2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Exploration driven by Local Potential Distortions

Edson Prestes and Paulo Martins Engel
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre - RS - Brazil
{prestes,engel}@inf.ufrgs.br

Abstract— **Boundary Value Problem (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. A BVP planner has been recently used to compute trajectories in known inhomogeneous environments, corresponding to different degrees of traveling difficulties. In this case, we locally distort the potential field generating regions with high or low preferences for navigating. In this paper, we extend this idea, generating an strategy allowing the robot to explore an unknown environment dynamically considering environment preferences. Several experiments demonstrate that our method can be used as a core in an integrated exploration approach.**

I. Introduction

Environment exploration is the task of active map building, involving the process of knowledge-acquiring from the robot sensors signal to create an environment map, and the exploitation of this knowledge to guide the robot through unexplored regions of the environment, corresponding to the controller that implements an exploration strategy. In this paper we present an exploration-exploitation strategy to explore an unknown environment taking into account environmental features that impose a certain preference on the regions to be explored. Our approach is based on a potential field that can be distorted by means of a parameter that locally specifies a distortion intensity. Although the distortion parameter is dynamically and locally acquired, our approach based on the incremental calculation of a parameterized potential function doesn't produce local minima. The sign of the distortion parameter regulates the potential curvature, turning it convex or concave corresponding respectively to a negative or a positive distortion intensity.

Our exploration strategy is an extension of the Path Planner based on Boundary Value Problem [1]. The core equation of this planner has a term that is a homogenous function of the potential gradient. With this 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 scenarios like robotic soccer [2]. Other interesting property is that in a hierarchy of preferred regions with different difficulty of traveling, the robot will always choose to pass in the more preferred region.

The environment preferences can be used together with localization strategies to improve robot localization in problems of global localization [3]. In our case, we use the preferences to guide the robot in its mapping task favoring certain regions. We intend to integrate our proposal with SLAM in order to generate an Integrated Exploration Approach. As discussed in other works, integrated exploration techniques [4], [5], [6], [7], [8] associate to each environment frontier an utility measure that provides an estimate of the environment information gain and/or the localization accuracy at that position [9]. This information is used by the robot during its exploratory task. These techniques tend to produce more quality maps than the maps generated by solution that does not consider localization, mapping, exploration altogether.

This paper is divided as follows. In Section II, we present the fundamentals of our BVP Path Planner. In Section III, we discuss our approach to explore unknown environments taking into account environment preferences. In Section IV, we present simulation results obtained in different environments and finally Section V presents our conclusions.

II. Theoretical Approach

The original BVP path planner [10] generates paths using the potential information computed from the numeric solution of

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

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 [10], [11]. After the potential convergence, the robot uses the gradient descent of this potential to reach the goal position.

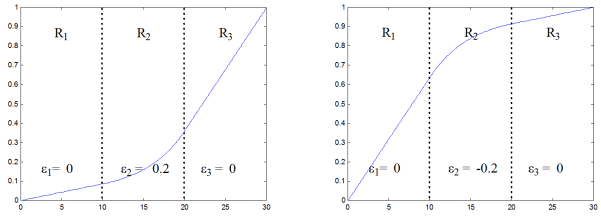
Eq. 1 indicates that, different from Harmonic Potentials, the potential $p(\mathbf{r})$ is locally convex or concave[1]. The Lapla-

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

978-1-61284-456-5/11/$26.00 ©2011 IEEE

cian 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 [12].

The effect of changing the curvature of a limited region can be better visualized in a 1D environment as shown in Figures 1(a) and (b). 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.



(a) (b)

Fig. 1. Unidimensional mesh divided in 3 Regions: R_1, R_2, R_3. (a) Regions R_1, R_2, R_3 have $\epsilon_1 = 0$, $\epsilon_2 = 0.2$, and $\epsilon_3 = 0$, respectively. (b) Regions R_1, R_2, R_3 have $\epsilon_1 = 0$, $\epsilon_2 = -0.2$, and $\epsilon_3 = 0$, respectively.

Recently, we [1] proposed a new version of Eq. 1 changing it to

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

which implies 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 involving just single components,

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

to obtain a more efficient equation,

$$\nabla^2 p(\mathbf{r}) = \epsilon(\mathbf{r}) \left(\left|\frac{\partial p(\mathbf{r})}{\partial x}\right| + \left|\frac{\partial p(\mathbf{r})}{\partial y}\right| \right) \tag{3}$$

Observe that the r.h.s. of Eq. 3 is still a homogeneous function of the potential gradient , so it does not present local minima. When $\epsilon(.)$ is positive the potential concavity increases, i.e., the 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 2(b), (c) and (d) show the level curves of the potential when

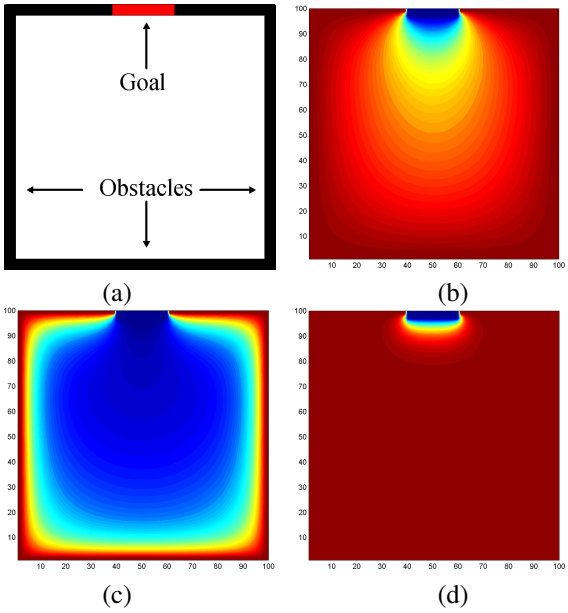


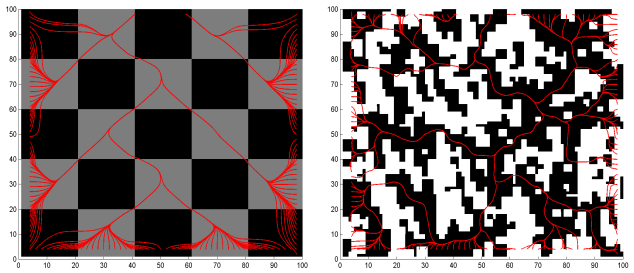
Fig. 2. Level curves of the potential in a (a) Test environment using (b) $\epsilon = 0$, which corresponds to Laplace Equation, (c) $\epsilon = 0.2$ and (d) $\epsilon = -0.2$.

$\epsilon(\mathbf{r}) = 0$, $\epsilon(\mathbf{r}) = 0.2$ and $\epsilon(\mathbf{r}) = -0.2$, respectively, over the environment ($\forall \mathbf{r}$) shown in Figure 2(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 2(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 2(d). The results in [1] 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.

Figure 3 illustrates the paths followed by the robot using Eq. 3 in two different environments. Figure 3(a) presents a checkerboard configuration of Low Preference (dark) and High Preference (grey) zones, LP and HP zones, respectively. Whereas Figure 3(b) 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 3(a), $\epsilon = 1.0$ for HP and $\epsilon = -1.0$ for LP zones. While in Figure 3(b), $\epsilon = 1.0$ for HP and $\epsilon = -1.5$ for LP zones. These paths are exclusively produced by the concavity and convexity of the regions since this environment has no obstacles.

III. Exploration Method using Preferences

Our approach for exploration is based on the incremental calculation of a potential function that defines an activation window, that travels with the robot, with roughly the size of its average sensor range [10]. This window indicates the grid cells that are recruited for update. The set of activated cells



(a) (b)

Fig. 3. Path followed by the robot in environments with (a) checkerboard configuration of LP (dark) and HP (gray) regions and (b) random distribution of HP (dark) and LP (white) regions.

that compose the explored space we call activated region or potential region. Cells that were never inside the activation window indicate unexplored regions. Their potential values are set to zero and define the boundary(frontier) between explored and unexplored space. Obstacles boundaries are set to 1. At same time, the robot analyzes the environment regions and determines if they should be classified as HP or LP region, associating a specific ϵ value. This value is used during the potential calculation to distort the potential field in order to generate regions more or less attractive.

As the robot moves according to the gradient of the potential, the detection of the nearest unexplored frontier comes naturally from the potential calculation. In our approach the potential region increases while the robot explores thus defining a global map.

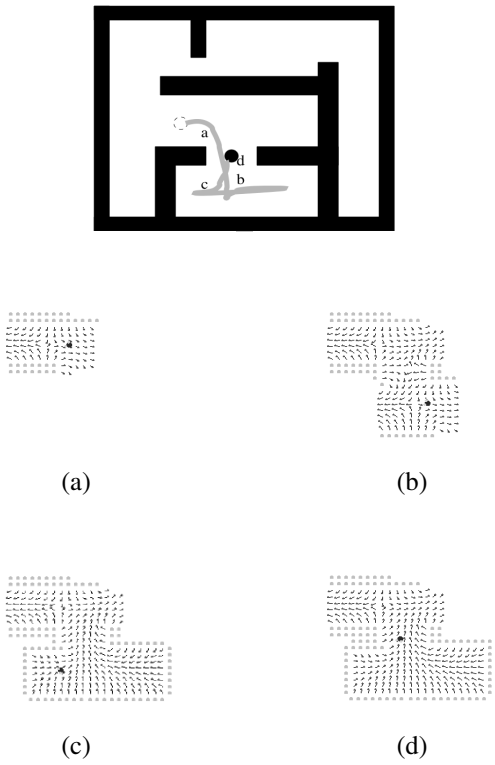
Figure 4 illustrates the exploratory behavior generated by our controller. The robot starts in the position indicated by the dashed circle, and sequentially occupies the positions a, b, c and d, with the final position indicated by the black circle. Figure 4 also shows snapshots of the internal map at the corresponding positions. At any instant, the vector field points to an unexplored place in the environment. In Figures 4(c) and (d) we see how the vector field changes pulling the robot from a dead-end.

A. Running

When the exploration begins all cells have their property state set to *Not Explored*; certainty set to 0 potential equals to 0, and $\epsilon = 0$ (see [10] for details about these parameters).

Throughout the exploration process the robot performs the algorithm described below:

1) Activate the sonar sensors.
2) Identify objects and perform local update of the map.
3) Determine the preferences of the regions in the map and associate a ϵ value to the map cells.
4) Update the potential over the visited positions.
5) Calculate the gradient vector in the position occupied by the robot.
6) Move itself in the direction defined by the gradient descent on the potential.
7) Repeat this procedure until all the environment is explored.



(a) (b) (c) (d)

Fig. 4. Exploration Process: On the top the trajectory followed by the robot. (a)-(d) show snapshots of the field and grid configurations for each corresponding position in the trajectory.

The robot activates the sonar sensors to get information over the environment around it. The acquired data are expressed in polar coordinates: range and position of the sensor in the ring. From these data, the robot updates the map.

Figure 5 shows how the sonar signal from a wall is interpreted and incorporated into the map. The sensor detects an object at a distance *d* from the robot. Using the knowledge about its own position and the direction to the obstacle the robot increases by 3 the certainty value of each cell in a region $\partial\gamma$, that corresponds to the limits of the sensors view cone. The other cells inside the view cone are decreased by 1. A cell has its attribute changed from *Dynamic* to *Static*, when its certainty value is bigger than 2. This process allows an easy treatment of dynamic and static objects. The cells inside the coneview are classified according to the environment preferences and have their ϵ parameter updated.

After the map updating, we perform the potential field update of the visited cells. Each cell c has a position (x, y) in the mesh; a potential $p(c)$ and a parameter $\epsilon(c)$ according to Eq. 3. Discretizing Eq. 3, we obtain an update equation to the potential of each cell c,

$$p(c) \leftarrow h(c) - \frac{\epsilon(c)}{4} d(c) \tag{4}$$

where $h(c) = \frac{1}{4}(p(c_n) + p(c_s) + p(c_w) + p(c_e))$ and $d(c) = \left(\left|\frac{p(c_n)-p(c_s)}{2}\right| + \left|\frac{p(c_w)-p(c_e)}{2}\right|\right)$. The value

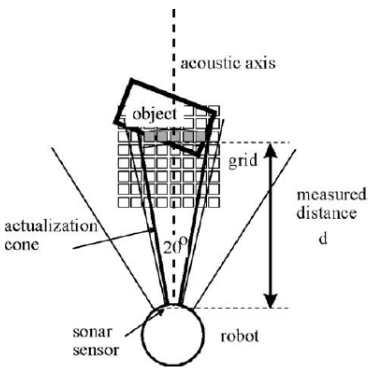


Fig. 5. Actualization of the grid: the figure shows an schematic diagram of an object detection by one of the robot sonar sensors.

$h(c)$ corresponds to the harmonic potential, $\nabla^2 p(c) = 0$, and the term $\epsilon(c)d(c)/4$ corresponds to the distortion $\epsilon(\mathbf{c})\left(\left|\frac{\partial p(\mathbf{c})}{\partial x}\right| + \left|\frac{\partial p(\mathbf{c})}{\partial y}\right|\right)$. The cells c_n, c_s, c_w, c_e have positions $(x, y+1)$, $(x, y-1)$, $(x-1, y)$, $(x+1, y)$, respectively.

A single actualization of the field consists in performing the update of all grid points inside the potential region. Here the procedure is repeated at least 10 times before the robot moves. After that, the negative gradient is computed for the cell containing the current position of the robot. Then the robot moves Δd after adjusting its head direction to the one defined by this gradient.

IV. Experiments

This section presents several results obtained in simulations to demonstrate the ideas discussed in the previous sections. We performed the experiments in the environments illustrated in Figure 6. These environments are represented using grids of the same size, i.e., 72×61 cells. The HP and LP regions are labeled as H_i and L_i, respectively, for $i \in \mathcal{N}$. The black regions correspond to obstacles while the white regions are free-space regions. In both cases, the cells's potential of these regions is not distorted ($\epsilon = 0.0$).

These environments have been chosen to analyze the robot behavior in a sparse area (Fig.6 (a)) and maze-like areas (Fig.6 (b) and (c)). We vary the ϵ parameter of the HP and LP regions, denoted by ϵ_H and ϵ_L, respectively, according to the following sets $\mathcal{L} = \{0.0, -0.2, -0.4, -0.6, -0.8, -1.0\}$ and $\mathcal{H} = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Observe that ϵ_H is used in all HP regions while ϵ_L is used in all LP regions. In each experiment, we select 10 robot starting positions at random and use them for each pair (ϵ_L, ϵ_H). The exploration process takes place and when the robot visits completely the environment, we compute the average number of robot steps and the number of robot visits in each LP/HP region.

At this moment, we are simulating the module of preference aquisition, step 3, commented in Section III-A . So, the robot will acquire the ϵ of a region when this region is under its sensor field.

A. Experiment in a Sparse Environment

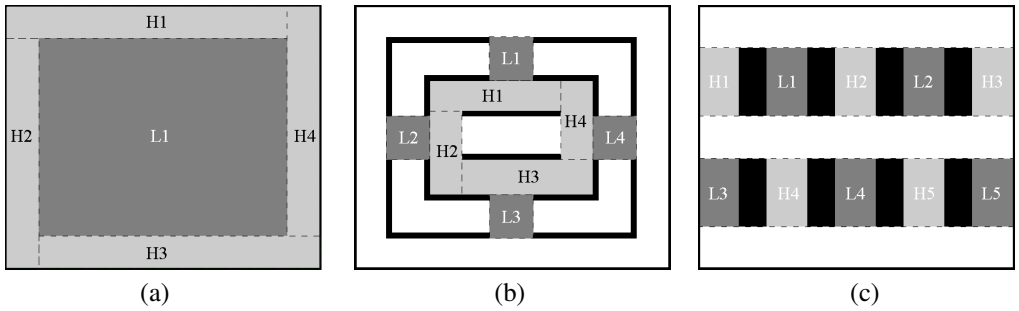
Due to space paper limitation, in this experiment, LP regions are associated only to the two extreme ϵ_L values : $\epsilon_L = 0$ and $\epsilon_L = -1.0$. For each ϵ_L, we varied ϵ_H value according to set $\mathcal{H}$, i.e., $\epsilon_H \in \mathcal{H}$. Figure 7(a) and (b) show the average number of robot steps and the number of robot visits, respectively, using $\epsilon_L = 0.0$. Figures 7(c) and (d) show the average number of robot steps and the number of robot visits, respectively, using $\epsilon_L = -1.0$. In Figures 7(a)-(d), the line corresponds to the results obtained with $\epsilon_L = 0.0$ and $\epsilon_H = 0.0$, i.e., all environment regions have the same preference for exploration. We observe that when $\epsilon_L = 0$, the robot navigates several steps inside region L_1. This region is sparse and does not help the robot to localize itself properly [3]. While the other regions $H_1 - H_4$ are disregarded even though these regions have been visited many times (Figure 7(b)). At each visit, the robot enters, moves a small number of steps and goes out. This happens because the potential field generates a vector field that pushes the robot towards the sparse region. This effect has been observed in our previous work [13] and results in an inefficient exploratory behavior.

In our experiments with $\epsilon_L = -1.0$, we observed that the sparse region became repulsive and the robot tended to move in regions with higher preferences, $H_1 - H_4$. The average number of steps in L_1 region diminished while this number increased in regions $H_1 - H_4$ (see Figure 7(c)). The average number of robot visits decreased (compare Figures 7(b) and (d)) because the robot tends to move more in a high preference region, even when an obstacle is detected. The remarkable increase in the number of visits for $\epsilon_H = 1$ is due to the oscillations in robot trajectory near the border of high and low preferences regions. In this experiment, the robot exhibited a behavior similar to wall-following.

B. Experiment in Maze 1

In this experiment, the robot moves in the environment shown in Figure 6(b). Similarly to the previous experiment, LP region has been assigned to only two ϵ_L values : $\epsilon_L = 0$ and $\epsilon_L = -1.0$. For each ϵ_L, we varied ϵ_H according to set $\mathcal{H}$. Figures 8(a) and (b) show the average number of robot steps and the number of robot visits, respectively, using $\epsilon_L = 0.0$. While Figures 8(c) and (d) show the average number of robot steps and the number of robot visits, respectively, using $\epsilon_L = -1.0$. In Figures 8(a)-(d), the line corresponds to results obtained with $\epsilon_L = 0.0$ and $\epsilon_H = 0.0$. We observe that setting $\epsilon_L = -1.0$, the average amount of robot steps in LP regions diminishes (compare Figures 8(a) and (c)). Whereas, the higher the ϵ_H, the bigger is the number of robot steps in HP regions. In this case, the robot tends to visit several times HP regions (see Figures 8(b) and (d)) while minimizing the visits of LP regions. The remarkable increase in the number of visits for $\epsilon_H = 1$ is due to the oscillations in robot trajectory near the border of high and low preferences regions, similarly to the experiments in the previous section.

In Figure 8(a), we see that there is a small difference between the number of robot steps in LP and HP regions,



(a) (b) (c)

Fig. 6. Environments used in the Experiments. All regions labeled as H_i are considered as High Preference regions and those labeled as L_i are Low preference regions, for $i \in \mathcal{N}$. (a) Sparse environment. (b) Maze 1. (b) Maze 2.

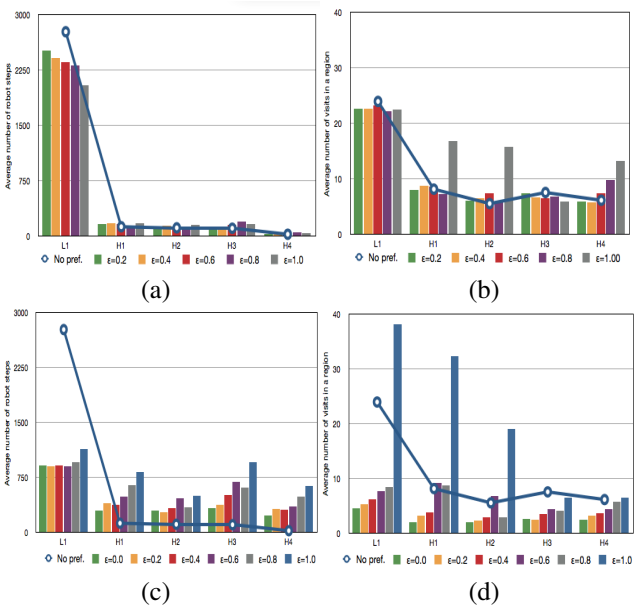


Fig. 7. Experiment in the sparse environment shown in Figure 6(a). L_i represents a LP region while H_i corresponds to HP region. Number of robot steps (a) and robot visits (b) using $\epsilon_L = 0$. Number of robot steps (c) and robot visits (d) using $\epsilon_L = -1.0$.

except when ϵ_H is near to 1. This happens because LP regions do not exert a repulsive force that pushes the robot away. Even in this case, the robot still prefer to visit HP regions. This preference augments insofar ϵ_H increases.

When the robot is inside a dense environment, we can expect that the best solution(or algorithm) should generate the shortest robot path to explore completely the environment. As discussed in several articles [4], [5], [6], [7], sometimes the robot needs to revisit specific places to minimize odometric errors and to generate consistent maps. For closing loops in cyclical environments the robot needs to revisit several times some places until the environment map is consistent. In our approach, when $\epsilon_H = \epsilon_L = 0.0$, all regions have the same preference and the robot tends to minimize only the time needed to explore all the environment using a greedy exploration. This case is equivalent to our approach based on harmonic functions [11]. When the robot associates a ϵ value to environment regions, it is breaking the balance of number of visits in the exploration process, giving, respectively, more or less attention to specific regions.

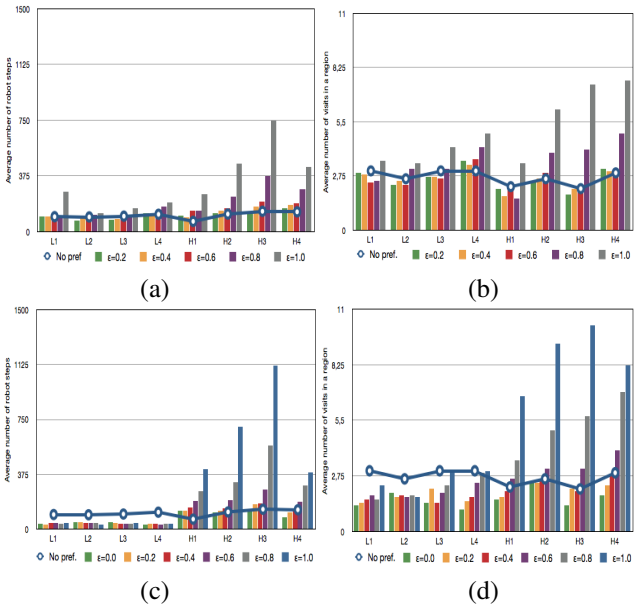


Fig. 8. Experiments in the Maze shown in Figure 6(b). L_i represents a LP region while H_i corresponds to HP region. Number of robot steps (a) and robot visits (b) using $\epsilon_L = 0$. Number of robot steps (c) and robot visits (d) using $\epsilon_L = -1.0$.

C. *Experiment in Maze* 2

In this experiment, the robot moves in the environment shown in Figure 6(c). In this experiment, the parameters ϵ_L and ϵ_H vary according to the sets $\mathcal{L}$ and $\mathcal{H}$, respectively. Figure 9(a) and (b) show the average number of robot steps using $\epsilon_L = 0.0$ and $\epsilon_L = -1.0$, respectively, and $\epsilon_H \in \mathcal{H}$. Figure 9(c) and (d) show the average number of robot steps using $\epsilon_H = 0.0$ and $\epsilon_H = 1.0$, respectively, and $\epsilon_L \in \mathcal{L}$. In Figures 9(a)-(d), the line corresponds to results obtained with $\epsilon_L = 0.0$ and $\epsilon_H = 0.0$. These results are equivalent to the one obtained with our approach based on harmonic functions [11].

In all figures, the average number of robot steps in LP regions is similiar, independent of the ϵ_L used. This ocurrs because these regions are repulsive but not prohibited for

navigation, then the robot tries to avoid them whenever it is possible, but it visits them only to complete its exploration task. An interesting situation appears in Figure 9(a) and (b), the smaller the ϵ_L the more attractive the HP regions. In Figure 9(a), the average number of steps in HP zones is smaller than the one in Figure 9(b). This effect is caused by the potential concavity of the LP region that increases and consequently, also increases the attraction force of the HP regions.

When $\epsilon_H = 0$ and $\epsilon_L \in \mathcal{L}$, the robot visits uniformly the environment (see Figure 9(c)). When $\epsilon_H = 1.0$, the HP regions are more prefered for visiting and the LP regions are crossed equally.

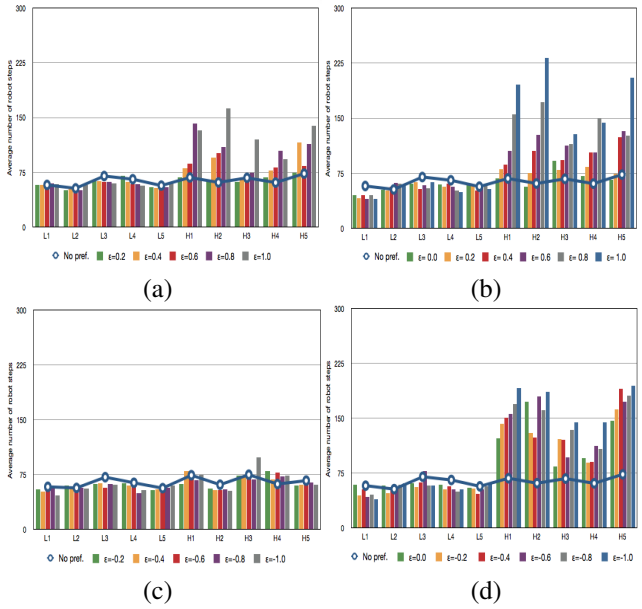


Fig. 9. Experiments in the Maze shown in Figure 6(c). L_i represents a LP region while H_i corresponds to HP region. Number of robot steps using (a) $\epsilon_L = 0$, (b) $\epsilon_L = -1.0$, (c) $\epsilon_H = 0$ and (d) $\epsilon_H = 1.0$

V. Conclusions

In this work, we present an extension of our BVP Path Planner [1] for exploratory tasks. This extension permits that a robot autonomously explores an unknown environment in an informed way. Instead of performing a greedy exploration, going always towards the nearest frontier, the robot analyzes the environment regions to identify that regions are more useful to be revisited. In our experiments, the module that analyzes these regions is not yet implemented. We simulated it in the following way. We have a prior knowledge of the importance of the region that should be explored and we associated a specific ϵ making some region more preferred for being visited than others. The robot obtains the preference information only when the region is under its sensor field. In the experiments, we observe that the use of environmental preferences impacts directly in the number of robot visits and robot steps in the regions. A HP is more frequently visited than LP regions. This can be seen in the experiments illustrated in Figures 7, 8 and 9.

When a region is defined as LP or HP, the potential field associated to it is distorted by a ϵ parameter that increases or decreases locally the potential convexity/concavity. As we see in [1], when $\epsilon(.) > 0$, the potential decays quickly from the obstacles augmenting the influence of the frontiers, i.e., the region is more attractive and pulls the robot with a greater 'attraction force". On the other hand, when $\epsilon(.) < 0$, the influence of the frontier diminishes and the region becomes less attractive to be visited. As the robot always follows the negative gradient of the potential, it will always tend to pass in regions where the potential decays quicker, in this case, the HP regions.

The use of HP and LP regions imposes in certain way a order for environment visitation. For instance, in the experiment shown in Figures 7, the robot exhibited a wall following behavior when $\epsilon_L \rightarrow -1.0$. When the preferences are not considered, i.e., $\epsilon_H = 0$ and $\epsilon_L = 0$, the effect is equivalent to our approach based on Harmonic Functions, i.e., an inefficient exploratory behavior [13].

Acknowledgment

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

References

[1] E. Prestes and M. Idiart, "Computing navigational routes in inhomogeneous environments using bvp path planner," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

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

[3] 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.

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

[5] 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.

[6] 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.

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

[8] L. Carlone, J. Du, M. E. K. Ng, B. Bona, and M. Indri, "An application of kullback-leibler divergence to active slam and exploration with particle filters," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 287 – 293.

[9] J. G. Jose-Luis Blanco, Juan-Antonio Fernndez-Madrigal, "An entropy-based measurement of certainty in rao-. blackwellized particle filter mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 3550 – 3555.

[10] 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.

[11] 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.

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

[13] E. Prestes, M. Idiart, P. M. Engel, and M. Trevisan, "Oriented exploration in non-oriented sparse environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.