Nonholonomic Motion Planning in Dynamically Changing Environments

Rodrigo Cuautle, Abraham Sánchez and Maria Osorio

Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570
Puebla, Pue., México
asanchez, aosorio@cs.buap.mx

Abstract. This paper presents a motion planner for mobile robots in dynamically changing environments with both static and moving obstacles. This planner is based on lazy PRM method and the reactive control by DVZ (Deformable Virtual Zone). The planner first computes a feasible free-collision path with respect to the static obstacles, using the lazy PRM method. Then, it uses the reflex commands in order to avoid dynamic changes. Experimental results are discussed to show the effectiveness of the proposed planner.

1 Introduction

The research in robot motion planning can be traced back to the late 60's, during the early stages of the development of computer-controlled robots. Nevertheless, most of the effort is more recent and has been conducted during the 80's. Within the 80's, roboticians addressed the problem by devising a variety of heuristics and approximate methods. Motion planning can be split into two classes: holonomic motion planning and non-holonomic motion planning.

In non-holonomic motion planning, any path in the free configuration space does not necessarily correspond to a feasible one. Non-holonomic motion planning turns out to be much more difficult than holonomic motion planning. This is a fundamental issue for most types of mobile robots.

From path planning to trajectory control, the motion planning problem for mobile robots has been thoroughly investigated in the case of structured environments. Moving among unknown or badly modeled environments, practically induces the necessity of taking unscheduled and dynamic events into account and reacting as the living beings would do. Therefore, reactive behaviors play a fundamental role when the robot has to move through unstructured and dynamic environments.

Artificial reflex actions for mobile robots can be defined as the ability to react when unscheduled events occur, for instance when they move in unknown and dynamic environments. For the last fifteen years, the scientific community has been interested in the problem of reactive behaviors for collision avoidance in the domain of mobile robots [1], [2]. Another important approach that it deals

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Artificial Intelligence
Research in Computing Science 20, 2006, pp. 155-166

with artificial reflex actions, is the potential method developed by O. Khatib, many years ago [3].

Probabilistic roadmap method (PRM) is a general planning scheme building probabilistic roadmaps by randomly selecting configurations from the free configuration space and interconnecting certain pairs by simple feasible paths. The method has been applied to a wide variety of robot motion planning problems with remarkable success [4], [5]. The adaptation of PRM planners to environments with both static and moving obstacles has been limited so far. This is mainly because the cost of reflecting dynamic changes into the roadmap during the queries is very high. On the other hand, single-query variants, which compute a new data structure for each query, deal more efficiently with highly changing environments. They however do not keep the information reflecting the constraints imposed by the static part of the environment useful to speed up subsequent queries.

In this trend, this work aims at providing a practical planner that considers reflex actions and lazy techniques to account for planning with changing obstacles. The paper is organized as follows. Section II gives an overview of the DVZ principle. Section III explains the details of the proposed planner. The performance of the planner is experimentally evaluated in Section IV. Finally, the conclusions and future work are presented in Section V.

2 The DVZ Principle

This section describes the DVZ principle. We assume that the mobile robots has no model of its surrounding space but can measure any intrusion of information (proximity-type information) at least in the direction of its own motion. The vehicle is protected by a risk zone while the deformations of the latter are directly used to trigger a good reaction.

The robot/environment interaction can be described as a deformable virtual zone (DVZ) surrounding the robot. The deformations of this *risk zone* are due to the intrusion of proximity information and controls the robot interactions. The *robot internal state* is defined to be a couple (Ξ, π), where the first component Ξ is called the *interaction component*, which characterizes the geometry of the deformable zone and the second component π characterizes the robot velocities (its translational and rotational velocities). In the absence of intrusion of information, the DVZ, denoted by Ξ_h is supposed to be a one-one function of π. The *internal control*, or reactive behavior is a relation ρ, linking these two components, $\Xi_h = \rho(\pi)$. In short, the risk zone, disturbed by the obstacle intrusion, can be reformed by acting on the robot velocities.

The geometric interaction between the moving n-dimensional robot and its moving n-dimensional environment, that is, the deformable zone surrounding the vehicle, can be viewed as an imbedding of the $(n-1)$-dimensional sphere S^{n-1} into the Euclidean n-dimensional space $\mathbb{R}^n$.

The main interest in the use of this formalism lies in the fact that each imbedding of S^{n-1} can be continuously transformed into another imbedding.

Thus, the deformations of the risk zone due to the intrusion of obstacles in the robot workspace or to the modifications of the robot velocities π (through the relation $\Xi = \rho(\pi)$) lead to the same mathematical entity (the imbedding). Fig. 1 shows different cases of the one-sphere deformations. These zones represent the various shapes of the DVZ, depending on the translational and rotational velocities of the robot. The first diagram illustrate a deformed DVZ due to the presence of an obstacle. The remaining diagrams show how the mobile robot can rebuild its DVZ, (b) by reducing the translational velocity, (c) by turning to the right, or (d) by turning to the left.

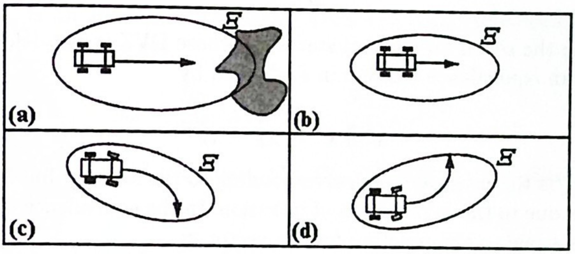

Fig. 1. Deformations of a 1-dimensional DVZ

The first cause of deformation in an interaction component is the information of intrusion due to the proximity of moving obstacles. The second cause is the internal control ($\Xi_h = \rho(\pi)$) of the robot for compensating this intrusion. The control of the internal state is done by comparing a reference interaction component Ξ_o with the deformed component Ξ. This reference depends on the accepted risk taken by the vehicle and is a matter of choice. Therefore, the reactive behavior can be modeled by a two-fold scheme. The control problem consists in generating the second deformation by internal control, with two possibilities:

- by integrally rebuilding the initial state interaction component Ξ through an action on the robot rotation (dynamic avoidance),
- by modifying the robot velocities to attain another acceptable stable state Ξ'.

2.1 Derivation of The State Equation

This subsection provides the general framework for the derivation of the state equation. This equation is formally seen as a two-fold control differential equation and imbedded in the theory of differential games.

Let $\chi = \binom{\Xi}{\sigma}$ be the vector that represents the internal state of the robot and let $\mathcal{E}$ be the state space, which is the set of all the vectors χ. The DVZ is defined by $\Xi = \begin{pmatrix} \Xi_1 \\ \Xi_2 \\ \vdots \\ \Xi_c \end{pmatrix}$ and the robot velocities vector σ is defined by $\sigma = \binom{v}{\dot\theta}$.

Where each component Ξ_i is the norm of the vector corresponding to the border's distance in the DVZ. These vectors belong to the straight lines that correspond to the main directions of the c proximity sensors c_i. Generally speaking, we assume that we can control the derivative ϕ of a function π for the robot velocities σ. Therefore, the control vector will be written

$$\phi = \dot\pi \tag{1}$$

Let $\mathcal{H}$ be the set of all internal states χ_h whose DVZ is not deformed. This set induces an equivalence relation in $\mathcal{E}$, defined by

$$\chi^1 \tilde{_{\mathcal{H}}} \chi^2 \Leftrightarrow \chi_h^1 = \chi_h^2 \tag{2}$$

where χ_h^i is the internal state corresponding to the state χ^i but without any deformation due to the information of intrusion. In the equivalence class $[\chi]$, the vector χ_h is a one to one function for the vector π:

$$\chi_h = \rho(\pi) \tag{3}$$

which can be written (by separation of the two sets of variables)

$$\begin{cases} \Xi_h = \rho_\Xi(\pi) \\ \sigma = \rho_\sigma(\pi) \end{cases} \tag{4}$$

The derivative of eq. (4) provides the state equation when no deformation occurs (when the state vector stays on $\mathcal{H}$):

$$\dot\chi_h = \rho'(\pi)\dot\pi = \rho'(\pi)\phi \tag{5}$$

This equation is the first part of the general state equation. If we now consider deformations of the DVZ, due to the intrusion of information, we will obtain the second part of the state equation. To do it, we will denote the deformation of the state vector by Δ and study the variations of this deformation with respect to the intrusion of information. This new vector represents the deformed DVZ, which is defined by

$$\Xi = \Xi_h + \Delta \tag{6}$$

Let $I = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_c \end{pmatrix}$ be the c-dimensional intrusion vector, where $I_i = d_{imax} - d_i$.

The sensor provides the measure $d_i = d_{imax}$, in the absence of obstacles.

Let $\Delta = \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_c \end{pmatrix}$ be the c-dimensional deformation vector, where

$$\Delta = \alpha(\Xi_h, I) \tag{7}$$

with $\alpha(\Xi_h, I)$ being a c-dimensional vector. Each element Δ_i is defined by

$$\Delta_i = \alpha(d_{h_i}, I_i) \begin{cases} 0 & \text{if } d_i > d_{h_i} \\ d_{h_i} - d_i & \text{if } d_i \le d_{h_i} \end{cases} \tag{8}$$

where d_{h_i} is an element of the intact DVZ (Ξ_h). By differentiating ec. (6) with respect to time, we get

$$\dot{\Delta} = \frac{\partial \alpha}{\partial \Xi_h}(\Xi_h, I)\dot{\Xi}_h + \frac{\partial \alpha}{\partial I}(\xi_h, I)\dot{I} \tag{9}$$

By letting $\psi = \dot{I}$, and using eqs. (4), (5), (6) and (9), we obtain the next control equation

$$\begin{cases} \dot{\Xi} = \left(\frac{\partial \alpha}{\partial \Xi_h}(\Xi_h, I) \times \rho'_{\Xi}(\pi) + \rho'_{\Xi}(\pi)\right)\phi + \frac{\partial \alpha}{\partial I}(\Xi_h, I)\psi \\ \dot{\sigma} = \rho'_{\sigma}(\pi)\phi \end{cases} \tag{10}$$

with

$$\begin{cases} \dot{\Xi} = \rho'_{\Xi}(\pi)\phi \\ \dot{\pi} = \phi \\ \dot{I} = \psi \end{cases} \tag{11}$$

The inputs of eq. (10) are the two control vectors ϕ and ψ. The first comes from the control module of the robot and the second from the environment itself.

3 A Reactive Lazy PRM Planner

The proposed planner integrates the lazy PRM planning method and the reactive control by DVZ in the following way: a collision-free feasible path for a mobile robot is calculated by the lazy PRM method, the robot starts moving (under the permanent protection of its DVZ), in the absence of dynamic obstacles, the control is performed by the lazy PRM method and does not require reflex commands. If there are dynamic obstacles in its path, the reactive method takes the control and generates commands to force the robot to move away from the intruder obstacles and gives back its DVZ to the original state.

In this point, the robot has lost its original path, and it is necessary to search for a reconnection path to reach its goal. The new path found is a single collision-free curve of Reeds & Shepp. If the attempt of reconnection is successful, the robot executes its new path towards the goal. The new alternative path was obtained with the lazy PRM method by using the information stored in the current robot's configuration, but if a deformation appears, the processes are interrupted by reflex actions that forces the planner to go back to the previous state. The algorithm can finish in three forms: i) the robot executes its path successfully, ii) the reflex action is not sufficient and a collision occurs, or iii) the robot does not find an alternative path to conclude its task. Figure 2 shows a high-level description of the proposed approach.

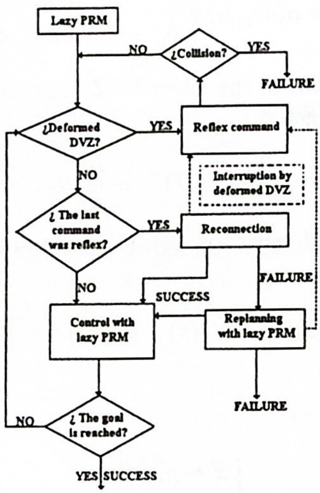

Fig. 2. High-level description of our planner

The following subsections detail the most important stages of the proposed planner. For more details, you can see [2].

3.1 Lazy PRM for Nonholonomic Robots

Lazy PRM approach for nonholonomic motion planning was presented in [6]. The algorithm is similar to the work presented by Bohlin and Kavraki [7], in the sense that the aim of our approach is to minimize the number of collision checks and calls to local method while searching the shortest feasible path in the roadmap.

Once a start-goal query is given, the planner performs A^* search on the roadmap to find a solution. If any of the solution edges are in collision, they are removed from the roadmap and then A^* search is repeated. Eventually, all edges may have to be checked for collisions, but often the solution is found before this happens. If no solution is found, more nodes may need to be added to the roadmap [8]. The most important advantage of this approach, is that the collision checking is only performed when needed. In this case, all edges don't have to be collision checked as in the original PRM case. Experiments show that only a very small fraction of the graph must be explored to find a feasible path in many cases. Single queries are handled very quickly, indeed, no preprocessing is required.

3.2 Generation of Reflex Commands

The DVZ form is used in our experimental design according to equations (1) to (12).

$$d_{h_i} = K_1 V_1^2 \cos^2(\beta_i + K_2\dot{\theta}) + d_i^{sec} \tag{12}$$

where K_1 and K_2 are constants, V_1 and $\dot{\theta}$ are the velocities of the robot, β_i is the angle of the sensor c_i with respect to the transverse axis of the robot, and d_i^{sec} is a safe distance in the direction of the sensor c_i.

For the first case in equation (8), ($d_i > d_{h_i}$), the DVZ is not deformed by the environment, the control is performed by the lazy PRM method and the reflex actions are not require. For the second case, when ($d_i < d_{h_i}$), a reflex action is necessary, the path executed by the lazy PRM method is suspended and the robot control is taken by the DVZ method. When the DVZ is in control, it has the task of taking the robot to a state free of deformations, indicating the kinematics attitudes that should continuously have the robot. These attitudes constitute the vector π, and the control is adapted in the following way.

Let $f_i[n]$ a vector in the direction of the sensor c_i to be defined as

$$f_i[n] = \begin{cases} \Delta_i[n] - \Delta_i[n-1] & \text{if } \Delta_i[n] - \Delta_i[n-1] > 0 \\ 0 & \text{if } \Delta_i[n] - \Delta_i[n-1] \leq 0 \end{cases} \tag{13}$$

Let $F[n]$ be the addition of the vectors $f_i[n]$

$$F[n] = \sum_{i=1}^{c} f_i[n] \tag{14}$$

then, the vector $\pi[n]$ is given by

$$\pi[n] = \begin{cases} V_1[n] = V_1[n-1] + Kv* \parallel F[n] \parallel *sign(\cos(\hat{F}[n])) \\ \dot{\theta} = \dot{\theta}[n-1] + Kt * \sin(\hat{F}[n]) \end{cases} \tag{15}$$

3.3 Reconnection

After a successful reflex action, the mobile robot recovers the intact state of its DVZ, but its initial planned path will be lost (Fig. 3-b). The lazy PRM method needs to have a path to push the mobile robot to the goal and it will be necessary to provide a path for such aim. Due to the high computational cost of a complete replanning, the method will avoid it by executing a process that uses a single collision-free Reeds & Shepp curve [9] (Fig. 3-c) to reconnect with the planned path.

Initially, the algorithm tries a local path that it is interrupted by a dynamic object. The algorithm will execute a reflex action in order to reconnect with the closest point that is collision-free in the original path. If it can not reconnect after a certain number of attempts, maybe because the possible reconnection paths are blocked with obstacles, the robot will remain immovable for a certain time before executing a new attempt (see Fig. 3-d). The process will be repeated several times, but if the DVZ was deformed by an intrusion, the reconnection process will be modified and will execute the reflex commands.

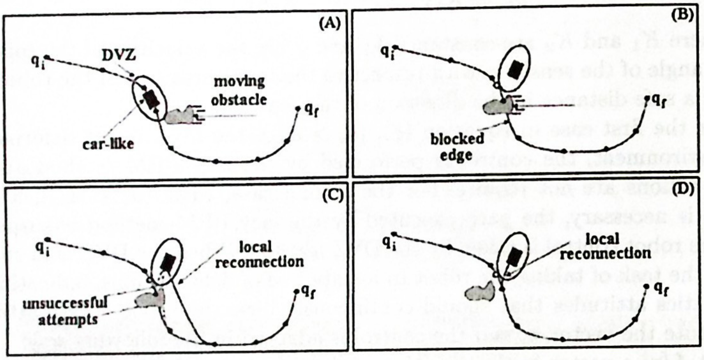

Fig. 3. Cases of the reconnection process

3.4 Replanning

If the reconnection attempts fails, it may happen that paths are blocked by many dynamic objects, or a moving object is parked obstructing the planned path. In this case, the planner executes the lazy PRM method (the initial configuration is the current configuration in the robot). The lazy PRM will be called several times until it returns a collision-free path. If after some attempts a collision-free path can not be found, the planner reports failure.

The model cannot distinguish if an intrusion is caused by a moving or a static obstacle because the DVZ method does not use any model of the environment. To solve this problem, it is necessary to use an auxiliary image that represents the environment and it is updated every time the replanning or reconnection procedures are called. When the sensors in the robot detect an obstacle that deforms the DVZ, the intruder object coordinates are revised to see if there was already an obstacle, registered in the auxiliary image; if this is the case, the system assumes the presence of a fixed obstacle and there is no need for a reflex action, otherwise, it will certainly assume that the object is in movement.

4 Experimental Results

This section presents experimental results for car-like robots obtained by using the planner described above to different scenes. The planner has been implemented in Builder C++ and the tests were performed on an Intel © Pentium IV processor-based PC running at 2.4 GHz with 512 MB RAM.

After having executed our planner in different scenes, in the majority of the cases the motion planning problem is solved satisfactorily. Our planner produces a first roadmap by sampling configurations spaces uniformly. It computes the shortest path in this roadmap between two query configurations and test it for collision. The robot starts moving under the permanent protection of its DVZ. In absence of dynamic obstacles, the robot does not require reflex commands and the control is executed with lazy PRM. If there are dynamic obstacles in its path, the reactive method takes the control and generates commands to force the robot to move away from the intruder obstacles and gives back its DVZ to the original state. The moving obstacles have a square form and move at constant velocity in straight line. Whenever they collide with another object they assume a new random direction in their movement.

Fig. 4 shows an environment that contains a circular obstacle, the scene is completely closed. This example also contains 10 dynamic obstacles moving randomly at the same velocity than the mobile robot.

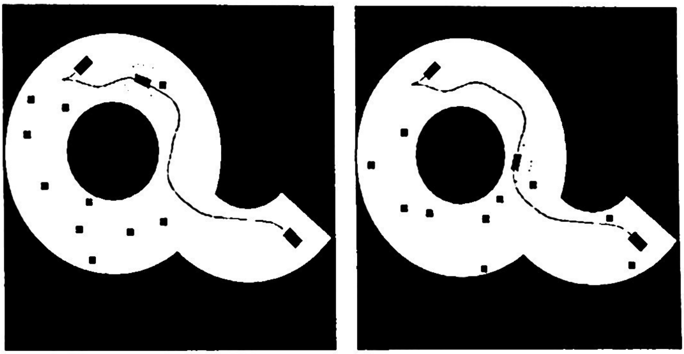

Fig. 4. An example of a query and its solution path in an environment with 10 moving obstacles. The robot starts moving under the permanent protection of its DVZ

In order to evaluate the performance of the planner, we performed tests on the environment of Fig. 5 for several roadmap sizes and different number of moving obstacles. The different settings are summarized in the tables 1, 2 and 3.

In fact, the method's performance can be considered satisfactory if it presents a fast planning phase, reflex actions based on sensors that do not require expensive algorithms, an effective process of reconnection performed in milliseconds,

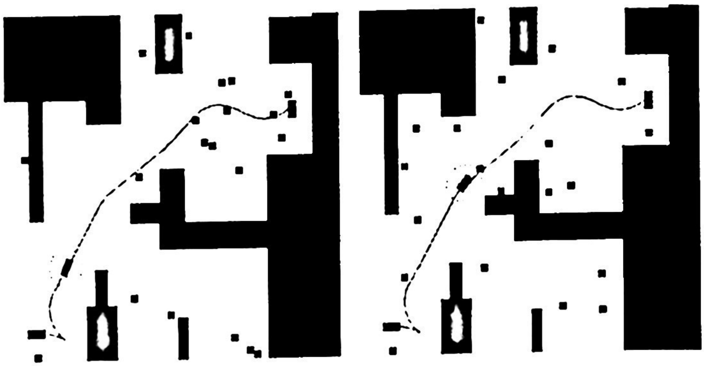

Fig. 5. An environment composed of narrow passages with 20 dynamic obstacles

Table 1. Performance data for Lazy PRM

Settings	50 nodes	50 nodes	50 nodes	100 nodes	100 nodes	200 nodes
Steering angle	25	35	45	35	25	45
Graph building	0.007	0.006	0.01	0.01	0.01	0.02
Graph searching	0.003	0.004	0.03	0.01	0.005	0.03
Coll. checking	380	425	1300	650	365	1481
Total Time (s)	0.01	0.01	0.04	0.02	0.015	0.05

Table 2. Performance data with 20 moving obstacles

Reconnections	Time for reconnection	Replanning	Time for replanning	Collision	Success
29	0.010	0	0.000	no	ok
39	0.015	2	0.000	no	ok
57	0.023	1	0.000	no	ok
5	0.010	0	0.000	no	ok
37	0.012	3	0.000	ok	no

Table 3. Performance data with 15 moving obstacles

Reconnections	Time for reconnection	Replanning	Time for replanning	Collision	Success
3	0.010	0	0.000	no	ok
10	0.020	0	0.000	no	ok
36	0.030	1	0.000	no	ok
40	0.040	2	0.000	no	ok
12	0.010	0	0.000	ok	no

and a process of replanning that is executed if the Lazy PRM and DVZ' s parameters are appropriate.

The planning time is reduced due to the incomplete collision detector whose work is complemented with the robot's sensors during the path execution. On the other hand, the assignation of direction angles to the nodes that conform the shortest paths obtained by the algorithm A^*, produces curves that allow the algorithm to omit the optimization process (i.e., the smoothing process). With respect to the reconnection process, the paths obtained with the planner are conformed by a single Reeds & Shepp curve and based on the incomplete collision detector, making short the time and close to optimal the curves obtained with the algorithm. Since the reflex actions are provided by the DVZ method, it is possible to interrupt the reconnection and replanning processes if necessary, without incurring in bigger problems. If the execution's parameters for the Lazy PRM and DVZ methods are adapted, the replanning process will not be called very often and will be successful in the absence of narrow passages.

Figure 6 presents a case where the reflex actions were not sufficient. The presence of narrow passages is an important problem to being considered.

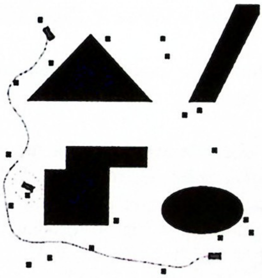

Fig. 6. The reflex actions were not sufficient, the mobile robot collides with a moving obstacle

5 Conclusions and Future Work

Even in the absence of obstacles, planning motions for nonholonomic systems is not an easy task. So far, no general algorithm exists for planning the motions of any nonholonomic system, that guarantees to reach a given goal. The only existing results deal with approximation methods, that is, methods that guarantees to reach a neighborhood of the goal, and exact methods for special classes of nonholonomic systems. Obstacle avoidance adds a second level of difficulty: not only does one have to take into account the constraints imposed by the kinematic nature of the system, but also the constraints due to the obstacles. It appears necessary to combine geometric techniques addressing the obstacle avoidance with control techniques addressing nonholonomic motions.

The results obtained in the evaluation of the reactive lazy PRM planner proposed in this work, show the importance of finding a solution for the complex problem of motion planning for nonholonomic robots in dynamic environments.

A reactive lazy PRM planner for dynamically changing environments is presented in this paper. Although some promising results are shown in its present form, the planner could be improved in a number of important ways. This approach can be extended to use real robots and to solve the problem posed by small static obstacles. Besides, some cases where the reflex action was not sufficient to avoid collisions were observed during the evaluation tests. Theses cases are difficult because they require a more intelligent behavior in order to avoid the robot to be trapped. In those cases, it can be necessary to add a process that computes the trajectories of moving objects and corrects the path in real time.

Finally, a very interesting topic in robotics, is the study of non-structured environments. This methodology can be extended to solve those cases.

References

1. Zapata R. "Quelques aspects topologiques de la planification de mouvements et des actions réflexes en robotique mobile", *Thèse d'Etat*, University of Montpellier II (in French), (1991)
2. Cuautle P. R. "Planificación de movimientos en ambientes dinámicos utilizando un enfoque lazy PRM reactivo", *Tesis de Maestria, FCC-BUAP*, (in spanish), (2005)
3. O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots", *International Journal of Robotic Research*, Vol. 5, No. 1, (1986) 90-98
4. Kavraki L. E., Švetska P., Latombe J. C., and Overmars M. H. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996) 566-580
5. Švestka P. and Overmars M. H. "Motion planning for car-like using a probabilistic learning approach", *The International Journal of Robotics Research*, Vol. 16, No. 2, (1997) 119-143
6. Sánchez A., Zapata R., Arenas A. "Motion planning for car-like robots using a lazy probabilistic roadmap method", *Advances in Artificial Intelligence, Springer Verlag, Vol. 2313*, (2002) 1-10
7. Bohlin R. and Kavraki L. E. "Path planning using lazy PRM", *Proc. of the IEEE Robotics and Automation Conference*, (2000) 521-528
8. Lanzoni C., Snchez L. A., and Zapata, R. "A single-query motion planner for car-like nonholonomic mobile robots", *3rd International Symposium on Robotics and Automation (ISRA)*, (2002) 267-274
9. Reeds J. A. and Shepp R. A. "Optimal paths for a car that goes both forward and backwards", *Pacific Journal of Mathematics*, Vol. 145, No. 2, (1990) 367-393
10. Sánchez A., Zapata R., and Lanzoni C. "On the use of low-discrepancy sequences in nonholonomic motion planning", *Proc. of the IEEE Robotics and Automation Conference*, (2003) 3764-3769