The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Computing Navigational Routes in Inhomogeneous Environments using BVP Path Planner

Edson Prestes[†] and Marco Idiart[‡]
[†]Instituto de Informática and [‡]Instituto de Física
Universidade Federal do Rio Grande do Sul
Porto Alegre - RS - Brazil
{prestes}@inf.ufrgs.br,{idiart}@if.ufrgs.br

***Abstract*— BVP Path Planners generate potential fields whose gradient descent represents navigational routes from any point of the environment to a goal position. The resulting trajectories are smooth and free of local minima. In this paper, we proceed with our comprehensive study on possible forms for the core equation of the planner. The main goal is to allow the planner to deal with inhomogeneous environments. The navigational potential is calculated independently in different patches of the environment. Each patch has its own bias for potential concavity (or convexity), then creating regions with higher or lower traveling preferences. By using simulated experiments we compare the performance of the new BVP with some recent proposals.**

I. Introduction

Path Planners based on Boundary Value Problems (BVP) have two important properties. First, if the core partial differential equation(PDE) is well chosen the resulting navigational potential is smooth and free from local minima. Second, the shape of the potential carries global environment information. If a passage is interrupted in a maze, the whole map changes and with it a new path towards the goal is found, if it exists.

A possible criticism that can be attributed to BVP planners is the lack of intuitive understanding of the relationship between the terms in the PDE and the behavior of the robot in the environment. In other words, it is relatively hard to program behaviors using this kind of potentials. The BVP planners have two forms of programming: we can either adjust the boundary conditions that are related to the way that the potential connects to the obstacles and goals, or we can adjust the core PDE itself. In the later years we have systematically explored the second possibility and we are beginning to understand the concepts that can help to create a user friendly way of potential programming.

The first planner based on BVP was the Harmonic Functions Path Planner [1], [2] with the Laplace Equation as the core equation. The resulting trajectories represent a compromise between length and safety, since the paths reduce the obstacle hitting probability[1]. A certain but limited freedom to redesign the paths can be obtained by changing the boundary conditions from Dirichlet to Neumann. For instance, using Neumann conditions, the robot exhibits the tendency to stay close to the c-space obstacles surface [2]. In cases where the robot needs to avoid specific environment regions or to constrain its motion direction an anisotropic harmonic potential field can be used[3], [4].

Some years ago we proposed an extension on the Path Planner based on Laplace's Equation [5]. We added to the PDE a new term that is a homogenous function of the potential gradient. It therefore vanishes when the gradient vanishes what prevents local minima. With the new term, adjustable parameters were introduced and assigned to patches of the environment. That makes possible to control the robot behavior in a specific way in different environment regions. This is what is expected, for instance, in Robotic Soccer[6] or in pedestrians behavior in realistic outdoors scenes[7]. This could not be done with the Laplace's Equation since it lacks controllable parameters.

There is some freedom for the choice of the gradient term and we have tried a few functional forms over the years [5], [8]. In this paper, we introduce yet a new form with some advantages over the last proposal discussed in [8]. First, we eliminate the need of the knowledge of the harmonic potential vector field to be used as input to the PDE. Second, in a hierarchy of preferred regions the robot always chooses to pass in the more preferred region, while in the previous form the robot would sometimes cut corners passing in a less preferred region. As consequence the new results have been significantly improved.

All improvements can be easily incorporated in the mapping or localization strategies already described in [9], [5], [10]. The preferences could guide the robot in its mapping task favoring regions where self-localization is more accurate. This gives rise to integrated exploration techniques [11], [12], [13], [14] that produce more quality maps than the maps generated by solution where this integration is absent.

This paper is divided as follows. In Section II, we discuss the basic idea behind the method by showing results in 2D environments. In Section III, we present results in different environments and finally in Section IV, we conclude.

II. Theoretical Approach

The original BVP path planner [5] generates paths using the potential information computed from the numeric solution of

$$\nabla^2 \, p(\mathbf{r}) \; = \; \epsilon \mathbf{v} \cdot \nabla p(\mathbf{r}) \; , \qquad (1)$$

with Dirichlet boundary conditions, where $\mathbf{v} \in \Re^2$ and $|\mathbf{v}| = \mathbf{1}$ corresponds to a vector that inserts a perturbation in the potential field; $\epsilon \in \Re$ is the intensity of the perturbation produced by $\mathbf{v}$; and $p(\mathbf{r})$ is the potential at position $\mathbf{r} \in \Re^2$, respectively. Both $\mathbf{v}$ and ϵ must be defined before computing this equation.

In the numeric implementation, the environment is divided into a regular mesh, where each mesh cell is associated to a square environment region and stores a potential value. By using the Dirichlet boundary conditions, the cells that are associated to obstacles store high potential values (1), whereas the cells associated to the goal position store a low potential value (0). Other cells have their potential computed iteratively using a finite difference equation[5], [9]. After the potential convergence, the robot uses the gradient descent of this potential to reach the goal position.

Eq. 1 indicates that, different from Harmonic Potentials, the potential $p(\mathbf{r})$ is locally convex or concave[1]. The Laplacian is the trace of the Hessian matrix, then if it is positive the function is locally concave and if negative it is locally convex. Therefore the curvature of the potential can be controlled by ϵ and $\mathbf{v}$. But, if a function has constant convexity or concavity it will eventually present an extremum (a minimum or a maximum). That is the reason why the added term has to be proportional to the gradient (and therefore a homogenous function of the gradient), so the convexity/concavity vanishes if the function approaches to an extremum. Apart from degenerate cases when the second derivative also vanishes, this is enough to guarantee that the potential does not present local minima or maxima[15].

The effect of changing the curvature of a limited region can be better visualized in a 1D environment as shown in Figures 1 and 2. They show three regions where the central region has the parameters set to present a curvature while the others have just the harmonic potential (the straight line in this case). The results illustrate how the inclination of the potential in the external regions adapts to the curvature of the central region. As the BVP Planner uses the direction of the gradient descent for navigation, the convexity/concavity of the potential is irrelevant in 1D environments. But it is determinant in 2D since adjacent regions with different curvatures could steer paths in many different ways as we will demonstrate in Section III.

Eq. 1 in 2D has an extra control parameter that is the direction ϕ of $\mathbf{v} = (\cos\phi, \sin\phi)$. The curvature of the potential is therefore also dependent on the angle θ that $\mathbf{v}$ makes with the streamlines of the potential represented by its gradient $\nabla p(\mathbf{r})$. In terms of θ we can write the r.h.s. of Eq. 1 as

$$\epsilon \mathbf{v} \cdot \nabla p(\mathbf{r}) \quad = \quad \epsilon |\nabla p(\mathbf{r})| \; \cos(\theta(\mathbf{r})) \tag{2}$$

For instance, if $\mathbf{v}$ is parallel to the gradient ($\theta(\mathbf{r}) = 0$), increasing a positive ϵ would in principle make the laplacian more positive, and the potential more concave with a faster ascent (see Figure 1). But, in 2D, a change in ϵ might also change the streamlines and therefore the angle $\theta(\mathbf{r})$. Observe that ϕ is a controllable parameter, but $\theta(\mathbf{r})$ is a function of ϵ and ϕ. This coupling makes hard to determine in advance suitable values for ϕ, specially if the environment is complex.

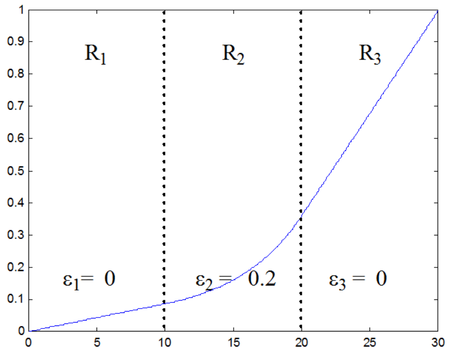


Fig. 1. Unidimensional mesh divided in 3 Regions: R_1, R_2, R_3. The regions R_1, R_2, R_3 have $\epsilon_1 = 0$, $\epsilon_2 = 0.2$, and $\epsilon_3 = 0$.

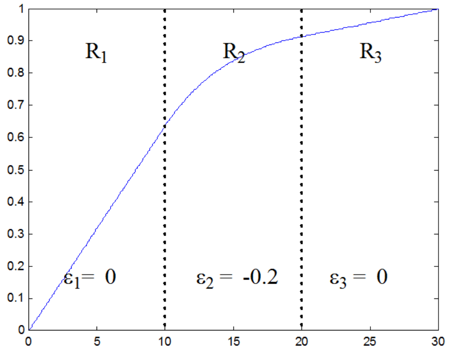


Fig. 2. Unidimensional mesh divided in 3 Regions : R_1, R_2, R_3. The regions R_1, R_2, R_3 have $\epsilon_1 = 0$, $\epsilon_2 = -0.2$, and $\epsilon_3 = 0$.

In order to avoid this incertitude we proposed in [8] to set $\mathbf{v}$ parallel to the streamlines of the harmonic potential of the environment. Then, Eq. 1 was changed to

$$\nabla^2 \; p(\mathbf{r}) \;=\; \epsilon(\mathbf{r}) \; \mathbf{v}(\mathbf{r}) \cdot \nabla p(\mathbf{r}), \tag{3}$$

where $\epsilon : \Re^2 \rightarrow \Re$ associates a specific distortion intensity to the position $\mathbf{r}$; $\mathbf{v}(\mathbf{r}) = \nabla p_h(\mathbf{r}) / |\nabla p_h(\mathbf{r})|$ is the input vector, defined previously, and $\nabla p_h(\mathbf{r})$ is the gradient of the harmonic potential at position $\mathbf{r}$. The solution of Eq. 3 is obtained computing first the Harmonic Potential through the Eq. 3 with $\epsilon(\mathbf{r}) = 0 \; \forall \mathbf{r}$ and saving its vector field as $\mathbf{v}$. After, we use this field together with the desired function $\epsilon(.)$ to calculate $p(\mathbf{r})$.

Like in 1D, Eq. 3 relies on a single parameter. When $\epsilon(.)$ is positive the potential concavity increases, i.e., potential is steeper near obstacles and flatter close to the goal. Whereas when $\epsilon(.)$ is negative the potential convexity increases, i.e., the potential is flatter near obstacles and steeper close to the goal. Figures 3(b), (c) and (d) shows the level curves of the potential when $\epsilon(\mathbf{r}) = 0$, $\epsilon(\mathbf{r}) = 0.2$ and $\epsilon(\mathbf{r}) = -0.2$,

[1] In this work, convexity and concavity are defined as if the potential is the surface of a solid object.

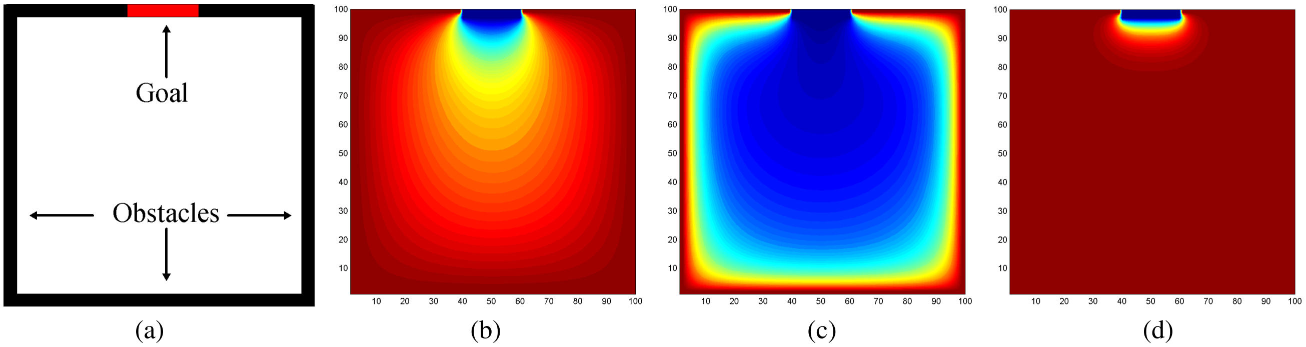


Fig. 3. Level curves of the potential in a (a) Test environment using (b) $\epsilon = 0$, which corresponds to Laplace Equation, (c) $\epsilon = 0.2$ and (d) $\epsilon = -0.2$.

respectively, over the environment ($\forall \mathbf{r}$) shown in Figure 3(a). This environment is limited by four walls, that correspond to obstacles, and contains a goal region at the top wall. When $\epsilon(.) > 0$, we observe that the potential in Figure 3(c) decays quickly from the obstacles augmenting the influence of the goal, indicated by regions where the potential is close to 0 (blue zone). When $\epsilon(.) < 0$, the influence of the goal is diminished as we can see in Figure 3(d).

The results in [8] showed that through the change of curvature we can manipulate the region traversing preferences. A low preference region can be created by locally increasing its potential convexity, while higher preference is linked to higher concavity. Fig. 4(a) shows a set of navigational paths to an environment similar to the one in Fig. 3 (a). In this environment, $\epsilon(\mathbf{r}) = -1.2$ for the dark region (indicating that this region has low preference) and $\epsilon(\mathbf{r}) = 0$ for the white region. We can see that some paths produced by Eq. 3 can cross low preference regions, even for low values of $\epsilon(.)$. Observe that to avoid local minima, we should use $|\epsilon(.)| < 2$, as discussed in [5] .

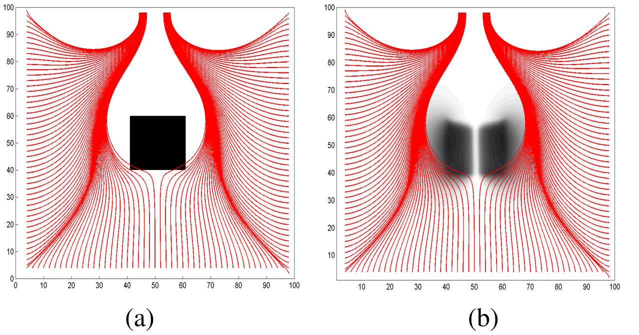


Fig. 4. Environment with one low preference region (dark region). (a) paths (in red) followed by robot to reach the goal at top. (b) paths (in red) followed by robot and the cosine(in gray scale) between the vector field generated by Eq. 3 and the vector field generated by Harmonic Functions.

The solution to these drawbacks is obtained making a deep analysis of Eq. 3 and rewriting it as

$$\nabla^2 p(\mathbf{r}) = \epsilon(\mathbf{r})|\nabla p(\mathbf{r})| \cos(\theta_h(\mathbf{r})) \quad (4)$$

where $\theta_h(\mathbf{r})$ is now the angle between the distorted potential and the harmonic potential streamlines. The factor $|\nabla p(\mathbf{r})|$ is necessary to avoid local minima, since the concavity/convexity diminishes or decreases proportionally to the variation of the potential. But the factor $\cos(\theta_h(\mathbf{r}))$ can be a nuisance in the regions where the desired potential vector field is approximately orthogonal to the harmonic potential vector field. Figure 4(b) shows in gray scale the value of the cosine between these vector fields for the example of Figure 4(a). White and black correspond to cosine equals to 1 and 0, respectively. In the dark regions in Fig. 4(b), the right side of Eq. 4 is near to 0 and as consequence we loose the curvature control, what causes some paths to cross the undesired region. To circumvent this problem, in this paper we propose to eliminate the term $\cos(\theta_h(\mathbf{r}))$ altogether from Eq. 4 changing it to

$$\nabla^2 p(\mathbf{r}) = \epsilon(\mathbf{r})|\nabla p(\mathbf{r})| \quad (5)$$

what is equivalent to assume that $\mathbf{v}(\mathbf{r})$ is parallel to $\nabla p(\mathbf{r})$.

Since the computation of $|\nabla p(\mathbf{r})|$ is numerically time consuming, here we use the triangular inequality,

$$|\nabla p(\mathbf{r})| \leq \left|\frac{\partial p(\mathbf{r})}{\partial x}\right| + \left|\frac{\partial p(\mathbf{r})}{\partial y}\right|$$

to obtain a more efficient equation,

$$\nabla^2 p(\mathbf{r}) = \epsilon(\mathbf{r}) \left(\left|\frac{\partial p(\mathbf{r})}{\partial x}\right| + \left|\frac{\partial p(\mathbf{r})}{\partial y}\right| \right) \quad (6)$$

Observe that r.h.s. of Eq. 6 is still a homogeneous function of the potential gradient , so it does not present local minima.

A. Algorithm

The basic relaxation algorithm is simple and can be executed in parallel with other modules of the robotic system. Basically, it iteratively computes the harmonic potential (Line 10) until its convergence. After, the function $\epsilon(c)$ is updated(Line 14-18) by the preference defined by the user, or computed by the robot according to the information gathered by its sensors. This function is used to distort the harmonic potential (Line 12). This relaxation algorithm can be easily integrated in the step 3 of the algorithm presented in [9]. That makes our method a solution for path planning problem in known environment or for exploration and mapping of unknown environment.

In the algorithm, each cell c has a position (x, y) in the mesh; a potential $p(c)$ and a parameter $\epsilon(c)$ according to Equation 6. Cells c_n, c_s, c_w, c_e have position $(x, y+1)$, $(x, y-1)$, $(x-1, y)$, $(x+1, y)$, respectively. The value $h(c)$ corresponds to the harmonic potential, $\nabla^2 p(c) = 0$, and the term $\epsilon(c)d(c)/4$ corresponds to the distortion $\epsilon(\mathbf{c}) \left(\left| \frac{\partial p(\mathbf{c})}{\partial x} \right| + \left| \frac{\partial p(\mathbf{c})}{\partial y} \right| \right)$.

Algorithm 1 Basic Algorithm

```
for all cell $c$ that does not contain the goal do
    set its potential value to high potential        ▷ $p(c) \leftarrow 1$
    initialize $\epsilon(c) \leftarrow 0$
end for
for all cell $c$ that contain the goal do
    set its potential value to low potential         ▷ $p(c) \leftarrow 0$
end for
while TRUE do
    for all cell $c$ that represents an environment free-space do
        $h(c) \leftarrow \frac{1}{4}(p(c_n) + p(c_s) + p(c_w) + p(c_e))$
        $d(c) \leftarrow \left( \left| \frac{p(c_n) - p(c_s)}{2} \right| + \left| \frac{p(c_w) - p(c_e)}{2} \right| \right)$
        $p(c) \leftarrow h(c) - \frac{\epsilon(c)}{4} d(c)$
    end for
    if potential converged then
        for all cell $c$ do
            $\epsilon(c) \leftarrow preference(c)$
        end for
    end if
end while
```

The algorithm extension to handle dynamic environment or preferences collected in real-time by the robot is straightforward. We need only to add lines to save the cells preferences from the data collected by robot sensors or to save the new features in the map and update the potential of the corresponding cells.

III. Results

This section presents several results obtained in simulation to demonstrate the ideas discussed in the previous sections. All experiments use the environment shown in Figure 3(a) represented by an array of 100×100 cells.

A. Experiment with Laplace Equation

Figure 5 shows an experiment using Laplace Equation. Several paths(black lines) starting near the bottom wall are displayed. We can see that the robot tends to approach the environment center, since it diminishes the hitting probability with obstacles.

In what follows we discuss the role of the distortions in concavity/convexity and its relation with preferences in crossing a region. In order illustrate that, we produce in Figures 6 and 7 the two-dimensional equivalents of Figures 1 and 2. In these figures, a small square region is introduced at the center of the environment shown in Figure 3(a). In Figure 6 we investigate the effect in the paths when we distort the potential using $\epsilon > 0$ while in Figure 7, $\epsilon < 0$ is used.

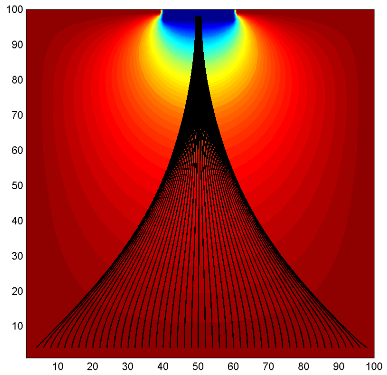


Fig. 5. Experiment with Laplace Equation. Paths (black lines) produced by Laplace Equation from several robot starting positions.

B. Zone with High Preference (HP): $\epsilon > 0$

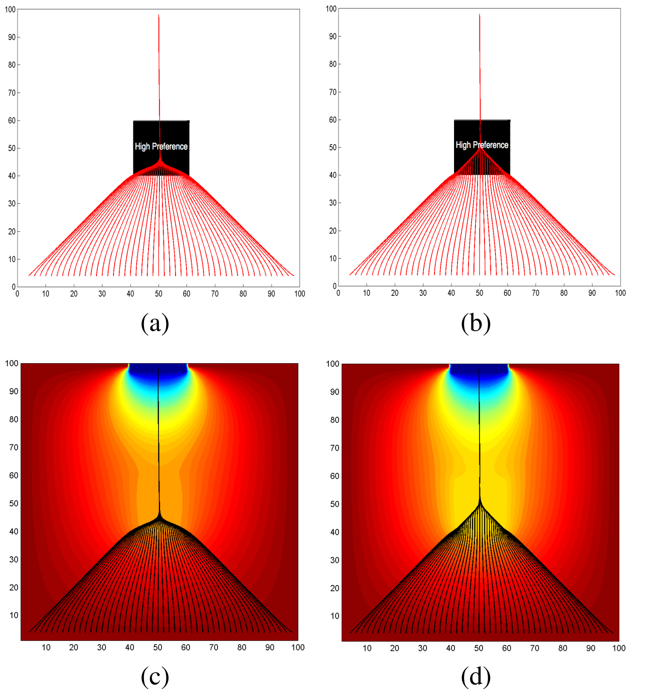
Figures 6 (a) and (b) show the paths followed by the robot using Eq. 3 and Eq. 6, respectively. In both cases, there exists a HP region at the environment center with $\epsilon(\mathbf{r}) = 1.2$ for all position $\mathbf{r}$ at this region. Observe that the concavity of the potential in this region is increased giving rise to an effective attractive force that pulls the trajectories towards this region. The bigger the ϵ the more the paths approach the center of the region. Figures 6(c) and (d) illustrate the paths (black lines) over the level curves of the potential field computed in (a) and (b), respectively. Observe that the low potential zone is stretched diminishing the high potential zone, as compared to the harmonic potential (Figure 5). The distortion produced by Eq. 6 , show in (c), is stronger than the distortion generated by Eq. 3, show in (d). As a consequence, this region becomes still more attractive since the potential decays more quickly augmenting the influence of the goal.

C. Zones with Low Preference (LP): $\epsilon < 0$

Figures 7(a) and (b) show the paths followed by the robot using Eq. 3 and Eq. 6, respectively, in an environment with a low preference region with $\epsilon(\mathbf{r}) = -1.2$. We can observe that increasing the convexity in a region decreases its preference for path planning. Figures 7(c) and (d) illustrate the paths (black lines) over the level curves of the potential field computed in (a) and (b), respectively. The high potential zone is stretched diminishing the low potential zone, as compared to the harmonic potential (Figure 5). The paths tend to avoid the convex region. Again, we can see that Eq. 6 produces a distortion stronger than Eq.3 for the same ϵ value. Besides, in (a), we can see that some paths cross the low preference region. In (b), this does not happen and the robot changes its direction before getting too close to the low preference region.

D. Several zones of different preference levels

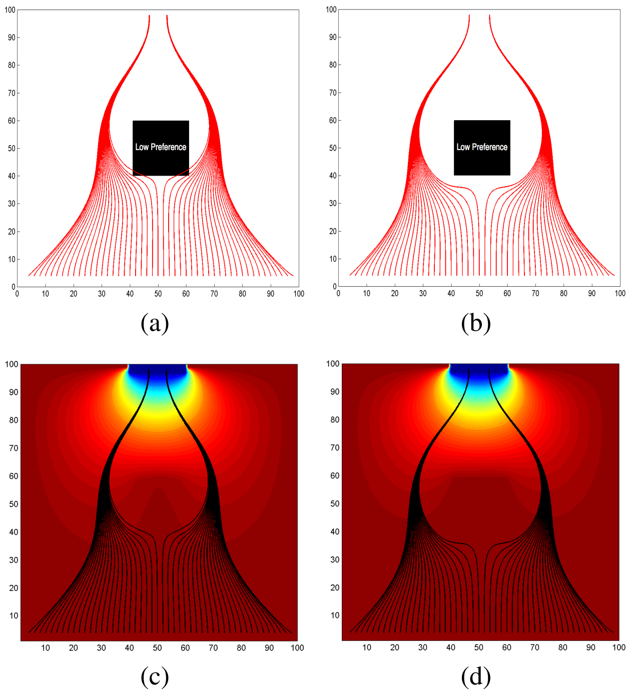
Figures 8(a) and (b) show an experiment in an environment with a barrier of alternating LP and very low preference (VLP) zones using Eq. 3 and Eq. 6, respectively. In both cases, $\epsilon(\mathbf{r}) = -0.6$ at the LP region and $\epsilon(\mathbf{r}) = -1.2$ at the VLP region. Observe the robot crosses LP regions to reach



(a) (b)

(c) (d)

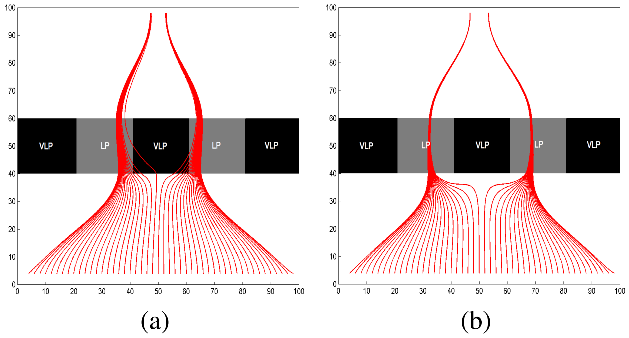
Fig. 6. Environment with one high preference region. This figure shows the path followed by robot to reach the goal at top using the potential field computed with (a) Eq. 3 and (b) Eq. 6 . In both cases, $\epsilon(\mathbf{r}) = 1.2$ for all position $\mathbf{r}$ at high preference region. Whereas (c) and (d) show the level curves of the potential field used in (a) and (b), respectively.



(a) (b)

(c) (d)

Fig. 7. Environment with one low preference region. This figure shows the path followed by robot to reach the goal at top using the potential field computed with (a) Eq. 3 and (b) Eq. 6 . In both cases, $\epsilon(\mathbf{r}) = -1.2$ for all position $\mathbf{r}$ at low preference region. Whereas (c) and (d) show the level curves of the potential field used in (a) and (b), respectively.



(a) (b)

Fig. 8. Environment with alternating zones of low (LP) and very low preference(VLP). This figure shows the path followed by robot to reach the goal at the top using the potential field computed with (a) Eq. 3 and (b) Eq. 6. In both cases, $\epsilon(\mathbf{r}) = -0.6$ at the LP region and $\epsilon(\mathbf{r}) = -1.2$ at the VLP region.

the goal, since the environment does not provide a viable path that avoids them[2]. This indicates that what determines the region to be crossed is the relative convexity/concavity between regions and not its absolute value. In (b), the robot tends to pass in the middle of LP region avoiding the VLP region. In (a), some paths cross the VLP region and others pass very close to it. This happens mainly due to the cosine term discussed in Section II. This term diminishes the distortion applied to the harmonic potential field, and as result we do not obtain a strong "repulsion force".

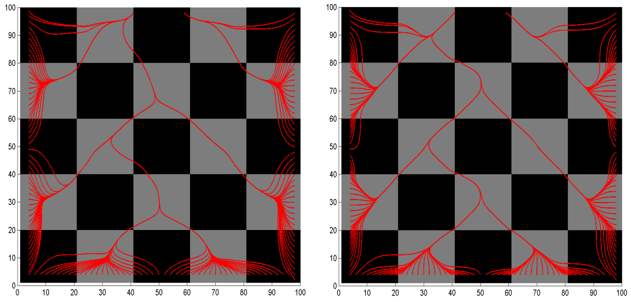
Figures 9 and 10 illustrate the differences between the paths generated by Eq. 3 and Eq. 6 in two different environments with higher complexity. Figure 9 presents a checkerboard configuration of LP (dark) and HP (gray) zones. Whereas Figure 10 shows an example of an environment with a random distribution of LP (white) and HP (dark) zones. To permit a better visualization of the paths and the regions, we have only two preference degrees. In Figure 9, $\epsilon = 1.0$ for HP and $\epsilon = -1.0$ for LP zones. While in Figure 10, $\epsilon = 1.0$ for HP and $\epsilon = -1.5$ for LP zones.

In both figures, Eq. 3 produces good results controlling the region convexities(concavities) of the potential field. However, the results produced by Eq. 6 are still better. For instance, in Figures 9(a) and 10(a), Eq. 3 makes the robot to travel big distances over LP zones, while in Figures 9(b) and 10(b), Eq. 6 minimizes expressively the amount of steps in the LP zones, leading the robot whenever it's possible to HP zones. In some cases, the robot has to cross LP zones, because there is no viable path towards the goal that crosses only HP zones. These paths are exclusively produced by the concavity and convexity of the regions since this environment has no obstacles.

IV. Conclusions

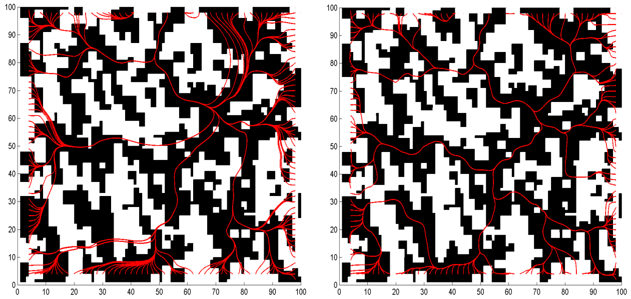
In this work, we introduce a new PDE to be used as core of our BVP Path Planner. This new equation eliminates the need of prior knowledge of the vector field to be used

[2] If these regions were considered obstacles, the potential below them would be flat with a null vector field that does not provide any information for the robot to navigate.



(a) (b)

Fig. 9. Environment with a checkerboard configuration of LP (dark) and HP (gray) regions. In red the path followed by robot to reach the goal at the top using the potential field computed with (a) Eq. 3 and (b) Eq. 6 . In both cases, $\epsilon(\mathbf{r}) = -1.0$ at the LP region and $\epsilon(\mathbf{r}) = 1.0$ at the HP region.



(a) (b)

Fig. 10. Environment with a random distribution of HP (dark) and LP (white) regions. In red the path followed by robot to reach the goal at the top using the potential field computed with (a) Eq. 3 and (b) Eq. 6 . In both cases, $\epsilon(\mathbf{r}) = -1.5$ at the LP region and $\epsilon(\mathbf{r}) = 1.0$ at the HP region.

as input to the PDE. That corresponds to an advantage over our last proposal discussed in [8], since the use of a static vector field minimizes the intensity of the attraction or repulsion force associated to the HP and LP zones, respectively. As consequence, this new equation generates better results minimizing the amount of robot steps in LP regions.

The attraction and repulsion forces are defined based on local convexity and concavity, respectively. Two adjacent regions compete based on their potential curvatures, and the winner is the one where the potential decays more quickly towards the goal potential. This competition can happen with regions both convex or both concave. What defines if a region is more preferred than another is its relative curvature. The preference of a region does not imply a rigid constrain in the sense the boundary conditions are. A LP region is different from an obstacle since it can be crossed by the robot if the local conditions of the potential demand it. This fact can be seen in Figure 8

Our proposal is not constrained to path planning in inhomogeneous environments, i.e., weighted region problem[16], but it can be integrated to exploratory tasks [5], [9] and other domains like robotic soccer[6] or simulation of pedestrians behavior in realistic outdoors scenes[7]. Basically, in all these domains, local control of convexity/concavity permits to endow the robot or agent with different behaviors according to environment features or sensors information.

As the potential field needs several steps to converge, the method presented in this paper is expensive computationally. To atenuate this problem, in [17], we propose a strategy that computes the potential field hierarchically using the Full Multigrid Method[18].

Acknowledgment

The authors thank the financial funds provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico-CNPq-Brasil.

References

[1] C. I. Connolly, “Harmonic functions and collision probabilities,” *International Journal of Robotic Research*, pp. 497–507, 1994.

[2] C. I. Connolly and R. A. Grupen, “On the application of harmonic functions to robotics,” *Journal of Robotic Systems*, vol. 10, pp. 931–946, 1993.

[3] S. A. Masoud and A. A. Masoud, “Motion planning in the presence of directional and obstacle avoidance constraints using nonlinear anisotropic, harmonic potential fields - a physical metaphor,” *IEEE Transactions on Systems, Man and Cybernetics, Part A - Systems and Humans*, vol. 32, no. 6, 2002.

[4] A. A. Masoud, “Managing the dynamics of a harmonic potential field-guided robot in a cluttered environment,” *IEEE Transactions on Industrial Eletronics*, vol. 56, no. 2, 2009.

[5] M. Trevisan, M. A. P. Idiart, E. Prestes, and P. M. Engel, “Exploratory navigation based on dynamic boundary value problems,” *Jounal of Intelligent and Robotic Systems*, vol. 45, no. 2, pp. 101–114, 2006.

[6] G. Faria, E. Prestes, M. A. P. Idiart, and R. A. F. Romero, “Multi robot system based on boundary value problems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 424–429.

[7] R. Silveira, F. Dapper, E. Prestes, and L. Nedel, “Natural steering behaviors for virtual pedestrians,” *The Visual Computer (available online)*, 2010.

[8] E. Prestes and M. Idiart, “Sculpting potential fields in the bvp path planner,” in *IEEE International Conference on Robotics and Biomimetics(ROBIO)*, 2009, pp. 183–188.

[9] E. P. e Silva Junior, P. M. Engel, M. Trevisan, and M. A. Idiart, “Exploration method using harmonic functions,” *Robotics and Autonomous Systems*, vol. 40, no. 1, pp. 25–42, 2002.

[10] E. Prestes, M. Ritt, and G. Fhr, “Improving monte carlo localization in sparse environments using structural environment information,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[11] L. Freda and G. Oriolo, “Frontier-based probabilistic strategies for sensor-based exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[12] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using rao-blackwellized particle filters,” in *Proceedings of Robotics: Science and Systems*, Cambridge, 2005.

[13] C. Leung, S. Huang, and G. Dissanayake, “Active slam using model predictive control and attractor based exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[14] F. Amigoni, “Experimental evaluation of some exploration strategies for mobile robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[15] K. Binmore and J. Davies, *Calculus Concepts and Methods*. Cambridge University Press, 2002.

[16] J. S. B. Mitchell and C. H. Papadimitriou, “The weighted region problem,” in *Third Annual Symposium on Computational Geometry*, 1987.

[17] R. Silveira, E. Prestes, and L. P. Nedel, “Fast path planning using multi-resolution boundary value problems,” in *Accepted for publication in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[18] A. Brandt, “Multi-level adaptive solutions to boundary-value problems,” *Mathematics of Computation*, vol. 31, pp. 330–390, 1977.