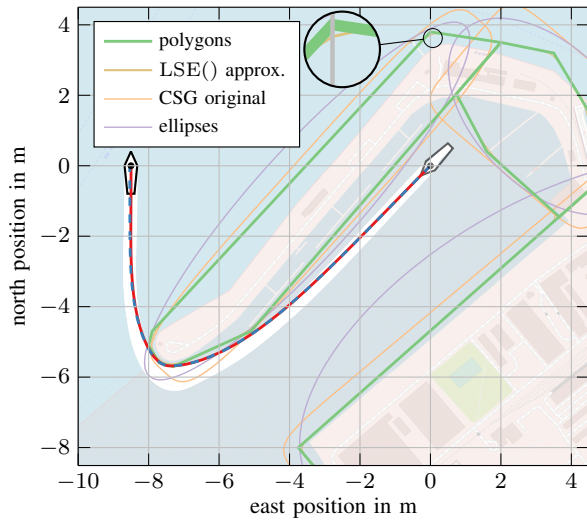
(a) Local NED frame with vessel and all obstacle shapes used for the simulation study. Additionally the swept volume of the optimization result for the proposed formulation (10) using a hard maximum and the union over all obstacles taking system's shape into account is given (white).
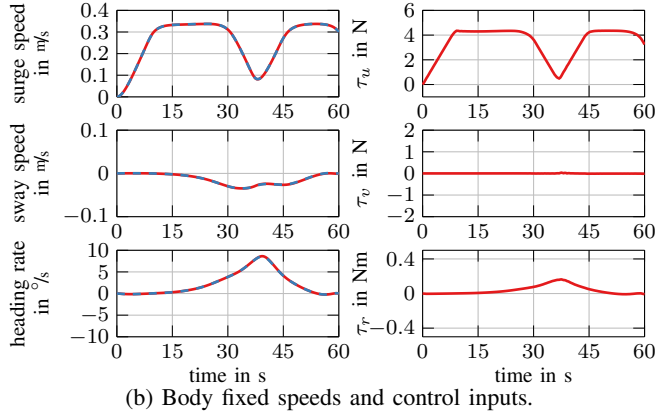
(b) Body fixed speeds and control inputs.

Fig. 1: Benchmark scenario overview including optimization results (red) and and simulation results using MATLAB's `ode45()` (blue, dashed) for the proposed formulation (8) using a hard maximum and the union over all obstacles.

is defined with the origin fixed on the vessel position at $t_0 = 0$. The scenario is chosen such that the vessel is at rest initially, $x_0 = [0, 0, -135°, 0, 0, 0]^\mathsf{T}$, and a trajectory to leave the harbor area is planned with $t_e = 60\,\text{s}$ and $x_e = [0, -8.5\,\text{m}, 0, 0.3\,\text{m/s}, 0, 0]^\mathsf{T}$. The OCP is solved with a total number of 61 equally spaced sample points using the NLP solver SNOPT [22]. To allow for an equal comparison the harbor structures are enclosed in 3 ellipses for the ellipsoidal formulation, in 3 rounded rectangular shapes for the original CSG formulation, compare [19, eq. (14)], or in 3 polygons with a total number of 18 vertices for all other cases. The vessel can be enclosed by a symmetrical polygon with a length of 1.2 m and a width of 0.36 m and 5 vertices, see Fig. 1a. The polygons are described in the form of (5), using half planes as geometric primitives for the novel CSG based method. Run times for a MATLAB implementation

on a computer with Intel Core i5-6200U CPU clocked at 2.30 GHz are reported in Table Ia for the point system case and in Table Ib for the methods that allow to consider the system's shape. All reported times are the average over ten runs. The maximum obstacle penetration of the optimization output is calculated for the actual obstacle shapes, not the smoothed approximations, where relevant. The number of obstacle constraint function calls and the run time of the solver itself, i.e., the time required to solve the problem not taking into account the time spent in user defined functions, are given to demonstrate that the effect of the code efficiency of the user defined functions on the qualitative assessment of the results is negligible. For reference the body fixed speeds and control inputs are given in Fig. 1b.

The results show that the novel CSG based obstacle avoidance constraint formulation is able to efficiently encode the required constraints and that it does perform very well for a point system and also if the system dimension is relevant. This is especially obvious in comparison to the ellipsoidal method, not even taking into account the reduced optimality of the results due to the limitations in shaping the obstacle constraints. While for the original CSG based method and the formulation using a hard maximum it appears advantageous to reduce the number of constraints by taking the union over all obstacles, for the LSE-smoothed version this is not the case, most likely there the increased nonlinearity is more relevant. In all cases an educated initial guess using the A* based concept described in [19, Sec. III-B] is used, including the dual variables. Nevertheless the dual signed distance obstacle formulation performs poorly for the point system case and scales badly as evident when considering the system's shape. The direct signed distance approach on the other hand benefits from the efficiency of the highly refined CSG and EPA algorithms, as highlighted by the small difference between solver run time and total run time for a similar number of user function calls as the proposed CSG based method when considering the system dimension. It is to be expected that this advantage becomes more important with increasing number and complexity of the shapes, however there is still room for improvement in the implementation of the proposed CSG based method by using iterative methods to calculate (8) and (10). It appears that the initial guess is slightly better in the case where the system shape is considered, as solver iterations (indicated by user function calls) and solution times do not reflect the increased complexity for all methods alike.

VI. CONCLUSION

A novel method to formulate the collision avoidance constraints for OCPs of mobile autonomous systems is proposed that combines the strengths of commonly used concepts by building on the CSG method. For the two cases, neglecting system's shape as well as considering it, it is proven that the collision avoidance constraints may be implemented using a lower bound on the signed distance to individual obstacles, a key difference to existing CSG based formulations that do not make the connection to the signed distance and therefore

TABLE I: Results of the simulation study. The novel formulation for the collision avoidance constraints according to Section IV is denoted by 'proposed hard $\max()$' for (8) and 'proposed LSE()' if the approximation (11) is used.

(a) Point system case.

	proposed hard $\max()$		proposed LSE()		ellipsoidal	CSG original		signed distance	
	union	separate	union	separate		union	separate	dual	direct
obstacle constraints	61	183	61	183	183	61	183	366	183
optimization variables	189	189	189	189	1189	189	189	1287	189
constraint function calls	31	35	25	34	40	34	26	81	47
max. obstacle penetration (m)	1.17e-15	2.66e-15	6.39e-2	6.39e-2	0	0	0	1.78e-15	3.3e-15
cost function value	38.22	38.27	37.58	37.58	42.91	40.78	40.78	38.62	38.15
run time solver only (s)	1.29	1.39	1.93	1.48	1.38	1,37	1.46	2,28	1.67
run time total (s)	1.47	1.66	2.08	1.78	1.68	1.60	1.68	4.46	2.02
run time total (relative)	1	1.23	1.41	1.21	1.14	1.09	1.14	3.18	1.37

(b) System shape considered.

	proposed hard $\max()$		proposed LSE()		signed distance	
	union	separate	union	separate	dual	direct
obstacle constraints	61	183	61	183	732	183
optimization variables	189	189	189	189	2202	189
constraint function calls	22	24	82	26	123	23
max. penetration (m)	0	3.03e-10	0.11	0.11	2.6e-8	0
cost function value	40.18	40.18	38.90	38.90	39.92	40.18
run time solver only (s)	0.95	1.29	4.61	1.22	220	1.26
run time total (s)	1.55	1.90	7.13	2.02	248	1.52
run time total (relative)	1	1.23	4.60	1.30	160	0.98

struggle to consider the controlled system's shape. Furthermore, it is demonstrated how the use of the LSE()-function yields smooth gradients while confining the approximation error to strict bounds. Using a flatness based direct method comparative simulation results are reported for the proposed novel formulation as well as established methods based on ellipsoids, direct and indirect implementations of the signed distance collision avoidance constraints and the original CSG based formulation using a positive defining function. To this end a realistic scenario based on the real world maritime test area in Kiel, Germany is addressed. The versatility and efficiency of the novel concept is indicated for the point system case as well as for the case taking into account the system shape. Current research is focused on validation of the simulation results with a small scale USV.

REFERENCES

[1] Y.-C. Chen and M. Vidyasagar, "Optimal trajectory planning for planar n-link revolute manipulators in the presence of obstacles," in *Proceedings. 1988 IEEE Int. Conf. Robot. Autom.* IEEE Comput. Soc. Press, 1988, pp. 202–208.

[2] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *2016 IEEE Int. Conf. Robot. Autom.*, vol. 2016-June. IEEE, may 2016, pp. 2958–2964.

[3] G. Bitar, M. Breivik, and A. M. Lekkas, "Energy-Optimized Path Planning for Autonomous Ferries," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 389–394, 2018.

[4] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-Based Collision Avoidance," *IEEE Trans. Control Syst. Technol., early access*, 2020.

[5] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Rob. Res.*, vol. 33, no. 9, pp. 1251–1270, aug 2014.

[6] E. Gilbert and D. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *IEEE J. Robot. Autom.*, vol. 1, no. 1, pp. 21–30, 1985.

[7] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[8] A. Ricci, "A constructive geometry for computer graphics," *Comput. J.*, vol. 16, no. 2, pp. 157–160, feb 1973.

[9] Q. Gong, L. R. Lewis, and I. M. Ross, "Pseudospectral Motion Planning for Autonomous Vehicles," *J. Guid. Control. Dyn.*, vol. 32, no. 3, pp. 1039–1045, may 2009.

[10] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *2017 Am. Control Conf.* IEEE, may 2017, pp. 5568–5573.

[11] Y. Wang, D. M. Lane, and G. J. Falconer, "Two novel approaches for unmanned underwater vehicle path planning: Constrained Optimization and Semi-infinite Constrained Optimization," *Robotica*, vol. 18, no. 2, pp. 123–142, 2000.

[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE Int. Conf. Robot. Autom.*, vol. 2. IEEE, 1985, pp. 500–505.

[13] E. Romon and D. Koditschek, "Exact robot navigation in geometrically complicated but topologically simple spaces," in *Proceedings., IEEE Int. Conf. Robot. Autom.*, vol. 3. IEEE Comput. Soc. Press, may 1990, pp. 1937–1942.

[14] S. Helling, M. Lutz, and T. Meurer, "Flatness-based MPC for underactuated surface vessels in confined areas," *IFAC World Congress, Berlin 2020, to appear*.

[15] V. Shapiro, "Real functions for representation of rigid solids," *Comput. Aided Geom. Des.*, vol. 11, no. 2, pp. 153–175, apr 1994.

[16] R. T. Farouki, "Conic Approximation of Conic Offsets," *J. Symb. Comput.*, vol. 23, no. 2-3, pp. 301–313, feb 1997.

[17] G. Van Den Bergen, *Collision detection in interactive 3D environments*. CRC Press, 2003.

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

[19] M. Lutz and T. Meurer, "Optimal Trajectory Planning and Model Predictive Control of Underactuated Marine Surface Vessels using a Flatness-Based Approach," *Am. Control Conf. (ACC), accepted,*, 2021, arXiv preprint arXiv:2101.12730 [eess.SY], jan 2021.

[20] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, jun 1995.

[21] K. D. Do and J. Pan, "Global robust adaptive path following of underactuated ships," *Automatica*, vol. 42, pp. 1713–1722, oct 2006.

[22] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, pp. 99–131, 2005.