Towards Locally Computable Polynomial Navigation Functions for Convex Obstacle Workspaces

Grigoris Lionis, Xanthi Papageorgiou and Kostas J. Kyriakopoulos

Abstract— **In this paper we present a polynomial Navigation Function (NF) for a sphere world that can be constructed almost locally, with partial knowledge of the environment. The presented navigation function is C^2 and as a result the computational complexity is very low, while the construction uses local knowledge and information. Moreover, an almost locally computable diffeomorphism between convex obstacles and spheres is presented, allowing the NF scheme to be used in a workspace populated by convex obstacles. Our approach is not strictly local in the ϵ sense, i.e. the field around a point is not influenced only by an ϵ region around the point, but rather it is local in the sense that the NF around each obstacle is influenced only by the obstacle and the adjacent obstacles. In particular, we require, in the vicinity of an obstacle, the distance between the obstacle and the adjacent obstacles. Simulations are presented to verify this approach.**

I. Introduction

A rather large portion of robotics problems are, in essence motion planning problems, cast in the form of a point agent moving in a static or dynamic environment while simultaneously avoiding collisions. The agent is either a robot, a physical agent moving in a 2-D or 3-D workspace, or it represents a much more complicated system moving in a complex environment.

A variety of different methods have been used for the robot motion planning problem, ranging from artificial intelligence, discrete motion planning, differential geometric methods to methods arising from topological considerations. Excellent introductions to the field are, among others, the books of Latombe [1], and LaValle [2].

The method we focus on is a closed-loop approach, where trajectory generation and trajectory tracking are fused together. The approach is based on defining a special function on the workspace, called a navigation function (NF). The core concept of an NF is to establish a vector field on the workspace, or more precisely on the free space (actually we seek a potential function that generates this vector field) that steers the agent from any initial position to a specified point of the workspace, the goal configuration, while on the same time avoids collisions with the obstacles. NF's were introduced by Rimon and Koditschek in a series of papers [3] as an exact way of solving the navigation problem in a fully known workspace eliminating the problem of local minima, and offer a very elegant solution to the problem of robotic navigation. Potential function for robotic navigation have been introduced earlier in the past by Khatib [4] as a low-level obstacle avoidance scheme that used local sensing, preventing the robot from colliding to nearby obstacles, but in these initial approaches, the problem of local minima prevailed. NF, which by construction do not have any local minima, have been used for solving a variety of robot motion problems, including among others high dimensional spaces, non-holonomic vehicles, multi agent navigation, manipulator control etc., [5], [6], [7].

Besides NF with global information,NF based control schemes that require only local knowledge have been presented in the literature. In [8] a closed loop navigation scheme for multiple agents, that requires limited sensing capabilities for the agents is presented, while in [9] 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. A main difference is that our NF does not need a tedious adjustment of a tuning parameter, required for most NF based schemes, and as a result the practicality of this approach is enhanced.

Rimon and Koditschek in their original work, required strong analytic conditions for their functions, even though technically only C^2 function are required. As a result the process of generating this function is relatively complex, while it is not possible to utilize their construction if the workspace is not completely know. The analytic requirement results a global definition of NF, that cannot be used in real-time given lack of global information. To understand this, consider a Taylor expansion around any point. The convergence of the Taylor expansion, means that information for the complete domain exists in any point of the NF, therefore it is not local.

In this work, we extend our previous approach, [10]. We present a new NF, following the same design, but being simply C^2 instead of being smooth. This new approach, leads to more efficient form of the NF, as the function is now a polynomial instead of a complex exponential function, allowing easier computations and tuning. Moreover, some of the geometric constraints have been relaxed. Finally, in this work, we present a scheme for mapping diffeomorphically

This research project is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%). Also, 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

Proc. IEEE Int'l Conf. on Robotics & Automation, (ICRA-2008), Pasadena, CA, USA, May 19-23, 2008, p.p. 3725-3730

a workspace populated with convex obstacles to a sphere world, while keeping the computation of the diffeomorphism local. This allows our local NF scheme to be used in a workspace with convex obstacles.

II. Problem Statement

We have a workspace $\mathcal{W} \in \mathbb{R}^n$. The coordinates[1] 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 seek a feedback control law, such that, the trajectories of the closed loop system are continuous functions that solve the motion planning problem, i.e. steer the robot to the goal while avoiding collisions.

We start, 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 $cl(\mathcal{W}) \cap \mathcal{O}_i = \emptyset$.

Then we extent this result in a workspace populated with arbitrary convex obstacles, by utilizing a series of transformations, as in [11], a sphere world represents an abstracted model, of the actual realistic workspace.

III. A Glimpse of the NF Methodology

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.

By following the negated gradient of a NF, 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 lead to the saddle points of the NF. The existence of the saddle points and of the set of initial configurations leading to these saddles is a necessity for any continuous NF. Rimon and Koditschek, in their seminal papers, established a number of properties of these functions, the most useful of which is that the NF properties of a function are invariant under diffeomorphisms. This property, allows us, 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 to 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.

In this work, we first present a locally computable NF for a sphere world. The novelty of this work (compared to our previous result) is a new NF on a sphere world, that it is a rational function and can thus be exactly computed with minimal computational cost, while for its construction only partial knowledge of the environment is required, in the sense that the robot needs information only for the obstacles in its immediate vicinity. Our approach has the added benefit that trivializes the tuning of an NF.

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

Finally, we present a local C^2 diffeomorphism from a world populated with convex obstacles, to a sphere world. By doing so, we establish a closed loop control scheme for solving the motion planning problem in a workspace populated by convex obstacles. This is a crucial step towards applying this methodology to more realistic environments.

IV. The Construction of Navigation Function

In the original definition of a NF, the field is manipulated by using an exponent k [3] to decouple the obstacles from each other. This exponent, in effect, makes the obstacle field resembling -when close to an obstacle- to the field produced by a single obstacle, so that all "problematic" points of the NF field are manipulated to become saddle. This approach is the most widely used approach to establishing NF properties. Our approach lies in constructing the NF field 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 every point of the workspace, the robot can see exactly either one or zero obstacles. This attribute of our NF allows the robot to compute the NF using only local information. Following, [3], we can define 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,as a map $\varphi : \mathcal{F} \rightarrow [0, 1]$ with the following properties:

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).

We start with a navigation function defined on a sphere world, i.e. on a workspace populated with spherical obstacles. A sphere world has the benefit of grasping all the topological and navigation properties of a workspace with arbitrarily shaped, distinct non overlapping obstacles. The local transformation from the convex shaped obstacle world, to the sphere world is presented in another section. For the sphere world, we propose the following navigation function:

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

[1] We stick subsets of $\mathbb{R}^n$ to avoid topological considerations.

where $\gamma_d(q) : \mathcal{F} \to [0, \infty)$, is a distance to the goal, defined as $\gamma_d(q) = \|q - q_d\|^2$, 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 $\beta(q) : \mathcal{F} \to [0, 1]$, which is a function that vanishes only in case of the agent collides with an obstacle. In [3], $\beta(q)$ function was defined as the product of several "obstacle functions", $\beta(q) = \prod \beta_i$, with β_i is vanishing when the agent is in contact with one of the environmental obstacles.

Following their line of reasoning, we construct function $\beta(q)$, as the product of several "obstacle functions", one for each obstacle. Prior to defining $\beta(q)$ we give some auxiliary definitions.

Definition 1: We construct the function

$$P(x) = a_n \cdot x^n + \ldots + a_1 \cdot x \tag{2}$$

which n is odd, and $x \in [0, 1]$. The odd coefficients of the polynomial are arbitrary positive number, $a_n, a_{n-2}, \ldots, a_1 > 0$, that sum up to one, $a_n + a_{n-2} + \ldots + a_3 = 1$. The even coefficients are negative and are chooser as follows $a_{n-1} = -\frac{n}{n-2} \cdot a_n$, $\ldots$, $a_2 = -3 \cdot a_3$, and $a_1 = n \cdot \left(\frac{n-1}{n-2} - 1\right) \cdot a_n + \ldots + 3 \cdot a_3$.

The exact form of this function depends on the choice of n and on the distribution of the odd parameters $a_n, a_{n-2}, \ldots, a_3$.

***Lemma** 1:* For any P as in (2), $P'(x) > 0$, $\forall x \in [0, 1)$.

***Lemma** 2:* For all $x \in (0, 1)$, $P(x) > 0$.

***Lemma** 3:* $P''(1) = 0$.

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 $\mathcal{O} = \bigcup \mathcal{O}_i$, where $i = 1, \ldots, M$, with M the number of obstacles. Furthermore, define the *free space*, as the space remains after removing all the obstacles from the workspace $\mathcal{F} \triangleq \mathcal{W} - \mathcal{O}$.

***Lemma** 4:* Let ν, δ be at least twice differentiable, and define $\rho \triangleq \frac{\nu}{\delta}$. At a critical point c of ρ, [12]

$$\nabla^2 \rho \mid_c = \frac{1}{\delta^2} \left[\delta \nabla^2 \nu - \nu \nabla^2 \delta \right]$$

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.

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

$$\begin{aligned} x &= r \sin \vartheta \\ y &= y_i + r \cos \vartheta \end{aligned} \tag{3}$$

Function γ_d becomes

$$\gamma_d = y_i^2 + r^2 + 2 r y_i \cos \vartheta \tag{4}$$

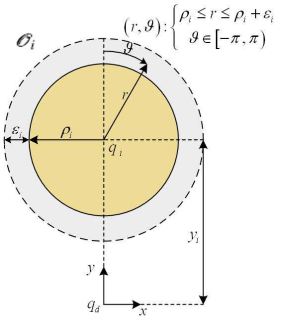
$(r,\vartheta) : \begin{cases} \rho_i \le r \le \rho_i + \varepsilon_i \\ \vartheta \in [-\pi, \pi) \end{cases}$
$\mathcal{O}_i$
ϑ
r
ε_i
ρ_i
q_i
y_i
y
q_d
x

Fig. 1. Coordinate frame around one obstacle.

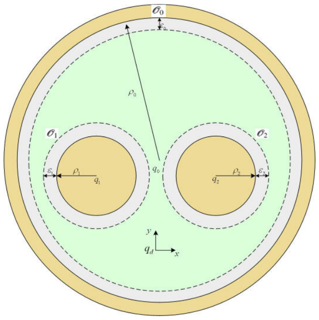

Fig. 2. Navigation Function around two obstacles.

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

$$\beta_i = \begin{cases} P(z_i) & , \ \rho_i \le r \le \rho_i + \varepsilon_i \\ 1 & , \ r > \rho_i + \varepsilon_i \end{cases}$$

where $z_i = \frac{r - \rho_i}{\varepsilon_i}$ (it holds that, for $r = \rho_i$, $z_i = 0$, and for $r = \rho_i + \varepsilon_i$, $z_1 = 1$), ρ_i is the radius of i^{th} obstacle, and ε_i is a positive constant. This function is smooth because $P(1) = 1$ (Lemma 2), and C^2 since $P'(1) = P''(1) = 0$ (Lemma 1, 3, respectively).

The gradient of γ_d is computed as $\nabla \gamma_d = \begin{bmatrix} 2r + 2y_i \cos \vartheta & -2 y_i r \sin \vartheta \end{bmatrix}^T$ and the Hessian of γ_d can be computed as

$$\nabla^2 \gamma_d = \begin{bmatrix} 2 & -2 y_i \sin \vartheta \\ -2 y_i \sin \vartheta & -2 y_i r \cos \vartheta \end{bmatrix} \tag{5}$$

Furthermore, the gradient and the Hessian of β_i are respectively $\nabla \beta_i = \begin{bmatrix} \frac{\partial \beta_i}{\partial r} & 0 \end{bmatrix}^T$ and

$$\nabla^2 \beta_i = \begin{bmatrix} \frac{\partial^2 \beta_i}{\partial r^2} & 0 \\ 0 & 0 \end{bmatrix} \tag{6}$$

We can compute the derivatives of β_i:

$$\frac{\partial \beta_i}{\partial r} = \beta_r = \begin{cases} P'(z_i) & , \ \rho_i \le r \le \rho_i + \varepsilon_i \\ 0 & , \ r > \rho_i + \varepsilon_i \end{cases}$$

$$\frac{\partial^2 \beta_i}{\partial r^2} = \beta_{rr} = \begin{cases} P''(z_i) & , \ \rho_i \le r \le \rho_i + \varepsilon_i \\ 0 & , \ r > \rho_i + \varepsilon_i \end{cases}$$

where

$$P'(z_i) = \frac{1}{\varepsilon_i} \cdot \ n \cdot a_n \cdot z_i^{n-1} + \ldots + 2 \cdot a_2 \cdot z + a_1$$

$$P''(z_i) = \frac{1}{\varepsilon_i^2} \cdot \ n \cdot (n-1) \cdot a_n \cdot z_i^{n-2} + \ldots + 6 \cdot a_3 \cdot z_i + 2 \cdot a_2$$

In the next proposition we show that the navigation function φ is a Morse function (all critical points are saddles).

***Proposition** 2:* For every $\varepsilon_i > 0$, all the critical points of φ are non-degenerate.

B. Multi-Dimensional Sphere Worlds

As shown 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 the same, 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 $x_1 = r\cos\phi_1, \ldots, x_n = r\sin\phi_1 ... \sin\phi_{n-2}\sin\phi_{n-1}$. The center of the obstacle is located at $p_c = [0\ 0\ ...\ 0]^T$ and that the goal configuration is at $p_d = [-d\ \ 0...\ \ 0]^T$. Thus the goal hyperspherical coordinates are $p_d = [-d\ 0\ ...\ 0]^T$. Obviously, the function β_i is a function of coordinate r. We examine the behavior of γ. Function γ in a random point $p \in \mathbb{R}^n$ with hyperspherical coordinates $p = [r\ \phi_1\ ...\ \phi_{n-1}]^T$.

We have that $\gamma^2(p) = (r\cos\phi_1 + d)^2 + (r\sin\phi_1\cos\phi_2)^2 + \ ...\ + (r\sin\phi_1\ ...\ \sin\phi_{n-2}\cos\phi_{n-1})^2 + (r\sin\phi_1\ \ ...\ \ \sin\phi_{n-2}\sin\phi_{n-1})^2$. 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 recursively the final result will be $\gamma_d = (r\cos\phi_1 + d)^2 + (r\sin\phi_1)^2 \rightarrow \gamma_d = d^2 + r^2 + 2dr\cos\phi_1$. Thus, 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. Convex obstacle workspace

In this section we will show how the NF defined above can be used in the case of a workspace populated with convex obstacles. We will stick to the 2-D case and we will assume that the obstacle set is also smooth.

The main idea is to transform each convex set to a circle, to calculate the NF on the transformed workspace, and then to compute the NF on the original workspace. This approach is based on the approach suggested by Rimon and Koditschek in their work [11] and can work, as the navigation function properties are preserved under a C^2 diffeomorphism. So, the approach, depicted in Fig. 3 works in the following way

- Around each obstacle, a patch of the workspace is chosen, so that the workspace is divided into patches that do not intersect, and that each contains a single obstacle.
- A diffeomorphism is calculated that transforms each patch, to patch with the same outer boundary, but in which the inner boundary -corresponding to the obstacle- becomes a circle.
- Outside the patch, the diffeomorphism becomes the identity map.
- The composition of all these diffeomorphisms defines a global diffeomorphism that renders the convex obstacle workspace to a sphere world.
- This diffeomorphism can be computed *locally*.
- The complete construction is a C^2 diffeomorphism, ensuring that the NF properties are inherited from the sphere world to the actual workspace

A. Derivation of the diffeomorphism

We will assume that obstacle $\mathcal{O}_i$ is described by function $b_i(q)$ with the following properties $b_i(q) = 1$ for $q \in \partial\mathcal{O}_i$, $b_i(q) > 0$ for $q \notin \mathcal{O}_i$, and $\frac{\partial b}{\partial r} \leq 1$. One final assumption, is that the sets $\beta_i(q) = \mu$, for $\mu > 1$ corresponds to a closed line, having the same shape as the internal obstacle (Fig. 3). We will use for this part only polar coordinates, and as we stick to the 2-D case, $q = [\rho\ \theta]^T$. The patch around the obstacle that will be transformed is defined by the equation $\mathcal{P}_i = \{q : b_i q \leq \mu_i\}, \mu_i > 0$. Our goal is to transform $\mathcal{P}_i$ into a set with the same outer boundary, but where the internal boundary -i.e. the obstacle- will be a circle. In order to do so, we introduce the real valued function

$$S(x) = \begin{cases} 1,\ x \leq 0 \\ -6\cdot x^5 + 15\cdot x^4 - 10\cdot x^3 + 1,\ 0 < x < 1 \\ 0,\ x \geq 1 \end{cases}$$

We can easily verify S is C^2 everywhere in $\mathbb{R}$. This function acts as a smooth switch, between 0 and 1. We assume that we know the radius ρ of a circle that can be inscribed into the obstacle. The following function transforms the boundary of the obstacle to a circle of radius ρ

$$\tau_i : \begin{bmatrix} r \\ \theta \end{bmatrix} \rightarrow \begin{bmatrix} b\cdot\rho \\ \theta \end{bmatrix}$$

, while the rest of the workspace is also shrunk. The complete transformation is defined as

$$T_i(r,\theta) = \begin{matrix} T^1 \\ T^2 \end{matrix} = \begin{matrix} S(\frac{b-\mu_1}{\mu-\mu_1})b\cdot\rho + (1 - S(\frac{b-\mu_1}{\mu-\mu_1}))\cdot r \\ \theta \end{matrix}$$

where μ_1 is a positive constant $\mu > \mu_1 > 1$ that represents a slice of $\mathcal{P}_i$ in which the shrinking transformation "fades out" to the identity map. The rationale behind this transformation is to shrink the obstacle diffeomorphically to a disk, to shrink the part of $\mathcal{P}_i$ that is outside $\mathcal{O}_i$ and to leave untouched the outer border of $\mathcal{P}_i$ where the transformation becomes (in a C^2 way) equal to the unity map. In this way, only the "patch" of the workspace containing the obstacle is transformed, while the rest of the workspace remains the same. Using this transformation, we can compute locally the diffeomorphism as outside $\mathcal{P}_i$ the diffeomorphism becomes the unity map. We proceed on showing that it is a diffeomorphism. This is obviously a bijection, and to establish it as a diffeomorphism it suffices to show that its Jacobian is always invertible.

$$JT = \begin{bmatrix} T_r^1 & T_\theta^1 \\ T_r^2 & T_\theta^2 \end{bmatrix} = \begin{bmatrix} T_r^1 & T_\theta^1 \\ 0 & 1 \end{bmatrix}.$$

Due to the form of the Jacobian, its invertibility is equivalent to $T_r^1 > 0, \forall(r,\theta)$. By a direct computation of T_r^1 we have that ($\kappa = \frac{1}{\mu-\mu_1}$),

$$T_r^1 = \kappa \cdot S' \cdot b \cdot b_r \cdot \rho + S \cdot b_r \cdot \rho + 1 - \kappa \cdot S' \cdot r - S.$$

It holds that $S \cdot b_r \cdot r > 0$ as all terms are positive, $1 - S > 0$ since S by definition is between 0 and 1. Finally we have that as b is a shrinking operation $b \cdot \rho < r$. Formally this comes from the fact that the circle to which we transform to is inscribed into the convex shape and by the fact that $b_r \leq 1$. So we have that $\kappa S'(b \cdot b_r \cdot \rho - r) \geq 0$ as $S' \leq 0$. So all terms of T_r^1 are positive, and therefore the jacobian is always invertible. This completes the proof. The transformation is depicted in Fig. 3. Point A is transformed to point A'. The NF in the transformed domain is given by $\tilde{\phi}_{A'}$ and the its by $\nabla\tilde{\phi}_{A'}$. In the original domain, the navigation function and the associated gradient are given by $\phi_A = \tilde{\phi}_{A'}$, and $\nabla\phi_A = DT^T|_A \cdot \nabla\tilde{\phi}_{A'}$.

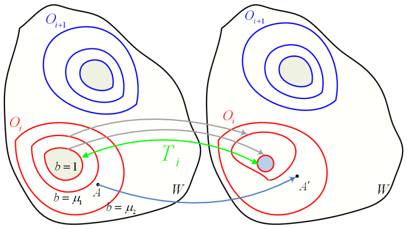

Fig. 3. Transformation from $\mathcal{P}_i$ to a set with the same external boundary but with a spherical obstacle. The circle to which the obstacle is transformed to is inscribed into the obstacle. Point A is transformed to point A', and $1 < \mu_1 < \mu_2$.

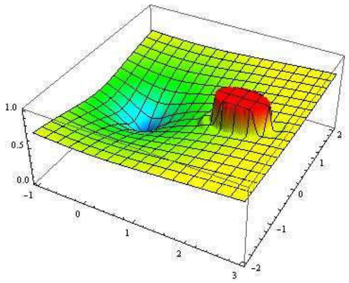

Fig. 4. NF field around an convex obstacle. The repulsive field is smooth, steers the robot around the obstacle to the goal, and it is exactly zero away from the obstacle.

VII. Simulation Results

To validate our approach, we present some simulation results, that clearly show how our proposed NF can handle a very large number of obstacles, for a spherical world. 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.

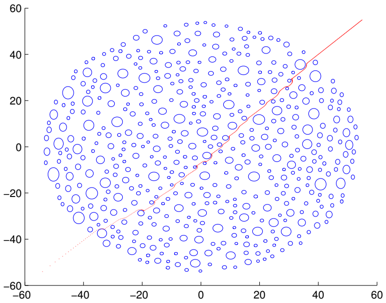

Fig. 5. Navigating in a workspace of ~ 500 obstacles.

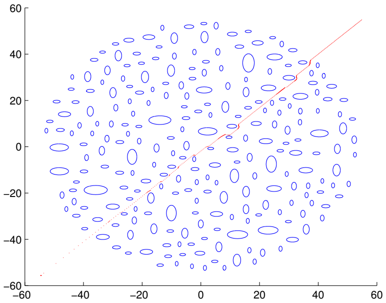

Fig. 6. Navigating in a ~ 250 convex obstacle workspace.

Fig. 5 represents a navigation in a sphere world of ~ 500 obstacles. The initial configuration of the robot is at the upper right corner, and the final configuration at the lower left cornet. 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 obstacles.

Fig. 6 depicts a simulation of a navigational task in a large workspace comprised of ~ 250 convex obstacles. The locally computable diffeomorphism is utilized to map -locally- each obstacle to a circle, and the navigation function is computed in the sphere world.

VIII. Conclusions and Future Works

In this work we presented a new locally computable NF, extending our previous result. We derived a new class of NF that combine local computability, along with rational functions (polynomial function, and therefore easily computable). Our NF is constructed as a combination of a large number of individual obstacle functions, one for each obstacle, and a global attraction potential. The obstacle functions are designed to have no effect outside a zone surrounding each obstacle, and the scheme is easily proven to be a NF as each obstacle NF has all the necessary requirements. We have shown that a large family of Polynomial functions can be used as NF, allowing us, for example, to tune the gradient of the NF using the width of the sensing zone, the order of the polynomial function and the magnitude of the coefficients of the polynomial.

We have also presented a locally computable diffeomorphism, between convex obstacles -although star shaped could also work- and sphere worlds. The diffeomorphism works exactly using the same design as the NF. A C^2 switch is used to nullify the diffeomorphism outside a zone surrounding the obstacles, instead of using analytic switches.

This approach is not completely local (i.e. knowledge of an ϵ region around a point does not suffices to build the NF) but it is local on the sense that it requires complete information only for the current and the adjacent obstacles, making it local in a graph theoretic way. Our current research focuses exactly on making the approach strictly local, i.e. constructing the NF and the transformation field based only on sensor measurements.

APPENDIX

A. *Proof of Lemma 1*

The first and the second order derivatives of $P(x)$ are calculated as $P'(x) = n \cdot a_n \cdot x^{n-1} + \ldots + 2 \cdot a_2 \cdot x + a_1$, and $P''(x) = n \cdot (n-1) \cdot a_n \cdot x^{n-2} + \ldots + 6 \cdot a_3 \cdot x + 2 \cdot a_2$. First of all, $P''(x) < 0$, $\forall x \in [0,1]$, since by the parameters definition (substituting the even parameters from Definition 1) it holds that, $n \cdot (n-1) \cdot a_n \cdot x^{n-2} + (n-1) \cdot (n-2) \cdot a_{n-1} \cdot x^{n-3} < 0$, $\ldots$, $6 \cdot a_3 \cdot x + 2 \cdot a_2 < 0$.

Thus, $P'(x)$ is strictly decreasing. Furthermore, as $P'(1) = n \cdot a_n + (n-1) \cdot a_{n-1} + \ldots + 3 \cdot a_3 + 2 \cdot a_2 + a_1 = n \cdot a_n \cdot \left(1 - \frac{n-1}{n-2}\right) + \ldots + 3 \cdot a_3 \cdot (1-2) + a_1 = 0$. On the other hand, $P'(0) = a_1 > 0$, by definition. Thus, $P'(x) > 0$, $\forall x \in [0,1]$.

B. *Proof of Lemma 2*

It holds that $P'(x) > 0$ (Lemma 1). Thus, $P(x)$ is strictly increasing for $x \in [0,1]$. Also, $P(0) = 0$ and $P(1) = a_n + \ldots + a_1$. By taking the pairs of odd and even terms, we have that, $a_n + a_{n-1} = \left(1 - \frac{n}{n-2}\right) \cdot a_n, \ldots, a_3 + a_2 = (1-3) \cdot a_3$, and from Definition 1, it holds that, $\left(1 - \frac{n}{n-2}\right) \cdot a_n + n \cdot \left(\frac{n-1}{n-2} - 1\right) \cdot a_n = a_n, \ldots, (1-3) \cdot a_3 + 3 \cdot a_3 = a_3$. Therefore, $P(1) = a_n + a_{n-2} + \ldots + a_3 = 1$. Thus, $P(x) > 0$, $\forall x \in [0,1]$.

C. *Proof of Lemma 3*

We have that $P''(1) = n \cdot (n-1) \cdot a_n + (n-1) \cdot (n-2) \cdot a_{n-1} + \ldots + 6 \cdot a_3 + 2 \cdot a_2$. By using the pairs of add and even terms according to Definition 1, it holds that, $n \cdot (n-1) \cdot a_n + (n-1) \cdot (n-2) \cdot a_{n-1} = 0, \ldots, 6 \cdot a_3 + 2 \cdot a_2 = 0$. Thus, $P''(1) = 0$, $\forall x \in [0,1]$.

D. *Proof of Proposition 1*

The gradient of function φ of (1) is $\nabla\varphi(q_d) = \frac{1}{(\gamma_d+\beta)^2}\left[(\gamma_d + \beta)\nabla\gamma_d - \gamma_d\nabla(\gamma_d + \beta)\right]|_{q_d} = 0$ since, both γ_d and $\nabla\gamma_d$ vanish at q_d. The Hessian of φ at q_d, by using Lemma 4, will be

$$\nabla^2\varphi = \frac{1}{(\gamma_d + \beta)^2} \ (\gamma_d + \beta)\nabla^2\gamma_d - \gamma_d\nabla^2(\gamma_d + \beta) \ |_{q_d} \qquad (7)$$

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.

E. *Proof of Proposition 2*

The critical points of the NF are located on the line connecting the goal with the center of the obstacle -as only on this line the gradient of the two fields are anti-parallel. By appropriate choosing the beginning of the local coordinate frame, we have on the critical points $\varphi = 0$. According to (7), we can take the numerator, and by using (5), and (6), we have that:

$$\left(\beta_i\nabla^2\gamma_d - \gamma_d\nabla^2\beta_i\right)|_{C_\varphi} =$$

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

$$\begin{bmatrix} 2\beta_i|_{C_\varphi} - \gamma_d|_{C_\varphi}\beta_{rr} & 0 \\ 0 & -2\beta_i|_{C_\varphi}y_i r \end{bmatrix}$$

It is a diagonal matrix. Since we are at a critical point, and $\gamma_d > 0$, and $\beta_i > 0$ (since $P'(x) > 0$, by Lemma 2), and $\beta_{rr} \leq 0$ (since $P''(x) < 0$, by Lemma 1), 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.

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] H. Tanner, "Switched uav-ugv cooperation scheme for target detection," *IEEE International Conference on Robotics and Automation, 2007 pp. 3457-3462*, pp. 3457–3462, 2007.

[8] 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.

[9] H. G. Tanner and A. Kumar, "Formation stabilization of multiple agents using decentralized navigation functions," *Robotics: Science and Systems I*, p. pp 4956, 2005.

[10] G. Lionis, X. Papageorgiou, and K. Kyriakopoulos, "Locally computable navigation functions for sphere worlds," *2007 IEEE International Conference on Robotics and Automation*, pp. 1998–2003, 2007.

[11] 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.

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