The Navigation Transformation

Savvas G. Loizou

***Abstract*—This work introduces a novel approach to the solution of the navigation problem by mapping an obstacle-cluttered environment to a trivial domain called the *point world*, where the navigation task is reduced to connecting the images of the initial and destination configurations by a straight line. Due to this effect, the underlying transformation is termed the "*navigation transformation*." The properties of the navigation transformation are studied in this work as well as its capability to provide—through the proposed feedback controller designs—solutions to the motion- and path-planning problems. Notably, the proposed approach enables the construction of temporal stabilization controllers as detailed herein, which provide a time abstraction to the navigation problem. The proposed solutions are correct by construction and, given a diffeomorphism from the workspace to a sphere world, tuning free. A candidate construction for the navigation transformation on sphere worlds is proposed. The provided theoretical results are backed by analytical proofs. The efficiency, robustness, and applicability of the proposed solutions are supported by a series of experimental case studies.**

***Index Terms*—Autonomous robots, motion planning, temporal stabilization, time abstraction.**

I. Introduction

The concept of transforming geometrically complex but topologically simple spaces to geometrically simple spaces was initially proposed by Rimon and Koditschek [23]–[25] for transforming *star worlds* to *sphere worlds*. This transformation was essential for implementing *navigation functions* [12] in complicated geometries, since navigation functions were constructed on sphere worlds. Navigation functions, a special class of potential functions free from local minima, were then used to generate through their negated gradient, a vector field on the sphere world. This was diffeomorphically mapped to the initial geometrically complicated domain and served as a control input to asymptotically stabilize, almost everywhere, a system whose dynamics were governed by a simple-holonomic integrator. Numerous applications of navigation functions to systems of increasing complexity have appeared in the literature [5], [7], [8], [20], [24], [28], [29]. Works that provide the necessary conditions for existence of navigation functions of the Rimon–Koditschek form on nonspherical worlds have recently appeared in the literature [9].

While navigation functions are an established technique for robotic navigation, one major issue that exists in closed-form solutions that are based on the Koditschek–Rimon's [12] seed function $\frac{\gamma_d^k}{\beta}$ is that the solution is not correct by construction. In fact, the closed-form solution becomes a navigation function only if the tuning parameter k is selected above a certain threshold whose existence is theoretically guaranteed, but its computation is not straightforward.

Time abstraction plays an important role in time-based scheduling operations [13], [21], [22] especially as these arise in symbolic control and motion planning [1]–[3], [10], [11]. From the robotics point of view, motion tasks need to be completed on time in order for a spatiotemporal scheduling operation to be carried out; hence, guarantees on the time performance of individual navigation tasks are essential. In closed-form feedback-based online motion planners, the generated trajectories depend on the initial condition and on external disturbances. This implies that the length of the trajectory is not known *a priori*, and as a consequence, the time required to carry out a task is uncertain. Several lines of work toward temporal stabilization have appeared in the literature, e.g., [26], [27], [30] that exploit stable limit cycles of Hopf oscillators to generate timed trajectories. To the best of the author's knowledge, this is the first work that provides a time abstraction to the closed-form feedback-based online motion-planning problem with analytically guaranteed global stability and completeness properties.

This work proposes a novel approach to the solution of the navigation problem, by introducing the navigation transformation. The navigation transformation operates on workspaces—restricted to sphere-world diffeomorphs in this work—by diffeomorphically mapping them to geometrically simple domains, where the navigation problem almost always[1] admits a trivial solution. Pulling back the solution to the original workspace yields a solution to the original problem. The importance of this solution is that, once the diffeomorphism has been established, the resulting solution is correct by construction in the sense that no tuning is needed (e.g., to eliminate local minima). In addition, in the transformed domain, the trajectory's length is immediately available, enabling the design of time-abstracting navigation controllers.

In summary, the innovations and contributions of this work are the following.[2]

1) Introduction of the navigation transformation, analysis of its properties, and proposition of a candidate navigation transformation applicable to sphere-world diffeomorphs.
2) Correct-by-construction solutions to the path- and motion-planning problems.
3) Analytically guaranteed time-abstracted solution to the motion-planning problem.
4) Experimentally validated results on mobile robot hardware.

The concept of navigation transformation, which is introduced and studied in the current work, has been successfully exploited in recent works for solving an array of problems in robotic navigation including

Manuscript received November 18, 2016; revised May 24, 2017; accepted June 15, 2017. Date of publication July 28, 2017; date of current version December 14, 2017. This paper was recommended for publication by Associate Editor H. Kress-Gazit and Editor C. Torras upon evaluation of the reviewers' comments. Portions of this work originally appeared in [16]. *(Corresponding author: Savvas G. Loizou.)*

The author is with the Faculty of Mechanical Engineering and Materials Science and Engineering, Cyprus University of Technology, Limassol 3041, Cyprus (e-mail: savvas.loizou@cut.ac.cy).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material consists of a video, showing experimental demonstration of four case studies presented in the paper. Contact savvas.loizou@cut.ac.cy for further questions about this work.

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

Digital Object Identifier 10.1109/TRO.2017.2725323

[1] i.e., up to a set of measure zero of initial conditions

[2] For contributions 1–3, the navigation transformation concept, construction, and the solutions to the motion- and path-planning problems were originally proposed in [16] and are presented here in a refined archival quality form. A novel approach to the time-abstraction problem is introduced in the current paper.

tuning-free—given a diffeomorphism from the workspace to a sphere world—navigation functions in [15], navigation in complex non sphere-diffeomorphic 3-D workspaces [17] and, more recently, for solving the decentralized multiagent navigation problem with arbitrarily shaped agents and obstacles [18]. The price one pays for the navigation transformation is that initial and destination configurations cannot reside on the workspace's (internal) boundary and initial conditions cannot reside in sets of measure zero, since the navigation vector field will grow unbounded.

The rest of this paper is organized as follows. Section II introduces the necessary preliminaries, while Section III introduces basic concepts and properties of the navigation transformation. Section IV exploits the properties of the navigation transformation to design motion-planning and time-abstracting controllers. Section V proposes the construction of a navigation transformation candidate, while providing a rigorous analysis of its properties. Section VI presents the experimental setup and an array of selected case studies. Eventually, Section VII concludes this paper.

II. Preliminaries

In this section, the necessary terminology and definitions for the development of the methodology are introduced.

If K is a set, then $\overline{K}$ denotes its closure, K^c its complement, and $\mathring{K}$ its interior. Denote with ∂K the boundary of K, and let $\mathbb{S}^n$ denote the n-dimensional ball. Denote with $\mathcal{S}^n$ the n-dimensional *sphere world* as this is defined in [12]. Given a function $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, denote the Jacobian matrix of this function as $J_f(\cdot)$ and the Jacobian determinant by $\|J_f(\cdot)\|$.

For a finite $M \in \mathbb{Z}^+$, let $P_i \in \mathbb{R}^n$, $i \in \{1, \dots M\}$ be M discrete elements of $\mathbb{R}^n$. Then:

Definition 1: A point world is defined as a manifold $\mathcal{P}^n \subseteq \mathbb{R}^n \setminus \bigcup_{i=1}^M P_i$.

Definition 2: A point world with spherical boundary is a manifold $\tilde{\mathcal{P}}^n \subseteq \mathcal{P}^n \setminus (\mathring{\mathbb{S}}^n)^c$, where $\bigcup_{i=1}^M P_i \in \mathring{\mathbb{S}}^n$.

Let us restrict our attention to workspaces that are as follows:

Definition 3: A *workspace* (with external boundary) $\mathcal{W} \subset \mathbb{R}^n$ is a manifold such that $\mathring{\mathcal{W}}$ is diffeomorphic to $\mathcal{P}^n$ ($\tilde{\mathcal{P}}^n$).

Although this definition fully captures the free-space topology of 2-D workspaces, it poses restrictions in higher (≥ 3) dimensions, capturing only topologies diffeomorphic to sphere worlds. To see this, observe that for a 2-D workspace, the Euler characteristic χ of a workspace $\mathcal{W}$ with M disjoint obstacles (simply connected subsets of $\mathcal{W}$) is the same with the Euler characteristic of the point world $\mathcal{P}^2$ (or $\tilde{\mathcal{P}}^2$), that is, $\chi = 1 - M$. Now, since $\mathcal{W}$ and $\mathcal{P}^2$ (or $\tilde{\mathcal{P}}^2$) possess the same orientability, then according to the surface classification theorem in topology (see, e.g., [6, Th. 7.19]), these surfaces are homeomorphic (and thus a diffeomorphism can be established upon surface smoothness assumptions). Unfortunately, due to the implications of the surface classification theorem [4], [17], higher dimensional workspaces are fundamentally more complicated than the 2-D ones, since the Euler characteristic of workspaces of dimension $n > 2$ match the Euler characteristic of point worlds only in the case that the workspace is a sphere-world diffeomorph ($\chi = 1 - (-1)^n M$) [12], which of course is just a special case.

Let $\mathcal{O} \triangleq \partial\mathcal{W}$. Then, $\mathcal{O}$ consists of the mutually disjoint sets of the *obstacle boundaries*, $\mathcal{O}_i, i \in \{1, \dots M\}$, and (if such exists) the disjoint "external" boundary $\mathcal{O}_0$, such that $\mathcal{O} = \bigcup_{j=0\dots M} \mathcal{O}_j$.

Assume a system described by the first-order holonomic kinematic model

$$\dot{x} = u \tag{1}$$

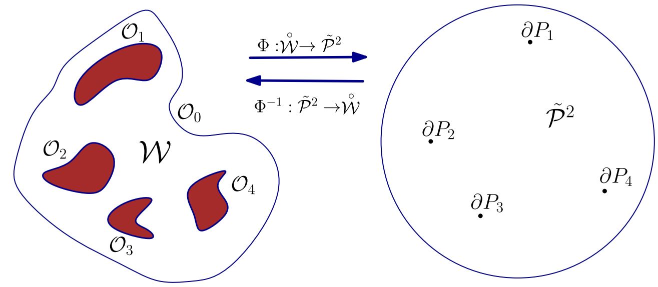

Fig. 1. Navigation transformation $\Phi(\cdot)$ of a bounded 2-D workspace $\mathcal{W}$ to a spherically bounded point world $\tilde{\mathcal{P}}^2$.

where $x \in \mathbb{R}^n$ is the robot's position, and $u \in \mathbb{R}^n$ is the velocity control input. The initial configuration of the robot is denoted as $x_0 \in \mathring{\mathcal{W}}$, and the destination configuration as $x_d \in \mathring{\mathcal{W}}$. We will need the following definitions.

Definition 4 (Motion-planning problem): Given a static workspace, determine (if it exists) a set of control actions $u(t)$, $t \geq t_0$ that will drive system (1) from any initial configuration $x(t_0) = x_0$ to a given destination configuration $x(t_f) = x_d$, for some $t_f > t_0$, avoiding collisions.

Definition 5 (Path-planning problem): Given a static workspace, determine (if it exists) a collision-free path connecting any initial configuration with the destination configuration.

Definition 6 (Time abstraction of the motion-planning problem): Given a static workspace and a finite duration $T > 0$, determine (if it exists) the set of control actions $u(t)$, $t \in [t_0, t_0 + T]$ that will drive system (1) from any initial configuration $x(t_0) = x_0$ to a given destination configuration $x(t_0 + T) = x_d$, avoiding collisions.

III. Navigation Transformation

A. Definition

The concept of the navigation transformation for a 2-D workspace is depicted in Fig. 1.

Intuitively, the navigation transformation squashes workspace obstacles into singularities in the point world, while preserving the topological information of the original workspace. Formally, the navigation transformation is defined as follows.

Definition 7: A navigation transformation is a diffeomorphism $\Phi : \mathring{\mathcal{W}} \to \mathcal{P}^n$ ($\Phi : \mathring{\mathcal{W}} \to \tilde{\mathcal{P}}^n$) that maps the interior of the workspace to a point world (with a spherical boundary).

B. Properties

The navigation transformation has some characteristics that are of interest to the field of robotic navigation.

The first property of the navigation transformation is that it provides up to a set of measure zero a global solution to the motion- and path-planning problems, with the capability to immediately identify whether an initial condition is in this set of measure zero. The existence of this set implies that there is zero probability of having an initial condition randomly assigned in this set. Note that such sets also appear in navigation functions [12], where it is shown that almost global navigation is the best that can be achieved with a smooth vector field.

Let $P_i, i \in \{1, \dots M\}$, be such that $\lim_{x \to \mathcal{O}_i} \Phi(x) = P_i$. The following result provides a solution to the path-planning problem, as stated in Definition 5.

Proposition 1: Let $q_0 \triangleq \Phi(x_0)$ and $q_d \triangleq \Phi(x_d)$. Let $\xi \in [0, 1]$. Define the line segment $\lambda(\xi) \triangleq q_0 \cdot (1 - \xi) + q_d \cdot \xi$. Then,

$h(\xi) = \Phi^{-1}(\lambda(\xi))$ is a solution to the path-planning problem, for almost all (i.e., up to a set of measure zero of) initial and destination configurations.

Proof: Since $\Phi(\cdot)$ is a diffeomorphism, it will also be a homeomorphism. Hence, Φ^{-1} exists, and the free space in the transformed domain (point world) is homeomorphically mapped to the free workspace. Since $P_i,\ i \in \{1, \ldots M\}$, is not in the range of Φ, the inverse transformation is not defined there, and any initial condition mapped in the set

$$\mathcal{Z} = \bigcup_{i \in \{1,\ldots M\}} \{z \in \mathcal{P}^n \,|\, z = P_i + (P_i - q_d) \cdot \zeta, \quad \zeta > 0\}$$

(or in the equivalent set $\tilde{\mathcal{Z}}$ for $z \in \tilde{\mathcal{P}}^n$) will fail to provide a feasible solution to the path-planning problem. However, since M is finite, and $P_i,\ i \in \{1, \ldots M\}$ are discrete points, we have that $\mu(\mathcal{Z}) = 0$, where $\mu(\cdot)$ denotes the Lebesgue measure of a set.

Remark 1: By the definition of the set of measure zero $\mathcal{Z}$ (as defined in the proof of Proposition 1), one can directly determine whether an initial position x_0 is mapped in this set, i.e., if $\Phi(x_0) \in \mathcal{Z}$.

Remark 2: Since deriving the closed-form expression of $\Phi^{-1}(\cdot)$ might not always be feasible, the solution to the path-planning problem can be obtained by evaluating the integral:

$$h(\xi) = x_0 + \int_{\tau=0}^{\xi} J_\Phi^{-1}\left(\Phi^{-1}(\lambda(\tau))\right)(q_d - q_0)\, d\tau$$

where J_Φ is the Jacobian matrix of Φ, and λ, ξ as defined in Proposition 1. A possible numerical iterative scheme could be of the form

$$h_{k+1} = h_k + J_\Phi^{-1}(h_k)(q_d - q_0)\,\Delta\xi \tag{2}$$

where $\Delta\xi$ is a suitable discretization step for ξ.

Define the destination vector in the point world as

$$\bar{d}_p(x) \triangleq \Phi(x_d) - \Phi(x). \tag{3}$$

The following result provides a solution to the motion-planning problem by backprojecting the destination vector field flows from the point world to the workspace.

Proposition 2: Let Φ be a navigation transformation. Then, system (1) under the feedback control law

$$u(x) = kJ_\Phi^{-1}(x) \cdot \bar{d}_p(x) \tag{4}$$

where k a positive scalar gain, is globally exponentially stable at x_d, almost everywhere.

Proof: Set up the following quadratic Lyapunov function candidate:

$$V(x) = \frac{1}{2}\bar{d}_p(x)^T \cdot \bar{d}_p(x).$$

Its time derivative is provided as: $\dot{V}(x) = \bar{d}_p(x)^T \cdot \dot{\bar{d}}_p(x)$. Noting that $\dot{\bar{d}}_p = -J_\Phi \dot{x}$, and substituting the control law (4), we get $\dot{V} = -k\bar{d}_p^T J_\Phi J_\Phi^{-1} \bar{d}_p$. Since Φ is a diffeomorphism, then J_Φ is nonsingular; hence, we get $\dot{V} = -2kV$, which implies global exponential stability for the system, for all the trajectories, where $\Phi(\cdot)$ is defined.

One can easily verify that the flows of system (1) under the control law (4) coincide with the paths produced by Proposition 1, since both generators $J_\Phi^{-1}\left(\Phi^{-1}(\lambda(\xi))\right) \cdot \bar{d}_p(x_0)$ and $J_\Phi^{-1}(x) \cdot \bar{d}_p(x)$ for the same initial condition x_0 will produce the same tangent space. This implies that the control law (4) will inherit the same set of measure zero, $\mathcal{Z}$, and any initial condition in this set will not be stable. However, since $\mathcal{Z}$ is a set of measure zero, the system will be globally exponentially stable almost everywhere.

Remark 3: The comments mentioned in Remark 1 apply here as well.

IV. Control Design Utilizing the Navigation Transformation

A. Time Abstraction

It is usually the case that we need to establish *a priori* the duration of a navigation task, e.g., for scheduling purposes. However, for a system that is based on the Koditschek–Rimon construction [12] of navigation functions, such a requirement would require for each initial condition the integration over the flow lines of the system. By using the navigation transformation, as we will show below, one can have a time abstraction for the motion-planning problem without the need to integrate over the flow lines.

Definition 8: A *scheduling function* is a decreasing $\mathcal{C}^1$ function $s_T : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, with $T > 0$, that satisfies

1) $s_T(0) = \left\|\bar{d}_p(x(0))\right\|$;
2) $s_T(T) = 0, \quad t \geq T$;
3) $s_T(t) > 0, \quad 0 \leq t < T$.

The scheduling function denotes the desired distance to the target in the point world at time $t \in [0, T]$. This is a generic definition for $s_T(\cdot)$, allowing flexibility on the scheduling function selection according to the scheduling task requirements. Specific selections are proposed in the experiments section. Define the destination direction vector in the point world as

$$\tilde{d}_p(x) \triangleq \begin{cases} \widehat{d}_p(x), & x \neq x_d \\ \mathbf{0}, & x = x_d. \end{cases}$$

Here, $\widehat{d}_p(x) \triangleq \frac{\bar{d}_p(x)}{\left\|\bar{d}_p(x)\right\|}$ is the unit vector of $\bar{d}_p(x)$. We have the following result that solves the time-abstraction problem by backprojecting the appropriately scaled destination direction vector field flows, from the point world to the workspace:

Proposition 3: System (1) under the control law

$$u(x,t) = J_\Phi^{-1}(x) \cdot \tilde{d}_p(x)\left(-\dot{s}_T(t) + k\left(\left\|\bar{d}_p(x)\right\| - s_T(t)\right)\right) \tag{5}$$

where k is a positive gain, and $s_T(\cdot)$ a scheduling function, provides for almost all initial conditions a *time-abstracted* solution to the motion-planning problem, with duration T.

Proof: Define the scheduling error as

$$\epsilon(t) \triangleq \left\|\bar{d}_p(x(t))\right\| - s_T(t).$$

This function quantifies whether the system is on schedule ($\epsilon = 0$), ahead of schedule ($\epsilon < 0$) or behind schedule ($\epsilon > 0$). Let us now consider the scheduling error dynamics (parentheses dropped for notational brevity):

$$\dot{\epsilon} = -\widehat{d}_p^{\,T} J_\Phi \dot{x} - \dot{s}_T.$$

Following the Filippov solution concept (due to the discontinuity of $\tilde{d}_p(x)$ at x_d), we have that $\dot{x} \in K[f](x)$, where

$$K[f](x) \triangleq \overline{co}\left\{\lim u(x_i, t) \| x_i \to x, x_i \neq x_d\right\}.$$

Away from the discontinuity, the differential inclusion is a singleton. Hence, substituting control law (5), we get

$$\dot{\epsilon} \overset{a.e.}{=} -k\left(\left\|\bar{d}_p\right\| - s_T\right) = -k\epsilon$$

for $x \neq x_d$, and for $x_0 \notin \mathcal{Z}$. Hence, $\epsilon(t) = \epsilon(0)e^{-kt}$. Due to requirement (1) of Definition 8 for $x \neq x_d$, we have that $\epsilon(0) = 0$, which implies $\epsilon(t) = 0$, that is, our system is always and for almost all initial conditions (i.e., all initial conditions up to a set of measure zero) on schedule. Now, we need to study the behavior around x_d when $t < T$. Assume $x_i \to x_d$. Since $\epsilon = 0$ from the preceding analysis, and $\bar{d}_p(x_i) \to 0$, we have that $s_T \to 0$. From properties 2 and 3 of

Definition 8 and the C^1 requirement, we have that $\dot{s}_T \to 0$. Observing control law (5), and examining the differential inclusion $K[f](x_d)$, we conclude that it contains a single element, i.e., $K[f](x_d) = \{0\}$. Moreover, due to requirement properties 2 and 3 of Definition 8, we will have $x = x_d$ only when $t := T$.

Although the analysis provided above is sufficient to support Proposition 3, it is useful to consider the stability properties of x_d under control law (5) when $t \geq T$:

Corollary 1: System (1) under control law (5) is almost everywhere exponentially stable at x_d for $t \geq T$.

Proof: Let

$$V = \frac{1}{2}\left\|\bar{d}_p\right\|^2$$

be a Lyapunov function candidate. Then, for almost all initial conditions

$$\dot{V} = -\bar{d}_p^T J_\Phi \dot{x}$$

where following the Filippov solutions concept, $\dot{x} \in K[f](x)$, where

$$K[f](x) \triangleq \overline{co}\left\{\lim u(x_i,t)|x_i \to x, x_i \neq x_d\right\}.$$

Since for $t \geq T$, we have by Definition 8 that $s_T(t) = \dot{s}_T(t) = 0$, then $K[f](x)$ contains a single element: $K[f](x) = \left\{kJ_\Phi^{-1}(x) \cdot d_p(x)\right\}$, and hence, $\dot{V} = -2kV$ recovering a.e. exponential convergence.

V. Construction of a Navigation Transformation

A navigation transformation is specified by the requirements of Definition 7. Several navigation transformation candidates and variants have been proposed by the author in previous works [17], [18]. The navigation transformation candidate that is presented and analyzed in this work is a two-step transformation that initially appeared in [16]. The proposed candidate operates on sphere-diffeomorphic n-dimensional star-worlds as these are defined in [23] and [24].

The first step of the construction is to transform the diffeomorphic-to-sphere (star-world) workspace to a sphere world. This can be achieved using, e.g., the diffeomorphism proposed in [23] and [24] to map a star-shaped- (or forest-of-stars-) world to a sphere world or using any other workspace to sphere-world diffeomorphism. Note that the construction proposed in [23] and [24] becomes a diffeomorphism after a parameter λ is appropriately tuned. However, this does not exclude the possibility of tuning-free diffeomorphisms (e.g., toward the lines of [18]). Since this step is well documented in the literature, we will assume here that

$$h_\lambda : \mathcal{W} \to \mathcal{S}^n$$

is such a diffeomorphic transformation that maps each workspace obstacle $\mathcal{O}_i$ to the ball $\mathbb{S}_i^n$ of radius r_i and centered at P_i, where $i \in \{1, \ldots M\}$ and (if such exists) the external workspace boundary $\mathcal{O}_0$ to the boundary of the external bounding ball $\mathbb{S}_0$ of $\mathcal{S}^n$. The tuning-free claim for the navigation transformation is based on the assumption of existence of a tuning-free h_λ construction.

Several assumptions are required for the following.

Assumption 1: The minimum distance between obstacle images in $\mathcal{S}^n$ is

$$\mu_a \triangleq \min_{\substack{i,j\in\{1,\ldots M\}\\ i\neq j}} \left\{\|P_i - P_j\| - (r_i + r_j)\right\} > 0.$$

Moreover, if there exists an external boundary in $\mathcal{S}^n$ centered at P_0 with radius r_0, its minimum distance from obstacle images in $\mathcal{S}^n$ is

$$\mu_0 \triangleq \min_{i\in\{1,\ldots M\}} \left\{r_0 - \|P_i - P_0\| - r_i\right\} > 0.$$

Note that existence of $h_\lambda(\cdot)$ implies Assumption 1 (see, e.g., [23] and [24]), since intersecting obstacles in $\mathcal{W}$ are mapped to a single-obstacle image in $\mathcal{S}^n$. Assumption 1, although redundant, is presented here for presentation completeness. In order to have a valid destination configuration, we need that:

Assumption 2: The minimum distance between any obstacle image in $\mathcal{S}^n$ and the image of the destination configuration $\tilde{q}_d = h_\lambda(x_d)$ in $\mathcal{S}^n$ is

$$\mu_d \triangleq \min_{i\in\{1,\ldots M\}} \left\{\|P_i - \tilde{q}_d\| - r_i\right\} > 0.$$

Now, define

$$\mu \triangleq \frac{1}{2}\min\left\{\mu_a, 2\mu_0, 2\mu_d\right\}$$

that encodes the influence region around each obstacle image in $\mathcal{S}^n$.

We may now proceed to the second step to construct a diffeomorphic transformation from a sphere world to a point world. Some preliminary constructions are required. Let $\tilde{q} = h_\lambda(x)$, where $x \in \mathcal{W}$. Define

$$b_i(\tilde{q}) \triangleq \|\tilde{q} - P_i\| - r_i, \quad i \in \{1, \ldots M\}$$

that is the distance of $\tilde{q}$ from the boundary of the ith obstacle image in $\mathcal{S}^n$. Let

$$N_i \triangleq \left\{\tilde{q} \in \mathring{\mathcal{S}}^n \,|b_i(\tilde{q}) < \mu\right\}$$

where $i \in \{1, \ldots M\}$ is the ith obstacle proximal region in $\mathcal{S}^n$. We have that:

Corollary 2: For the obstacle proximal regions, it holds that

$$N_i \bigcap N_j = \emptyset, \quad i \neq j, \quad i,j \in \{1, \ldots M\}.$$

Proof: This can be seen by noting that μ is at most half the distance between any two obstacle images in $\mathcal{S}^n$ (we will henceforth refer to obstacle images as obstacles in $\mathcal{S}^n$). Since N_i is open, the intersection on any two obstacle proximal regions will be the empty set.

Define the obstacle proximal region as

$$\mathcal{N} \triangleq \bigcup_{i\in\{1,\ldots M\}} N_i.$$

Define the set away from the spherical obstacles as

$$\mathcal{F} \triangleq \mathring{\mathcal{S}}^n - \mathcal{N}.$$

The region $\mathcal{F}$ away from the obstacles and the obstacle proximal region $\mathcal{N}$ form a partition of $\mathring{\mathcal{S}}^n$.

Define the smooth function

$$\sigma(x) \triangleq \begin{cases} e^{-1/x}, & x > 0 \\ 0, & x \leq 0. \end{cases}$$

Define the smooth switch function

$$\eta(x,\delta) \triangleq \frac{\sigma(x)}{\sigma(x) + \sigma(\delta - x)}.$$

Define the function

$$s(x,\delta) \triangleq \frac{x}{\delta}\left(1 - \eta(x,\delta)\right) + \eta(x,\delta).$$

Now, we can define a contraction-like transformation:

$$v_i(\tilde{q}) \triangleq \left[1 - s\left(b_i(\tilde{q}), \mu\right)\right](P_i - \tilde{q}).$$

Finally, for the transformation from a sphere world to the point world, the following transformation is proposed:

$$T(\tilde{q}) \triangleq \mathrm{id}(\tilde{q}) + \sum_{i=1}^{M} v_i(\tilde{q}). \tag{6}$$

Let us now proceed with the analysis of the proposed transformation. We have that:

Corollary 3: $T : \mathring{\mathcal{S}}^n \to \tilde{\mathcal{P}}^n$ is a diffeomorphism in $\mathcal{F} \subset \mathring{\mathcal{S}}^n$.

Proof: Let $f \in \mathcal{F}$. Since $s(b_i(f), \mu) = 1$, $i \in \{1, \ldots M\}$, $T(\cdot)$ reduces to the identity transformation, i.e., $T(f) = f$, which is a diffeomorphism.

Before proceeding, we will need the following result:

Corollary 4: $s(x, \delta)$ is a strictly increasing function of x for $x \in [0, \delta)$

Proof: Since $\eta(x, \delta) \in [0, 1)$ for $x \in [0, \delta)$, we have that

$$\begin{aligned}\frac{\partial s(x,\delta)}{\partial x} &= \frac{1}{\delta}(1-\eta(x,\delta)) + \frac{\partial \eta(x,\delta)}{\partial x}\left(1-\frac{x}{\delta}\right) \\ &> \frac{\partial \eta(x,\delta)}{\partial x}\left(1-\frac{x}{\delta}\right).\end{aligned}$$

The term in parentheses is positive for $x \in [0, \delta)$, and noting that

$$\frac{\partial \sigma(x)}{\partial x} = \frac{1}{x^2}\sigma(x), \quad \frac{\partial \sigma(\delta - x)}{\partial x} = -\frac{1}{(d-x)^2}\sigma(\delta - x)$$

we have

$$\frac{\partial \eta(x,\delta)}{\partial x} = \frac{\frac{1}{x^2}\sigma(x)\sigma(\delta-x) + \frac{1}{(\delta-x)^2}\sigma(x)\sigma(\delta-x)}{(\sigma(x)+\sigma(\delta-x))^2} > 0.$$

Hence, for $x \in [0, \delta)$, we have that $\frac{\partial s(x,\delta)}{\partial x} > 0$.

We are now in place to state the following.

Lemma 1: $T : \mathring{\mathcal{S}}^n \to \tilde{\mathcal{P}}^n$ is a diffeomorphism in $\mathcal{N} \subset \mathring{\mathcal{S}}^n$.

Proof: We will prove this by showing first that T is a bijection, and then, that the Jacobian of T is nondegenerate. Assume a point $\tilde{q} \in \mathcal{N}$. Since the $N_j, j \in \{1, \ldots M\}$ subsets of $\mathcal{N}$ are disjoined according to Corollary 2, we assume without loss of generality that $\tilde{q} \in N_i$ for some $i \in \{1, \ldots M\}$. Then, $v_j = 0$ for $j \neq i$, and the transformation becomes

$$T(\tilde{q}) = \tilde{q} + v_i(q).$$

Write $\tilde{q}$ as

$$\tilde{q} = \tilde{q}(r, \hat{u}) = P_i + (r_i + r)\hat{u}$$

where $\hat{u} \triangleq \frac{\tilde{q}-P_i}{\|\tilde{q}-P_i\|}$, and $r \in (0, \mu)$ is the distance of $\tilde{q}$ from the boundary of the obstacle image i. Then

$$b_i(\tilde{q}) = r$$

and

$$T(\tilde{q}(r, \hat{u})) = (r_i + r)\, s(r, \mu)\, \hat{u} + P_i. \tag{7}$$

In order to prove that $T(\cdot)$ is a bijection, it is sufficient to establish that r and $\hat{u}$ are functions of T. According to Corollary 4, $s(r, \mu)$ is an increasing function of r and, so, will be the multiplication with a positive increasing function of r, i.e., the function $\mathcal{K}(r) \triangleq (r_i + r)\, s(r, \mu)$ will be an increasing function. Hence, the inverse $\mathcal{K}^{-1}(\cdot)$ will also be a function. Hence, from (7), r can be recovered as

$$r = \mathcal{K}^{-1}(\|T - P_i\|)$$

and also, $\hat{u}$ can be recovered as

$$\hat{u} = \frac{T - P_i}{\|T - P_i\|}.$$

Hence, $T^{-1}(\cdot)$ is also a function, which implies the bijection property of T.

In order to show that the Jacobian matrix of the transformation is nonsingular, we will show that each eigenvalue is nonzero. To this extend, let $\mathcal{P} \triangleq \mathrm{span}\{\hat{u}\}$ and $\mathcal{P}^{\perp} \triangleq \mathrm{span}\{\hat{\mathbb{V}}_1, \ldots \hat{\mathbb{V}}_{n-1}\}$ such that $\mathbb{R}^n \triangleq \mathcal{P} \oplus \mathcal{P}^{\perp}$ and $B \triangleq [\hat{u}, \hat{\mathbb{V}}_1, \ldots \hat{\mathbb{V}}_{n-1}]$ is an orthonormal basis of $\mathbb{R}^n$. A vector in $v \in \mathcal{S}^n \subset \mathbb{R}^n$ using the basis B can be written as $v_B = [r, s_1, \ldots s_{n-1}]^T$. The transformation T can be written as

$$T(r, s_1, \ldots s_{n-1}) = \mathcal{K}(r)\, \hat{u} + P_i.$$

Observe that since $\hat{u}$ is a unit vector, its differential will live in $\mathcal{P}^{\perp}$. More specifically, $d\hat{u} = \hat{\mathbb{V}}_1 ds_1 + \ldots + \hat{\mathbb{V}}_{n-1} ds_{n-1}$. Taking the total differential of T, we get

$$dT = \frac{\partial \mathcal{K}(r)}{\partial r}\hat{u}\, dr + \mathcal{K}(r)\sum_{i=1}^{n-1}\hat{\mathbb{V}}_i ds_i.$$

The last equation implies that the Jacobian $J_T^{(B)}$ of T in the B basis can be written as

$$J_T^{(B)} = \begin{bmatrix} \frac{\partial \mathcal{K}(r)}{\partial r} & 0 & \ldots & 0 \\ 0 & \mathcal{K}(r) & \ldots & 0 \\ & & & \\ 0 & \ldots & \ddots & 0 \\ 0 & \ldots & 0 & \mathcal{K}(r) \end{bmatrix}.$$

Since $\mathcal{K}(r) > 0$, and $\frac{\partial \mathcal{K}(r)}{\partial r} > 0$ for $r > 0$, the Jacobian will be nonsingular in $\mathcal{N} \subset \mathring{\mathcal{S}}^n$, and the proof is complete.

Combining the previous results, we have the following.

Proposition 4: $T : \mathring{\mathcal{S}}^n \to \tilde{\mathcal{P}}^n$ is a diffeomorphism.

Proof: By Corollary 3 and by Lemma 1, $T(\cdot)$ is a diffeomorphism in $\mathcal{F}$ and in $\mathcal{N}$, respectively. Moreover, $\mathcal{F}$ and $\mathcal{N}$ are a partition of $\mathring{\mathcal{S}}^n$, and since by construction, $T(\cdot)$ is smooth everywhere in $\mathring{\mathcal{S}}^n$, then $T : \mathring{\mathcal{S}}^n \to \tilde{\mathcal{P}}^n$ is a diffeomorphism.

Remark 4: By following the same analysis, the conclusion of Proposition 4 can be directly extended to $T : \mathring{\mathcal{S}}^n \to \mathcal{P}^n$.

Since by construction $h_\lambda(\partial \mathcal{W}) = \partial \mathcal{S}^n$, by restricting the domain of $h_\lambda(\cdot)$ to the interior of the workspace, we have $\mathring{h}_\lambda : \mathring{\mathcal{W}} \to \mathring{\mathcal{S}}^n$. We hereby propose the following construction of a navigation transformation candidate:

$$\Phi(x) = \left(T \circ \mathring{h}_\lambda\right)(x). \tag{8}$$

We may now state our main result.

Proposition 5: The transformation $\Phi : \mathring{\mathcal{W}} \to \tilde{\mathcal{P}}^n$ is a navigation transformation.

Proof: From Proposition 4, we have that $T : \mathring{\mathcal{S}}^n \to \tilde{\mathcal{P}}^n$ is a diffeomorphism; hence, the composition $\Phi(x) = (T \circ \mathring{h}_\lambda)(x)$ will also be a diffeomorphism, since $\mathring{h}_\lambda(\cdot)$ is a diffeomorphism. Noting, moreover, that $\lim_{x \to \partial \mathbb{S}_i^n} T(x) = P_i$, where $i \in \{1, \ldots M\}$, all the requirements of Definition 7 are satisfied.

Remark 5: By following the same analysis, the conclusion of Propositions 5 and Proposition 4 can be directly extended to $\Phi : \mathring{\mathcal{W}} \to \mathcal{P}^n$

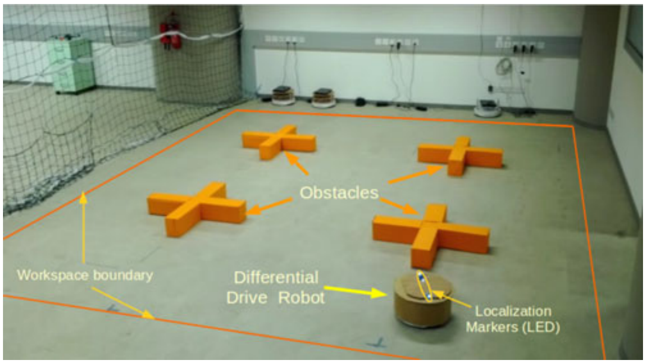

Fig. 2. Experimental setup. A star-shaped workspace with four obstacles and a differential drive robot.

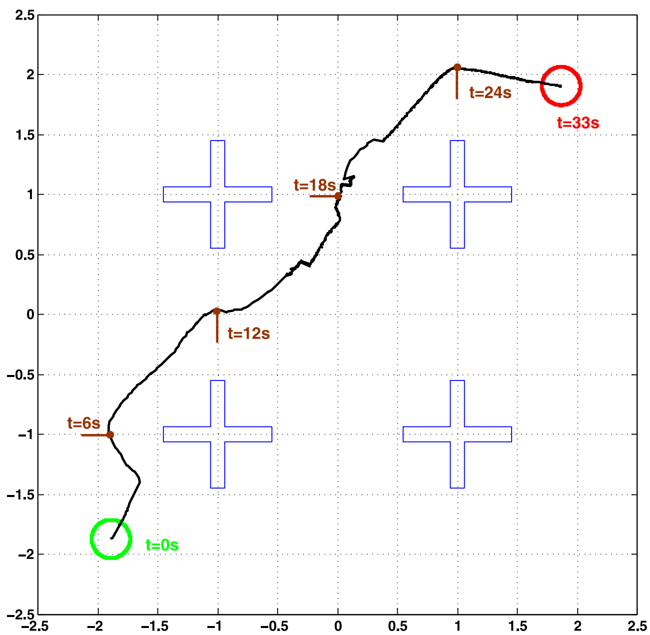

Fig. 3. Case study 1: Robot trajectory under control law (4). Green and red circles denote initial and destination configurations, respectively. Visitation times shown along the trajectory.

VI. Experiment Results

A. Setup

In order to validate the effectiveness of the proposed methodology, a set of experiments were carried out. The experimental setup is depicted in Fig. 2. The robot is a differential drive Turtlebot [31] variant based on the i-Robot Roomba 531 fitted with three LED markers for localization. Two ceiling mounted GigE cameras were used to determine the posture of the robot. All the software development (vision-based localization and robot controllers) was performed in C++ on the Ubuntu Linux OS, under the Robotic Operating System (ROS Fuerte) framework.

An Ubuntu 12.04 Linux server (Intel i7 quad-core) computer was responsible for the localization service utilizing information only from the two overhead cameras, and with accuracy of at least 5 cm. The controllers were realized on a networked (via WiFi 802.11n) client Ubuntu Linux PC (Intel Atom dual-core) that was installed on the Turtlebot. The workspace was populated with four star-shaped obstacles, as depicted in Fig. 2. A square boundary was considered around the limits of the experimentation area, as depicted in Fig. 2. The volume of the robot

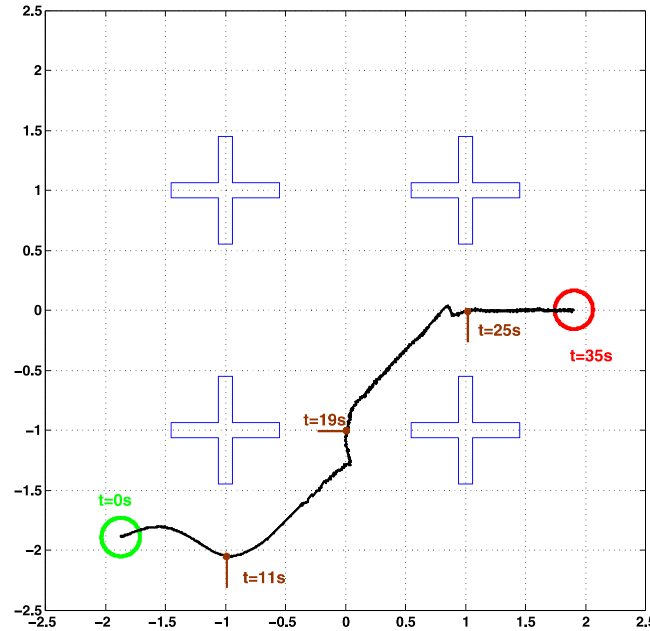

Fig. 4. Case study 2: Robot trajectory under control law (5). Green and red circles denote initial and destination configurations, respectively. Visitation times shown along the trajectory.

was accounted for by considering the grown obstacles [14]. The transformation $h_\lambda(\cdot)$ was performed as described in [23] and [24] to yield an S^2 world with boundary, that was subsequently fed to the $T(\cdot)$ transformation to yield the navigation transformation $\Phi(\cdot)$. Obstacle positions were considered *a priori* knowledge and were coded in the software. Due to the nonholonomic nature of the differential drive robot and the robot hardware limitations regarding its actuation capabilities, a feedback linearization controller with input saturation, as the one mentioned in [19], was implemented to map the derived control actions to the robot hardware. Note that analytical consideration of issues pertaining to the experimental implementation like kineto-dynamic constraints, localization uncertainties, and communication delays are beyond the scope of the current work, hence not specifically tackled in the experiment beyond the provisions mentioned above, in order to better demonstrate the system performance under nontrivial conditions.

B. Case Study 1

In the first case study, control law (4) provided in Proposition 2 was experimentally evaluated. The video of the experiment is provided in the multimedia attachment. The resulting path is depicted in Fig. 3.

As can be seen, control law (4) successfully solves the motion-planning problem.

C. Case Study 2

In the second case study, control law (5) provided in Proposition 3 was experimentally evaluated. The video of the experiment is provided in the multimedia attachment. A sinusoidal scheduling function was chosen as

$$s_T(t) := \left\|\bar{d}_p(x(0))\right\| \left[\frac{\cos(\frac{t\pi}{T})+1}{2}\right]. \tag{9}$$

The task duration was set at $T = 35$ s. The resulting path is depicted in Fig. 4. Fig. 5 depicts the distance to the destination versus time for the experiment.

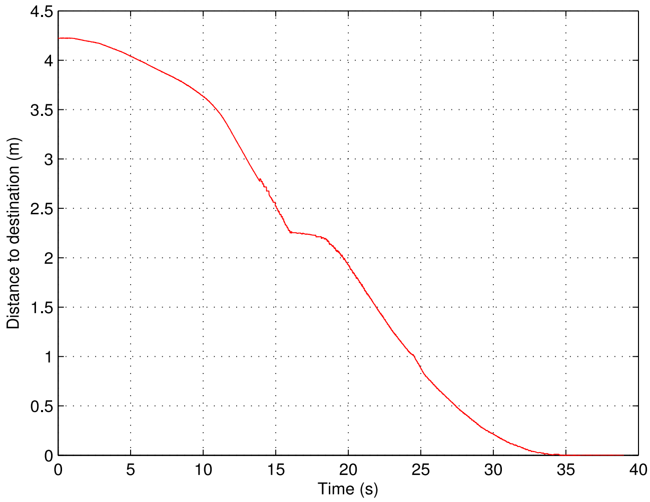

Fig. 5. Case study 2: Distance to destination versus time under control law (5).

As can be seen, control law (5) successfully solves the motion-planning and time-abstraction problems.

VII. Conclusion

In this work, a novel solution to the navigation problem has been proposed. Navigation is accomplished by diffeomorphically mapping the workspace to a domain called the *point world*, wherein the navigation problem admits a trivial solution. The solution to the original navigation problem is then recovered by transforming the trivial solution in the point world, back to the initial domain. It is shown that this approach provides a solution to the path-planning problem, through an iterative scheme, but most importantly, it provides a solution to the motion-planning problem by means of an exponentially converging feedback controller. Due to the trivial nature of the solution in the point world, a time abstraction of the navigation task can be achieved by means of a "time-abstracting" controller. Time abstractions of the navigation problem are important for scheduling purposes and for motion task planning where one needs to satisfy spatiotemporal constraints of motion tasks. Another important aspect of the proposed technique is that in contrast to the navigation-function-based techniques, the proposed technique is tuning-free conditioned upon the existence of such a tuning-free h_λ construction.

A navigation transformation candidate was proposed, and a rigorous analysis of its properties has been provided. The candidate is built on top of the existing star-world-to-sphere-world transformations to map any valid star world to a point world. The price one pays for the navigation transformation is that initial and destination configurations cannot reside on the workspace's (internal) boundary and initial conditions cannot reside in sets of measure zero, since the navigation vector field will grow unbounded. The issue of boundedness of the navigation vector field under a navigation transformation is treated in [15].

A series of experimental case studies on a nontrivial setup have been provided along with a multimedia attachment. The experimental results prove the validity, effectiveness, and robustness of the proposed solution in real-time performance. The applicability of the methodology is established through experiments on hardware with limited capabilities regarding computation (on the client's side), localization accuracy, communication delays, limited actuation capabilities, and nonholonomic kinematic constraints.

Further research includes developing a single-step transformation Φ that can transform a workspace directly to a point world that is tuning free (the h_λ transformation proposed in [23] and [24] needs tuning of the λ parameter) and, also, research into extensions of the developed technique to appropriate topologies in order to account for the robot's orientation, extensions to more complicated dynamics, moving obstacles, multirobot navigation, and exploitation of the developed methodology in spatiotemporal symbolic motion task planners.

References

[1] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [Grand Challenges of Robotics]," *IEEE, Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, Mar. 2007.

[2] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 864–874, Oct. 2005.

[3] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2689–2696.

[4] William Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry, Revised, Volume 120 (Pure and Applied Mathematics)*, 2nd ed. New York, NY, USA: Academic, 2002.

[5] N. J. Cowan, "Navigation functions on cross product spaces," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1297–1302, Jul. 2007.

[6] M. D. Crossley, *Essential Topology*. London, U.K.: Springer-Verlag, 2005.

[7] D. V. Dimarogonas, "Sufficient conditions for decentralized potential functions based controllers using canonical vector fields," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2621–2626, Oct. 2012.

[8] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.

[9] I. Filippidis and K. J. Kyriakopoulos, "Navigation functions for focally admissible surfaces," in *Proc. Amer. Control Conf.*, Washington, DC, USA, Jun. 17–19, 2013, pp. 994–999.

[10] E. A. Gol and C. Belta, "Time-constrained temporal logic control of multi-affine systems," *Proc. IFAC Conf. Anal. Des. Hybrid Syst.*, vol. 45, no. 9, pp. 102–107, 2012.

[11] M. Guo, J. Tumova, and D. V. Dimarogonas, "Hybrid control of multi-agent systems under local temporal tasks and relative-distance constraints," in *Proc. 54th IEEE Conf. Decision Control*, 2015, pp. 1701–1706.

[12] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, pp. 412–442, 1990.

[13] J. M. Krause and S. C. Vestal, "Automatic scheduling challenges in guidance, navigation and control," in *Proc. IEEE Symp. Comput.-Aided Control Syst. Des.*, 1992, pp. 293–300.

[14] J. C. Latombe., *Robot Motion Planning*. Norwell, MA, USA: Kluwer, 1991.

[15] S. G. Loizou, "Closed form navigation functions based on harmonic potentials," in *Proc. 50th IEEE Conf. Decision Control/Eur. Control Conf.*, 2011, pp. 6361–6366.

[16] S. G. Loizou, "The navigation transformation: Point worlds, time abstractions and towards tuning-free navigation," in *Proc. 19th Mediterranean Conf. Control Autom.*, 2011, pp. 303–308.

[17] S. G. Loizou, "Navigation functions in topologically complex 3-D workspaces," in *Proc. Amer. Control Conf.*, 2012, pp. 4861–4866.

[18] S. G. Loizou, "The multi-agent navigation transformation: Tuning-free multi-robot navigation," in *Proc. Robot.: Sci. Syst. Conf.*, Berkeley, CA, USA, Jul. 2014.

[19] S. G. Loizou and V. Kumar, "Mixed initiative control of autonomous vehicles," *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1431–1436.

[20] S. G. Loizou and K. J. Kyriakopoulos, "Navigation of multiple kinematically constrained robots," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 221–231, Feb. 2008.

[21] F. Mastrogiovanni, A. Paikan, and A. Sgorbissa, "Semantic-aware real-time scheduling in robotics," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 118–135, Feb. 2013.

[22] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI Commun.*, vol. 29, no. 1, pp. 151–162, 2015.

[23] E. Rimon and D. E. Koditschek, “The construction of analytic diffeomorphisms for exact robot navigation on star worlds,” *Trans. Amer. Math. Soc.*, vol. 327, no. 1, pp. 71–115, 1991.
[24] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
[25] E. Rimon and D. E. Koditschek, “Exact robot navigation in geometrically complicated but topologically simple spaces,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, vol. 3, pp. 1937–1942.
[26] C. Santos and M. Ferreira, “Timed trajectory generation using dynamical systems: Application to a puma arm,” *Robot. Auton. Syst.*, vol. 57, no. 2, pp. 182–193, Feb. 2009.
[27] C. P. Santos, “Generating timed trajectories for an autonomous vehicle: A non-linear dynamical systems approach,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 3741–3746.
[28] H. G. Tanner and A. Boddu, “Multiagent navigation functions revisited,” *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1346–1359, Dec. 2012.
[29] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, “Nonholonomic navigation and control of cooperating mobile manipulators,” *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Feb. 2003.
[30] M. Tuma, I. Iossifidis, and G. Schöner, “Temporal stabilization of discrete movement in variable environments: An attractor dynamics approach,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 863–868.
[31] 2016. [Online]. Available: http://www.turtlebot.com