# Robot Navigation Based on Electrostatic Field Computation

Luciano C. A. Pimenta[1], Alexandre R. Fonseca[1], Guilherme A. S. Pereira[2], Renato C. Mesquita[2], Elson J. Silva[2], Walmir M. Caminhas[2], and Mario F. M. Campos[3]

[1]Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Departamento de Engenharia Eletrica, Belo Horizonte 31270-901, Brazil
[2]Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte 31270-901, Brazil
[2]Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte 31270-901, Brazil

**This paper addresses the problem of mobile robot navigation using artificial potential fields. Many potential field based methodologies are found in the robotics literature, but most of them have problems with spurious local minima, which cause the robot to stop before reaching its target position. Although some free of local minima methodologies are found in the literature, none of them are easy to implement and generalize for complex shaped environments and robots. We propose a perfect analogy between electrostatic field computation and robot path planning. Thus, an easy solution to the problem, which is based on standard finite-element methods, can be applied with generic geometries and can even take into account the robot's orientation. To demonstrate the elegance of the proposed methodology, several experimental results with actual mobile robots are included.**

***Index Terms*—Finite element methods, mobile robots, motion-planning.**

## I. INTRODUCTION

THE mobile robot navigation problem has attracted attention of many researchers along the years and many methodologies have been proposed. Considering a static environment, where the shape and position of the obstacles are known (by means of a map, for example), the navigation problem can be stated as follows:

"*Given a generic shaped robot in an environment with generic shaped obstacles, drive the robot to a target position in this environment by avoiding collisions with the obstacles*."

A wide survey on methodologies to solve the problem stated above can be found in [1]. Most of the approaches are based on the concept of *configuration space* $\mathcal{C}$ [1]. Consider a mobile robot navigating in a planar surface, and a reference point $G$ which is fixed on the robot. The robot's configuration $q = (x, y, \theta)$ is composed by the $x$ and $y$ coordinates of $G$ and the robot's orientation $\theta$. Then, by definition, the configuration space, $\mathcal{C}$, is the set of all possible configurations of the robot, while the robot's trajectory is a continuous sequence of configurations in $\mathcal{C}$. Fig. 1 shows these concepts. Obstacles are represented in the robot's configuration space as a set of forbidden configurations. The regions of the configuration space free of collisions are referred to as the free configuration space, $\mathcal{F}$ [1]. The main advantage of solving the navigation problem in the configuration space is that the robot can be considered as point, since its shape is considered only during the construction of the space. The computation of the robot's configuration space can be performed by growing the obstacles by the size of the robot using *Minkowski Sums* [2].

After computing the configuration space we must plan the robot's trajectory and then control the robot to follow this trajectory. A practical approach to plan trajectories and control the robot is based on *artificial potential fields* [3]. In this approach a scalar field $\phi(q)$, called potential function, is defined over the robot's free configuration space, $\mathcal{F}$. The negative gradient of the potential function $-\nabla\phi(q)$ is then treated as an artificial force acting on the robot (represented by its configuration $q$), and the resultant force may be used to control the robot. The most basic instance of this approach is to assign an *attractive potential* to the goal and a *repulsive potential* to the obstacles and add them together in order to compose $\phi(q)$. The integral curves of the vector field formed by $-\nabla\phi(q)$ define implicit paths from every start configuration in $\mathcal{F}$ to the target configuration $q^d$.

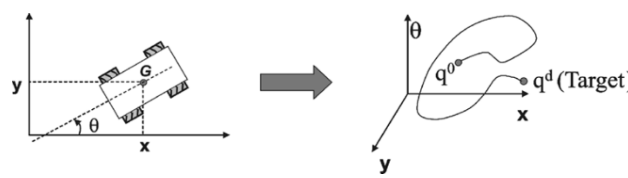


Fig. 1. Robot is represented by a point in its configuration space. A trajectory is then a continuous sequence of configurations that starts at $q^0$ (the initial robot's configuration) and ends at $q^d$ (the desired configuration).

The main drawback of most potential field approaches is that, due to the presence of spurious local minima in the potential function, convergence to the target is not guaranteed [1]. In [4] a free of local minima potential function named *navigation function* was proposed, and an analytical method to build such a function was developed in [5]. The main difficulty with this method is its implementation for generic shaped and high dimensional (i.e., more than two) configuration spaces. At about the same time the navigation functions were proposed, Connolly *et al.* [6] proposed the use of *harmonic functions*, which are solutions to the *Laplace's Equation*, as navigation functions. Differently from the authors of [5], who pursuit analytical solutions, Connolly *et al.* propose the numerical integration of the Laplace's equation. The authors discretize the domain in a homogeneous rectangular grid and applies *finite differences* to obtain the solution.

Digital Object Identifier 10.1109/TMAG.2006.870931

Many other authors have proposed different methodologies to construct numerical navigation functions [7]–[9]. The work [7] proposes computing minimal cost navigation functions by using a linear programming algorithm. In [8], the authors compute navigation functions for two-dimensional (2-D) domains using an artificial electrostatic potential field developed by means of a resistor network, which is derived to represent the environment. All these algorithms are based on regular grid discretization, which is not suitable for representing complex geometries. Furthermore, they are not practical for high dimensions and do not address the robot's orientation.

Most of the issues found in the previous approaches are not present in ours. We use finite elements methods to compute robot navigation functions based on electrostatic fields. A deeper discussion about this computation is in the next section. Section III presents the robot's control law based on these fields. Experimental results with actual robots are in Section IV. Conclusions and future work are finally presented in Section V.

## II. Navigation Function Computation

Similarly to [6], we propose computing an artificial potential field without spurious local minima by solving the Laplace's equation

$$\nabla^2 \phi = 0 \tag{1}$$

which is valid in the domain $\Omega \subset \mathbb{R}^n$. In this paper, we consider the function domain to be the robot's free configuration space, $\mathcal{F}$. Therefore, our approach to the navigation problem is to consider an equivalent electrostatic problem, where the navigation function $\phi$ is analogous to a scalar electric potential over a domain free of electric static charges and constituted by a single dielectric isotropic material.

Differently from previous works, we solve the robotics problem using finite elements methods. Since finite elements work properly with unstructured meshes, we use these meshes to discretize the configuration space and efficiently compute potential functions for generic shaped robots in highly complex domains. Because we focus in planar navigation of mobile robots our domains are $2D(x, y)$ (when we are not concerned about robot's orientation) or $3D(x, y, \theta)$. Moreover, differently from [9], that also addresses robot's orientation and complex environments, but applies searching algorithms in the configuration space, we present a closed-form solution based on the gradient of the navigation function, as originally proposed in [3].

In order to guarantee uniqueness in the solution, we must impose boundary conditions to the domain boundary. In our approach, we consider that the target boundary conditions are constant Dirichlet with value equals to zero, and obstacles boundary conditions are also constant Dirichlet with identical positive values. This implies that the negative gradient of the resultant field close to obstacles is perpendicular outward of the obstacles surface.

It is interesting to notice that, since all the boundaries, except the target ones, receive identical Dirichlet boundary conditions, it does not matter the condition value by itself, regardless it is positive. This value produces different computed potential values but the directions of the negative gradient vectors remain the same. As we show in the next section, we make the robot follow the normalized negative gradient vectors. Thus, only the gradient orientation is needed.

The inclusion of robot's orientation into the navigation problem is needed when complex shaped robots are used, or when the robot must reach the target with a given orientation. To consider robot's orientation, we must consider a three-dimensional (3-D) domain, where the $\theta$ axis is periodic over each interval of $(2\pi)/(k)$ radians, $k \geq 1$. The value of $k$ depends on the axis of symmetry of the robot. For a totally asymmetric robot we have $k = 1$. It is easy to represent this periodic property by limiting the domain such that $0 \leq \theta < (2\pi)/(k)$ and imposing a periodic boundary condition in the planes $\theta = 0$ and $(2\pi)/(k)$

$$\phi(x, y, 0) = \phi\left(x, y, \frac{2\pi}{k}\right). \tag{2}$$

## III. Robot Control

We consider that the motion of a robot is described by a simple kinematic model of the form

$$\dot{q} = u(q) \tag{3}$$

where $u$ is the input vector of the system. It is desirable to steer the robot from its initial configuration $q^0 \in \mathcal{F}$ at time $t = t_0$ to the desired configuration $q^d \in \mathcal{F}$ at some time $t = t_f > t_0$, such that $q \in \mathcal{F} \, \forall t \in (t_0, t_f]$, where $\mathcal{F} \subseteq \mathcal{C}$ is the free configuration space.

Given the potential function $\phi$, the following control law can be used to solve the navigation problem

$$u(q) = \begin{cases} -\mathbf{K} \cdot \frac{\nabla \phi(q)}{\|\nabla \phi(q)\|}, & \text{if } \nabla \phi \neq 0 \\ 0, & \text{if } \nabla \phi = 0 \end{cases} \tag{4}$$

where $\mathbf{K}$ is a $2 \times 2$ or a $3 \times 3$ diagonal matrix used to scale the solution to a robot compatible velocity value, and $\nabla \phi(q)$ is the gradient of $\phi(q)$.

Because our artificial robot navigation functions are similar to scalar electric potentials, the use of (4) makes the robot velocity to be parallel to the electric field evaluated from this potential. This analogy allows us to guarantee that our approach always drives the robot to the target independently of the initial robot's configuration. This is because field lines in an electrostatic field are oriented from higher to lower potential points. Since we have defined boundary conditions such that a unique global minimum is placed at the target, the robot can reach the target from every initial point by following those field lines.

## IV. Examples

This section presents illustrative examples with actual mobile robots. Our testbed consists of remotely controlled robots with different shapes observed by an overhead camera. A single computer is responsible to perform all the computation needed to localize and control the robots. A pair transmitter/receiver is responsible for sending the velocity vector to the robots in each

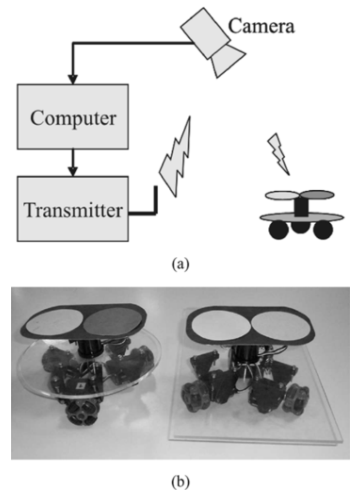


Fig. 2. (a) Diagram of the testbed used to generate the results. (b) Picture of the robots.

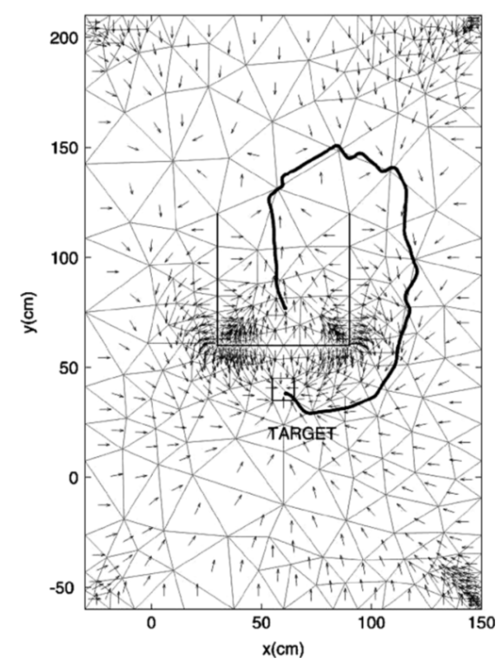


Fig. 3. Typical robot trajectory in a U-shaped workspace. Robot's trajectories are parallel to the arrows (normalized electric field) inside each element.

time step. Fig. 2(a) shows a diagram of our testbed. Fig. 2(b) shows the robots used in the examples.[1]

Fig. 3 shows a typical trajectory of an actual circular robot [see Fig. 2(b)] in a U-shaped environment. In this figure, the arrows represent the normalized electric field in each element. Since the robot is circular it is not necessary to take into account the robot's orientation and so it is sufficient to use a 2-D domain. This example is useful to visualize how the proposed approach works. Since we are using first order elements, for each element of the mesh we have a constant electric field. Thus, observe that inside each triangular element, the robot's trajectory is mostly parallel to the vector representing the field. However, due to the robot dynamics, a delay can be observed, i.e., after the robot is localized in an element it takes some time until its velocity assumes the correct field in that element. Also, due to localization

[1] Some movies are available at the web page http://www.cpdee.ufmg.br/~lucpim/compumag2005.

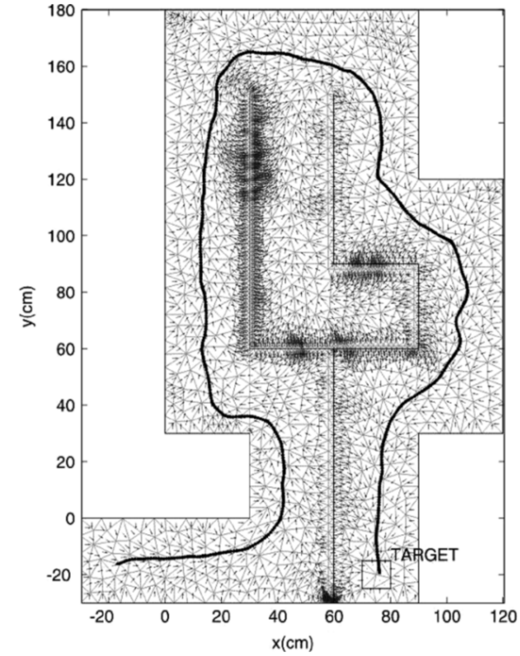


Fig. 4. Typical robot trajectory in a maze-like environment.

errors, it is possible for the robot to follow a vector that is not the one presented in the element where the robot is. In spite of this, it can be observed that, since our potential function is computed for all points in the configuration space, the method is robust to small localization errors.

A further example with the same circular robot is presented in Fig. 4. In this case a maze-like environment shows how the approach can be successfully used in more complex problems. Despite the apparent complexity of this environment, it was quite easy to construct a navigation function for this workspace using a finite element method.

Fig. 5 presents a result for a more complicated problem: an actual rectangular shaped robot [see Fig. 2(b)] moving in a workspace with a narrow passage and an oriented target. In order to facilitate the visualization of the trajectory we have fixed an arrow at the robot center. In this example the robot must reach the target with orientation $(\pi)/(2)$ rad. Although the figure shows a bidimensional view of the robot's trajectory, it should be clear that this problem is solved in a 3-D configuration space, since the robot is not circular and it must reach the target with a specific orientation. Periodic boundary conditions are also used in order to treat robot's rotation properly.

## V. Conclusion and Discussion

We have proposed a solution to the problem of computing navigation functions for complex shaped robots and environments. The problem is transformed into an equivalent electrostatic problem in the $(x, y, \theta)$ space, subject to Dirichlet and periodic boundary conditions. Finite elements are used to solve the resulting problem, and its solution is used in mobile robots navigation. Our approach is able to solve problems that could not be solved efficiently by previous methodologies. Since a path to the target exists, our approach guarantees that the robot reaches the target, independently of the initial robot's configuration. Furthermore, due to the fact that the artificial potential function is

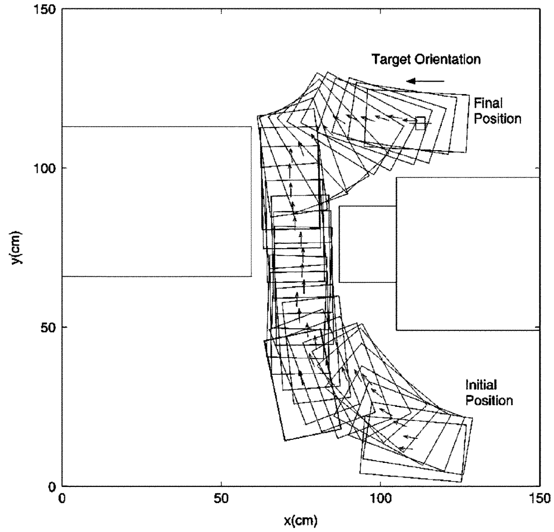


Fig. 5. Typical trajectory of a rectangular robot in a complicate workspace with an oriented target.

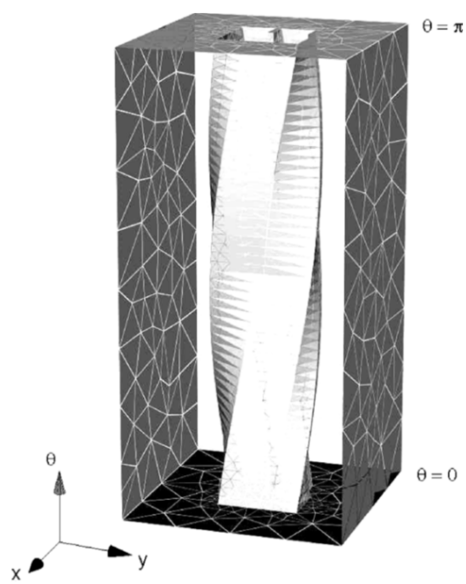


Fig. 6. Representation of a U-shaped obstacle in the configuration space of a rectangular robot.

defined over the free configuration space, the implicit paths determined by the computed vector field are free of collisions. The introduction of the periodic boundary condition and the control law proposed in Section III makes the treatment of the robot's orientation in a closed form possible.

Since we can establish a perfect analogy between electromagnetic problems and robotic problems, the numerical techniques developed in the field of electromagnetics can be directly applied to solve problems in the field of robotics. Moreover, due to the nature of the environments considered in the robotics field, their respective configuration spaces have a quite complex geometry and thus, could be used as a benchmark for mesh generation algorithms. Just to illustrate the complexity of such spaces we show in Fig. 6 the representation of a U-shaped obstacle in the configuration space of a rectangular robot. In this figure it can be also seen a boundary box used to limit the domain. Due to the symmetry of the rectangular robot, the domain must be confined in the interval $0 \leq \theta < \pi$. Since the robot is not allowed to navigate inside the obstacle, this obstacle is represented as a hole in the domain.

The main current limitations of our approach are two: 1) good knowledge about the environment must be provided, i.e., all obstacles must be modeled previously and 2) only static obstacles are treated.

Future works include real-time finite element implementation for allowing unknown stationary obstacles. The treatment of this type of unknown obstacles is useful when the robot has partial or no knowledge of the environment. In situations like that the robot must use its sensors (cameras, ultrasound, infrared, etc.) to update the environment map in real time.

A more complicated problem concerning moveable obstacles is also a possible extension of this work. This problem appears when the robot environment is populated by humans and other robots. We believe that the use of meshes limits the application of finite elements for such situations. It seems that mesh free methods such as element free Galerkin (EFG) methods [10], [11] are promising for this case.

Finally, one should notice that even though we presented results in two and three dimensions, this is not a limitation of our approach. Actually, our immediate plans are to apply this approach for robots with more degrees of freedom such as manipulators and flying and underwater robots. The main challenge in these applications is to generate good meshes in domains with dimensions higher than three.

## ACKNOWLEDGMENT

This work was supported in part by CNPq and FAPEMIG, Brazil.

## REFERENCES

[1] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.
[2] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry Algorithms and Applications*, 2nd ed. New York: Springer-Verlag, 2000.
[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
[4] E. Rimon and D. E. Koditschek, "Exact robot navigation using cost functions: The case of distinct spherical boundaries in $E^n$," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1988, pp. 1791–1796.
[5] ——, "Exact robot navigation using artificial potencial functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–517, 1992.
[6] C. I. Connnolly, J. B. Burns, and R. Weiss, "Path planning using laplace's equation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 2102–2106.
[7] K. Konolige, "A gradient method for realtime robot control," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2000, pp. 639–646.
[8] K. P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis, "Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, no. 2, pp. 187–196, Mar. 2000.
[9] Y. Wang and G. S. Chirikjian, "A new potential field method for robot path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 977–982.
[10] T. Belytschko, Y. Y. Lu, and L. Gu, "Element free Galerkin methods," *Int. J. Numer. Meth. Eng.*, vol. 37, pp. 229–256, 1994.
[11] G. F. Parreira, A. R. Fonseca, E. J. Silva, and R. C. Mesquita, "The element-free Galerkin method in 3-dimensional electromagnetic problems," in *Proc. 15th Conf. Comput. Eletromag. Fields*, vol. 1, 2005, pp. 212–213.

Manuscript received June 20, 2005 (e-mail: renato@ufmg.br).