

Locally Computable Navigation Functions for Sphere Worlds*

Grigoris Lionis, Xanthi Papageorgiou and Kostas J. Kyriakopoulos

Abstract—In this paper we present a new Navigation Function for a sphere world that can be computed locally with limited knowledge of the environment. By requiring smooth and not analytic NF, the effect of each obstacle is exactly nullified outside a sensing zone around the obstacle (the only required parameter is the width of the sensing zone). This allows the computation of the navigation function using information from a single obstacle each time. We present simulations to verify the validity of this approach.

I. INTRODUCTION

Robot Motion Planning is a problem tightly integrated with the very nature of what robotics really is. A large part of robotic problems have to cope with a point agent having to move from one point to another, avoiding collisions with a static or dynamic environment. The point agent could be a physical agent, that moves in a variety of workspaces, or a much more complex multi physical agent system.

The robot motion planning problem has been attacked by a variety of methods, ranging from artificial intelligence heuristics and discrete logic approximations, to differential geometric methods. Excellent introduction in this field are among others the books of Latombe [1], and LaValle [2].

We will focus our attention to a closed-loop solution to the motion planning problem, by defining a special function on the workspace, called navigation function that completely solves the closed-loop motion planning problem. Navigation Functions have been from their very introduction to the literature, a great tool that allows an elegant and compact solution to the problem of robotic navigation. The concept of a NF is to find a vector field over the free space that steers the agent towards the target and repels it from the obstacles. NF's were introduced by Rimon and Koditschek in their seminal paper [3] as an exact way of solving the navigation problem in a fully known workspace eliminating the problem of local minima. On the other hand, potential fields, were introduced

by Khatib [4] as a low-level obstacle avoidance schema that used local sensing, preventing the robot from colliding to nearby obstacles. NF have been used in high dimensional spaces, for navigating non-holonomic vehicles and for multi agent navigation problems [5], [6].

Moreover, a number of NF based control schemes that require only local knowledge, have been presented in the literature. In [7] the authors present a closed loop navigation based scheme for multiple agents, that requires limited sensing capabilities for the agents, while in [8] the authors utilize navigation functions that require local knowledge, to stabilize a group of agents into a specific formation on the plane while avoiding static point obstacles. In this work, we present a NF scheme, that handles static, non point obstacles and contrary to most literature work, does not require sophisticated parameter tuning, other than the trivial adjustment of the minimum obstacle separation.

Rimon and Koditschek require strong conditions on their functions, and as a result the process of generating the Navigation Function is relatively complex, while it is not possible to utilize their technique if the workspace is not completely known. Mathematically the Rimon and Koditschek NF do not have to be more than double differentiable, as reported in their work, but the NF they propose is analytic. The analytic requirement results in non-local NF, i.e. in NF that cannot be used in real-time. Given an analytic function f defined over a subset of R^n , by expand using a Taylor series the value of f in an arbitrary point p , we can compute the value of f over the complete set, just by knowing how it behaves locally in p . Thus, there is an intrinsic non-locality associated with a analytic NF. This non-locality does global knowledge of the configuration. Moreover, most of the NF implementations found in the literature need special tuning in order to mathematically guarantee the Navigation properties, and this special tuning requires -most of the times- some global knowledge.

In this paper we present a new class on NF's that try to circumvent some of these problems. We drop the analytic requirement, and we define a smooth function that solves locally the navigation problem for sphere worlds.

The rest of the paper is organized as follows: Section II formally states the problem, while in section III we briefly review the N.F. methodology. Sections IV and V describe and analyses the proposed navigation function respectively, while section VI is devoted to simulations.

*This work is partially supported by the European Commission through contract "FP6 - IST - 001917 - NEUROBOTICS: The fusion of Neuroscience and Robotics", contract "FP6 IST 2002 507006 ISWARM: Intelligent Small World Autonomous Robots for Micro-Manipulation", by Eugenides Foundation Scholarship, and by the Greek State Scholarship Institute, IKY.

G. Lionis is a PhD Student in the Mechanical Engineering Department, National Technical University of Athens, Greece, glion@mail.ntua.gr

X. Papageorgiou is a PhD Student in the Mechanical Engineering Department, National Technical University of Athens, Greece, xpapag@mail.ntua.gr

K.J. Kyriakopoulos is with the Faculty of Mechanical Engineering, National Technical University of Athens, Greece, kkyria@mail.ntua.gr

II. PROBLEM STATEMENT

The general problem of robot motion planning, in an informal way, is the following. We have a workspace $\mathcal{W} \in \mathbb{R}^n$. The coordinates of the robot are a point $p \in \mathcal{W}$. Inside the workspace there is a number of obstacles $\mathcal{O}_i \subset \mathcal{W}$. The free space is defined as $\mathcal{F} = \mathcal{W} - \bigcup \mathcal{O}_i$. The navigation problem is to find a continuous function $T : [0, 1] \rightarrow \mathcal{W}$ steering the robot from $q_{initial} \in \mathcal{F}$ to $q_{final} \in \mathcal{F}$ that is $T(0) = q_{initial}, T(1) = q_{final}, T(t) \in \mathcal{F}, t \in [0, 1]$.

We are specifically interested in constructing a feedback control scheme, so that the trajectories of the closed loop system are continuous functions that solve the motion planning problem.

We limit the definition of the navigation problem to a subset of \mathbb{R}^n to avoid topological considerations.

We will limit the problem somehow, following closely the work of Rimon and Koditschek, by focusing on sphere worlds. Here, both W and O_i are hyperspheres defined on \mathbb{R}^n , and moreover we will assume that obstacle spheres do not intersect $\mathcal{O}_i \cap \mathcal{O}_j = \emptyset$ and that the obstacles are contained in the interior of the workspace $cl(\mathcal{W}) \cap \mathcal{O}_i = \emptyset$.

One could argue that this is an oversimplified terrain for navigation, but by utilizing a series of transformations, as in [9], a sphere world represents an abstracted model, of the actual realistic workspace. Having said that, we will concentrate on the sphere worlds that serve as the archetypal obstacle world, in the context of the Navigation Function methodology.

III. A LITTLE ON THE N.F. METHODOLOGY

We will discuss a little bit on the concept of a NF, and the usefulness of utilizing a NF in an actual motion planning problem. Given a workspace, populated with obstacles, a NF, is, informally, a real function defined on the workspace, that:

- is 0 -minimum value- at the target configuration
- is 1 -maximum value- on the boundary of the free space
- does not have any local minima on the workspace

These properties of the NF, allow us -if we have constructed a NF on the particular workspace, for the particular goal configuration- to “automatically” solve an infinite number of motion planning problems in this workspace. By following the negated gradient of a N.F., the agent converges to the goal configuration, avoiding all the obstacles, from almost any initial configuration, within the free space. There is a set of initial configurations -with zero measure- that steer to the saddle points that the N.F. necessarily -if it is continuous- possess. Rimon and Koditschek, in their seminal papers, established a number of properties of these functions on general manifolds, the most useful of which is that the NF property of a function is invariant under diffeomorphisms. This property, allows us, through a process proposed by Rimon and Koditschek, to reduce the motion planning problem posed in an initial realistic workspace, to an equivalent problem, in a sphere world, a spherical subset of an Euclidean space, populated by spherical objects that do not “touch” each other. The clearance between the spheres is

equivalent with the gap between the obstacles of the original space.

Thus the problem of navigating in the real workspace, is decomposed in to two parts:

- map diffeomorphically the original workspace to a sphere world
- find a NF in the sphere world

Upon doing this, a closed-loop control scheme for steering in the original workspace will have been derived.

Here, we examine only the construction of a NF in the sphere world. Our contribution is a NF on a sphere world, that can be easily computed, and with partial knowledge of the environment. In a number of NF proposed in the literature, carefully tuning of the NF was required, with a process that needed as input full knowledge of the workspace. Our approach circumvents this obstacle by defining a NF that does not require the number of obstacles in the workspace as an input.

This is a step towards utilizing the NF methodology to an actual real-time motion planner, which is both provably correct, closed loop and computationally tractable.

Of course, the other part of the NF construction is extremely interesting, if one tries to solve in step-wise, local, real-time manner. The robot has to be able to build a map of the environment and extract from the map the connectivity properties of the obstacles. Much work remains to be done in this respect.

IV. THE CONSTRUCTION OF NAVIGATION FUNCTION

While in the traditional definition of a NF, the NF field is manipulated by using an exponent k [3] in such a way as to decouple the obstacles from each other, in our approach the NF field is constructed in a decoupled way from the beginning. The field of each obstacle does not affect the fields from the other obstacles. In this way, in each point of the workspace, the robot sees exactly either one or zero obstacles. This attribute of our NF allows the robot to compute the NF using only local information. The authors in [3], have defined a navigation function $\varphi(q)$ on a compact connected analytic manifold with boundary, $\mathcal{F} \subset \mathbb{R}^n$, in the interior of which there is a target point q_d . The navigation function is a map $\varphi : \mathcal{F} \rightarrow [0, 1]$ if it is:

- 1) smooth on \mathcal{F} (at least a $C^{(2)}$ function)
- 2) polar on \mathcal{F} , (q_d is a unique minimum)
- 3) admissible on \mathcal{F} , (uniformly maximal on $\partial\mathcal{F}$)
- 4) a Morse function, (its critical points are nondegenerate)

The navigation function that we propose in this work has the following form:

$$\varphi(q) \triangleq \frac{\gamma_d(q)}{\gamma_d(q) + \beta(q)} \quad (1)$$

where $\gamma_d(q) : \mathcal{F} \rightarrow [0, \infty)$ is a distance to the goal, defined as

$$\gamma_d(q) = \|q - q_d\|^2 \quad (2)$$

taking the value zero only when the agent reach its destination point $q = q_d$, where $\|\cdot\|$ is the Euclidean norm.

The other component function of the proposed navigation function (1), is $\beta(q) : \mathcal{F} \rightarrow [0, 1]$, which is a function that vanishes only in case of the agent collides with an obstacle. In the original work of [3], $\beta(q)$ function was defined as the product of several ‘‘obstacle functions’’, $\beta(q) = \prod_i \beta_i$, with β_i is vanishing when the agent is in contact with one of the environmental obstacles.

In this work we present a $\beta(q)$ function, which is defined again, as the product of several ‘‘obstacle functions’’, as in [3], but each of obstacle function β_i has some more specific properties.

Let $B_\epsilon(\alpha)$ be an open ball of center α and radius ϵ . It is shown in [10], that there is a C^∞ function $g(x)$ on \mathbb{R}^n , which is positive on $B_{\epsilon+\rho}(x_0)$, identically 1 on $B_\rho(x_0)$, and 0 outside $B_{\epsilon+\rho}(x_0)$, where ρ, ϵ are positive constants, and x_0 is the radius. We have defined the function $h(t)$ of the form:

$$h(t) = \begin{cases} e^{-\epsilon/t}, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (3)$$

which is C^∞ , since we can prove by directly computation that all of its derivatives exist and are zero at $t = 0$, and since it is analytic for other values of t .

We define the obstacle function β_i that has the properties of **i)** vanishing on obstacle’s boundary, **ii)** increasing as the agent is moving away from the obstacle and **iii)** is identically 1, when the agent reach a predefined distance ϵ_i from the obstacle. Our obstacle function is the following

$$\beta_i(q) = \frac{h(\|q - q_i\| - \rho_i)}{h(\|q - q_i\| - \rho_i) + h((\epsilon_i + \rho_i) - \|q - q_i\|)} \quad (4)$$

where q_i is the center and ρ_i the radius of i^{th} obstacle, and ϵ_i a positive constant. It is easy to check that this is a C^∞ function.

V. PROOF OF CORRECTNESS

A. 2D Sphere World

In this section we will formally show that the function φ , constructed in the previous section is indeed a navigation function. Let the workspace \mathcal{W} with obstacles $\mathcal{O} = \bigcup_i \mathcal{O}_i$, where $i = 1, \dots, M$, with M the number of obstacles. Furthermore, define the *free space*, as the space remains after removing all the obstacles from the workspace $\mathcal{F} \triangleq \mathcal{W} - \mathcal{O}$. Let $\epsilon > 0$, define $\mathcal{B}_i(\epsilon) \triangleq \{q \mid 0 < \beta_i < \epsilon\}$. We have partition the free space into following five subsets:

- 1) The destination point: $\mathcal{F}_d \triangleq \{q \mid \gamma_d(q) = 0\}$
- 2) The free space boundary: $\partial\mathcal{F} \triangleq \beta^{-1}(0)$
- 3) The set ‘‘near the obstacles’’: $\mathcal{F}_0(\epsilon) \triangleq \bigcup_i \mathcal{B}_i(\epsilon) \setminus \mathcal{F}_d$
- 4) The set ‘‘near the workspace boundary’’: $\mathcal{F}_1(\epsilon) \triangleq \mathcal{B}(\epsilon) \setminus (\mathcal{F}_d \cup \mathcal{F}_0(\epsilon))$
- 5) The set ‘‘away from the obstacles’’: $\mathcal{F}_2(\epsilon) \triangleq \mathcal{F} \setminus (\mathcal{F}_d \cup \partial\mathcal{F} \cup \mathcal{F}_0(\epsilon) \cup \mathcal{F}_1(\epsilon) \cup \mathcal{F}_2(\epsilon))$

The following technical lemma, which gives formulas for the Hessian of a rational function at a critical point, will be very useful in the sequel of the proof.

Lemma 1: Let ν, δ be at least twice differentiable, and define $\rho \triangleq \frac{\nu}{\delta}$. At a critical point c of ρ , [11]

$$\nabla^2 \rho|_c = \frac{1}{\delta^2} [\delta \nabla^2 \nu - \nu \nabla^2 \delta]$$

Firstly, we will show that the goal configuration q_d is a non-degenerate local minimum of the navigation function.

Proposition 1: If the workspace is valid, the destination point, q_d , is a non-degenerate local minimum of φ .

Proof: The gradient of function φ of (1) is

$$\nabla \varphi(q_d) = \frac{1}{(\gamma_d + \beta)^2} [(\gamma_d + \beta) \nabla \gamma_d - \gamma_d \nabla (\gamma_d + \beta)]|_{q_d} = 0 \quad (5)$$

since, both γ_d and $\nabla \gamma_d$ vanish at q_d . The Hessian of φ at q_d , using Lemma 1, will be

$$\nabla^2 \varphi = \frac{1}{(\gamma_d + \beta)^2} [(\gamma_d + \beta) \nabla^2 \gamma_d - \gamma_d \nabla^2 (\gamma_d + \beta)]|_{q_d} \quad (6)$$

Since, γ_d vanish at q_d and it holds that $\nabla^2 \gamma_d = 2I$, we have that

$$\nabla^2 \varphi = 2\beta^{-1}I$$

which implies that q_d is a non-degenerate local minimum of φ . ■

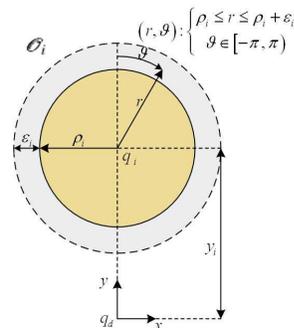


Fig. 1. Coordinate frame around one obstacle.

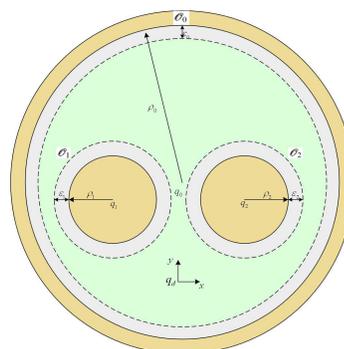


Fig. 2. Navigation Function around 2 obstacles.

We can assume by using coordinate transformation, that the obstacle is located (the center of obstacle) at $q_i = (0, y_i)$, $y_i > 0$. We define a new coordinate system, using polar coordinates, as depicted in Fig. 1, with r measuring the distance from the obstacle’s center, and with ϑ measuring the

angle associated with the point. The coordinate transformation is given by

$$\begin{aligned} x &= r \sin \vartheta \\ y &= y_i + r \cos \vartheta \end{aligned} \quad (7)$$

Function γ_d becomes

$$\gamma_d = y_i^2 + r^2 + 2ry_i \cos \vartheta \quad (8)$$

While function β becomes

$$\beta_i = \frac{h(r - \rho_i)}{h(r - \rho_i) + h(\epsilon_i + \rho_i - r)} \quad (9)$$

The gradient of γ_d is computed as

$$\nabla \gamma_d = \begin{bmatrix} 2r + 2y_i \cos \vartheta & -2y_i r \sin \vartheta \end{bmatrix}^T \quad (10)$$

and the Hessian of γ_d can be computed as

$$\nabla^2 \gamma_d = \begin{bmatrix} 2 & -2y_i \sin \vartheta \\ -2y_i \sin \vartheta & -2y_i r \cos \vartheta \end{bmatrix} \quad (11)$$

Furthermore, the gradient and the Hessian of β_i are respectively:

$$\nabla \beta_i = \begin{bmatrix} \frac{\partial \beta_i}{\partial r} & 0 \end{bmatrix}^T \quad (12)$$

$$\nabla^2 \beta_i = \begin{bmatrix} \frac{\partial^2 \beta_i}{\partial r^2} & 0 \\ 0 & 0 \end{bmatrix} \quad (13)$$

In order to compute the derivatives of β_i w.r.t r , it is convenient to introduce the following parameters:

$$h_1 = h(r - \rho_i) \quad , \quad h_2 = h(\epsilon_i + \rho_i - r) \quad (14)$$

By (9), we have that $\beta_i = \frac{h_1}{h_1 + h_2}$, and therefore, we can compute its derivatives:

$$\frac{\partial \beta_i}{\partial r} = \beta_r = \beta_i(1 - \beta_i) \left[\frac{\epsilon_i}{(r - \rho_i)^2} + \frac{\epsilon_i}{(\rho_i + \epsilon_i - r)^2} \right] \quad (15)$$

$$\frac{\partial^2 \beta_i}{\partial r^2} = \beta_i(1 - \beta_i)(1 - 2\beta_i) \left[\frac{\epsilon_i}{(r - \rho_i)^2} + \frac{\epsilon_i}{(\rho_i + \epsilon_i - r)^2} \right]^2 +$$

$$2\beta_i(1 - \beta_i) \left[\frac{\epsilon_i}{(\rho_i + \epsilon_i - r)^3} - \frac{\epsilon_i}{(r - \rho_i)^3} \right] = \beta_{rr} \quad (16)$$

Lemma 2: For every $\rho_i > 0$ there exists a positive ϵ_i s.t. the distance of all critical points associated with i^{th} obstacle is greater than three quarters of the boundary zone, i.e. $q \in C_{\varphi_i} \Rightarrow r < \rho_i + 3/4\epsilon_i$, where C_{φ_i} is the set of all critical points associated with obstacle i .

Proof: Obviously, due to the fact that the gradient of γ_d points to the target, and the gradient of β_i is radial, the critical point will always be at $\vartheta = 0$. So, in a critical point, from (5), the following equation holds:

$$\beta_i \nabla \gamma_d = \gamma_d \nabla \beta_i \quad (17)$$

By taking norms of the two sides, and using (8), (10), and (12), we have that

$$2\beta_i(r + y_i) = (r + y_i)^2 \beta_r \quad , \quad \vartheta = 0$$

So at a critical point we have that, $\frac{\beta_r}{\beta_i} = \frac{2}{r + y_i}$. But $\frac{2}{r + y_i} < \frac{1}{\rho_i}$, as $r > \rho_i$, and $y_i > \rho_i$.

It suffice to show that $\frac{\beta_r}{\beta_i} > \frac{1}{\rho_i}$, for $r < \rho_i + 3/4\epsilon_i$.

$$\frac{\beta_r}{\beta_i} = (1 - \beta_i) \left[\frac{\epsilon_i}{(r - \rho_i)^2} + \frac{\epsilon_i}{(\rho_i + \epsilon_i - r)^2} \right]$$

It holds, by direct computation, that $(1 - \beta_i) \geq 0.065$ for $r \leq \rho_i + 3/4\epsilon_i$, so

$$\frac{\beta_r}{\beta_i} \geq 0.065 \frac{\epsilon_i}{(r - \rho_i)^2} \geq \frac{0.11}{\epsilon_i}$$

Thus, by choosing $\epsilon_i < 0.11\rho_i$, all critical points are on the quarter of the boundary zone. ■

Lemma 3: For $\epsilon_i > 0 \Rightarrow \beta_i > (r - \rho_i)/\epsilon_i$, $r > \rho_i + \epsilon_i/2$.

Proof: Let $w = r - \rho_i$

$$\beta_i > w/\epsilon_i \Leftrightarrow \frac{e^{-\epsilon_i/w}}{e^{-\epsilon_i/w} + e^{-\epsilon_i/(\epsilon_i - w)}} > w/\epsilon_i \Leftrightarrow$$

$$e^{-\epsilon_i/w} > w e^{-\epsilon_i/w} / \epsilon_i + w e^{-\epsilon_i/(\epsilon_i - w)} / \epsilon_i \Leftrightarrow$$

$$e^{-\epsilon_i/w} (1 - w/\epsilon_i) > w e^{-\epsilon_i/(\epsilon_i - w)} / \epsilon_i \Leftrightarrow$$

$$e^{-(\epsilon_i/w) + (\epsilon_i/(\epsilon_i - w))} > w/(\epsilon_i - w)$$

We set as $p = \epsilon_i/(\epsilon_i - w)$ and $q = \epsilon_i/w$ and we have that

$$\beta_i > (r - \rho_i)/\epsilon_i \Leftrightarrow e^{p-q} > p/q \Leftrightarrow e^p/p > e^q/q$$

Since $w > \epsilon_i/2$ we have that $\epsilon_i - w < \epsilon_i/2 < w$ so $p > q$. We check the monotonicity of $f(v) = e^v/v$. We have that $f'(v) = e^v/v - e^v/v^2 = e^v(1/v - 1/v^2)$. For $v > 1$, we have that $f'(v) > 0$ and therefore $p > q \Leftrightarrow e^p/p > e^q/q$. $v > 1 \Leftrightarrow q > 1 \Leftrightarrow \epsilon_i/w > 1 \Leftrightarrow w < \epsilon_i \Leftrightarrow r - \rho_i < \epsilon_i$, which is always true. ■

Lemma 4: For $a, b > 0$, and $a - b > 0$, $a > \frac{3+\sqrt{5}}{2}b \Rightarrow (b - a) \left(\frac{1}{a^2} + \frac{1}{b^2} \right)^2 + 2 \left(\frac{1}{b^3} - \frac{1}{a^3} \right) < 0$.

Proof: We have that

$$\begin{aligned} (b - a) \left(\frac{1}{a^2} + \frac{1}{b^2} \right)^2 + 2 \left(\frac{1}{b^3} - \frac{1}{a^3} \right) &= \\ \frac{(b-a)(a^2+b^2)^2}{a^4b^4} + \frac{2(a^3-b^3)}{a^3b^3} &= \\ \frac{(b-a)(a^2+b^2)^2 + 2ab(a^3-b^3)}{a^4b^4} \end{aligned}$$

Let now take the numerator N :

$$\begin{aligned} N &= (b - a)(a^2 + b^2)^2 + 2ab(a - b)(a^2 + ab + b^2) = \\ (b - a)[a^4 + b^4 + 2a^2b^2 - 2ab(a^2 + ab + b^2)] &= \\ (b - a)[a^4 + b^4 - 2a^3b - 2ab^3] \end{aligned}$$

Let $A = a^4 + b^4 - 2a^3b - 2ab^3$, then we have that:

$$\begin{aligned} A &= (a - b)^4 + 2a^3b + 2ab^3 - 6a^2b^2 = \\ (a - b)^4 + 2ab(a^2 - 3ab + b^2) \end{aligned}$$

Therefore, it holds that $N = (b - a)A < 0$, since $a > \frac{3+\sqrt{5}}{2}b$ and according to the other predefined conditions, and consequently, the proof is completed. ■

Proposition 2: For every $\epsilon_i < 0.11\rho_i$, and $r > \rho_i + 3/4\epsilon_i \Rightarrow \beta_{rr} < 0$.

Proof: From (16) it suffices to show:

$$\begin{aligned} L(r) &= (1 - 2\beta_i)\epsilon_i \left[\frac{1}{(r - \rho_i)^2} + \frac{1}{(\rho_i + \epsilon_i - r)^2} \right]^2 + \\ 2 \left[\frac{1}{(\rho_i + \epsilon_i - r)^3} - \frac{1}{(r - \rho_i)^3} \right] &< 0 \end{aligned}$$

since $0 < \beta_i < 1$, for $r \in (\rho_i, \rho_i + \epsilon_i)$.

We have that $\beta_i > (r - \rho_i)/\epsilon_i$ according to Lemma 3, therefore, $(1 - 2\beta_i) \leq (2\rho_i + \epsilon_i - 2r)/\epsilon_i, \forall r > \rho_i + 3\epsilon_i/4$. Thus, we have that:

$$L(r) \leq (2\rho_i + \epsilon_i - 2r) \left[\frac{1}{(r-\rho_i)^2} + \frac{1}{(\rho_i+\epsilon_i-r)^2} \right]^2 + 2 \left[\frac{1}{(\rho_i+\epsilon_i-r)^3} - \frac{1}{(r-\rho_i)^3} \right]$$

Assuming that $r - \rho_i = a$ and $\rho_i + \epsilon_i - r = b$, and according to Lemma 4, based on the conditions $a - b = 2r - 2\rho_i - \epsilon_i > 0$, and $a > \frac{3+\sqrt{5}}{2}b$, it holds that $L(r) < 0$. The last condition is proved as follows:

$$a > \frac{3+\sqrt{5}}{2}b \Rightarrow 2a > (3 + \sqrt{5})b \Rightarrow r - \rho_i > \frac{3+\sqrt{5}}{5+\sqrt{5}}\epsilon_i$$

which is true since $r - \rho_i > 3/4 > \frac{3+\sqrt{5}}{5+\sqrt{5}}\epsilon_i$. ■

Finally, with the next proposition it is shown that the navigation function φ is a Morse function (it is ensured that the critical points are saddles).

Proposition 3: For $\epsilon_i < 0.11\rho_i$, all the critical points of φ are non-degenerate.

Proof: According to (6), we can take the numerator, and by using (11), and (13), we have that:

$$\begin{aligned} & (\beta_i \nabla^2 \gamma_d - \gamma_d \nabla^2 \beta_i)|_{C_\varphi} = \\ & \beta_i|_{C_\varphi} \begin{bmatrix} 2 & 0 \\ 0 & -2y_i r \end{bmatrix} - \gamma_d|_{C_\varphi} \begin{bmatrix} \beta_{rr} & 0 \\ 0 & 0 \end{bmatrix} = \\ & \begin{bmatrix} 2\beta_i|_{C_\varphi} - \gamma_d|_{C_\varphi} \beta_{rr} & 0 \\ 0 & -2\beta_i|_{C_\varphi} y_i r \end{bmatrix} \end{aligned}$$

It is a diagonal matrix. Since we are at a critical point, and since $\epsilon_i < 0.11\rho_i$ the critical point lies on the outer quarter of the boundary zone, according to Lemma 2. Since the critical point lies on the outer quarter of the boundary zone, according to Proposition 2. Since $\gamma_d > 0$, and $\beta_i > 0$, we conclude that $2\beta_i|_{C_\varphi} - \gamma_d|_{C_\varphi} \beta_{rr} > 0$. Moreover, since $\beta_i > 0, y_i > 0, r > 0$ we conclude that

$$-2\beta_i|_{C_\varphi} \cdot y_i \cdot r < 0$$

Thus, the Hessian of ϕ at the critical point is a diagonal matrix, with one positive and one negative element. The eigenvalues of this matrix are therefore positive and negative. Therefore, the critical point is not a local minima, but a saddle point. ■

B. Multi-Dimensional Sphere Worlds

The proof outlined above shows that the function constructed is indeed NF for a plane sphere world (a world defined on a subset of \mathbb{R}^2). We can easily generalize this result for a \mathbb{R}^n sphere world. The general form of the function defined remain as is, with the norms now being defined on \mathbb{R}^n . As the influence of each obstacle is only local, by appropriately choosing the depth of each individual obstacle field (ϵ_i) we can construct a NF such that the robot will only be influenced by either exactly one or exactly zero obstacles. Therefore, it suffices to check the NF properties for checking a single obstacle.

We will first show that the proposed control law, the motion of the agent -while traversing a single obstacle- is

bound on a single hyperplane of the workspace, the hyperplane defined by the goal configuration, the position of the agent and the center of the obstacle.

We use hyper-spherical coordinates, introducing the following coordinate transformation

$$\begin{aligned} x_1 &= r \cos \phi_1 \\ &\dots \\ x_n &= r \sin \phi_1 \dots \sin \phi_{n-2} \sin \phi_{n-1} \end{aligned}$$

The center of the obstacle is located at $p_c = [0 \ 0 \ \dots \ 0]^T$ and that the goal configuration is at $p_d = [-d \ 0 \ \dots \ 0]^T$. Thus the goal hyperspherical coordinates are $p_d = [-d \ 0 \ \dots \ 0]^T$. Obviously function β_i is a function of coordinate r . We will examine the behavior of γ . Let us examine function γ in a random point $p \in \mathbb{R}^n$ with hyperspherical coordinates $p = [r \ \phi_1 \ \dots \ \phi_{n-1}]^T$

We have that $\gamma^2(p) = (r \cos \phi_1 + d)^2 + (r \sin \phi_1 \cos \phi_2)^2 + \dots + (r \sin \phi_1 \dots \sin \phi_{n-2} \cos \phi_{n-1})^2 + (r \sin \phi_1 \dots \sin \phi_{n-2} \sin \phi_{n-1})^2$. It is not hard to see that the two last terms of the second part of the equality can be combined using the basic trigonometric equality, and the recursively the final result will be

$$\begin{aligned} \gamma_d &= (r \cos \phi_1 + d)^2 + (r \sin \phi_1)^2 \rightarrow \\ \gamma_d &= d^2 + r^2 + 2dr \cos \phi_1 \end{aligned}$$

In plain words, both functions that comprise the NF φ , namely the goal function γ and the obstacle function β are functions only of r and ϕ_1 which are by construction the coordinates of the plane generated by the goal configuration, the center of the obstacle and of the current point.

Since the motion of the robot is along the gradient of the NF, and since the gradient is zero along $\phi_i, i \neq 1$, the robot does not move along $\phi_i, i \neq 1$ but only along r and ϕ_1 . Moreover, in this plane, function φ is exactly equal with the NF that would have been produced if the robot was bound on this plane. Therefore, function φ is a NF in \mathbb{R}^n , since φ is in essence locally 2D.

VI. SIMULATION RESULTS

To validate our approach, we present some simulated results, that clearly show how our proposed NF can handle a very large number of obstacles, for a spherical world. In the simulation, the radius of all obstacle is identical, but this is for simplicity in the programming of the algorithm. The methodology can handle spherical obstacle of arbitrary radius.

The workspace is considered to be disk-like, and the robotic agent is considered omni-directional.

Fig. 3 presents a relatively small world, populated by ~ 50 obstacles. The initial configuration of the robot is at the upper right corner, and the final configuration at the lower left corner. This simulation clearly depicts the local behavior of our NF. The system “slides along” each obstacle it perceives in its immediate neighborhood. No global knowledge of the workspace is required, except the clearance between any two

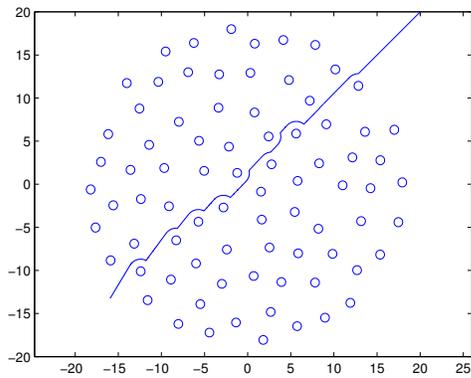


Fig. 3. Navigating around ~ 50 obstacles.

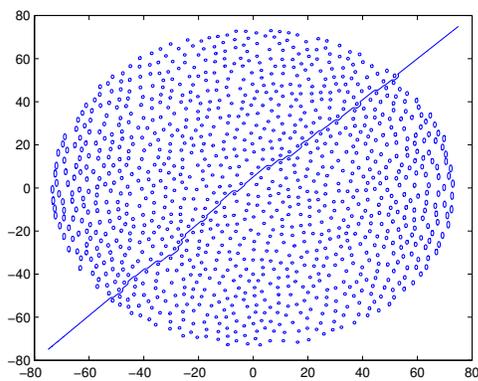


Fig. 4. Navigating in a ~ 1100 obstacle workspace.

obstacles. We are currently working on showing that real-time update of the clearance is possible.

Fig. 4 presents a simulation navigational task in a large workspace comprised of ~ 1000 obstacles. Due to the local nature of the NF, the computational complexity of the NF is $O(1)$, for each time step.

VII. CONCLUSIONS AND FUTURE WORKS

In this work we examined a class of navigation functions, that have the potential of resolving some difficulties of the original NF approach. By moving away from analytic function we derived a class of NF that are exactly locally computable, do not need a careful parameter tuning and are smooth -hence can be used in most navigation tasks. We exploited the nature of smooth functions -which can be exactly in one set and exactly zero in another set which is disjoint from the intimal- to construct a NF which has by construction exactly one saddle point per obstacle without the of special tuning to place the saddle points appropriately. Moreover, the computations needed -at each point- are $O(1)$, since the number of the obstacles is irrelevant. This NF can be used with completely local knowledge of the environment, and can be use as such for navigation with real-time obstacle avoidance. Obviously, the computational complexity is $O(1)$

only for defining a NF in the sphere world, and the computational complexity of generating a real-time diffeomorphism between the original workspace, and a sphere world, has not been assessed.

Moreover, the original constructions by Rimon and Koditschek for transforming a real obstacle workspace to a spherical workspace can also be used with these NF. Of course, in this case the topology of the obstacle space has to be know in order to apply the tree-world transformations. An extremely interesting research topic is a real-time algorithm for constructing the real-world to sphere world transformations based on local knowledge. This algorithm, used in conjunction with a local NF will result in a real time, provably correct, navigation scheme.

Another interesting research direction, is to utilize other kind of local NF, which can have more controlled properties. One problem with the Navigation Function proposed in this work, is that near the obstacles the field becomes very steep, making it difficult for the robot to follow the v.f. An interesting problem is to extend this NF, to others that limit the steepness of the NF.

ACKNOWLEDGMENT

We would like to thank all the anonymous reviewers for their valuable and insightful comments.

REFERENCES

- [1] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [2] S. M. LaValle, *Panning Algorithms*. Cambridge University Press, 2006.
- [3] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots." *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [5] D. Dimarogonas, S. Loizou, K. Kyriakopoulos, and M. Zavlanos, "Decentralized feedback stabilization and collision avoidance of multiple agents," NTUA, <http://users.ntua.gr/ddimar/TechRep0401.pdf>, Tech. Report, 2004.
- [6] H. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," *2005 IEEE International Conference on Robotics and Automation*, 2005.
- [7] D. V. Dimarogonas, K. J. Kyriakopoulos, and D. Theodorakatos, "Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.
- [8] H. G. Tanner and A. Kumar, "Formation stabilization of multiple agents using decentralized navigation functions," *Robotics: Science and Systems I*, p. pp 4956, 2005.
- [9] E. Rimon and D. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Trans. of the American Mathematical Society*, vol. 327, no. 1, pp. 71–115, September 1991.
- [10] W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 1986.
- [11] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances Appl. Math.*, vol. 11, pp. 412–442, 1990.