Creating Star Worlds
Modelling Concave Obstacles for Reactive Motion Planning
Albin Dahlin and Yiannis Karayiannidis

Abstract—Motion planning methods like navigation functions and harmonic potential fields provide (almost) global convergence and are suitable for obstacle avoidance in dynamically changing environments due to their reactive nature. A common assumption in the control design is that the robot operates in a disjoint star world, i.e. all obstacles are strictly starshaped and mutually disjoint. However, in real-life scenarios obstacles may intersect due to expanded obstacle regions corresponding to robot radius or safety margins. To broaden the applicability of aforementioned reactive motion planning methods, we propose a method to transform a workspace of intersecting obstacles to a disjoint star world. The algorithm is based on two novel concepts presented here, namely admissible kernel and starshaped hull with specified kernel, which are closely related to the notion of starshaped hull. The utilization of the proposed method is illustrated with examples of a robot operating in a 2D workspace using a harmonic potential field approach in combination with the developed algorithm.

Index Terms—Collision Avoidance, Reactive and Sensor-Based Planning, Motion and Path Planning, Mobile Manipulation

I. INTRODUCTION

One of the central problems in robotics is to plan the motion of the robot to a desired goal state while avoiding collisions with obstacles [1]. Motion planning remains an active topic of research as modern robots, such as collaborative robots, mobile robots, unmanned aerial vehicles or even combined robotic systems for mobile manipulation, are intended for use in dynamically changing environments. Traditionally, motion planning problems have been solved considering static maps, but when considering changes in the environments and moving obstacles the robot must constantly adjust its planned path to avoid crashes. A common method to tackle such problems is to construct closed form control laws formulated as dynamical systems that provide stability and convergence guarantees in combination with instant reactivity to changing environments. Specifically, artificial potential fields, introduced in [2], based on scalar functions, guiding the robot with an attractive force to the goal and repulsive forces from the obstacles, remain an active topic of research as modern robots, such as collaborative robots, mobile robots, unmanned aerial vehicles or even combined robotic systems for mobile manipulation, are intended for use in dynamically changing environments. One of the central problems in robotics is to plan the motion of the robot to a desired goal state while avoiding collisions with obstacles [1]. Motion planning remains an active topic of research as modern robots, such as collaborative robots, mobile robots, unmanned aerial vehicles or even combined robotic systems for mobile manipulation, are intended for use in dynamically changing environments. Traditionally, motion planning problems have been solved considering static maps, but when considering changes in the environments and moving obstacles the robot must constantly adjust its planned path to avoid crashes. A common method to tackle such problems is to construct closed form control laws formulated as dynamical systems that provide stability and convergence guarantees in combination with instant reactivity to changing environments. Specifically, artificial potential fields, introduced in [2], based on scalar functions, guiding the robot with an attractive force to the goal and repulsive forces from the obstacles, have become popular [3], [4]. However, a drawback of the additive artificial potential field methods is the possible existence of local minima other than the goal point, i.e. the robot could get stuck at a position away from the goal. To address this issue, harmonic potential fields have emerged [5]–[8]. Other approaches providing (almost) global convergence are based on navigation functions [9]–[14] which are a special subclass of potential functions designed to be bounded.

A repeated assumption [7]–[14] enabling the proof of (almost) global convergence is the premise of disjoint obstacles. However, in cluttered dynamic environments, closely positioned obstacles may be seen as having intersecting regions, violating the assumption of disjoint regions. This occurs since the obstacles need to be inflated by the robot radius due to a point mass modelling, and since it is common to add extra safety margin to prevent the robot touching the obstacles during obstacle circumvention or to address uncertainties in obstacle positions. To preserve the convergence properties, intersecting obstacles must therefore be combined into a single obstacle. Additionally, the obstacle shapes are restricted to be topological disks. In most cases they are considered as ellipses or Euclidean disks, but a more general premise is to consider all strictly starshaped sets (see example in Fig. 1a) - sets where there exists a point, a kernel point, where all rays emanating from the kernel point cross the boundary once and only once. While methods like [8] can operate directly in star worlds (a workspace of strictly starshaped obstacles), navigation functions can be used in star worlds by transforming the obstacles to Euclidean disks [9] or to points [15].

To apply the aforementioned motion planning methods in practice while preserving convergence properties, a star world representation of the workspace with mutually disjoint obstacles is thus needed, i.e. it is necessary to find a strict starshaped set which fully encapsulates non-starshaped as well as intersecting obstacles. Since all convex sets are starshaped, the convex hull is intuitively a candidate for “starifying” a non-starshaped region. However, using the convex hull introduces a more conservative enclosing than needed and the benefits of considering general starshapes become redundant. Moreover, it can result in expanding an obstacle such that the robot position is in the obstacle interior, as in Fig. 1b, which could inhibit a sound motion planning process. In [9] forest of stars was

Fig. 1

(a) Example of a (concave) strictly starshaped set with kernel shown in blue.
(b) Robot position, shown as red circle, is included in the convex hull, shown in green, of the intersecting obstacles.
proposed as a way to transform a set of intersecting starshaped obstacles into a set of disjoint starshaped obstacles. However, it is restricted to a specific structure where all intersecting obstacles are ordered in a parent-child relation. In particular, a parent must contain a kernel point of every child and no children with same parent are allowed to intersect. While these may be reasonable assumptions when modelling obstacles as a union of several simpler subshapes offline, it cannot be presumed to hold in a dynamic environment with moving objects.

As an alternative, the mathematical concept of starshaped hull introduced in [16], defined analogously to the convex hull [4], is reasonable to use in order to guarantee starshaped obstacle representation. A straightforward use of the starshaped hull to generate obstacle regions does not consider excluding points from the expanded obstacle and, as stated above for the convex hull, may be problematic. Moreover, by nature of the starshaped hull, the resulting set is not in general strictly starshaped and a direct use of the starshaped hull does not in general result in a star world.

In this paper, we present a method to transform a workspace of possibly intersecting obstacles into a workspace of disjoint strictly starshaped obstacles. The method is based on two concepts, both introduced here: admissible kernel and starshaped hull with specified kernel. The admissible kernel enables excluding points of interest from the starshaped hull, i.e. obstacles can be modelled as starshaped regions which do not contain the robot nor goal position. The starshaped hull with specified kernel expands on the idea of starshaped hull and will prove to be useful in generating strictly starshaped sets. Additionally, some general properties of the starshaped hull are established and are instrumental in the design of the proposed algorithm.

First, in Sec. II the notation used and brief theory of starshaped sets is presented, and in Sec. III the problem formulation is stated. In Sec. IV some important properties of the starshaped hull are presented, followed by the definition of the admissible kernel and a definition of the starshaped hull with specified kernel. An algorithm transforming a workspace of obstacles to disjoint starshaped obstacles is presented in Sec. V and in Sec. VI examples are provided where the algorithm is used in combination with a motion planner. Finally, in Sec. VII conclusions are drawn.

II. PRELIMINARIES

A. Mathematical notation

The closed line segment from point \( a \) to point \( b \) is denoted as \( l(a, b) \) and \( \overrightarrow{ab} \) denotes the vector from \( a \) to \( b \). The ray emanating from point \( a \) in the direction of \( \overrightarrow{bc} \) is denoted as \( r(a, \overrightarrow{bc}) \). The interior, the boundary and the exterior of a set \( A \subset \mathbb{R}^n \) are denoted by \( \text{int} A \), \( \partial A \) and \( \text{ext} A \), respectively. The convex hull of \( A \) is denoted \( CH(A) \). Given a closed convex set \( A \) and an exterior point \( x \in \text{ext} A \), a point \( a \in \partial A \) is called a tangent point of \( A \) through \( x \) if the ray emanating from \( x \) in direction of \( \overrightarrow{ax} \) does not intersect the interior of \( A \). That is, \( r(x, \overrightarrow{ax}) \cap \text{int} A = \emptyset \). The set of all tangent points of \( A \) through \( x \) is denoted by \( T_A(x) \). For any interior or boundary point \( x \in A \), \( T_A(x) \) is defined as the empty set. The linear cone containing all rays between two rays, \( r_1 \) and \( r_2 \), emanating from the same point, in counterclockwise (CCW) orientation, is denoted by \( C_L(r_1, r_2) \).

B. Starshaped sets

A set \( A \subset \mathbb{R}^n \) is starshaped with respect to \( x \) if for every point \( y \in A \) the line segment \( l(x, y) \) is contained in \( A \). The set \( A \) is said to be starshaped if it is starshaped with respect to (w.r.t.) some point \( x \), i.e. \( \exists x \) s.t. \( l(x, y) \subset A \), \( \forall y \in A \). The set of all such points is called the kernel of \( A \) (show in blue for the example in Fig. 1a) and is denoted \( \text{ker}(A) \), i.e. \( \text{ker}(A) = \{ x \in A : l(x, y) \subset A, \forall y \in A \} \). The kernel of \( A \) is a convex set and the set \( A \) is convex if and only if \( \text{ker}(A) = A \).

The set \( A \) is strictly starshaped with respect to \( x \) if it is starshaped w.r.t. \( x \) and any ray emanating from \( x \) crosses the boundary only once, i.e. \( r(x, \overrightarrow{xy}) \cap \partial A = \{ y \} \), \( \forall y \in \partial A \). We say that \( A \) is strictly starshaped if it is strictly starshaped w.r.t. some point. For a thorough survey on the theory starshaped sets, see [17].

III. PROBLEM FORMULATION

In this work, we consider a robot operating in the Euclidean space or Euclidean plane containing a collection of possibly intersecting obstacle regions, \( \mathcal{O} \). Depending on dimensionality, the robot workspace, \( W \), and each obstacle region, \( \mathcal{O}_i \), are

a) \( W = \mathbb{R}^3 \) and \( \mathcal{O}_i \subset \mathbb{R}^3 \) is a convex sets,

b) \( W = \mathbb{R}^2 \) and \( \mathcal{O}_i \subset \mathbb{R}^2 \) is a convex sets or a polygon.

Note that by allowing intersecting regions, the formulation does not restrict the obstacles to be convex since a single concave obstacle can be modelled as a combination of convex regions, e.g. a human/robot modelled as a kinematic chain of ellipsoids. In addition to this, any kind of (starshaped or non-starshaped) polygon shape is included in the formulation in \( \mathbb{R}^2 \). Obviously, the scenario when multiple obstacles are closely located such that their regions intersect when introducing margins, e.g. to adjust for robot radius, as in Fig. 2 is also considered.

The free configuration space is

\[
\mathcal{F} = W \setminus \bigcup_{\mathcal{O}_i \in \mathcal{O}} \mathcal{O}_i
\]

and similar to [9] we define a star world as follows:

**Definition 1.** A free configuration space, \( \mathcal{F} \), all of whose obstacles are strictly starshaped sets is a star world.

To distinguish between the scenario with intersecting obstacles and with mutually disjoint obstacles, we will call a star world where all obstacles are mutually disjoint a disjoint star world and a star world where two or several obstacles intersect an intersecting star world.

\[
^2\text{The definition of a star world in [9] in fact correspond to what we call a disjoint star world and does not include scenarios with intersecting obstacles.}
\]

\[
^3\text{The starshaped hull of a set } A \text{ with respect to some point } x \text{ is the smallest set which fully contains } A \text{ and where } x \text{ is a kernel point. See Sec. IV for a more detailed explanation.}
\]
Given the current position of a robot, \( x \) or \( x \), that of obstacle regions, \( O \). Consider a robot workspace, \( W \). Problem 1. should be created. The problem is stated as follows: 

The objective is to create a disjoint star world in the free configuration space, \( F^* \subseteq F \), such that motion planning methods operating in star worlds can be applied with guaranteed convergence to a goal position, \( x_g \). For sound motion planning, the robot position, \( x \), should remain in the free set, and for convergence to the goal, the goal should remain in the free set, i.e. \( x \in F^* \) and \( x_g \in F^* \). In cases when no such \( F^* \) exists, e.g. when the robot and/or goal are fully surrounded by several intersecting obstacles, the condition of disjoint obstacles is relaxed and an intersecting star world containing \( x \) and \( x_g \) should be created. The problem is stated as follows:

**Problem 1.** Consider a robot workspace, \( W \), and a collection of obstacle regions, \( O \), that are either

a) \( W = \mathbb{R}^3 \) and \( O_i \subseteq \mathbb{R}^3 \) is a convex set, or

b) \( W = \mathbb{R}^2 \) and \( O_i \subseteq \mathbb{R}^2 \) is a convex set or a polygon. Given the current position of a robot, \( x \in F \), and a goal position \( x_g \in F \), construct a disjoint star world \( F^* \subseteq F \) such that \( x \in F^* \) and \( x_g \in F^* \). That is, construct a collection of obstacles, \( O^* \), such that

\[
\bigcup_{O_i \in O} O_i \subset \bigcup_{O_i^* \in O^*} O_i^*
\]

(2a)

\( O_i^* \) is strictly starshaped, \( \forall O_i^* \in O^* \)

(2b)

\( x \notin O_i^* \), \( \forall O_i^* \in O^* \)

(2c)

\( x_g \notin O_i^* \), \( \forall O_i^* \in O^* \)

(2d)

\( O_i^* \cap O_j^* = \emptyset \), \( \forall O_i^* \neq O_j^* \in O^* \).

(2e)

If no such \( F^* \) exists, construct a star world \( F^* \subseteq F \) such that \( x \in F^* \) and \( x_g \in F^* \). That is, construct a collection of obstacles, \( O^* \), satisfying (2a)-(2d).

**IV. STARSHAPED HULL**

A definition of the starshaped hull was provided in [16]. In this work, it is defined with a minor modification as follows:

**Definition 2.** Let \( A \subseteq \mathbb{R}^n \) and \( x \in \mathbb{R}^n \). The starshaped hull of \( A \) with respect to \( x \), denoted \( SH_x(A) \), is the smallest starshaped set with respect to \( x \) containing \( A \).

In comparison, the starshaped hull is now well-defined for any \( x \in \mathbb{R}^n \) and not solely for \( x \in A \). The measure is conventionally considered in terms of Lebesgue measure. Proposition 3.2 in [16] still holds for the adjusted definition and we have

\[
SH_x(A) = \bigcup_{y \in A} I(x, y).
\]

(3)

We will interchangeably refer to the starshaped hull as a set and as an operation, i.e. the smallest starshaped set w.r.t \( x \) containing \( A \) vs the generation of this set. Here we provide some important properties of the starshaped hull which will be used in the subsequent sections.

**Property 1.** Let \( A \subseteq \mathbb{R}^n \) and let \( B \) be a collection of sets \( \forall B \subseteq \mathbb{R}^n \).

a. \( SH_x(A) = A \iff A \) is starshaped and \( x \in ker(A) \)

b. \( SH_x(A) \subseteq CH(A) \), \( \forall x \in CH(A) \)

c. \( SH_x \left( \bigcup_{B \in B} B \right) = \bigcup_{B \in B} SH_x(B) \)

**Proof.** See Appendix A.

As a consequence of Property A, we have \( SH_x(A) \neq A \) for any \( x \notin ker(A) \). That is, the starshaped hull of any set is a strict superset unless the set is starshaped and the hull is generated w.r.t. a kernel point. Property A ensures that the starshaped hull w.r.t. any point \( x \in CH(A) \) provides a less (or at most equally) conservative enclosing of the set \( A \) compared to the convex hull. In particular, any point \( x \in A \) can be used to generate a starshaped set which is upper bounded by the convex hull of \( A \). Property A simplifies finding the starshaped hull for complex regions which can be described as combinations of simpler subsets, e.g. as the union of several polygons and/or convex sets, as the hull can be computed separately for each subset.
An algorithm to find the starshaped hull of a polygon has been presented in [18]. For a convex set \( A_{\text{conv}} \) the starshaped hull w.r.t. \( x \) is given as
\[
SH_x(A_{\text{conv}}) = A_{\text{conv}} \cup CH(T_{A_{\text{conv}}}(x) \cup x),
\]
where \( CH \) is the convex hull of all tangent points of \( A \) through \( x \). As a consequence, the starshaped hull of a convex set is also convex.

In the case of \( A \subset \mathbb{R}^2 \) we have that \( CH(T_{A}(x) \cup x) \) is a triangle with vertices in \( x \) and in two tangent points of \( A \) through \( x \), see Fig. 4a. When \( A \subset \mathbb{R}^3 \) we have that \( CH(T_{A}(x) \cup x) \) is the solid cone with apex \( x \) and base as the planar intersection of \( A \) containing three, and thus all, tangent points of \( A \) through \( x \), see Fig. 4b.

Property 1. Let \( A \subset \mathbb{R}^n \) be a collection of starshaped sets \( A \subset \mathbb{R}^n, \forall A \in A \) with union \( A_u = \bigcup_{A \in A} A \) and kernel intersection \( K_\cap = \bigcap_{A \in A} \ker(A) \). If \( K_\cap \neq \emptyset \), the union of all sets, \( A_u \), is starshaped and \( K_\cap \subset \ker(A_u) \).

Proof. See Appendix B.

Property 2. Let \( A \subset \mathbb{R}^n \) be a starshaped set. \( A \) is strictly starshaped if the kernel of \( A \) has a nonempty interior. That is
\[
\text{int} \ker(A) \neq \emptyset \Rightarrow A \text{ is strictly starshaped}
\]

Proof. See Appendix C.

The implication of Property 1 is illustrated in Fig. 5a where three starshaped sets are intersecting. Note that the condition \( K_\cap \neq \emptyset \) in Property 1 is sufficient but not necessary for \( A_u \) to be starshaped. Consider for example the starshaped polygon \( A_1 \) and ellipse \( A_2 \) in Fig. 5b. The kernels of the starshaped sets are disjoint, i.e. \( K_\cap = \emptyset \), but the union \( A_u \) is starshaped with \( \ker(A_u) = \ker(A_1) \).

\[ A. \text{ Excluding points from the starshaped hull} \]

In some scenarios it is desired to find a starshaped set containing a set \( A \), while ensuring that some points of interest, \( \bar{X} \), are not included in the resulting starshaped set. For instance, in Problem 1 we need all obstacles to be starshaped but the robot and goal positions should remain outside the extended starshaped obstacles. The starshaped hull of \( A \) provides a starshaped enclosing of \( A \), but it does not inherently provide any way to exclude specific points. The shape of the set depends on the point selected for generating the hull and as a consequence, we have that the starshaped hull w.r.t. some points is disjoint from \( \bar{X} \), while it is not w.r.t. other points. To enable the exclusion of \( \bar{X} \) from the starshaped hull, we introduce the admissible kernel defined as follows:

Definition 3. Let \( A \subset \mathbb{R}^n \) and \( \bar{X} \subset \mathbb{R}^n \). The admissible kernel for \( A \) excluding \( \bar{X} \), denoted as \( \text{ad ker}(A, \bar{X}) \), is the set such that the starshaped hull of \( A \) at any \( x \in \text{ad ker}(A, \bar{X}) \) does not contain any point in \( \bar{X} \). That is,
\[
\text{ad ker}(A, \bar{X}) = \{ x \in \mathbb{R}^n : SH_x(A) \cap \bar{X} = \emptyset \}.
\]

Given the admissible kernel for the sets \( A \) and \( \bar{X} \), any point \( x \in \text{ad ker}(A, \bar{X}) \) can be used for the starshaped hull to generate a starshaped set which contains \( A \) and excludes all points in \( \bar{X} \). For computing the admissible kernel, the following two properties are useful:

Property 2. Let \( A \) be a collection of sets \( A \subset \mathbb{R}^n, \forall A \in A \) with union \( A_u = \bigcup_{A \in A} A \). The admissible kernel for \( A_u \) excluding a point set \( \bar{X} \) is the intersection of the admissible kernel excluding \( \bar{X} \) for all subsets. That is,
\[
\text{ad ker}(A, \bar{X}) = \bigcap_{A \in A} \text{ad ker}(A, \bar{X})
\]

Proof. See Appendix D.

Property 3. The admissible kernel for the starshaped hull of a set \( A \) excluding a point set \( \bar{X} \) is the intersection of all admissible kernels given each individual point \( \bar{x} \in \bar{X} \). That is,
\[
\text{ad ker}(A, \bar{X}) = \bigcap_{\bar{x} \in \bar{X}} \bigcap_{A \in A} \text{ad ker}(A, \{ \bar{x} \})
\]

Proof. See Appendix E.

As a consequence of Property 2 the admissible kernel for a set given as the union of several “simpler” subsets, i.e. for intersecting obstacles as in Problem 1 can be found by intersecting the admissible kernel for each subset. In Fig. 6 an example is shown of how it can be used to find the admissible kernel for intersecting obstacles as in Problem 1, can be found by intersecting the admissible kernels given each individual point \( \bar{x} \in \bar{X} \).
Eq. [7] is instrumental for simplifying the problem of finding the admissible kernel for a complex set excluding several points by decomposing it into subproblems of finding the admissible kernel for a simple set excluding a single point.

A point $\bar{x} \in \bar{X}$ can be classified into three distinct types w.r.t. $A$. It may be a point in the set, a bounded exterior point or a free exterior point. The difference between the two latter, illustrated as $\tilde{x}$ and $\bar{x}$ in Fig. 7B, is that there exists a ray in some direction emanating from a free exterior point which does not intersect $A$, while a bounded exterior point is fully surrounded by $A$. Obviously, any exterior point to a convex set is a free exterior point.

**Property 4.** The admissible kernel for the starshaped hull of $A$ excluding the singleton set $\{\bar{x}\}$ is nonempty if and only if $\bar{x}$ is a free exterior point of $A$.

**Proof.** See appendix E. □

The admissible kernel for any 2-dimensional set, $A \subset \mathbb{R}^2$, given any free exterior point, $\bar{x}$, is found as the cone

$$\text{ad ker}(A, \{\bar{x}\}) = \text{int}C_\subset \left(r(\bar{x}, t_1(\bar{x})), r(\bar{x}, t_2(\bar{x})), r(\bar{x}, t_3(\bar{x})), \ldots\right)$$

(8)

where $t_1(\bar{x}), t_2(\bar{x}) \in T_A(\bar{x})$. For a convex set specified by the boundary function $b(x) = 0$ iff $x \in \partial A$, the tangent points can be found by solving for $t$ such that $\frac{db}{dx}(t) \cdot (t - \bar{x}) = 0$ and $b(t) = 0$ and the tangent points are set such that the order $xt_1t_2$ is CW. For a polygon, the tangent points can be found as proposed in [19] by expressing the vertices in polar coordinates w.r.t. $\bar{x}$, and $t_1$ and $t_2$ are chosen as the vertices with maximum and minimum polar angles, respectively. Note that the admissible kernel is given by the interior, and does not include the boundary, of the generated cone in (8). Consider for example Figs. 7A and 7B where the admissible kernel for two sets $A \subset \mathbb{R}^2$ are shown. Since the starshaped hull w.r.t. any boundary point, $x \in \partial \text{ker}(A, \{\bar{x}\})$, would contain the line segments $l(x, t_1)$ and $l(x, t_2)$, it would also contain $\bar{x}$ as this is part of both lines. The admissible kernel for any convex set, $A_{\text{conv}} \subset \mathbb{R}^n$, $n \geq 2$, is found as

$$\text{ad ker}_{A_{\text{conv}}}(\bar{x}) = \mathbb{R}^n \setminus \left\{ r(\bar{x}, y\bar{x}) : y \in CH(T_{A_{\text{conv}}}(x)) \right\}$$

(9)

assuming $\bar{x}$ is an exterior point of $A_{\text{conv}}$. Since only rays emanating from $\bar{x}$ in directions from points in $A_{\text{conv}}$ are excluded in (9), the admissible kernel for a convex set fully contains the set. In Fig. 7C an example of the admissible kernel excluding a single point for an ellipsoid is illustrated.

**B. Starshaped hull for strictly starshaped sets**

Depending on the set $A$ and the point $x$ used for generating $SH_x(A)$, the kernel of the starshaped hull is the singleton $\text{ker}(SH_x(A)) = \{x\}$. Furthermore, $x$ is typically contained in a hyperplane with tangency along some planar region of $\partial SH_x(A)$. This is a natural effect from the manner that the
starshaped hull is constructed, i.e. by expanding the set with line segments from the single point \( x \). As a consequence, there may exist more than one boundary point along some direction from all (or the only) kernel points. In other words, it is not strictly starshaped. In Fig. 8a an example of this is illustrated where the starshaped hull is generated w.r.t. a point \( x \) and several boundary points of the hull are located along the four directions shown as red dashed lines from the singleton kernel \( x \). For this reason, the starshaped hull in its original definition is not appropriate to apply when strictly starshaped hulls are needed. To treat this issue, we introduce the starshaped hull with specified kernel.

**Definition 4.** Let \( A \subset \mathbb{R}^n \) and \( K \subset \mathbb{R}^n \). The starshaped hull of \( A \) with specified kernel \( K \), denoted as \( SH_{ker K}(A) \), is defined as the smallest starshaped set such that \( A \subset SH_{ker K}(A) \) and \( K \subset ker(SH_{ker K}(A)) \).

Property 1 for the starshaped hull w.r.t. a single point cannot be directly applied for the starshaped hull with specified kernel. Instead, we have the following properties.

**Property 5.** Let \( A \subset \mathbb{R}^n \) and let \( \mathcal{B} \) be a collection of sets \( B \subset \mathbb{R}^n \), \( \forall B \in \mathcal{B} \).

a. \( SH_{ker K}(A) = SH_{ker CH(K)}(A) = \bigcup_{k \in CH(K)} SH_k(A) \)

b. \( SH_{ker K}(A) = A \iff A \) is starshaped and \( K \subset ker(A) \)

c. \( SH_{ker K}(A) \subset CH(A), \forall K \subset CH(A) \)

d. \( SH_{ker K} \left( \bigcup_{B \in \mathcal{B}} B \right) = \bigcup_{B \in \mathcal{B}} SH_{ker K}(B) \)

**Proof.** See Appendix G.

While it is sufficient to have \( K \subset ker(SH_{ker K}(A)) \) by Definition 4, Property 5 states that \( CH(K) \) is also contained in \( ker(SH_{ker K}(A)) \) in all cases, since \( SH_{ker K}(A) = SH_{ker CH(K)}(A) \). This will prove to be instrumental for generating sets which are guaranteed to be strictly starshaped. Additionally, Property 5 provides a direct relation between the starshaped hull with specified kernel and the starshaped hull w.r.t. a point. Properties 6-8 directly relate to Properties 1-4. Property 5 provides a guarantee that a starshaped set, \( A \), is not expanded by the operation \( SH_{ker K}(A) \) if the specified kernel points are selected within the kernel of \( A \) and Property 6 provide an upper bound if the specified kernel points are selected within the convex hull of \( A \). Property 6 simplifies finding the starshaped hull with specified kernel for complex regions which can be described as combinations of simpler subsets, e.g. as the union of several polygons and/or convex sets, since the hull can be computed separately for each subset.

Using Proposition 2 in combination with Property 5, a sufficient condition on the specified kernel can be derived for \( SH_{ker K}(A) \) to be strictly starshaped as stated in the following property.

**Property 6.** Let \( A \subset \mathbb{R}^n \) and let \( K \subset \mathbb{R}^n \). The starshaped hull of \( A \) with specified kernel \( K \), \( SH_{ker K}(A) \), is strictly starshaped if \( K \) contains \( n + 1 \) affinely independent points.

**Proof.** See Appendix H.

From Property 6 we have that any set \( A \subset \mathbb{R}^n \), can be enclosed by a strictly starshaped set with \( SH_{ker K}(A) \) given that \( K \) is chosen as \( n + 1 \) affinely independent points. Specifically, in \( \mathbb{R}^2 \), it is sufficient to select \( K \) as three points which are not collinear.

As stated in the previous section, the admissible kernel provides a useful instrument when choosing the kernel point \( x \) for generating the starshaped hull, \( SH_x(A) \), such that some specified points, \( \bar{x} \), are excluded. However, the admissible kernel does not provide such a guarantee when the starshaped hull with specified kernel, \( SH_{ker K}(A) \), is used. This is evident from Fig. 9a where \( \bar{x} \) is contained by \( SH_{ker K}(A) \) even though the kernel points are selected within the admissible kernel,
\( K \subset \text{ad ker}(A, \{\bar{x}\}) \). To extend the applicability of the admissible kernel to the starshaped hull with specified kernel, consider the following property.

**Property 7.** Let \( A \subset \mathbb{R}^n \), \( \bar{X} \subset \mathbb{R}^n \) and \( \text{ad ker}(A, \bar{X}) \) be the admissible kernel for any \( A \) excluding \( \bar{X} \). If \( CH(K) \) is contained by \( \text{ad ker}(A, \bar{X}) \), no point \( \bar{x} \in \bar{X} \) is included in the starshaped hull of \( A \) with specified kernel \( K \). That is,

\[
CH(K) \subset \text{ad ker}(A, \bar{X}) \Rightarrow \text{SH}_{\text{ker}K}(A) \cap \bar{X} = \emptyset. \tag{10}
\]

**Proof.** See Appendix I.

From Properties 6 and 7, we can now conclude that the admissible kernel for \( A \subset \mathbb{R}^n \) excluding \( \bar{X} \subset \mathbb{R}^n \), \( \text{SH}_{\text{ker}K}(A) \) is guaranteed to be a strictly starshaped set which does not contain any \( \bar{x} \in \bar{X} \) if \( K \) is chosen as \( n+1 \) affinely independent points such that \( CH(K) \subset \text{ad ker}(A, \bar{X}) \).

Before deriving the expressions for the starshaped hull with specified kernel, note that the naive approach to separately generate the starshaped hull w.r.t. each \( k \in K \) and combine them does not provide the desired result, i.e. \( \text{SH}_{\text{ker}K}(A) \neq \bigcup_{k \in K} \text{SH}_{k}(A) \) in general. This is evident from Fig. 9b, where, given an ellipse, a set of \( K = \{k_1, k_2\} \), the union \( B = \text{SH}_{k_1}(A) \cup \text{SH}_{k_2}(A) \) is shown. Since \( l(k_1, k_2) \notin \) \( B \) neither \( k_1 \) nor \( k_2 \) belongs to the kernel of \( B \). Instead, the starshaped hull should be generated w.r.t. each \( k \in CH(K) \) according to Property 5a. However, as \( CH(K) \) is a continuous set for any non-singleton set \( K \), it becomes intractable in practice to generate the starshaped hull w.r.t. each \( k \in CH(K) \) and an explicit method to find \( \text{SH}_{\text{ker}K}(A) \) is needed. For a convex set, \( A_{\text{conv}} \), the extended region of the starshaped hull with specified kernel \( K \) is based on the convex hull of the kernel points and their corresponding points of tangency. That is,

\[
\text{SH}_{\text{ker}K}(A_{\text{conv}}) = A_{\text{conv}} \cup CH \left( \bigcup_{k \in K} T_k(A_{\text{conv}}) \cup k \right). \tag{11}
\]

Naturally, this is equivalent to (4) for a single specified kernel point. In Fig. 9b, the case with an ellipse and two specified kernel points is again considered and \( \text{SH}_{\text{ker}K}(A) \) is depicted. As the resulting shape is convex, it follows that \( K \subset \text{ker}(\text{SH}_{\text{ker}K}(A)) \).

Inspired by the approach in [18], an algorithm to find the starshaped hull with a specified kernel for a polygon has been developed and is given in Algorithm 1. In Fig. 8b, the starshaped hull with three affinely independent points, i.e. a triangle, is depicted for a polygon, \( P \). In contrast to the starshaped hull w.r.t. \( x \), shown in Fig. 8a, \( \text{SH}_{\text{ker}K}(P) \) is strictly starshaped in accordance with Property 6 and therefore only one boundary point exists in each direction from any interior point of \( CH(K) \).

**Remark.** For a single specified kernel point, i.e. \( K \) is a singleton set, Algorithm 1 simplifies to the algorithm for generating the starshaped hull of a polygon w.r.t. a point presented in [18] with the instrumental distinction that all vertices, and not only convex vertices, are considered in the iteration.

**Algorithm 1:** Finding starshaped hull with desired kernel for a polygon

**Input:** A polygon \( P \) and a finite point set \( K \)

**Output:** The minimum starshaped polygon \( \text{SH}_{\text{ker}K}(P) \) s.t. \( P \subset \text{SH}_{\text{ker}K}(P) \) and \( K \subset \text{ker}(\text{SH}_{\text{ker}K}(P)) \)

Initialize \( P^* \) as empty list;

\( k^*, \bar{v}, e_1, e_2 \leftarrow \) centroid of \( K \);

Order \( P = v_1, v_2, \ldots, v_N \) s.t. \( v_1 \) is the vertex with largest \( x \)-value and the order \( v_Nv_1v_2 \) is CCW;

**foreach vertex** \( v \in P \) **do**

- **if** \( r(v, \bar{v}) \) **does not intersect interior of** \( P \), \( \forall k \in K \) **then**
  - Append \( v \) to \( P^* \);
  - \( v' \leftarrow \) vertex preceding \( v \) in \( P \);
  - **if** \( \exists k \in K \) s.t. \( l(k, v) \) **intersects** \( l(e_1, e_2) \) **then**
    - \( e_1 \leftarrow \) closest intersection to \( e_2 \) of \( l(k, v) \) and \( l(e_1, e_2) \), \( \forall k \in K \);
  - **else**
    - **foreach** \( k \in K \) **do**
      - **if** \( l(k, v) \) **intersects** **interior of** \( P \), \( \forall k' \in K, k' \neq k \) **then**
        - **if** \( \exists k' \in K, k' \neq k \) s.t. \( l(k', \bar{v}) \) **intersects** \( l(u, v) \) **then**
          - \( u \leftarrow \) intersection of \( l(k', \bar{v}) \) and \( l(u, v) \);
          - Append \( u \) to \( P^* \);
          - \( e_1 \leftarrow u; \)
          - \( e_2 \leftarrow v; \)
          - **if** \( uvv' \) **is CCW** **then**
            - Swap last two elements of \( P^* \);
          - **else**
            - **if** \( r(k, \bar{v}) \) **does not intersect** **interior of** \( P \), \( \forall k' \in K, k' \neq k \) **then**
              - **if** \( \exists k \in K \) s.t. \( kvv' \) **is CCW** **then**
                - Swap last two elements of \( P^* \);
        - **else**
          - **foreach** consecutive vertices \( v, \bar{v}' \in P^* \) **do**
            - **if** \( \exists k \in K \) s.t. \( kvv' \) **is CCW** **then**
              - Insert \( k \) in \( P^* \) between \( v \) and \( v' \);
    - **end**
  - **end**
- **end**
- \( \bar{v} \leftarrow \) last element of \( P^* \);
- **return** \( P^* \);
The optimization problem in (12). That is, the global starshaped admissible kernel can be used to restrict the search space for dimensional sets.

The global starshaped hull coincides with the minimum-area where \( \lambda \) defined in Definition 2, and equivalently to Definition 5 the global starshaped hull can be given as

\[
\bigcup_{k \in \mathcal{K}} \text{SH}_k(A).
\]

Let these aspects more closely related to the convex hull. Obviously, there is a close relation to the starshaped hull as defined in Definition 2, and equivalently to Definition 5 the starshaped hull of \( A \) excluding \( X \), denoted \( \text{SH}_{X}(A) \), can be defined as the smallest starshaped set which contains \( A \) such that \( \text{SH}_{X}(A) \cap \bar{X} = \emptyset \) and we have

\[
\text{SH}_{X}(A) = \min_{x \in \text{ad ker}(A, \bar{X})} \lambda(\text{SH}_x(A)).
\] (13)

With a slight modification of the method to find the global starshaped hull for a polygon, \( P \), presented in [18], it may be applied to find \( \text{SH}_{X}(P) \). In particular, only cells from the cell decomposition that lie inside the admissible kernel should be considered in the minimization step.

\section{V. Forming Disjoint Star Worlds}

Since any convex set with nonempty interior is strictly starshaped, we have from the formulation of Problem 4 that \( F = \mathcal{V} \backslash \bigcup_{O_i \in \mathcal{O}} O_i \) is a star world satisfying (2a) if each polygon obstacle is strictly starshaped. A simple solution in case no disjoint star world exists satisfying all conditions is hence to construct \( \mathcal{O}^* \) as

\[
\mathcal{O}^* = \{ \text{CD}(O_i) : O_i \in \mathcal{O} \}
\] (14)

where \( \text{CD}(\cdot) \) is a convex decomposition of the considered set. For any convex obstacle we trivially have \( \text{CD}(O_i) = O_i \), while for any concave polygon a convex decomposition can for instance be found using Hertel Mehlhorn algorithm [20].

In an attempt to find a disjoint star world solving Problem 1 we present Algorithm 2 which is discussed in the following section.

\subsection{A. Algorithm}

The fundamental idea of Algorithm 2 is to create clusters of obstacles by combining intersecting obstacles followed by a generation of starshaped obstacles which fully contain each cluster in an iterative manner. In Algorithm 2 the notation \( C_l \) is used for the set of all clusters, \( c_l \subset \mathcal{O} \), containing one or several obstacles. Each iteration in the algorithm can be divided into three main steps: computation of admissible kernels for each cluster, generation of starshaped obstacles containing each cluster, and clustering of intersecting starshaped obstacles. The number of clusters does in this way never increase and the algorithm converges whenever the number of clusters remains the same after an iteration. For the initial iteration, each single obstacle is considered as a separate cluster.

The steps are illustrated in Fig. 10. First, the admissible kernel is found for each cluster given the excluding points defined as the robot and goal position. This is shown for one ellipse and the polygon in Fig. 10a and 10b. Next, in Fig. 10c new strictly starshaped obstacles are generated using the starshaped hull with three (\( n = 2 \)) specified kernel points. Since all starshaped obstacles intersect in Fig. 10c the obstacles are combined into one cluster in Fig. 10d. As the number of clusters has been reduced, the process is iterated once again as illustrated in Figs. 10e and 10f. Obviously, no change is made in the clustering stage when only one cluster is considered and the algorithm terminates.
Algorithm 2: Forming (disjoint) star worlds

Input: A set of obstacles, \( \mathcal{O} \), as in Problem 1, the robot position, \( x \), and goal position, \( x_g \).
Output: A set of (disjoint) strictly starshaped obstacles, \( \mathcal{O}^* \).

1. \( \bar{X} \leftarrow \{x, x_g\} \);  
2. \( Cl \leftarrow \mathcal{O} \);
3. do
4. \( N_{Cl} \leftarrow \) #clusters in \( Cl \);
5. \( \mathcal{O}^* \leftarrow \emptyset \);
6. foreach \( cl \in Cl \) do
7. if \( \text{ad ker}(cl, \bar{X}) = \emptyset \) then
8. \( \text{return} \ \mathcal{O}^* \) as in (14);
9. \( K \leftarrow n + 1 \) affinely independent points s.t. \( CH(K) \subset \text{ad ker}(cl, \bar{X}) \);
10. Add \( SH_{\text{ker}}(cl) \) to \( \mathcal{O}^* \);
11. \( Cl^* \leftarrow \) clusters of \( \mathcal{O}^* \) s.t. no region in one cluster intersects with a region in another;
12. \( Cl \leftarrow \emptyset \);
13. foreach \( cl^* \in Cl^* \) do
14. Add \( cl = \{O \in \mathcal{O} : O \subset Cl^*\} \) to \( Cl \);
15. while \( N_{Cl} \neq \) #clusters in \( Cl \);
16. return \( \mathcal{O}^* \);

1) Admissible kernel: According to \( \mathcal{O}^* \) the admissible kernel for a cluster can be found as the intersection of the admissible kernels for the corresponding cluster obstacles, derived using (8) and (9) for the \( \mathbb{R}^2 \) and \( \mathbb{R}^3 \) case, respectively. Consequently, for computational efficiency, the admissible kernel for each obstacle can be computed once outside the loop and the admissible kernel for all clusters in each iteration can be found by pure intersections. As discussed in Section \( \text{V-A} \), the admissible kernel may be the empty set if \( x \) and/or \( x_g \) are bounded exterior points for the evaluating set, i.e. there exists no starshaped set which contains the cluster obstacle(s) and at the same time does not contain \( x \) and \( x_g \). In such case, the algorithm is terminated by providing a solution based on convex decomposition (line 8) resulting in an intersecting star world.

2) Starshaped hull: To generate a starshaped set containing each cluster, the starshaped hull with specified kernel is applied (line 10). In particular, \( n + 1 \) affinely independent points with corresponding convex hull contained by the admissible kernel are selected as the specified kernel points (line 9), forming a triangle or polyhedron, depending on the dimensionality. This ensures that the resulting set is strictly starshaped and does not contain \( x \) nor \( x_g \) according to Properties 6 and 7. The starshaped hull is found using (11) for convex sets and Algorithm 1 for polygons. For clusters with more than one obstacle, Property 5.1 can be leveraged to compute the starshaped hull of the union of all cluster obstacles.

3) Clustering: The clustering is applied on the original obstacle set, \( \mathcal{O} \), (line 14) but is determined by the intersection of the current starshaped obstacles, \( \mathcal{O}^* \), (line 11). For instance, even though the original polygon is disjoint from the ellipses, all four original obstacles are combined into one cluster in Fig. 10d since the starshaped obstacles in \( \mathcal{O}^* \) containing them are intersecting in Fig. 10c.

Unless Algorithm 2 terminates prematurely in line 8 \( \mathcal{O}^* \) consists of mutually disjoint strictly starshaped obstacles which all contain the original obstacles but not the robot nor goal position, i.e. \( \mathcal{F}^* = \mathcal{W} \setminus \bigcup_{O_i \in \mathcal{O}^*} O_i \), is a disjoint star world such that \( x \in \mathcal{F}^* \) and \( x_g \in \mathcal{F}^* \), and provides a solution to Problem 1. A premature termination occurs either if a single polygon obstacle surrounds the robot and/or goal, as for \( \bar{z} \) in Fig. 7c or if the combination of all obstacles in a cluster surrounds the robot and/or goal. When a single polygon or when intersecting obstacles in combination surround \( x \) and/or \( x_g \), there exists no solution with a disjoint star world to Problem 1. However, disjoint obstacles may as well be combined into one cluster leading to a termination with an intersecting star world if the combination of the clustered obstacles surround \( x \) and/or \( x_g \). This is exemplified in Fig. 11 with two polygons which in combination surround \( x \) and where the generated hulls of the two polygons, when following the kernel point selection in Algorithm 3 as discussed below, intersect such that they are combined into one cluster. As the clustered obstacles now surround \( x \), the algorithm terminates with an intersecting star world.

---

**Fig. 10:** Steps in Algorithm 2 when four obstacles are combined into one starshaped obstacle in two iterations. Each cluster is identified with a separate color.

(a) Admissible kernel for the leftmost ellipse.
(b) Admissible kernel for the polygon.
(c) Starshaped obstacles after one iteration. The convex hull of the specified kernel is shown (f) Starshaped hull of the new cluster. The convex hull of the specified kernel is shown in blue.
(d) Clustering of original obstacles contained in intersecting starshaped obstacles into one single cluster.
(e) Admissible kernel for the new cluster.

world as depicted in Fig. 11b jeopardizing convergence to the goal as will be exemplified in Section VI. As seen in Fig. 11c there exists a disjoint star world solving Problem 1 in this scenario.

With the analysis above in mind, a sufficient, but not necessary, condition to obtain a disjoint star world \( F^* \) is that the set of all obstacles, \( \bigcup_{i \in O} O_i \), does not surround \( x \) nor \( x_g \), i.e. there exist rays emanating from \( x \) and \( x_g \) which do not intersect with any obstacle.

**B. Excluding obstacle points**

In Fig. 12a all obstacles are combined into one although the polygon is disjoint from the ellipses. It is desired to maintain disjoint obstacles when possible, additional excluding points can be introduced in the computation of the admissible kernel for a cluster. Specifically, the additional excluding points should be representative for all obstacles which are not in the cluster. The points can for instance be selected as the vertices for a polygon and the two extreme points in each axis for an ellipse. When adopting this approach for the example considered in Fig. 12a the algorithm terminates with a disjoint star world which is upper bounded by the convex hull according to property 3c. For computational efficiency the convex hull can be omitted and a selection from ad ker \((\bar{X}_i) \cap CH(\bar{X})\) resulting in a starshaped hull which is unbounded.

**C. Kernel point selection**

Algorithm 2 does not offer a constructive way to select the kernel points, but rather a condition which they must fulfill (line 9). However, the choice of specified kernel points may have a big impact on the resulting hull shape as already seen and they should not be selected randomly in the admissible kernel. In particular, in cases where the admissible kernel is unbounded (as for both clusters in Fig. 12a and 12b) random selection is directly unsuitable. A reasonable selection can be found in the set ad ker \((\bar{X}_i) \cap CH(\bar{X})\) resulting in a starshaped obstacle.

The authors’ recommendation as a general strategy is to select the kernel points as a small equilateral triangle (regular tetrahedron in \( \mathbb{R}^3 \)) with centroid coinciding with the centroid of any of the specified sets discussed above. The side length of the triangle should be selected such that \( CH(K) \) is contained in this set. A procedure to obtain this is shown in Algorithm 3.

When Algorithm 2 is used in an online fashion to treat a dynamic workspace with moving obstacles it may be favorable to keep track of clusters and corresponding kernel points from previous time step in order to maintain the obstacle shapes as similar as possible in sequential time steps. That is, for a cluster consisting of the same obstacles as a cluster in the
A. Obstacle representation

The obstacle representation is generated with Algorithm 2 and used when deciding on the goal position, \( x_g \), for motion planning based on the following setting. Given a present some results when it is used as obstacle modelling total area coverage of the obstacles. (compared to one single obstacle in Fig. 10) with a smaller bottom, as in Fig. 12b, would in this specific case lead to in number of obstacles in \( O \). For instance, in Fig. 10 the kernel points are selected by in the line segment between the robot and goal position, \( l(x, x_g) \). In other words, convergence is guaranteed if the center point of each obstacle lies in its convergence center point set defined as \( X_i = \ker(O_i^\star) \setminus l(x, x_g) \). Since \( \ker(K_i) \) is an open nonempty set of dimension \( n \) according to Property 5, the set \( X_i = \ker(K_i) \setminus l(x, x_g) \subset X_i^c \) is also nonempty and any \( x_{c,i} \in X_i^c \) is a valid center point selection which guarantees convergence to the goal. In the following examples \( x_{c,i} \) is chosen as the centroid of \( K_i \), given that it does not lie on the line segment \( l(x, x_g) \). Otherwise a random point in \( X_i^c \) is chosen.

In addition to a center point, the description of each obstacle must be given in terms of an obstacle function, \( \beta \), in the form

\[
\beta(x) < 0, \quad \forall x \in \ker(O_i^\star),
\]
\[
\beta(x) = 0, \quad \forall x \in \partial O_i^\star,
\]
\[
\beta(x) > 0, \quad \forall x \in \text{ext } O_i^\star.
\]

Each starshaped obstacle in Algorithm 2 generated by a cluster of \( M \) original obstacles can be seen as a combination of \( M^\star \leq 2M \) primitives which are either convex sets or starshaped polygons, i.e. \( O_i^\star = \bigcup_{j=1}^{M^\star} o_j \) where \( o_j \) is a convex set or starshaped polygon with obstacle function defined as \( \beta_j \).

This is realized from the generation of starshaped hull with specified kernel (Algorithm 1 and 11) where any original polygon obstacle is represented by a single starshaped polygon and any original convex obstacle, \( O_{\text{conv}} \), is represented by the convex region \( O_{\text{conv}} \) and the convex polygon (polyhedron in \( \mathbb{R}^3 \)) \( CH(\bigcup_{k \in K} T_k(O_{\text{conv}}) \cup K) \). Thus, an obstacle function can easily be constructed also for the complex starshaped obstacle as \( \beta_i(x) = \min_{j=1,\ldots, M^\star} \beta_j(x) \).

B. Examples

1) Intersecting obstacles: Consider the scenario in Fig. 14 with three intersecting ellipses. The vector field for the motion planner is depicted in Fig. 14a for the case when Algorithm 2 is not used, i.e. with \( O^\star = O \). Clearly, the obstacles are not disjoint, which is a condition for guaranteed convergence to the goal. As a result, there exists two attractors, apart from the goal position, at points of obstacle intersection. When Algorithm 2 is applied on the original obstacles as in Fig. 13 the resulting set is disjoint (a single obstacle for the case in Fig. 14b) and the robot converges to the goal.
2) Moving obstacles: In Fig. 15 a scenario with three humans walking around in an area containing two walls, modelled as three moving circle obstacles and two static polygon obstacles, is illustrated. Since the motion planner treats the robot as a point mass and since an additional safety margin is desired, all obstacles are inflated by the robot radius and the safety margin. As a result, the obstacles may intersect when a human is close to a wall or another human. Initially, in Fig. 15a all obstacles are disjoint and are treated as five starshaped obstacles. Note that one of the polygons is non-starshaped and the original obstacles cannot be directly used by the motion planner without some preprocessing. This is solved when Algorithm 2 is utilized which applies the starshaped hull of the polygon. As two humans are walking towards and passing by the corner of a wall, depicted in Fig. 15b three obstacles are intersecting and are combined into one single starshaped obstacle, ensuring maintained convergence to the goal for the motion planner. When the leftmost circle intersects with the convex polygon, in Fig. 15c the specified kernel points are chosen in the intersecting region so that no intersection occurs. As two humans are walking around in an area containing two walls, in Fig. 15d the specified kernel points as well as the center point are kept static such that a similar shape of the starshaped obstacle is obtained to enable a smooth trajectory for the robot, as discussed in Sec. V-C.

VII. CONCLUSION

In this work we have considered the problem of transforming a set of intersecting obstacles to a set of disjoint strictly starshaped obstacles such that motion planning methods operating in star worlds can be applied while maintaining convergence properties. To this end, we have elaborated on the concept of starshaped hull and its properties. The admissible kernel for a set has been introduced to enable excluding points of interest from the starshaped hull and the starshaped hull with specified kernel has been introduced to ensure that the resulting set is strictly starshaped. Using the concepts of admissible kernel and starshaped hull with specified kernel, an algorithm has been designed to transform a workspace of intersecting obstacles to a workspace of disjoint starshaped obstacles. A sufficient condition for obtaining a valid solution to the problem using the algorithm is provided, although this is achieved in most scenarios even when the condition is not fulfilled. The utilization of the proposed method has been illustrated with examples of a robot operating in a 2D workspace using a motion planner in combination with the developed algorithm.

In this work we laid the foundations for generating star worlds, a process that can support the design of real-time control algorithms for a variety of robots operating in dynamical environments. The robot workspace is assumed to be the full Euclidean space. In many scenarios this is not the case, e.g. for a robot operating in a closed room, and an extension of the method to also consider workspace boundaries would be beneficial. Moreover, the algorithm is not complete in the sense that it may in some scenarios provide an intersecting star world even though a disjoint star world exists. Further investigation and modification of the method is needed to address complex scenarios where the robot and goal are surrounded by obstacles.

APPENDIX

A. Proof of Property [7]

a. (⇒): \(SH_x(A)\) is a starshaped set w.r.t. \(x\) by definition, i.e. \(SH_x(A)\) is starshaped with \(x \in \ker(SH_x(A))\). Since \(A = SH_x(A)\) also \(A\) is starshaped with \(x \in \ker(A)\).

(⇐): Since \(x \in \ker(A)\), \(A\) is starshaped w.r.t. \(x\) from the definition of the starshaped kernel. Obviously, \(A\) is the
minimum starshaped set which contains A and we have \( SH_x(A) = A \).

b. \( \text{SH}_x(A) = \bigcup_{y \in A} l(x,y) \subset \bigcup_{y \in CH(A)} l(x,y) \subset \bigcup_{x \in CH(A)} \bigcup_{y \in CH(A)} l(x,y) = CH(A) \)

c. \( \text{SH}_x \left( \bigcup_{B \in \mathcal{B}} B \right) = \bigcup_{y \in \bigcup_{B \in \mathcal{B}} B} l(x,y) = \bigcup_{B \in \mathcal{B}} \bigcup_{y \in B} l(x,y) = \bigcup_{B \in \mathcal{B}} \text{SH}_x(B) \)

B. Proof of Proposition 7

Let \( x \in K_n \). Since \( x \in \text{ker}(A) \), \( \forall A \in A \) we have that \( A = \text{SH}_x(A) \), \( \forall A \in A \) according to Property 1. Thus we have \( A \cup \text{SH}_x(A) = \text{SH}_x(\bigcup_{A \in A} A) = \text{SH}_x(A) \) from Property 1 which is starshaped by definition. From Property 2 we have \( x \in \text{ker}(A) \). This holds for any selection \( x \in K_n \) and we can conclude \( K_n \subset \text{ker}(A) \).

C. Proof of Proposition 2

Let \( x \in \text{int ker}(A) \) and assume \( A \) is not strictly starshaped. Since \( A \) is not strictly starshaped there exists more than one boundary point in some direction, i.e. there exist two boundary points \( x_{b1}, x_{b2} \) such that \( \frac{x_{b2} - x_{b1}}{\|x_{b2} - x_{b1}\|} = \frac{x_{b3} - x_{b1}}{\|x_{b3} - x_{b1}\|} \). Let \( z_1 \) be the point, \( x_{b1} \) or \( x_{b2} \), which is furthest away from \( x_{b1} \) and \( z_2 \) be the closest one, such that \( z_2 \in l'(x_{b1}, z_1) \) where \( l'(x_{b1}, z_1) \) is the open line segment from \( x_{b1} \) to \( z_1 \). As int ker(A) is nonempty, there exists an \( n \)-ball, \( B_n(x_{b1}) \), with radius \( n \) around \( x_{b1} \), such that \( B_n(x_{b1}) \subset \text{ker}(A) \). From the starshapedness of \( A \) we have \( l(x, z_1) \subset A \), \( \forall x \in \text{ker}(A) \), and the cone \( C_{z_1} = \bigcup_{z \in B_n(x_{b1})} l(x, z) \) is hence contained by \( A \). The cone \( C_{z_1} \) is centered around the axis aligned with \( l(x_{b1}, z_1) \), and \( l'(x_{b1}, z_1) \subset \text{int} C_{z_1} \subset \text{int} A \). Thus \( z_2 \in \text{int} A \). That is, the closest point to \( x_{b1} \) of \( x_{b1} \) and \( x_{b2} \) is an interior point of \( A \). This is a contradiction to \( x_{b1}, x_{b2} \in \partial A \) and we can conclude that \( A \) is strictly starshaped.

D. Proof of Property 2

We have \( x \in \text{ad ker}(A) \Rightarrow \text{SH}_x(A) \cap \tilde{x} = \emptyset, \forall A \in A \) since

\[
\text{ad ker}(A, \tilde{x}) = \{ x \in \mathbb{R}^n : \tilde{x} \cap \text{SH}_x(A) = \emptyset \},
\]

Thus, \( x \in \text{ad ker}(A) \) if and only if \( x \in \bigcap_{A \in A} \text{ad ker}(A, \tilde{x}) \).
\( S_1 \subset S_2 \) it is also the smallest set satisfying these conditions and we have \( S_2 = S_1 \). Similarly, \( S_{ij} \) is a starshaped set containing \( A \) with \( K \) contained by its kernel. Since \( S_{ij} \subset S_1 \) it is also the smallest set satisfying these conditions and we have \( S_1 = S_{ij} \).

b. \( \Rightarrow \): \( SH_{\ker K}(A) \) is a starshaped set w.r.t. \( \forall k \in K \) by definition, i.e. \( SH_{\ker K}(A) \) is starshaped with \( k \in \ker(SH_{\ker K}(A)) \). Thus, if \( A = SH_{\ker K}(A) \), \( A \) is also starshaped with \( K \subset \ker(A) \).

\( \Leftarrow \): Since \( K \subset \ker(A) \), \( A \) is starshaped w.r.t. \( \forall k \in K \). Obviously, \( A \) is the minimum starshaped set which contains \( A \) and we have \( SH_{\ker K}(A) = A \).

c. A convex set is a starshaped set with kernel given as the set itself. Thus, \( CH(A) \) is a starshaped set with \( A \subset CH(A) \) and for any \( K \subset CH(A) \) we have \( K \subset \ker(CH(A)) \). Thus, under assumption that \( K \subset CH(A) \), \( CH(A) \) is a starshaped set containing \( A \) with \( K \) contained by its kernel. If it is the smallest such set, we have \( SH_{\ker K}(A) = CH(A) \), otherwise \( SH_{\ker K}(A) \subset CH(A) \). Thus, \( SH_{\ker K}(A) \subset CH(A) \), \( \forall K \subset CH(A) \).

d. We have

\[
SH_{\ker K}( \bigcup_{B \in B} B ) \supseteq \bigcup_{k \in CH(K)} SH_k( \bigcup_{B \in B} B ) = \bigcup_{k \in CH(K)} \bigcup_{B \in B} SH_k(B) \supseteq \bigcup_{B \in B} SH_{\ker K}(B).
\]

H. Proof of Property \[1\]

The convex hull of \( n + 1 \) affinely independent points is of dimension \( n \) and thus \( CH(K) \) is guaranteed to have a nonempty interior. Knowing from Property 3 that \( CH(K) \subset \ker(SH_{\ker K}(A)) \), we therefore have that \( \text{int} \ \ker(SH_{\ker K}(A)) \neq \emptyset \). Using Proposition 2 \( SH_{\ker K}(A) \) can be concluded to be a strictly starshaped set.

I. Proof of Property \[2\]

From the definition of the admissible kernel we have \( CH(K) \subset \text{ad} \ \ker(A, X) \Rightarrow SH_k(A) \cap X = \emptyset, \forall k \in CH(K) \), and it follows that \( \bigcup_{k \in CH(K)} SH_k(A) \cap X = \emptyset \). Using Property 3 this can be written as \( SH_{\ker K}(A) \cap X = \emptyset \).

REFERENCES

[1] S. LaValle, “Motion planning: The essentials,” Robotics & Automation Magazine, IEEE, vol. 18, pp. 79–89, 03 2011.
[2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in Proceedings, 1985 IEEE International Conference on Robotics and Automation, vol. 2, pp. 500–505, 1985.
[3] M. Ginesi, D. Meli, A. Calanca, D. Dall’Alba, N. Sansonetto, and P. Fiorini, “Dynamic movement primitives: Volumetric obstacle avoidance,” in 2019 19th International Conference on Advanced Robotics (ICAR), pp. 234–239, 2019.
[4] S. Stavridis, D. Papageorgiou, and Z. Doulgeri, “Dynamical system based robotic motion generation with obstacle avoidance,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 712–718, 2017.
[5] C. Connolly, J. Burns, and R. Weiss, “Path planning using laplace’s equation,” in Proceedings., IEEE International Conference on Robotics and Automation, pp. 2102–2106 vol.3, 1990.
[6] H. Feder and J.-J. Slotine, “Real-time path planning using harmonic potentials in dynamic environments,” in Proceedings of International Conference on Robotics and Automation, vol. 1, pp. 874–881 vol.1, 1997.
[7] R. Daily and D. M. Bevly, “Harmonic potential field path planning for high speed vehicles,” in 2008 American Control Conference, pp. 4609–4614, 2008.
[8] L. Huber, A. Billard, and J.-J. Slotine, “Avoidance of convex and concave obstacles with convergence ensured through contraction,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1462–1469, 2019.
[9] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” IEEE Transactions on Robotics and Automation, vol. 8, no. 5, pp. 501–518, 1992.
[10] R. Conn and M. Kam, “Robot motion planning on n-dimensional star worlds among moving obstacles,” IEEE Transactions on Robotics and Automation, vol. 14, no. 2, pp. 320–325, 1998.
[11] S. G. Loizou, “Closed form navigation functions based on harmonic potentials,” in 2011 50th IEEE Conference on Decision and Control and European Control Conference, pp. 6361–6366, 2011.
[12] S. Paternain, D. E. Koditschek, and A. Ribeiro, “Navigation functions for convex potentials in a space with convex obstacles,” IEEE Transactions on Automatic Control, vol. 63, no. 9, pp. 2944–2959, 2018.
[13] S. Hacohen, S. Shoval, and N. Shvalb, “Probability navigation function for stochastic static environments,” International Journal of Control, Automation and Systems, vol. 17, 05 2019.
[14] S. G. Loizou and E. D. Rimon, “Correct-by-construction navigation functions with application to sensor based robot navigation,” 2021.
[15] S. G. Loizou, “The navigation transformation: Point worlds, time abstractions and towards tuning-free navigation,” in 2011 19th Mediterranean Conference on Control Automation (MED), pp. 303–308, 2011.
[16] M. Beltagy and A. El-Araby, “On convex and starshaped hulls,” Kyungpook mathematical journal, vol. 40, 01 2000.
[17] G. Hansen, I. Herbut, H. Martini, and M. Moszyńska, “Starshaped sets,” Aequationes mathematicae, vol. 94, 12 2020.
[18] E. M. Arkin, Y.-J. Chiang, M. Held, J. B. M. Mitchell, V. S. Adinolfi, S. Skiena, and T.-H. Yang, “On minimum-area hulls,” Algoritmita, vol. 21, pp. 119–136, 1998.
[19] H. Freeman and P. P. Loutrel, “An algorithm for the solution of the two-dimensional “hidden-line” problem,” IEEE Transactions on Electronic Computers, vol. EC-16, no. 6, pp. 784–790, 1967.
[20] S. Hertel and K. Mehlhorn, “Fast triangulation of simple polygons,” in Foundations of Computation Theory (M. Karpinski, ed.), (Berlin, Heidelberg), pp. 207–218, Springer Berlin Heidelberg, 1983.
[21] D. T. Lee and F. P. Preparata, “An optimal algorithm for finding the kernel of a polygon,” J. ACM, vol. 26, p. 415–421, jul 1979.