Reactive Navigation in Partially Known Non-Convex Environments*

Vasileios Vasilopoulos[1] and Daniel E. Koditschek[2]

[1] Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104
[2] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104
{vvasilo,kod}@seas.upenn.edu

Abstract. This paper presents a provably correct method for robot navigation in 2D environments cluttered with familiar but unexpected non-convex, star-shaped obstacles as well as completely unknown, convex obstacles. We presuppose a limited range onboard sensor, capable of recognizing, localizing and (leveraging ideas from constructive solid geometry) generating online from its catalogue of the familiar, non-convex shapes an implicit representation of each one. These representations underlie an online change of coordinates to a completely convex model planning space wherein a previously developed online construction yields a provably correct reactive controller that is pulled back to the physically sensed representation to generate the actual robot commands. We extend the construction to differential drive robots, and suggest the empirical utility of the proposed control architecture using both formal proofs and numerical simulations.

Keywords: Motion and Path Planning · Collision Avoidance · Vision and Sensor-based Control.

1 Introduction

1.1 Motivation and Prior Work

Recent advances in the theory of sensor-based reactive navigation [2] and its application to wheeled [3] and legged [28] robots promote its central role in provably correct architectures for increasingly complicated mobile manipulation tasks [29, 31]. The advance of the new theory [2] over prior sensor-based collision avoidance schemes [5–8, 10, 11, 15, 20, 26] was the additional guaranteed convergence to a designated goal which had theretofore only been established for reactive planners possessing substantial prior knowledge about the environment [19, 24]. A key feature of these new (and other recent parallel [14, 21])

* This work was supported by AFRL grant FA865015D1845 (subcontract 6697371). The authors thank Dr. Omur Arslan for many formative discussions and for sharing his simulation and presentation infrastructure.

approaches is that they trade away prior knowledge for the presumption of simplicity: unknown obstacles can be successfully negotiated in real time without losing global convergence guarantees if they are "round" (i.e., very strongly convex in a sense made precise in [3]). The likely necessity of such simple geometry for guaranteed safe convergence by a completely uninformed "greedy" reactive navigation planner is suggested by the result that a collision avoiding, distance-diminishing reactive navigation policy can reach arbitrarily placed goals in an unknown freespace *only if* all obstacles are "round" [3, Proposition 14].

This paper offers a step toward elucidating the manner in which partial knowledge may suffice to inform safe, convergent, reactive navigation in geometrically more interesting environments. Growing experience negotiating learned [12] or estimated [17,27] environments suggests that reasonable statistical priors may go a long way toward provable stochastic navigation. But in this work we are interested in what sort of deterministic guarantees may be possible. Recent developments in semantic SLAM [9] and object pose and triangular mesh extraction using convolutional neural net architectures [16, 18, 22] now provide an avenue for incorporating partial prior knowledge within a deterministic framework well suited to the vector field planning methods reviewed above.

1.2 Contributions and Organization of the Paper

We consider the navigation problem in a 2D workspace cluttered with unknown convex obstacles, along with "familiar" non-convex, star-shaped obstacles [23] that belong to classes of known geometries, but whose number and placement are unknown, awaiting discovery at execution time. We assume a limited range onboard sensor, a sufficient margin separating all obstacles from each other and the goal, and a catalogue of known star-shaped sets, along with a "mapping oracle" for their online identification and localization in the physical workspace. These ingredients suggest a representation of the environment taking the form of a "multi-layer" triple of topological spaces whose realtime interaction can be exploited to integrate the geometrically naive sensor driven methods of [2] with the offline memorized geometry sensitive methods of [24]. Specifically, we adapt the construction of [23] to generate a realtime smooth change of coordinates (a *diffeomorphism*) of the mapped layer of the environment into a (locally) topologically equivalent but geometrically more favorable model layer relative to which the reactive methods of [2] can be directly applied. We prove that the conjugate vector field defined by pulling back the reactive model space planner through this diffeomorphism induces a vector field on the robot's physical configuration space that inherits the same formal guarantees of obstacle avoidance and convergence. We extend the construction to the case of a differential drive robot, by pulling back the extended field over planar rigid transformations introduced for this purpose in [2] through a suitable polar coordinate transformation of the tangent lift of our original planar diffeomorphism and demonstrate, once again, that the physical differential drive robot inherits the same obstacle avoidance and convergence properties as those guaranteed for the geometrically simple model

robot [2]. Finally, to better support online implementation of these constructions, we adopt modular methods for implicit description of geometric shape [25].

The paper is organized as follows. Section 2 describes the problem and establishes our assumptions. Section 3 describes the physical, mapped and model planning layers used in the constructed diffeomorphism between the mapped and model layers, whose properties are established next. Based on these results, Section 4 describes our control approach both for fully actuated and differential drive robots. Section 5 presents a variety of illustrative numerical studies and Section 6 concludes by summarizing our findings and presenting ideas for future work. Finally, the reader is referred to Appendix A and Appendix B of the accompanying technical report [30] for the proofs of our main results and a sketch of the ideas from computational geometry [25] underlying our modular construction of implicit representations of polygonal obstacles, respectively.

2 Problem Formulation

We consider a disk-shaped robot with radius $r > 0$, centered at $\mathbf{x} \in \mathbb{R}^2$, navigating a closed, compact workspace $\mathcal{W} \subset \mathbb{R}^2$, with known convex boundary $\partial\mathcal{W}$. The robot is assumed to possess a sensor with fixed range R, capable of recognizing "familiar" objects, as well as estimating the distance of the robot to nearby obstacles[3].

The workspace is cluttered by an unknown number of fixed, disjoint obstacles, denoted by $\mathcal{O} := (O_1, O_2, \ldots)$. We adopt the notation in [2] and define the *freespace* as

$$\mathcal{F} := \left\{ \mathbf{x} \in \mathcal{W} \,\middle|\, \overline{B(\mathbf{x}, r)} \subseteq \mathcal{W} \setminus \bigcup_i O_i \right\} \tag{1}$$

where $B(\mathbf{x}, r)$ is the open ball centered at $\mathbf{x}$ with radius r, and $\overline{B(\mathbf{x}, r)}$ denotes its closure. To simplify our notation, we neglect the robot dimensions, by dilating each obstacle in $\mathcal{O}$ by r, and assume that the robot operates in $\mathcal{F}$. We denote the set of dilated obstacles by $\tilde{\mathcal{O}}$.

Although none of the positions of any obstacles in $\tilde{\mathcal{O}}$ are à-priori known, a subset $\tilde{\mathcal{O}}^* \subseteq \tilde{\mathcal{O}}$ of these obstacles is assumed to be "familiar" in the sense of having an à-priori known, readily recognizable star-shaped geometry [23] (i.e., belonging to a known catalogue of star-shaped *geometry classes*), which the robot can efficiently identify and localize instantaneously from online sensory measurement. Although the implementation of such a sensory apparatus lies well beyond the scope of the present paper, recent work on semantic SLAM [9] provides an excellent example with empirically demonstrated technology for achieving this need for localizing, identifying and keeping track of all the familiar obstacles encountered in the otherwise unknown environment. The à-priori unknown center of each catalogued star-shaped obstacle $\tilde{O}_i^*$ is denoted $\mathbf{x}_i^*$. Similarly to [24],

[3] We refer the reader to an example of existing technology [1] generating 2D LIDAR scans from 3D point clouds for such an approach.

each star-shaped obstacle $\tilde{O}_i^* \in \tilde{\mathcal{O}}^*$ can be described by an *obstacle function*, a real-valued map providing an implicit representation of the form

$$\tilde{O}_i^* = \{\mathbf{x} \in \mathbb{R}^2 \,|\, \beta_i(\mathbf{x}) \leq 0\} \tag{2}$$

which the robot must construct online from the catalogued geometry, after it has localized $\tilde{O}_i^*$. The remaining obstacles $\tilde{\mathcal{O}}_{convex} := \tilde{\mathcal{O}} \backslash \tilde{\mathcal{O}}^*$ are are assumed to be strictly convex but are in all other regards (location and specific shape) completely unknown to the robot, while nevertheless satisfying a curvature condition given in [2, Assumption 2].

For the obstacle functions, we require the technical assumptions introduced in [24, Appendix III], outlined as follows.

Assumption 1. *The obstacle functions satisfy the following requirements*

a) For each $\tilde{O}_i^* \in \tilde{\mathcal{O}}^*$, there exists $\varepsilon_{1i} > 0$ such that for any two obstacles $\tilde{O}_i^*, \tilde{O}_j^* \in \tilde{\mathcal{O}}^*$

$$\{\mathbf{x} \,|\, \beta_i(\mathbf{x}) \leq \varepsilon_{1i}\} \bigcap \{\mathbf{x} \,|\, \beta_j(\mathbf{x}) \leq \varepsilon_{1j}\} = \emptyset \tag{3}$$

i.e., the "thickened boundaries" of any two stars still do not overlap.

b) For each $\tilde{O}_i^* \in \tilde{\mathcal{O}}^*$, there exists $\varepsilon_{2i} > 0$ such that the set $\{\mathbf{x} \,|\, \beta_i(\mathbf{x}) \leq \varepsilon_{2i}\}$ does not contain the goal $\mathbf{x}_d \in \mathcal{F}$ and does not intersect with any other obstacle in $\tilde{\mathcal{O}}_{convex}$.

c) For each obstacle function β_i, there exists a pair of positive constants $(\delta_i, \varepsilon_{3i})$ satisfying the inner product condition[4]

$$(\mathbf{x} - \mathbf{x}_i^*)^\top \nabla \beta_i(\mathbf{x}) \geq \delta_i \tag{4}$$

for all $\mathbf{x} \in \mathbb{R}^2$ such that $\beta_i(\mathbf{x}) \leq \varepsilon_{3i}$.

For each obstacle $\tilde{O}_i^* \in \tilde{\mathcal{O}}^*$, we then define $\varepsilon_i = \min\{\varepsilon_{1i}, \varepsilon_{2i}, \varepsilon_{3i}\}$. Finally, we will assume that the range of the sensor R satisfies $R >> \varepsilon_i$ for all i.

Based on these assumptions and further positing first-order, fully-actuated robot dynamics $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x})$, the problem consists of finding a Lipschitz continuous controller $\mathbf{u} : \mathcal{F} \to \mathbb{R}^2$, that leaves the freespace $\mathcal{F}$ positively invariant and asymptotically steers almost all configurations in $\mathcal{F}$ to the given goal $\mathbf{x}_d \in \mathcal{F}$.

3 Multi-layer Representation of the Environment and Its Associated Transformations

In this Section, we introduce associated notation for, and transformations between three distinct representations of the environment that we will refer to as planning "layers" and use in the construction of our algorithm. Fig. 1 illustrates the role of these layers and the transformations that relate them in constructing and analyzing a realtime generated vector field that guarantees safe passage to

[4] A brief discussion on this condition is given in [30, Appendix B].

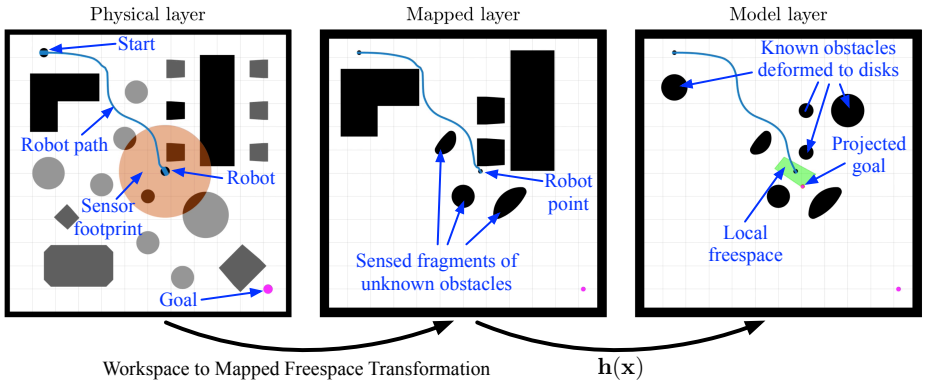

Fig. 1. Snapshot Illustration of Key Ideas. The robot in the physical layer (left frame, depicting in blue the robot's placement in the workspace along with the prior trajectory of its centroid) containing both familiar objects of known geometry but unknown location (dark grey) and unknown obstacles (light grey), moves towards a goal and discovers obstacles (black) with an onboard sensor of limited range (orange disk). These obstacles are localized and stored permanently in the mapped layer (middle frame, depicting in blue the robot's placement as a point in freespace rather than its body in the workspace) if they have familiar geometry or temporarily, with just the corresponding sensed fragments, if they are unknown. An online map $\mathbf{h}(\mathbf{x})$ is then constructed (Section 3), from the mapped layer to a geometrically simple model layer (right frame, now depicting the robot's placement and prior tractory amongst the $\mathbf{h}$-deformed convex images of the mapped obstacles). A doubly reactive control scheme for convex environments [2] defines a vector field on the model layer which is pulled back in realtime through the diffeomorphism to generate the input in the physical layer (Section 4).

the goal. The new technical contribution is an adaptation of the methods of [24] to the construction of a diffeomorphism, $\mathbf{h}$, where the requirement for fast, online performance demands an algorithm that is as simple as possible and with few tunable parameters. Hence, since the reactive controller in [2] is designed to (provably) handle convex shapes, sensed obstacles not recognized by the semantic SLAM process are simply assumed to be convex (implemented by designing $\mathbf{h}$ to resolve to the identity transformation in the neighborhood of "unfamiliar" objects) and the control response defaults to that prior construction.

3.1 Description of Planning Layers

Physical Layer The *physical layer* is a complete description of the geometry of the unknown actual world and while inaccessible to the robot is used for purposes of analysis. It describes the actual workspace $\mathcal{W}$, punctured with the obstacles $\mathcal{O}$. This gives rise to the freespace $\mathcal{F}$, given in (1), consisting of all placements of the robot's centroid that entail no intersections of its body with any obstacles. The robot navigates this layer, and discovers and localizes new obstacles, which are then stored in its *semantic map* if their geometry is familiar.

Mapped Layer The *mapped layer* $\mathcal{F}_{map}$ has the same boundary as $\mathcal{F}$ (i.e. $\partial\mathcal{F}_{map} := \partial\mathcal{F}$) and records the robot's evolving information about the environment aggregated from the raw sensor data about the observable portions of $N \geq 0$ unrecognized (and therefore, presumed convex) obstacles $\{\tilde{O}_1, \ldots, \tilde{O}_N\} \subseteq \tilde{\mathcal{O}}_{convex}$, together with the inferred star centers $\mathbf{x}_j^*$ and obstacle functions β_j of $M \geq 0$ star-shaped obstacles $\{\tilde{O}_1^*, \ldots, \tilde{O}_M^*\} \subseteq \tilde{\mathcal{O}}^*$, that are instantiated at the moment the sensory data triggers the "memory" that identifies and localizes a familiar obstacle. It is important to note that the star environment is constantly updated, both by discovering and storing new star-shaped obstacles in the semantic map and by discarding old information and storing new information regarding obstacles in $\tilde{\mathcal{O}}_{convex}$. In this representation, the robot is treated as a point particle, since all obstacles are dilated by r in the passage from the workspace to the freespace representation of valid placements.

Model Layer The *model layer* $\mathcal{F}_{model}$ has the same boundary as $\mathcal{F}$ (i.e. $\partial\mathcal{F}_{model} := \partial\mathcal{F}$) and consists of a collection of M Euclidean disks, each centered at one of the mapped star centers, $\mathbf{x}_j^*, j = 1, \ldots, M$, and copies of the sensed fragments of the N unrecognized visible convex obstacles in $\tilde{\mathcal{O}}_{convex}$. The radii $\{\rho_1, \ldots, \rho_M\}$ of the M disks are chosen so that $\overline{B(\mathbf{x}_j^*, \rho_j)} \subseteq \{\mathbf{x} \,|\, \beta_j(\mathbf{x}) < 0\}$, as in [24].

This metric convex sphere world comprises the data generating the doubly reactive algorithm of [2], which will be applied to the physical robot via the online generated change of coordinates between the mapped layer and the model layer to be now constructed.

3.2 Description of the C^∞ Switches

In order to simplify the diffeomorphism construction, we depart from the construction of analytic switches [23] and rely instead on the C^∞ function $\zeta : \mathbb{R} \to \mathbb{R}$ [13] described by

$$\zeta(\chi) = \begin{cases} e^{-1/\chi}, & \chi > 0 \\ 0, & \chi \leq 0 \end{cases} \tag{5}$$

with derivative

$$\zeta'(\chi) = \begin{cases} \frac{\zeta(\chi)}{\chi^2}, & \chi > 0 \\ 0, & \chi \leq 0 \end{cases} \tag{6}$$

Based on that function, we can then define the C^∞ switches for each star-shaped obstacle $\tilde{O}_j^*$ in the semantic map as

$$\sigma_j(\mathbf{x}) = \eta_j \circ \beta_j(\mathbf{x}), \quad j = 1, \ldots, M \tag{7}$$

with $\eta_j(\chi) = \zeta(\varepsilon_j - \chi)/\zeta(\varepsilon_j)$ and ε_j given according to Assumption 1. The gradient of the switch σ_j is given by

$$\nabla\sigma_j(\mathbf{x}) = (\eta_j' \circ \beta_j(\mathbf{x})) \cdot \nabla\beta_j(\mathbf{x}) \tag{8}$$

Finally, we define

$$\sigma_d(\mathbf{x}) = 1 - \sum_{j=1}^{M} \sigma_j(\mathbf{x}) \tag{9}$$

Using the above construction, it is easy to see that $\sigma_j(\mathbf{x}) = 1$ on the boundary of the j-th obstacle and $\sigma_j(\mathbf{x}) = 0$ when $\beta_j(\mathbf{x}) > \varepsilon_j$ for each $j = 1, \ldots, M$. Based on Assumption 1 and the choice of ε_j for each j, we are, therefore, led to the following results.

Lemma 1. *At any point $\mathbf{x} \in \mathcal{F}_{map}$, at most one of the switches $\{\sigma_1, \ldots, \sigma_M\}$ can be nonzero.*

Corollary 1. *The set $\{\sigma_1, \ldots, \sigma_M, \sigma_d\}$ defines a partition of unity over $\mathcal{F}_{map}$.*

3.3 Description of the Star Deforming Factors

The deforming factors are the functions $\nu_j(\mathbf{x}) : \mathcal{F}_{map} \to \mathbb{R}, j = 1, \ldots, M$, responsible for transforming each star-shaped obstacle into a disk in $\mathbb{R}^2$. Once again, we use here a slightly different construction than [23], in that the value of each deforming factor ν_j at a point $\mathbf{x}$ does not depend on the value of $\beta_j(\mathbf{x})$. Namely, the deforming factors are given based on the desired final radii $\rho_j, j = 1, \ldots, M$ as

$$\nu_j(\mathbf{x}) = \frac{\rho_j}{||\mathbf{x} - \mathbf{x}_j^*||} \tag{10}$$

We also get

$$\nabla \nu_j(\mathbf{x}) = -\frac{\rho_j}{||\mathbf{x} - \mathbf{x}_j^*||^3}(\mathbf{x} - \mathbf{x}_j^*) \tag{11}$$

3.4 The Map Between the Mapped and the Model Layer

Construction The map for M star-shaped obstacles centered at $\mathbf{x}_j^*, j = 1, \ldots, M$ is described by a function $\mathbf{h} : \mathcal{F}_{map} \to \mathcal{F}_{model}$ given by

$$\mathbf{h}(\mathbf{x}) = \sum_{j=1}^{M} \sigma_j(\mathbf{x}) \left[\nu_j(\mathbf{x})(\mathbf{x} - \mathbf{x}_j^*) + \mathbf{x}_j^*\right] + \sigma_d(\mathbf{x})\mathbf{x} \tag{12}$$

Note that the N visible convex obstacles $\{\tilde{O}_1, \ldots, \tilde{O}_N\} \subseteq \tilde{\mathcal{O}}_{convex}$ are not considered in the construction of the map. Since the reactive controller used in the model space $\mathcal{F}_{model}$ can handle convex obstacles and there is enough separation between convex and star-shaped obstacles according to Assumption 1-(b), we can "transfer" the geometry of those obstacles directly in the model space using the identity transformation.

Finally, note that Assumption 1-(b) implies that $\mathbf{h}(\mathbf{x}_d) = \mathbf{x}_d$, since the target location is assumed to be sufficiently far from all star-shaped obstacles.

Based on the construction of the map $\mathbf{h}$, the jacobian $D_{\mathbf{x}}\mathbf{h}$ at any point $\mathbf{x} \in \mathcal{F}_{map}$ is given by

$$D_{\mathbf{x}}\mathbf{h} = \sum_{j=1}^{M} \left\{ \sigma_j(\mathbf{x})\nu_j(\mathbf{x})\mathbf{I} + (\mathbf{x} - \mathbf{x}_j^*) \left[\sigma_j(\mathbf{x}) \nabla \nu_j(\mathbf{x})^\top + (\nu_j(\mathbf{x}) - 1) \nabla \sigma_j(\mathbf{x})^\top \right] \right\} + \sigma_d(\mathbf{x})\mathbf{I} \tag{13}$$

Qualitative Properties of the Map We first verify that the construction is a smooth change of coordinates between the mapped and the model layers.

Lemma 2. *The map* $\mathbf{h}$ *from* $\mathcal{F}_{map}$ *to* $\mathcal{F}_{model}$ *is smooth.*

Proof. Included in Appendix A.1 of the accompanying technical report [30]. ■

Proposition 1. *The map* $\mathbf{h}$ *is a* C^∞ *diffeomorphism between* $\mathcal{F}_{map}$ *and* $\mathcal{F}_{model}$*.*

Proof. Included in Appendix A.1 of the accompanying technical report [30]. ■

Implicit representation of obstacles To implement the diffeomorphism between $\mathcal{F}_{map}$ and $\mathcal{F}_{model}$, shown in (12), we rely on the existence of a smooth obstacle function $\beta_j(\mathbf{x})$ for each star-shaped obstacle $j = 1, \ldots, M$ stored in the semantic map. Since recently developed technology [16,18,22] provides means of performing obstacle identification in the form of triangular meshes, in this work we focus on polygonal obstacles on the plane and derive implicit representations using so called "R-functions" from the constructive solid geometry literature [25]. In [30, Appendix B], we describe the method used for the construction of such implicit functions for polygonal obstacles that have the desired property of being analytic everywhere except for the polygon vertices. For the construction, we assume that the sensor has already identified, localized and included each discovered star-shaped obstacle in $\mathcal{F}_{map}$; i.e., it has determined its pose in $\mathcal{F}_{map}$, given as a rotation $\mathbf{R}_j$ of its vertices on the plane followed by a translation of its center $\mathbf{x}_j^*$, and that the corresponding polygon has already been dilated by r for inclusion in $\mathcal{F}_{map}$.

4 Reactive Controller

4.1 Reactive Controller for Fully Actuated Robots

Construction First, we consider a fully actuated particle with state $\mathbf{x} \in \mathcal{F}_{map}$, whose dynamics are described by

$$\dot{\mathbf{x}} = \mathbf{u} \tag{14}$$

The dynamics of the fully actuated particle in $\mathcal{F}_{model}$ with state $\mathbf{y} \in \mathcal{F}_{model}$ are described by $\dot{\mathbf{y}} = \mathbf{v}(\mathbf{y})$ with the control $\mathbf{v}(\mathbf{y})$ given in [2] as[5]

$$\mathbf{v}(\mathbf{y}) = -k \left(\mathbf{y} - \Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})}(\mathbf{x}_d) \right) \tag{15}$$

[5] Here $\Pi_C(\mathbf{q})$ denotes the metric projection of $\mathbf{q}$ on a convex set C.

Here, the convex *local freespace* for $\mathbf{y}$, $\mathcal{LF}(\mathbf{y}) \subset \mathcal{F}_{model}$, is defined as in [2, Eqn. (30)]. Using the diffeomorphism construction in (12) and its jacobian in (13), we construct our controller as the vector field $\mathbf{u} : \mathcal{F}_{map} \to T\mathcal{F}_{map}$ given by

$$\mathbf{u}(\mathbf{x}) = [D_{\mathbf{x}}\mathbf{h}]^{-1} \cdot (\mathbf{v} \circ \mathbf{h}(\mathbf{x})) \tag{16}$$

Qualitative Properties First of all, if the range of the virtual LIDAR sensor used to construct $\mathcal{LF}(\mathbf{y})$ in the model layer is smaller than R, the vector field $\mathbf{u}$ is Lipschitz continuous since $\mathbf{v}(\mathbf{y})$ is shown to be Lipschitz continuous in [2] and $\mathbf{y} = \mathbf{h}(\mathbf{x})$ is a smooth change of coordinates. We are led to the following result.

Corollary 2. *The vector field $\mathbf{u} : \mathcal{F}_{map} \to T\mathcal{F}_{map}$ generates a unique continuously differentiable partial flow.*

To ensure completeness (i.e. absence of finite time escape through boundaries in $\mathcal{F}_{map}$) we must verify that the robot never collides with any obstacle in the environment, i.e., leaves its freespace positively invariant.

Proposition 2. *The freespace $\mathcal{F}_{map}$ is positively invariant under the law* (16).

Proof. Included in Appendix A.2 of the accompanying technical report [30]. ■

Lemma 3. *1. The set of stationary points of control law* (16) *is given as $\{\mathbf{x}_d\} \bigcup \{\mathbf{h}^{-1}(\mathbf{s}_j)\}_{j\in\{1,\dots,M\}} \bigcup_{i=1}^{N} \mathcal{G}_i$, where*[6]

$$\mathbf{s}_j = \mathbf{x}_j^* - \rho_j \frac{\mathbf{x}_d - \mathbf{x}_j^*}{||\mathbf{x}_d - \mathbf{x}_j^*||} \tag{17a}$$

$$\mathcal{G}_i := \left\{ \mathbf{q} \in \mathcal{F}_{map} \,\middle|\, d(\mathbf{q}, O_i) = r, \frac{(\mathbf{q} - \Pi_{\overline{O}_i}(\mathbf{q}))^\top (\mathbf{q} - \mathbf{x}_d)}{||\mathbf{q} - \Pi_{\overline{O}_i}(\mathbf{q})||\,||\mathbf{q} - \mathbf{x}_d||} = 1 \right\} \tag{17b}$$

with j spanning the M star-shaped obstacles in $\mathcal{F}_{map}$ and i spanning the N convex obstacles in $\mathcal{F}_{map}$.

2. The goal $\mathbf{x}_d$ is the only locally stable equilibrium of control law (16) *and all the other stationary points $\{\mathbf{h}^{-1}(\mathbf{s}_j)\}_{j\in\{1,\dots,M\}} \bigcup_{i=1}^{N} \mathcal{G}_i$, each associated with an obstacle, are nondegenerate saddles.*

Proof. Included in Appendix A.2 of the accompanying technical report [30]. ■

Proposition 3. *The goal location $\mathbf{x}_d$ is an asymptotically stable equilibrium of* (16)*, whose region of attraction includes the freespace $\mathcal{F}_{map}$ excepting a set of measure zero.*

Proof. Included in Appendix A.2 of the accompanying technical report [30]. ■

We can now immediately conclude the following central summary statement.

Theorem 1. *The reactive controller in* (16) *leaves the freespace $\mathcal{F}_{map}$ positively invariant, and its unique continuously differentiable flow, starting at almost any robot placement $\mathbf{x} \in \mathcal{F}_{map}$, asymptotically reaches the goal location $\mathbf{x}_d$, while strictly decreasing $||\mathbf{h}(\mathbf{x}) - \mathbf{x}_d||$ along the way.*

[6] Here $d(A, B) = \inf\{||\mathbf{a} - \mathbf{b}|| \,|\, \mathbf{a} \in A, \mathbf{b} \in B\}$ denotes the distance between two sets A, B.

4.2 Reactive Controller for Differential Drive Robots

In this Section, we extend our reactive controller to the case of a differential drive robot, whose state is $\overline{\mathbf{x}} := (\mathbf{x}, \psi) \in \mathcal{F}_{map} \times S^1 \subset SE(2)$, and its dynamics are given by[7]

$$\dot{\overline{\mathbf{x}}} = \mathbf{B}(\psi)\overline{\mathbf{u}} \tag{18}$$

with $\mathbf{B}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}^\top$ and $\overline{\mathbf{u}} = (v, \omega)$ with $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ the linear and angular input respectively. We will follow a similar procedure to the fully actuated case; we begin by describing a smooth diffeomorphism $\overline{\mathbf{h}} : \mathcal{F}_{map} \times S^1 \to \mathcal{F}_{model} \times S^1$ and then we establish the results about the controller.

Construction and Properties of the $SE(2)$ Diffeomorphism We construct our map $\overline{\mathbf{h}}$ from $\mathcal{F}_{map} \times S^1$ to $\mathcal{F}_{model} \times S^1$ as

$$\overline{\mathbf{y}} = (\mathbf{y}, \varphi) = \overline{\mathbf{h}}(\overline{\mathbf{x}}) := (\mathbf{h}(\mathbf{x}), \xi(\overline{\mathbf{x}})) \tag{19}$$

with $\overline{\mathbf{x}} = (\mathbf{x}, \psi) \in \mathcal{F}_{map} \times S^1$, $\overline{\mathbf{y}} := (\mathbf{y}, \varphi) \in \mathcal{F}_{model} \times S^1$ and

$$\varphi = \xi(\overline{\mathbf{x}}) := \angle\left(\mathbf{e}(\overline{\mathbf{x}})\right) \tag{20}$$

Here, $\angle\mathbf{e} := \text{atan2}(e_2, e_1)$ and

$$\mathbf{e}(\overline{\mathbf{x}}) = \Pi_{\mathbf{y}} \cdot D_{\overline{\mathbf{x}}}\overline{\mathbf{h}} \cdot \mathbf{B}(\psi) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = D_{\mathbf{x}}\mathbf{h} \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} \tag{21}$$

with $\Pi_{\mathbf{y}}$ denoting the projection onto the first two components. The reason for choosing φ as in (20) will become evident in the next paragraph, in our effort to control the equivalent differential drive robot dynamics in $\mathcal{F}_{model}$.

Proposition 4. *The map $\overline{\mathbf{h}}$ in (19) is a C^∞ diffeomorphism from $\mathcal{F}_{map} \times S^1$ to $\mathcal{F}_{model} \times S^1$.*

Proof. Included in Appendix A.2 of the accompanying technical report [30]. ■

Construction of the Reactive Controller Using (19), we can find the push-forward of the differential drive robot dynamics in (18) as

$$\dot{\overline{\mathbf{y}}} = \begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\varphi} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \xi(\overline{\mathbf{x}}) \end{bmatrix} = \left[D_{\overline{\mathbf{x}}}\overline{\mathbf{h}} \circ \overline{\mathbf{h}}^{-1}(\overline{\mathbf{y}})\right] \cdot \left(\mathbf{B} \circ \overline{\mathbf{h}}^{-1}(\overline{\mathbf{y}})\right) \cdot \overline{\mathbf{u}} \tag{22}$$

Based on the above, we can then write

$$\dot{\overline{\mathbf{y}}} = \begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\varphi} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \xi(\overline{\mathbf{x}}) \end{bmatrix} = \mathbf{B}(\varphi)\overline{\mathbf{v}} \tag{23}$$

[7] We use the ordered set notation $(*, *, \ldots)$ and the matrix notation $\left[* * \ldots\right]^\top$ for vectors interchangeably.

with $\overline{\mathbf{v}} = (\hat{v}, \hat{\omega})$, and the inputs $(\hat{v}, \hat{\omega})$ related to (v, ω) through

$$\hat{v} = ||\mathbf{e}(\overline{\mathbf{x}})||\, v \tag{24}$$

$$\hat{\omega} = v D_{\mathbf{x}}\xi \begin{bmatrix}\cos\psi \\ \sin\psi\end{bmatrix} + \frac{\partial \xi}{\partial \psi}\omega \tag{25}$$

with $D_{\mathbf{x}}\xi = \begin{bmatrix}\frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y}\end{bmatrix}$. The calculation of $D_{\mathbf{x}}\xi$ can be tedious, since it involves derivatives of elements of $D_{\mathbf{x}}\mathbf{h}$, and is included in [30, Appendix C].

Hence, we have found equivalent differential drive robot dynamics, defined on $\mathcal{F}_{model} \times S^1$. The idea now is to use the control strategy in [2] for the dynamical system in (23) to find *reference inputs* $\hat{v}, \hat{\omega}$, and then use (24), (25) to find the *actual inputs* v, ω that achieve those reference inputs as

$$v = \frac{\hat{v}}{||\mathbf{e}(\overline{\mathbf{x}})||} \tag{26a}$$

$$\omega = \left(\frac{\partial \xi}{\partial \psi}\right)^{-1}\left(\hat{\omega} - v D_{\mathbf{x}}\xi \begin{bmatrix}\cos\psi \\ \sin\psi\end{bmatrix}\right) \tag{26b}$$

Namely, our reference inputs $\hat{v}$ and $\hat{\omega}$ inspired by [2,4] are given as[8]

$$\hat{v} = -k \begin{bmatrix}\cos\varphi \\ \sin\varphi\end{bmatrix}^{\top}\left(\mathbf{y} - \Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})\cap H_{\parallel}}(\mathbf{x}_d)\right) \tag{27a}$$

$$\hat{\omega} = k \operatorname{atan}\left(\frac{\begin{bmatrix}-\sin\varphi \\ \cos\varphi\end{bmatrix}^{\top}\left(\mathbf{y} - \frac{\Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})\cap H_G}(\mathbf{x}_d) + \Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})}(\mathbf{x}_d)}{2}\right)}{\begin{bmatrix}\cos\varphi \\ \sin\varphi\end{bmatrix}^{\top}\left(\mathbf{y} - \frac{\Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})\cap H_G}(\mathbf{x}_d) + \Pi_{\mathcal{L}\mathcal{F}(\mathbf{y})}(\mathbf{x}_d)}{2}\right)}\right) \tag{27b}$$

with $k > 0$ a fixed gain, $\mathcal{L}\mathcal{F}(\mathbf{y}) \subset \mathcal{F}_{model}$ the convex polygon defining the local freespace at $\mathbf{y} = \mathbf{h}(\mathbf{x})$, and $H_{\parallel}$ and H_G the lines defined in [2] as

$$H_{\parallel} = \left\{\mathbf{z} \in \mathcal{F}_{model} \,\middle|\, \begin{bmatrix}-\sin\varphi \\ \cos\varphi\end{bmatrix}^{\top}(\mathbf{z} - \mathbf{y}) = 0\right\} \tag{28}$$

$$H_G = \{\alpha\mathbf{y} + (1-\alpha)\mathbf{x}_d \in \mathcal{F}_{model} \,|\, \alpha \in \mathbb{R}\} \tag{29}$$

[8] In (19), we construct a diffeomorphism $\overline{\mathbf{h}}$ between $\mathcal{F}_{map} \times S^1$ and $\mathcal{F}_{model} \times S^1$. However, for practical purposes, we deal only with one specific chart of S^1 in our control structure, described by the angles $(-\pi, \pi]$. As shown in [4], the discontinuity at $\pm\pi$ does not induce a discontinuity in our controller due to the use of the atan function in (27b). On the contrary, with the use of (27b) as in [2,4], the robot never changes heading in $\mathcal{F}_{model}$, which implies that the generated trajectories both in $\mathcal{F}_{model}$ and (by the properties of the diffeomorphism $\overline{\mathbf{h}}$) in $\mathcal{F}_{map}$ have no cusps, even though the robot might change heading in $\mathcal{F}_{map}$ because of the more complicated nature of the function ξ in (20).

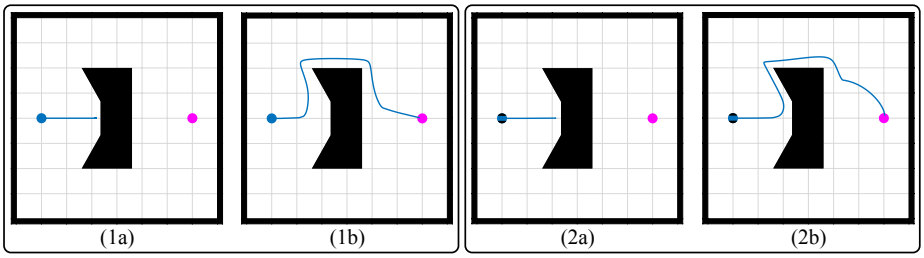

Fig. 2. Navigation around a U-shaped obstacle: 1) Fully actuated particle: (a) Original doubly reactive algorithm [2], (b) Our algorithm, 2) Differential drive robot: (a) Original doubly reactive algorithm [2], (b) Our algorithm.

Qualitative Properties The properties of the differential drive robot control law given in (26) can be summarized in the following theorem.

Theorem 2. *The reactive controller for differential drive robots, given in* (26)*, leaves the freespace* $\mathcal{F}_{map} \times S^1$ *positively invariant, and its unique continuously differentiable flow, starting at almost any robot configuration* $(\mathbf{x}, \psi) \in \mathcal{F}_{map} \times S^1$*, asymptotically steers the robot to the goal location* $\mathbf{x}^*$*, without increasing* $||\mathbf{h}(\mathbf{x}) - \mathbf{x}_d||$ *along the way.*

Proof. Included in Appendix A.2 of the accompanying technical report [30]. ■

5 Numerical Experiments

In this Section, we present numerical experiments that verify our formal results. All simulations were run in MATLAB using `ode45`, with control gain $k = 0.4$ and $p = 20$ for the R-function construction (see [30, Appendix B]). The reader is also referred to our video attachment for a visualization of the examples presented here and more numerical simulations.

5.1 Comparison with Original Doubly Reactive Algorithm

We begin with a comparison of our algorithm performance with the standalone version of the doubly reactive algorithm in [2], that we use in our construction. Fig. 2 demonstrates the basic limitation of this algorithm; in the presence of a non-convex obstacle or a flat surface, whose curvature violates [2, Assumption 2], the robot gets stuck in undesired local minima. On the contrary, our algorithm is capable of overcoming this limitation, on the premise that the robot can recognize the obstacle with star-shaped geometry at hand. The robot radius is 0.2m and the value of ε used for the obstacle is 0.3.

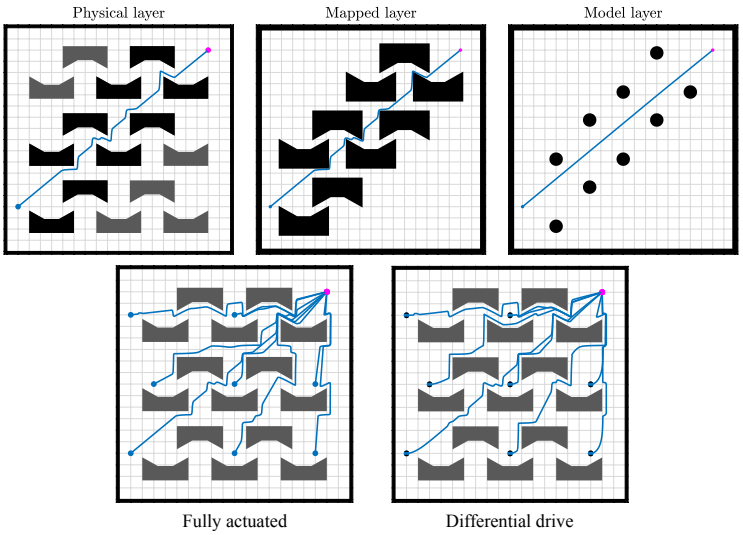

Fig. 3. Navigation in a cluttered environment with U-shaped obstacles. Top - Trajectories in the physical, mapped and model layers from a particular initial condition. Bottom - Convergence to the goal from several initial conditions: left - fully actuated robot, right - differential drive robot.

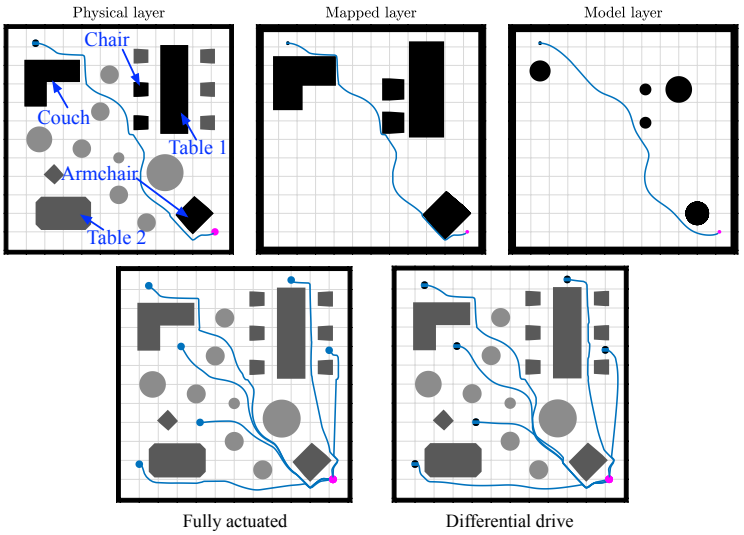

Fig. 4. Navigating a room cluttered with known star-shaped and unknown convex obstacles. Top - Trajectories in the physical, mapped and model layers from a particular initial condition. Bottom - Convergence to the goal from several initial conditions: left - fully actuated robot, right - differential drive robot. Mapped obstacles are shown in black, known obstacles in dark grey and unknown obstacles in light grey.

5.2 Navigation in a Cluttered Non-Convex Environment

In the next set of numerical experiments, we evaluate the performance of our algorithm in a cluttered environment, packed with instances of the same U-shaped obstacle, with star-shaped geometry, we use in Fig. 2. Both the fully actuated and the differential drive robot are capable of converging to the desired goal from a variety of initial conditions, as shown in Fig. 3. In the same figure, we also focus on a particular initial condition and include the trajectories observed in the physical, mapped and model layers. The robot radius is 0.25m and value of ε used for all the star-shaped obstacles in the environment is 0.3.

5.3 Navigation Among Mixed Star-Shaped and Convex Obstacles

Finally, we report experiments in an environment cluttered with both star-shaped obstacles (with known geometry) and unknown convex obstacles. We consider a robot of radius 0.2m navigating a room towards a goal. The robot can recognize familiar star-shaped obstacles (e.g., the couch, tables, armchair, chairs) but is unaware of several other convex obstacles in the environment. Fig. 4 summarizes our results for several initial conditions. We also include trajectories observed in the physical, mapped and model layers during a single run. The value of ε used for all the star-shaped obstacles in the environment is 0.3.

6 Conclusion and Future Work

In this paper, we present a provably correct method for robot navigation in 2D environments cluttered with familiar but unexpected non-convex, star-shaped obstacles as well as completely unknown, convex obstacles. The robot uses a limited range onboard sensor, capable of recognizing, localizing and generating online from its catalogue of the familiar, non-convex shapes an implicit representation of each one. These sensory data and their interpreted representations underlie an online change of coordinates to a completely convex model planning space wherein a previously developed online construction yields a provably correct reactive controller that is pulled back to the physically sensed representation to generate the actual robot commands. Using a modified change of coordinates, the construction is also extended to differential drive robots, and numerical simulations further verify the validity of our formal results.

Experimental validation of our algorithm with deep learning techniques for object pose and triangular mesh recognition [22] is currently underway. Next steps target environments presenting geometry more complicated than star-shaped obstacles, by appropriately modifying the purging transformation algorithm for trees-of-stars, presented in [24]. Future work aims to relax the required degree of partial knowledge and the separation assumptions needed for our formal results, by merging the "implicit representation trees" (e.g. see [30, Fig. 5]) online, when needed.

References

1. http://wiki.ros.org/pointcloud_to_laserscan
2. Arslan, O., Koditschek, D.E.: Sensor-Based Reactive Navigation in Unknown Convex Sphere Worlds. In: The 12th International Workshop on the Algorithmic Foundations of Robotics (2016)
3. Arslan, O., Koditschek, D.E.: Sensor-Based Reactive Navigation in Unknown Convex Sphere Worlds. International Journal of Robotics Research p. (to appear) (Jul 2018)
4. Astolfi, A.: Exponential Stabilization of a Wheeled Mobile Robot Via Discontinuous Control. Journal of Dynamic Systems, Measurement, and Control **121**(1), 121–126 (1999)
5. van den Berg, J., Lin, M., Manocha, D.: Reciprocal Velocity Obstacles for real-time multi-agent navigation. In: IEEE International Conference on Robotics and Automation. pp. 1928–1935 (2008)
6. van den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-Body Collision Avoidance, pp. 3–19. Springer Berlin Heidelberg (2011)
7. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots. IEEE Transactions on Systems, Man, and Cybernetics **19**(5), 1179–1187 (1989)
8. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation **7**(3), 278–288 (1991)
9. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic SLAM. In: IEEE International Conference on Robotics and Automation. pp. 1722–1729 (May 2017)
10. Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: IEEE International Conference on Robotics and Automation. pp. 341–346 (1999)
11. Fiorini, P., Shiller, Z.: Motion Planning in Dynamic Environments Using Velocity Obstacles. The International Journal of Robotics Research **17**(7), 760–772 (1998)
12. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: IEEE International Conference on Robotics and Automation. pp. 981–986 (2010)
13. Hirsch, M.W.: Differential Topology. Springer (1976)
14. Ilhan, B.D., Johnson, A.M., Koditschek, D.E.: Autonomous legged hill ascent. Journal of Field Robotics **35**(5), 802832 (Aug 2018). https://doi.org/10.1002/rob.21779
15. Johnson, A.M., Hale, M.T., Haynes, G.C., Koditschek, D.E.: Autonomous legged hill and stairwell ascent. In: IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 134–142 (2011)
16. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: IEEE International Conference on Computer Vision and Pattern Recognition. pp. 1966–1974 (2015)
17. Karaman, S., Frazzoli, E.: High-speed flight in an ergodic forest. In: IEEE International Conference on Robotics and Automation. pp. 2899–2906 (2012)
18. Kong, C., Lin, C.H., Lucey, S.: Using Locally Corresponding CAD Models for Dense 3D Reconstructions from a Single Image. In: IEEE International Conference on Computer Vision and Pattern Recognition. pp. 5603–5611 (2017)
19. Majumdar, A., Tedrake, R.: Funnel Libraries for Real-Time Robust Feedback Motion Planning. The International Journal of Robotics Research **36**(8), 947–982 (2017)

20. Paranjape, A.A., Meier, K.C., Shi, X., Chung, S.J., Hutchinson, S.: Motion primitives and 3D path planning for fast flight through a forest. The International Journal of Robotics Research **34**(3), 357–377 (2015)
21. Paternain, S., Koditschek, D.E., Ribeiro, A.: Navigation Functions for Convex Potentials in a Space with Convex Obstacles. IEEE Transactions on Automatic Control (2017). https://doi.org/10.1109/TAC.2017.2775046
22. Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K.: 6-DoF object pose from semantic keypoints. In: IEEE International Conference on Robotics and Automation. pp. 2011–2018 (May 2017)
23. Rimon, E., Koditschek, D.E.: The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds. Transactions of the American Mathematical Society **327**(1), 71–116 (1989)
24. Rimon, E., Koditschek, D.E.: Exact Robot Navigation Using Artificial Potential Functions. IEEE Transactions on Robotics and Automation **8**(5), 501–518 (1992)
25. Shapiro, V.: Semi-analytic geometry with R-functions. Acta Numerica **16**, 239–303 (2007)
26. Simmons, R.: The curvature-velocity method for local obstacle avoidance. In: IEEE International Conference on Robotics and Automation. vol. 4, pp. 3375–3382 (1996)
27. Trautman, P., Ma, J., Murray, R.M., Krause, A.: Robot navigation in dense human crowds: Statistical models and experimental studies of humanrobot cooperation. The International Journal of Robotics Research **34**(3), 335–356 (2015)
28. Vasilopoulos, V., Arslan, O., De, A., Koditschek, D.E.: Sensor-Based Legged Robot Homing Using Range-Only Target Localization. In: IEEE International Conference on Robotics and Biomimetics. pp. 2630–2637 (2017)
29. Vasilopoulos, V., Vega-Brown, W., Arslan, O., Roy, N., Koditschek, D.E.: Sensor-Based Reactive Symbolic Planning in Partially Known Environments. In: IEEE International Conference on Robotics and Automation. pp. 5683–5690 (2018)
30. Vasilopoulos, V., Koditschek, D.E.: Technical Report: Reactive Navigation in Partially Known Non-Convex Environments. In: arXiv preprint (2018), `https://arxiv.org/abs/1807.08432`
31. Vasilopoulos, V., Topping, T.T., Vega-Brown, W., Roy, N., Koditschek, D.E.: Sensor-Based Reactive Execution of Symbolic Rearrangement Plans by a Legged Mobile Manipulator. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3298–3305 (2018)