

This paper is published in:

Journal of intelligent and Robotic Systems, July 2010, vol. 59 (1), 57-73. ISSN: 0921-0296.  
DOI: <http://dx.doi.org/10.1007/s10846-009-9381-3>

© Springer Science + Business Media B.V. 2009

This work has been supported by the CAM Project S2009/DPI-1559/ROBOCITY2030 II, developed by the research team RoboticsLab at the University Carlos III of Madrid.

# Robotic Motion using Harmonic Functions and Finite Elements

Santiago Garrido, Luis Moreno, Dolores Blanco, Fernando Martín  
Carlos III University  
Madrid, Spain

## Abstract

The harmonic functions have proved to be a powerful technique for motion planning in a known environment. They have two important properties: given an initial point and an objective in a connected domain, a unique path exists between those points. This path is the maximum gradient path of the harmonic function that begins in the initial point and ends in the goal point. The second property is that the harmonic function cannot have local minima in the interior of the domain (the objective point is considered as a border). This paper proposes a new method to solve Laplace's equation. The harmonic function solution with mixed boundary conditions provides paths that verify the smoothness and safety considerations required for mobile robot path planning. The proposed approach uses the Finite Elements Method to solve Laplace's equation, and this allows us to deal with complicated shapes of obstacles and walls. Mixed boundary conditions are applied to the harmonic function to improve the quality of the trajectories. In this way, the trajectories are smooth, avoiding the corners of walls and obstacles, and the potential slope is not too small, avoiding the difficulty of the numerical calculus of the trajectory.

Results show that this method is able to deal with moving obstacles, and even for non-holonomic vehicles. The proposed method can be generalized to 3D or more dimensions and it can be used to move robot manipulators.

**Keywords:** Motion Planning, Harmonic Functions, Finite Elements

## 1 Introduction

Robot motion control can be viewed as an underconstrained problem. The robot exists at a certain configuration and must achieve a certain goal configuration using any free path. If properly specified, such underconstrained interpolation problems can be solved using Laplace's equation. Intuitively, Laplace's equation can be thought of as governing the shape of a thin membrane: obstacle positions are pulled up and goal positions are pulled down. If a ball bearing is dropped to the membrane, it will always fall into one of the goal positions, never hitting the obstacles. The movement of the ball corresponds to a change in the state of a robot (e.g., the movement of joints during a task). The trajectory of the ball is always a smooth obstacle-avoiding path.

The harmonic functions are solutions to Laplace's equation. The harmonic potentials have proved to be a powerful technique for robot motion planning in a known environment [3]. They have two important properties: given an initial point and an objective in a connected domain, a unique path exists between those points. This path is the maximum gradient path of the harmonic function that begins in the initial point and ends in the goal point. The second property is that the harmonic function cannot have local minima in the interior of the domain (the objective point is considered as a border). These properties mean that a path can always be found if it exists and it is impossible to be stuck in a local minima. These are the most important properties that make harmonic functions extremely interesting for motion planning purposes.

Our approach has the following advantages over the methods used previously (Finite Difference Methods): 1) It uses the Finite Elements Method to solve the partial differential equation (PDE) problem. This method permits complicated shapes of the obstacles and walls. 2) It uses mixed boundary conditions because in this way the trajectories are smooth, the potential slope is not too small (avoiding troubles of path tracing through a gradient-based strategy), and the trajectories avoid the corners of walls and obstacles. 3) It can be generalized to 3D or more dimensions and it can be used to move robot manipulators. 4) It can be used for motion planning of non-holonomic vehicles.

## 1.1 Previous work

Every method concerning robot motion planning has its own advantages and application domains, as well as its disadvantages and constraints. Therefore, it would be rather difficult either to compare methods or to motivate the choice of a method upon others.

One of the current directions in the robot motion research is to include considerations regarding obstacle avoidance with trajectory generation, for example methods such as the Rapidly-Exploring Random Tree (RRT) or extensions of Probabilistic Roadmap.

The first solution to this problem is robot motion planning through artificial potential fields (APF). It considers simultaneously the problems of obstacle avoidance and trajectory planning (Arkin [1][2]). In addition, the dynamics of the manipulator is directly taken into account, which leads, in our opinion, to a more natural motion of the robot. The approach of our method belongs to these kind of solutions.

The first use of the APF concept for obstacle avoidance was presented by Khatib [6]. He proposed a Force Involving an Artificial Repulsion from the Surface (FIRAS, in French) which should be non-negative, continuous, and differentiable. However, the potential field introduced exhibits more local minima than the goal position of the robot. To solve the preceding problem, Khosla and Volpe [14] developed new elliptical potential functions called "superquadric artificial potential functions", which do not generate local minima in the physical space. They have shown that superquadric potential fields can be constructed only for simple shapes as square or triangular figures. The problem of local minima remains because the superquadric potential functions do not generate local minima in the workspace but the local minima can occur in the C-space of the robot. The contributions of Koditschek in [7][8][11] are worth to be mentioned

because they introduce an analytical potential field in the C-space without local minima. However, the topology of the application range is limited to obstacles which have to be ball- or star-shaped, otherwise no solution can be found. The contributions of Connolly [4] and Khosla and Kim [5] are the most successful methods concerning robot motion planning with potential field. They used the harmonic functions to build a potential field in the C-space without local minima. The harmonic functions attain their extreme values at the boundary of the domain. Using a potential function to accomplish a certain motion implies that the trajectory of the robot is not known or calculated in advance, which means that the robot chooses autonomously the way to reach its goal. The main problem of this method is the complication of the Panel Method used, that means that the obstacles have to be very simple, such as rectangles.

The method of harmonic functions based on the Finite Difference Method has been used for guiding robots by Connolly and Gruppen [3]. PresteseSilva et al. [9] [10] used this method for exploration of unknown environments.

The method of harmonic functions is not very extended because, up to now, it has been based on the Finite Difference Method, and for this reason it is not easy to implement. It is quite slow and has to be used in static environments and the environment has to be very easy with obstacles and walls with parallel straight lines. The method that we propose to solve the Harmonic equation using Finite Elements addresses several issues, such as boundary conditions, complex shape of obstacles, and real-time performance.

## 2 Harmonic Functions

Equilibrium problems in two-dimensional and higher space, give rise to elliptic partial differential equations. A prototype is the famous Laplace's equation:

$$\nabla^2 \phi = 0$$

This equation holds for the steady temperature in an isotropic medium, characterizes gravitational or electrostatic potentials at points of empty space, and describes the velocity potential of an irrotational, incompressible fluid flow. The two-dimensional counterpart of this equation lies at the foundation of the theory of analytical functions of a complex variable.

A harmonic function  $\phi$  on a domain  $\Omega \in R$  is a function which satisfies Laplace's equation:

$$\nabla^2 \phi = \sum_{i=1}^n \frac{\partial^2 \phi}{\partial x_i^2} = 0$$

where  $x_i$  is the  $i$ -th Cartesian coordinate and  $n$  is the dimension. In the case of robot motion planning, the boundary  $\partial\Omega$  of  $\Omega$  is formed by all the walls, obstacles, and goal, but not by the start point.

One important property of the harmonic functions for our task is the *Maximum Principle*: A non-constant harmonic function  $\phi(x)$  defined on  $\Omega$  takes its Maximum value  $M$  on the boundary and guarantees that there are no local minima in the interior of  $\Omega$ .

## 2.1 Problems using Harmonic Functions

The use of harmonic functions as motion planning potentials is a powerful technique but it is not very extended because: first, the previous methods based on finite differences need that all obstacles and walls of the region are composed by parallel straight lines. Second, those methods are slow and difficult to implement in a dynamic environment that changes with time. Third, some implementations of those methods do not assure the existence of a trajectory. For example, when using Harmonic functions as a potential field, it is not possible to add an attractive potential to the goal and a repulsive potential from the obstacles because the domain is not the same. In the first case, the goal is one of the borders because this is the only way of being the minimum of the function. In the second case, the borders are the maximum and the goal cannot be one of them because the resulting addition function does not have the minimum at the goal. Because of that, the only way of doing a potential attractive to the goal and repulsive from the obstacles is defining the obstacles borders value at the maximum value, the goal border at the minimum value and to solve Laplace's function with these border values. Fourth, in case the obstacles are not fixed, it is necessary to recalculate the harmonic function and path continuously, each time a movement is registered, in order to obtain an adaptive version of the method.

## 3 Boundary Conditions

The different kinds of contour conditions imposed to Laplace's equation have a critical importance in the solution of the equation and the quality of the trajectory. The direction field associated to the partial derivatives problem gives the path that the robot has to track.

To achieve a good quality path to the goal, some convenient properties of the direction fields are:

1. The trajectories cannot be smooth and as short as possible.
2. The potential slope have not to be small.
3. The trajectories have to avoid the corners of walls and obstacles. Because of that, the gradient has to be perpendicular or tangent to the borders of the work space.

The solution can be considered as the potential function and the trajectories associated to the direction field of the potential, that is the path of the robot.

The contour conditions can be given in three ways: by the values of the potential function in the contour (Dirichlet), by the gradient values in the contour (Neumann), or by a linear combination of the two of them (Robin or mixed).

### Dirichlet

$$\phi = c(x, y), \quad \text{in the contour.}$$

Usually, it is taken as a constant function  $\phi = K$ .

### Neumann

$$\nabla(\phi) = c(x, y), \quad \text{in the contour.}$$

Usually, it is taken as a constant function  $\nabla(\phi) = K$ .

### Robin or Mixed

$$\vec{n} \cdot a \cdot \nabla(\phi) + b \cdot \phi = c(x, y) \quad \text{in the contour.}$$

Usually, it is taken as a constant function  $\vec{n} \cdot a \cdot \nabla(\phi) + b \cdot \phi = K$ .

If the solution of Laplace's equation is the velocity potential of an irrotational, incompressible fluid flow, then the Dirichlet conditions represent the different contour levels, the Neumann conditions correspond to the entrance or exit of fluid flow, and the Robin conditions correspond to the two of them at the same time.

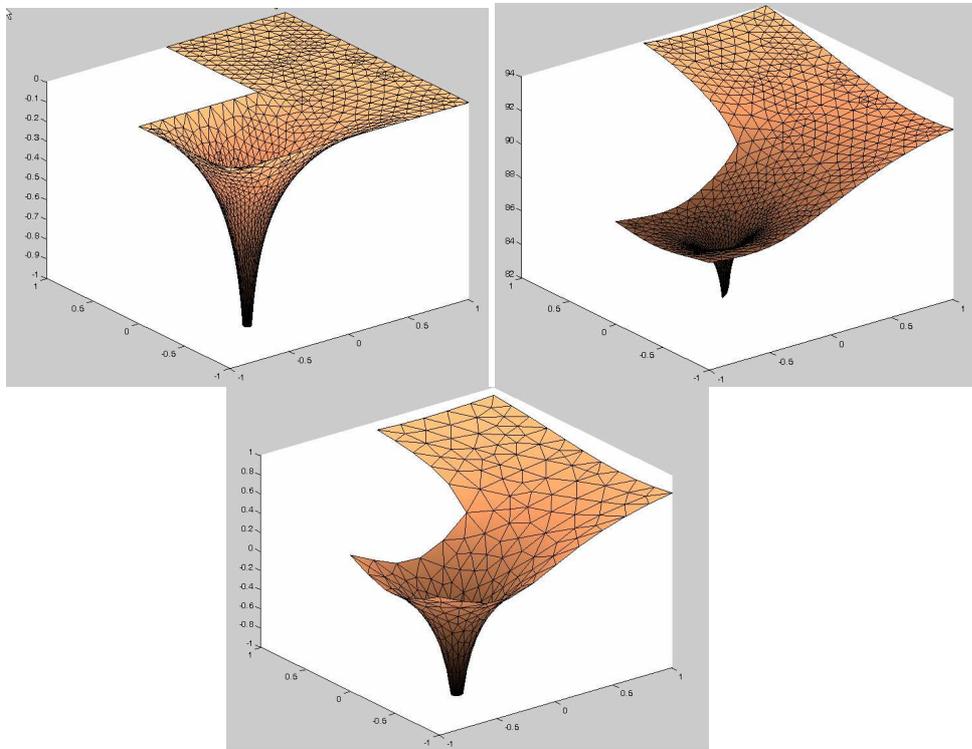


Figure 1: Solutions of Laplace's equation with Dirichlet's, Neumann's, and Robin's boundary conditions, respectively.

Subsection 2.1 treated some problems in the use of harmonic functions to solve path planning and the reasons why this method is not very extended. Other problems can come up because the way the boundary conditions are assigned.

### Dirichlet's Conditions.

In Dirichlet's conditions case, the boundary is maintained at a constant value higher than the goal point. As the boundary value is fixed, the fluid flow has to be along the normal to the boundary (see figure 2). Some authors [?] give a higher potential value to the initial point than the walls values because in regions far from the objective point the potential values are quite flat for Dirichlet's conditions, see figure 1. In this situation, the gradient is very small and computational errors can be produced (round off, quantization,...). The absence of an appreciable gradient will prevent the efficient tracing of a path because, if the magnitude of the gradient is not appreciable, i.e., the difference level between successive points of the path has a comparable magnitude to the algorithm error, the path will not have a good quality.

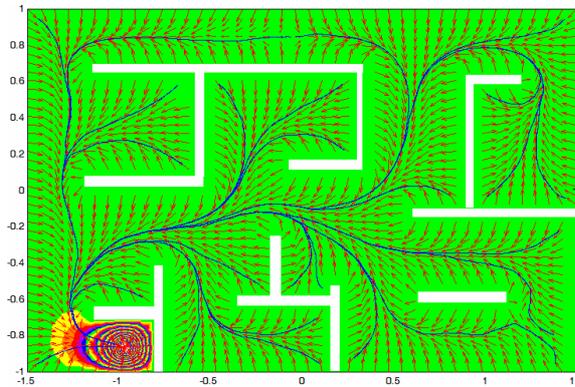


Figure 2: Robot's trajectories obtained by solving Laplace's equation with Dirichlet's boundary conditions.

This high potential value for the initial point is motivated for the logarithmic dependency of the distance to the objective point (in two dimensions the field is proportional to the logarithm of the distance and in three dimensions it is inverse proportional to the square of the distance).

Due to this reason, points distant to the objective have a small gradient, which makes difficult the numerical calculus of the trajectory.

An advantage of Dirichlet's conditions is the completeness (the algorithm is capable of finding a path between start and goal points, if it exists).

*Proposition.*- If the borders have a maximum potential value ( $=1$ ) and the objective point has a minimum value ( $=0$ ) in a connected domain, then there exist a unique trajectory from the initial point to the objective point.

This proposition shows the convenience of choosing Dirichlet's or Robin's contour conditions because they give fixed values to the potential conditions in the contour, see figure 3.

When Dirichlet's contour conditions are used, the solution is quite flat for values far from the objective point. For this reason, it is easy to have computational errors in the trajectory computation. Apart from that, the trajectories are very smooth and they

are not very close to the walls corners and obstacles.

Dirichlet's conditions represent the union between an attractive and a repulsive field (attractive to the goal point and repulsive from the walls and obstacles). Its only disadvantage is the planitude of the gradient in regions far from the goal point.

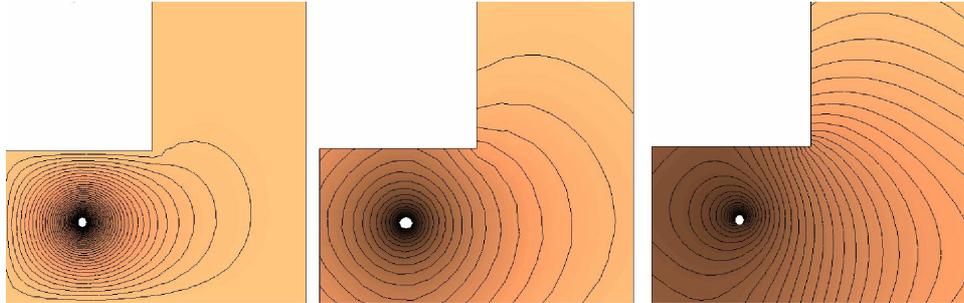


Figure 3: Contour of Laplace's equation with Dirichlet's, Neumann's, and Robin's boundary conditions, respectively.

#### **Neumann's Conditions.**

Neumann's conditions constrain the normal component of the gradient to be zero at the boundaries. As there is no normal component of fluid flow, the condition forces the flow to be tangential to the boundary. In Neumann's conditions case, the descent is smooth and continuous, with a slope not close to zero and, because of that, it is more difficult to have computational errors in the trajectory calculation.

The problem with Neuman's conditions is that the trajectories are so close to the contour corners that it is impossible to continue the calculation of the trajectory, as can be seen in figure 4.

This situation happens because the Maximum Principle says that the solution of Laplace's equation cannot have its maximum (and minimum) in an interior point, but it says nothing about the border. The only way not to have the end point of a trajectory in the walls is to give them the maximum value.

Neuman's conditions represent only the attractive field to the goal point, reason why there are problems to avoid obstacles and walls (especially corners).

#### **Robin's Conditions (or mixed).**

In order to assure that the trajectories end at the objective point and the solution gradient is not small, it is preferable to use Robin's boundary conditions. It has the advantages of using Dirichlet's and Neumann's conditions and none of their disadvantages: the solution has a smooth slope, continuous and big enough, and at the same time, the trajectories are not close to the corners. This let us an easy calculation of the trajectories, see figure 5.

The correct way to be sure that all the trajectories end at the objective point is to have only two values of the potential in the boundary conditions: the maximum value in the walls and obstacles and the minimum value in the objective point. It is important to note that the objective point, the obstacles, and the walls belong to the contour of Laplace's equation problem.

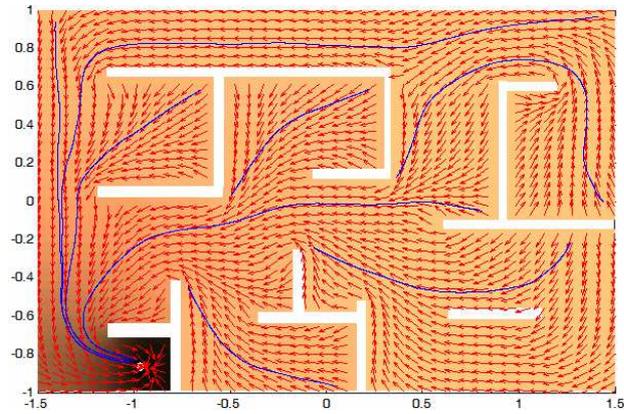


Figure 4: Robot's trajectories obtained to solved Laplace's equation with Neumann's boundary conditions.

It is also possible to give the highest potential value to the initial point. But in this case, the unicity of the trajectories is lost and a selection criteria will be necessary in order to choose the initial direction.

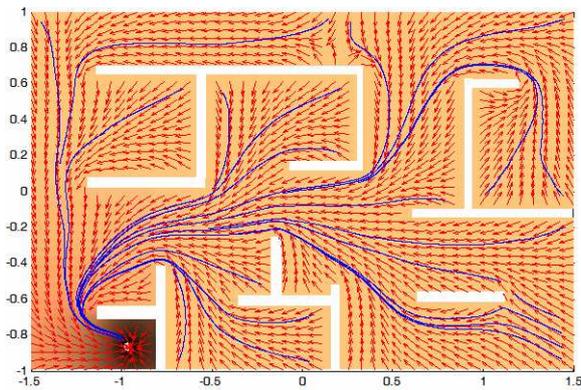


Figure 5: Robot's trajectories obtained by solving Laplace's equation with Robin's boundary conditions.

### 3.1 Comparison between different boundary conditions

We have applied our method using three different parameters to compare the boundary conditions. The following formula has been used in our algorithms:

$$\vec{n} \cdot a \cdot \nabla(\phi) + b \cdot \phi = K$$

Table 1: Comparison among different parameter values of the boundary conditions

	Type	Parameters (a, b, K)	d(%),I.P.=(0.5,0.8)	d(%),I.P.=(0.2,0.8)
1	Neumann	1, 1, 0	23	9
2	Robin	1, 1, 1	32	23
3	Robin	1, 1, 2	41	41
4	Robin	1, 1, 3	45	50
5	Robin	1, 1, 10	82	64
6	Dirichlet	0, 1, 0	100	100
7	Robin	1, 1, 100	100	100
8	Robin	1, 1, 1000	100	100

The values of  $a$ ,  $b$ , and  $K$  have been changed in order to obtain different boundary conditions. The results of our experiments are in the previous table. Two initial points have been selected. Note that case 6 corresponds to Dirichlet's conditions ( $\phi = K$ ) and case 1 to Neumann's conditions ( $\vec{n} \cdot \nabla(\phi) = K$ ). These two examples and case 5 for an initial position of (0.5,0.8) can be seen in figure 6. It can be observed that in case number 1 (represented in the top right image), the trajectory is too close to the corner.

To compare the results the variable  $d$  has been obtained:

$$d = \frac{d_i}{d_D} * 100[\%],$$

where  $d_i$  is the shortest distance between the trajectory calculated and the obstacle, and  $d_D$  is the shortest distance between the trajectory calculated and the obstacle with Dirichlet's conditions. The value of reference is  $d_D$  because Dirichlet's conditions guarantee a safe trajectory far from the obstacles.

It can be concluded that the trajectories calculated with Robin's conditions (considering the value of  $K$ ) are smooth and safe, and the potential field is not flat far from the objective.

## 4 Comparison between Finite Elements Method and Finite Difference Method

There are many elegant analytical solutions to Laplace's equation in special geometries, but nowadays real problems are usually solved numerically. The common approach to finding a numerical solution of Laplace's equation is the Finite Difference Method (FDM). The Finite Elements Method (FEM), however, has a number of advantages which justify the additional cost involved in its understanding and formulation.

The finite elements method (FEM) is particularly useful when a robust approximation is used to solve partial differential equations on an inhomogeneous mesh. Solid mathematical foundations and a great deal of generality allow different implementations.

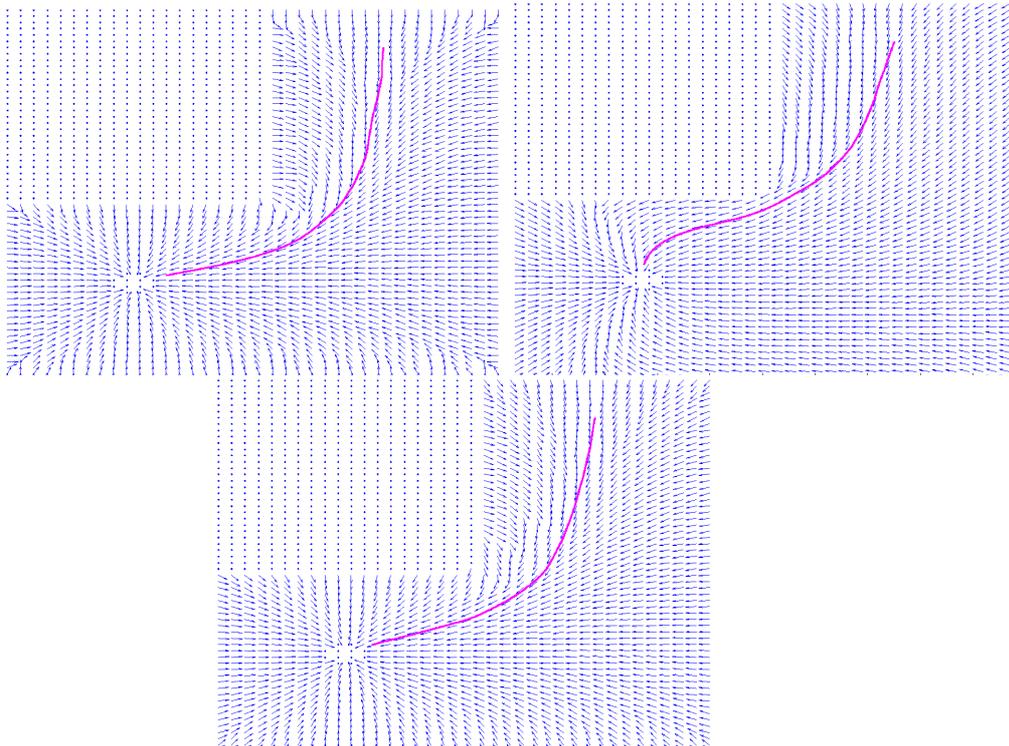


Figure 6: Trajectories calculated with Dirichlet's, Neumann's, and Robin's boundary conditions, respectively.

The most important difference between FEM and FDM is the treatment of the geometry of the domain. FDM uses rectangular meshes and FEM usually uses triangular meshes.

J. Rosell, in [12], uses a hierarchical decomposition of the configuration space, harmonic functions are computed over a non-regular rectangular mesh, but it can be awkward to try to work finite differences over non-uniform meshes. Generally speaking, FEM is more flexible with curved geometries (see figure 7).

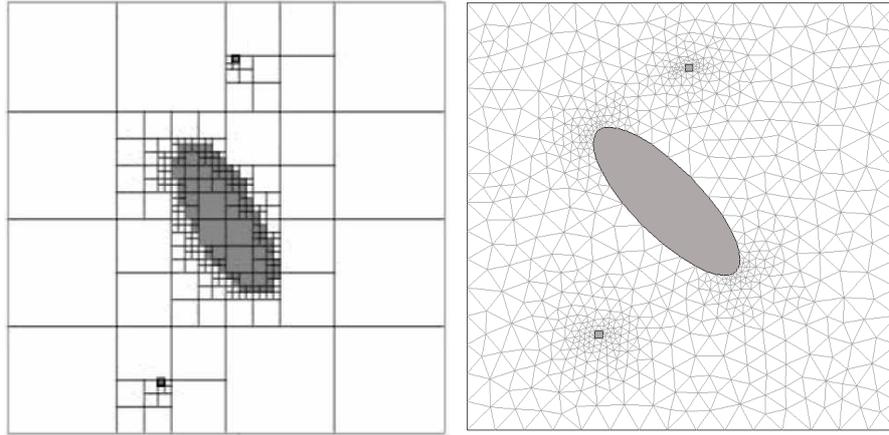


Figure 7: Hierarchical Finite Differences and Finite Elements Method decomposition of C-space

To illustrate the difference of computational time between Successive Over-Relaxation Method (SOR) and FEM, both methods have been applied to solve Laplace's Equation of the same workspace corresponding to figure 1, using a 1.6 Ghz laptop PC. The results are shown in table 2 (N is the number of grid points).

Table 2: Time comparison between SOR and FEM methods

	N=10	N=20	N=50
SOR (in seconds)	1.3720	5.2670	128.8550
FEM (in seconds)	0.2210	0.9910	7.5010

Clearly, the details of FEM are more complex than those of FDM. The end-points of both methods are essentially the same, namely a system of linear or non-linear algebraic equations. The finite element method, however, has a number of advantages:

- it is well suited to problems involving complex geometries.
- it can readily handle problems where the physical parameters vary with the position within the domain.

- it can also be used for non-linear and/or time-varying problems.
- complex boundary conditions can be readily dealt with.
- general computer programs to perform finite element calculations can be, and have been, developed using conventional numerical techniques.

In a few words, anything that can be done easily with Finite Differences can be done well with finite elements, but not the other way round.

PresteseSilva and his group have been working with Finite Differences in Exploratory Navigation [9][10]. The trajectories calculated are smooth and safe and their exploration method works very well in dense environments. However, it does not work well in sparse environments, and harmonic functions are no longer a good choice.

## 5 Implementation of the Finite Elements Method

This paper proposes the use of the general FEM with mixed boundary conditions to solve Laplace's equation.

The method starts with approximating the computational domain with a union of simple geometric objects, in the 2D case, triangles. The triangles form a mesh and each vertex is called a node. The mesh design has to strike a balance between the ideal rounded forms of the original sketch and the limitations of his simple building-blocks, triangles or quadrilaterals. If the result is not close enough to the original objects, we can always improve it using smaller blocks.

Now, using Laplace's equation (expressed in  $\Omega$ )

$$-\nabla^2 u = 0$$

If  $u_h$  is a piecewise linear approximation to  $u$ , it is not clear what the second derivative term means. Inside each triangle,  $\nabla u_h$  is a constant (because  $u_h$  is flat) and thus, the second-order term vanishes. At the edges of the triangles,  $\nabla u_h$  is in general discontinuous and a further derivative makes no sense.

This is the best approximation of  $u$  in the class of continuous piecewise polynomials. Therefore we test the equation for  $u_h$  against all possible functions  $v$  of that class. Testing means formally to multiply the residual against any function and then integrate, i.e., determine  $u_h$  such that  $\int_{\Omega} (\nabla u_h) v dx = 0$  for all possible  $v$ . The functions  $v$  are usually called test functions.

Partial integration (Green's formula) yields that  $u_h$  should satisfy

$$\int_{\Omega} \nabla u_h \nabla v - \int_{\partial\Omega} \vec{n} \cdot (c \nabla u_h) v ds = 0$$

where  $\partial\Omega$  is the boundary of  $\Omega$  and  $\vec{n}$  is the outward pointing normal on  $\partial\Omega$ . Note that the integrals of this formulation are well-defined even if  $u_h$  and  $v$  are piecewise linear functions.

Boundary conditions are included in the following way. If  $u_h$  is known at some boundary points (Dirichlet's boundary conditions), we restrict the test functions to  $v =$

0 at those points, and require  $u_h$  to attain the desired value at that point. For the rest of points, Robin's boundary conditions are used, i.e.,  $(\nabla u_h) \cdot \vec{n} + qu_h = g$ .

The FEM formulation is: to find  $u_h$  such that

$$\int_{\Omega} \nabla u_h \nabla v + \int_{\partial\Omega_1} qu_h v ds = \int_{\partial\Omega_1} g v ds \quad \forall v$$

where  $\partial\Omega_1$  is the part of the boundary with Robin's conditions. The test functions  $v$  must be zero on  $\partial\Omega - \partial\Omega_1$ .

Any continuous piecewise linear  $u_h$  is represented as a combination  $u_h(x) = \sum_{i=1}^N U_i \phi_i(x)$ , where  $\phi_i$  are some special piecewise linear basis functions and  $U_i$  are scalar coefficients. Choose  $\phi_i$  like a tent, such that it has the "height" 1 at the node  $i$  and the height 0 at all other nodes. For any fixed  $v$ , the FEM formulation yields an algebraic equation in the unknowns  $U_i$ . To determine  $N$  unknowns, we need  $N$  different instances of  $v$ . What better candidates than  $v = \phi_j$ ,  $j = 1, 2, \dots, N$ ? We find a linear system  $KU = F$  where the matrix  $K$  and the right side  $F$  contain integrals in terms of the test functions  $\phi_i$ ,  $\phi_j$  and the coefficients defining the problem:  $q$  and  $g$ . The solution vector  $U$  contains the expansion coefficients of  $u_h$ , which are also the values of  $u_h$  at each node  $x_i$ , since  $u_h(x_i) = U_i$ .

If the exact solution  $u$  is smooth, then FEM computes  $u_h$  with an error of the same size as that of the linear interpolation. It is possible to estimate the error on each triangle using only  $u_h$  and the PDE coefficients (but not the exact solution  $u$ , which in general is unknown). The FEM method provides functions that assemble  $K$  and  $F$  automatically.

To summarize, the FEM approach is to approximate the PDE solution  $u$  by a piecewise linear function which is expanded in a basis test-functions  $\phi_i$ , and the residual is tested against all the basis functions. This procedure yields a linear system  $KU = F$ . The components of  $U$  are the values of  $u_h$  at the nodes. For  $x$  inside a triangle,  $u_h(x)$  is found by linear interpolation from the nodal values.

### Extension of the method to 3D

Using the same techniques described in the previous sections, the method can be applied to 3D environments as shown in figure 8. The only difference is the use of a 3D grid in the Finite Elements Method. In the figure, the colour lines represent the paths to the goal point from different initial points. The trajectories in 3D maintain the good qualities of the 2D case.

## 6 Avoidance of stationary and moving obstacles

This technique can be used at a global level to plan a free obstacles trajectory by considering the environment information contained in a priory map. Besides, this technique can also be applied at a local level to consider sensor-based information (see figure 9, in which the algorithm is used with raw sensor data). This situation is required because, in spite of the fact that the robot has an initial global map of the environment and makes an initial motion planning from the initial point to the goal, unexpected obstacles can appear in the workspace. The map is not perfect and can have errors such as obstacles that do not appear in the map. These obstacles can be

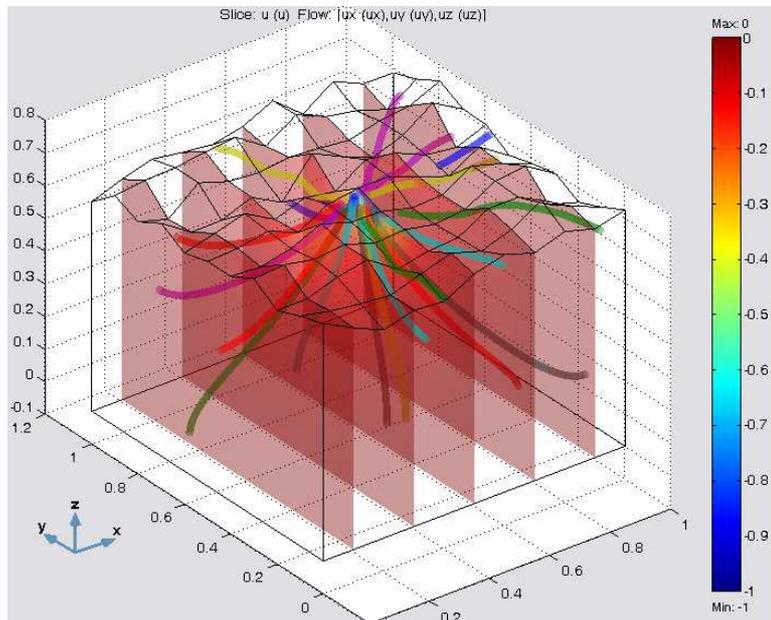


Figure 8: Extension of the method to 3-D.

static or they can move. In order to avoid them, the robot requires to consider local map information extracted from sensors. This local map is used in a reactive way to find the path avoiding the obstacles.

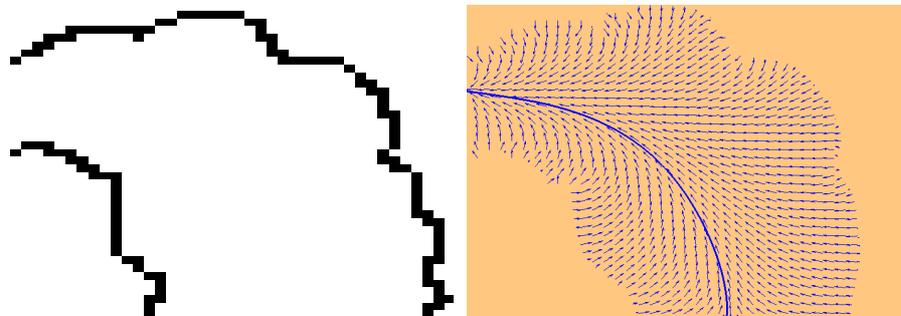


Figure 9: Local Motion Planning using the trajectory from the solution of Laplace's equation calculated by Finite Element Method, with sensor data.

When there are static obstacles, the followed strategy consists of the steps shown in figure 10. It starts with the calculation of the position of the obstacles that do not appear in the map. Second, Laplace's equation and the direction field given by the potential are calculated. Then, the trajectory to the objective is calculated, and the robot is moved

a small distance that depends on the distance to the next mobile object. Finally, if the robot has not found the objective, the first step starts again. In figure 11 an example of obstacle avoidance can be seen.

It is also possible to avoid dynamic obstacles, as can be seen in figure 12. In that figure an obstacle is moving close to the robot. If the robot detects the obstacle, it changes the trajectory previously calculated to avoid the collision.

The trajectory in the global map is done off-line and the trajectory in a local map of dimensions  $40 \times 40$  pixels, with a grid composed by 3200 triangles, lasts less than one second in a 1.6 Ghz laptop PC. In this way, the computational efficiency of the proposed motion planner makes it suitable for real-time implementation depending of the robot speed and the dynamic obstacles speed. A normal size for a grid map representation is  $10 \times 10 \text{ cm}$  cell size, that is, a local map of dimensions  $40 \times 40$  pixels represents a region of  $400 \times 400 \text{ cm}$ . The time required to plan a trajectory by the proposed method in this local map is less than one second.

This kind of algorithm is possible to be performed today because of the speed of current computers. The flexibility of the resolution of Laplace's equation with the Finite Elements Method permits all kind of shapes in walls and obstacles.

An important detail to be considered is that the gradient modulus is proportional to the objective distance (without obstacles). For this reason, it is better to use a constant speed in order to avoid big differences in speed of the robot.

## 7 Motion planning with non-holonomic vehicles.

This method can also be used with non-holonomic vehicles. To do that, starting from the initial position and orientation, the path is constructed step by step, according to the following order:

- The front wheels are aligned with the vector field in the midpoint of the front axis.
- The perpendicular lines to the front and rear wheels are considered and their intersection is taken as center of the step movement.
- With the previously calculated center, the vehicle is moved a circumference arc of length proportional to the vector modulus corresponding to that point.

The previous process is repeated from the new point until the destination point is reached. The final point and orientation is always reached because the funnel potential ends at this point and orientation.

The model of a non-holonomic vehicle and a typical trajectory can be observed in figure 13.

## 8 Conclusions

In this paper, a technique to solve the robot motion planning problem based on the resolution of Laplace's equation is proposed. The harmonic potential function is used

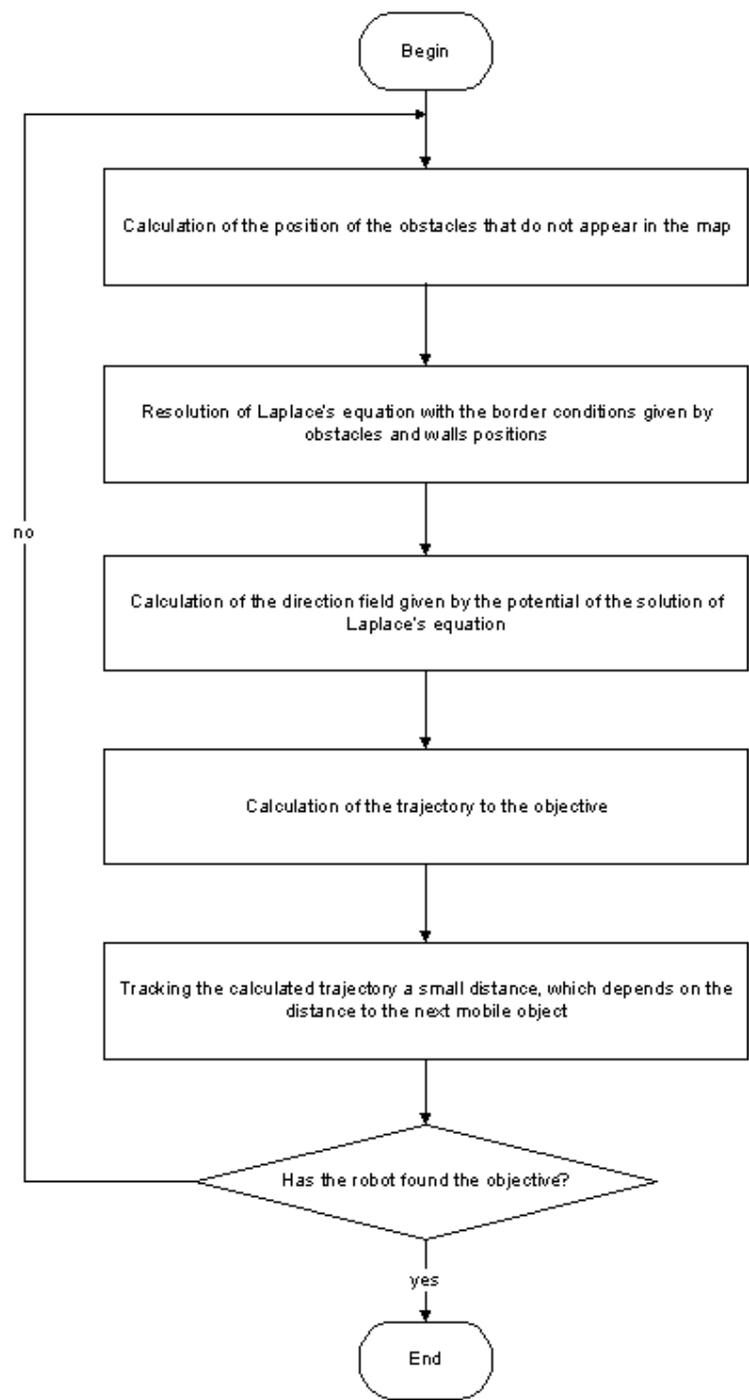


Figure 10: Flow chart of the algorithm.

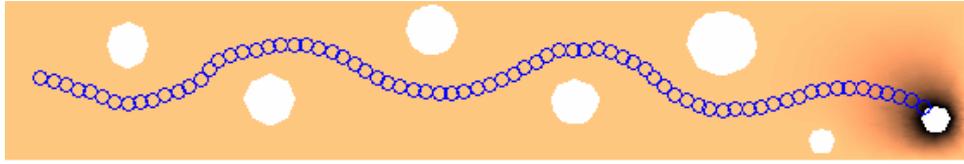


Figure 11: Avoidance of stationary obstacles.

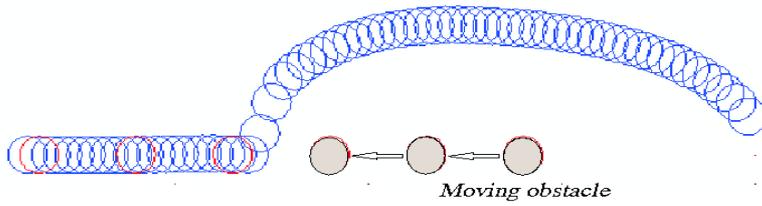


Figure 12: Avoidance of moving obstacles.

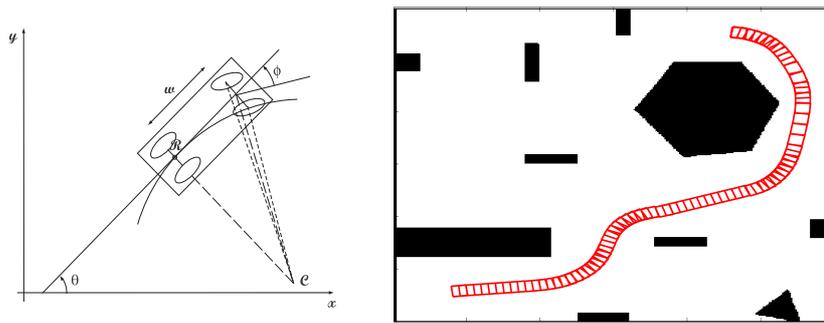


Figure 13: Motion planning with a non-holonomic vehicle.

to build potential fields for motion planning and moving obstacle avoidance. The use of harmonic functions eliminates the local minima even in very complicated environments.

The solution of Laplace's equation by means of the Finite Elements Method with mixed boundary conditions provides robot trajectories with suitable characteristics for the robot motion: smoothness and distance from the corners and walls to be placed in the safest region. The effectiveness of the proposed strategy has been demonstrated using this Finite Elements Method for robot navigation in environments with unexpected obstacles. The method is able to deal with all kind of shapes in walls and obstacles.

The computational efficiency of the proposed control scheme makes it suitable for real-time implementation.

Nowadays, the current computer speed gives us the possibility to solve these kind of problems that could not be confronted several years ago.

## Acknowledgment

The research leading to these results has received funding from the RoboCity2030-II-CM project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

## References

- [1] R.C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6, 1990.
- [2] R.C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.
- [3] C.I. Connolly and R.A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, Oct., 1993.
- [4] C. I. Connolly. *Harmonic functions as a basis for motor control and planning*. PhD thesis, University of Massachusetts, Amherst, MA, USA, 1994.
- [5] J.O. Kim and P. K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, Jun., 1992.
- [6] O. Khatib. *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. PhD thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE), Toulouse, France, 1980.
- [7] D.E. Koditschek. The control of natural motion in mechanical systems. *Journal of Dynamic Systems, Measurement, and Control*, 113:547–551, 1991.
- [8] D.E. Koditschek. Some applications of natural motion. *Journal of Dynamic Systems, Measurement, and Control*, 113:552–557, 1991.
- [9] E. PresteseSilva, P.M. Engel, M. Trevisan and M.A. Idiart. Exploration method using harmonic functions. *Robot. Auton. Syst.* 40(1):25–42, Jul., 2002.
- [10] M. Trevisan, M.A. Idiart, E. Prestes and P.M. Engel. Exploratory Navigation Based on Dynamical Boundary Value Problems. *J. Intell. Robotics Syst.* 45(2):101–114, Feb., 2006.
- [11] Rimon, E. and Koditschek, D.E.. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.*, 8(5):501–518, Oct., 1992.
- [12] J. Rosell and P. Iñiguez. Path planning using Harmonic Functions and Probabilistic Cell Decomposition. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, ICRA2005*, 1803–1808, 2005.
- [13] R. Suárez and J. Rosell. Searching Form-Closure Fixturing Points on Objects Described by Triangular Meshes. *2007 IEEE International Symposium Conference on Assembly and Manufacturing, ISAM'2007*, 7–12, 2007.

- [14] P. Khosla and R. Volpe. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1423–1436, 1990.