We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



23

Reactive Motion Planning for Mobile Robots

Abraham Sánchez[1], Rodrigo Cuautle[1], Maria A. Osorio[1] and René Zapata[2]
[1]*Autonomous University of Puebla,* [2]*LIRMM – Université Montpellier II*
[1]*México,* [2]*France*

1. Introduction

Motion planning refers to the ability of a system to automatically plan its motions. It is considered central to the development of autonomous robots. In the last decade, much research effort was done on the application of probabilistic roadmaps methods (PRM) for different types of problems (Kavraki et al., 1996; Svestka & Overmars, 1997; Bohlin & Kavraki, 2000; Sánchez & Latombe, 2002).

There are two main classes of PRM planners: multiple-query and single-query planners. A multiple-query planner pre-computes a roadmap and then uses it to process many queries (Kavraki et al., 1996; Svestka & Overmars, 1997). In general, the query configurations are not known in advance and the roadmap must be distributed over the entire free configuration space (C-space). On the other hand, a single-query planner computes a new roadmap for each query (Bohlin & Kavraki, 2000; Sánchez & Latombe, 2002; Sánchez et al., 2002). Its only goal is to find a collision-free path between two query configurations. Looking for the smallest space to explore before finding a path. Planners that can answer single queries very quickly and with a little preprocessing are of particular interest. Such planners can be used to re-plan paths in applications where the configuration space obstacles can change. This occurs, for instance, when the robot changes tools, grasps an object, or a new obstacle enters in the workspace. These kinds of planners are more suitable in environments with frequent changes. The adaptation of PRM planners to environments with both static and moving obstacles has been limited so far (Jaillet & Siméon, 2004).

The planner proposed by Jaillet and Siméon (Jaillet & Siméon, 2004), uses a combination of single and multiple queries techniques. The proposed planner builds a roadmap of valid paths, considering only the static obstacles, when dynamic changes occurs, the planner uses lazy-evaluation mechanisms combined with a single-query technique as local planner to rapidly update the roadmap.

A novel real-time motion planning framework was proposed in (Brock & Kavraki, 2001). It is particularly well suited for planning problems and it decomposes the original planning into simpler sub-problems. The paradigm addresses the planning problems in which a minimum clearance to obstacles can be guaranteed along the solution path.

A method for generating collision-free paths for robots operating in changing environments was presented in (Leven & Hutchinson, 2000). The method begins by constructing a graph that represents a roadmap in the configuration space, but this graph is not constructed for a specific workspace. Later, the method constructs the graph for an obstacle-free workspace, and encodes the mapping from workspace cells to nodes and arcs in the graph. When the

environment changes, this mapping are used to make the appropriate modifications to the graph, and new plans can be generated by searching the modified graph.
A dynamic structure to enrich any non-holonomic motion planner for car-like robots with the capacity of reactiveness to environment changes was proposed by (Jaouni et al., 1998). The main advantage of the star elastic band proposed in their work, is that it allows a better reactivity than the ball band (the elastic band approach was proposed by Quinlan and Khatib in 1993 (Quinlan & Khatib, 1993)). The elastic band approach is a dynamic trajectory modification that maintains a permanent flexible and deformable path between initial and final robot configurations.
This work aims at providing a practical planner that considers reflex actions and planning with lazy techniques to account for obstacle changes. A collision-free feasible path for a mobile robot is computed using the lazy PRM method. The robot starts moving (under the permanent protection of its deformable virtual zone (DVZ)), in a free of dynamic obstacles trajectory, it does not require reflex commands and the control is performed by the lazy PRM method. 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.

2. The DVZ principle

Artificial reflex actions for mobile robots can be defined as the ability to react when unscheduled events occurs, for instance when they move in unknown and dynamic environments. For the last seventeen years, we have been interested in the problem of reactive behaviours for collision avoidance in the domain of mobile robotics (Zapata, 1991; Zapata et al., 1994; Cacitti & Zapata, 2001). This section describes the DVZ principle. We assume that the mobile robot has not 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.
In what follows, n will denote the dimension of the robot world (Euclidean space), $\mathbb{R}$ the real line, $\|y\|$ the Euclidean norm of vector y, and $(\partial\Xi/\partial x)$ the Jacobian of the vector-valued function Ξ. 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 control 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.

Let $\chi = \begin{pmatrix} \Xi \\ \sigma \end{pmatrix}$ be the vector that represents the internal state of the robot and let ε be the state space, which is the set of all the vectors χ. The DVZ is defined by $\Xi = \begin{pmatrix} \Xi_1 \\ \vdots \\ \Xi_c \end{pmatrix}$ and the robot velocities vector σ is defined by $\sigma = \begin{pmatrix} v \\ \dot{\theta} \end{pmatrix}$, 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 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 H be the set of all internal states χ_h whose DVZ is not deformed. This set induces an equivalence relation in ε, defined by

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

where χ_h^i is the internal state corresponding to the state χ^i but without any deformation due to 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 as, (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 equation (4) provides the state equation when no deformation occurs (when the state vector stays on 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 intrusion, we will obtain the second part of the state equation. To do it, we denote the deformation of the state vector by Δ and study the variations of this deformation with respect to intrusion. This new vector represents the deformed DVZ, which is defined by

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

Let $I = \begin{pmatrix} I_1 \\ \vdots \\ I_c \end{pmatrix}$ be the *c*-dimensional intrusion vector, where $I_i = d_{i\max} - d_i$. The sensor provides the measure $d_i = d_{i\max}$, in the absence of obstacles.

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

$$\Delta_i = \alpha(d_{hi}, I_i) = \begin{cases} 0 & if\ d_i > d_{hi} \\ d_{hi} - d_i & if\ d_i \le d_{hi} \end{cases} \tag{7}$$

where d_{hi} is an element of the intact DVZ (Ξ_h). Figures 1 and 2 illustrate this function.

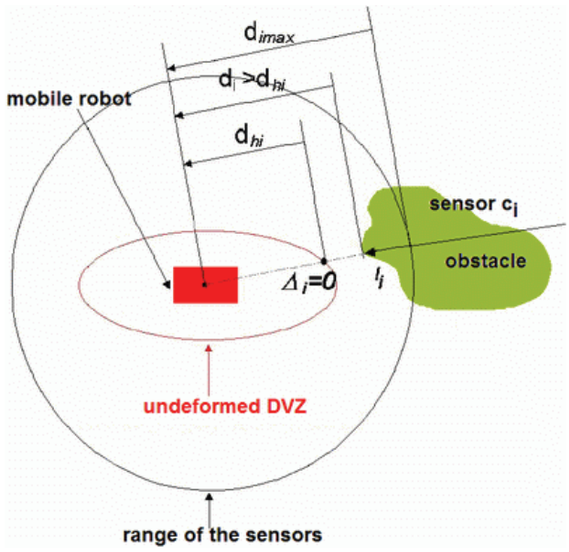


Figure 1. The deformation vector $\Delta_i = 0$

By differentiating equation (6) with respect to time, we get

$$\dot{\Delta} = \frac{\partial \alpha}{\partial \Xi_h}(\Xi_h, I)\dot{\Xi}_h + \frac{\partial \alpha}{\partial I}(\Xi_h, I)\dot{I} \tag{8}$$

By letting $\psi = \dot{I}$ and using equations (4), (5), (6) and (8), we obtain the next control equation

$$\begin{cases} \dot{\Xi} = \left(\frac{\partial \alpha}{\partial \Xi_h}(\Xi_h, I) \times \rho'_{\Xi}(\pi) + \rho'_{\Xi}(\pi) \right)\phi + \frac{\partial \alpha}{\partial I}(\Xi_h, I)\psi \\ \dot{\sigma} = \rho'(\pi)\phi \end{cases} \tag{9}$$

with

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

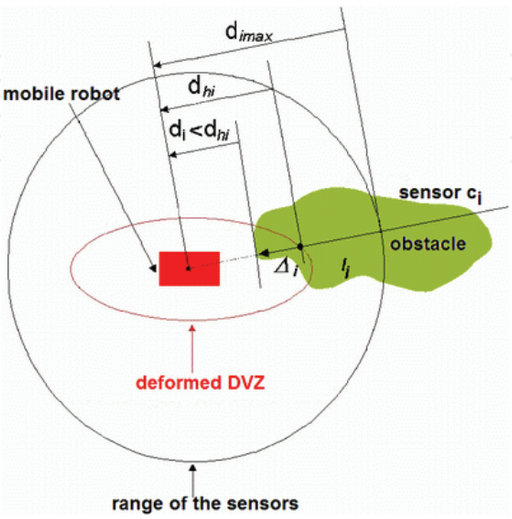


Figure 2. The obstacle deforms the DVZ

The inputs of equation (9) 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.
We can consider the matrix A with dim(A)=$[c \times n]$ (c sensors, k control variables) and the matrix B with dim=(B)=$[c \times c]$ as follows:

$$\begin{aligned} A &= \frac{\partial \alpha}{\partial \Xi_h}(\Xi_h, I)\rho'_{\Xi}(\pi) \\ B &= \frac{\partial \alpha}{\partial I}(\Xi_h, I) \end{aligned} \tag{10}$$

By replacing equation (9) and (10) in equation (8), we obtain the evolution of the deformation

$$\dot{\Delta} = A\Phi + B\Psi \tag{11}$$

The DVZ control algorithm consists of choosing the desired evolution $\dot{\Delta}_{des}$ of the deformation. Given $\dot{\Delta}_{des}$, the best control vector $\breve{\phi}$ in the sense of least-squares that minimizes the function $\left\| \dot{\Delta}_{des} - \dot{\Delta} \right\|^2$ is obtained by inverting equation (11):

$$\breve{\phi} = A^{+}(\dot{\Delta}_{des} - B\psi) \tag{12}$$

where A^{+} is the pseudo-inverse of A.
A simple and efficient control law consists of choosing the desired deformation as proportional to the real deformation and its derivative:

$$\dot{\Delta}_{des} = -K_p \Delta - K_d \dot{\Delta} \tag{13}$$

where the two matrices K_p and K_d are respectively the proportional and derivative gains and are tuned in order to carry out the avoidance task. In this work, we define an ellipse as DVZ parameterized by the linear velocity and the steering angle of vehicle.

3. Lazy PRM for non-holonomic mobile robots

A Lazy PRM approach for non-holonomic motion planning was presented in (Sánchez et al., 2002). The algorithm is similar to the work presented by Bohlin and Kavraki (Bohlin & Kavraki, 2000), in the sense that the aim is to find the shortest path in a roadmap generated by randomly distributed configurations. In a later work, Sánchez et al., 2003 showed that the use of deterministic sampling improved remarkably the results obtained with random sampling.

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. 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 (see figure 3). Experiments show that, in many cases, only a very small fraction of the graph must be explored to find a feasible path.

Planners, based on lazy strategy (Bohlin & Kavraki, 2000; Sánchez & Latombe, 2002) always use the straight-line segment (Euclidean distance) as steering method. Much research has been done on motion planning for nonholonomic car-like robots (see Laumond, 1998 for a review). Svestka and Overmars used the RTR paths as a steering method (Svestka & Overmars, 1997). An alternative choice is to use steering method that constructs the shortest paths connecting two configurations (Reeds & Shepp, 1990; Sanchez et al., 2002). Reeds & Shepp have provided a sufficient family of shortest paths for the car-like robots moving both forward and backward. Figure 4 shows the Reeds & Shepp paths and an example computed with this approach.

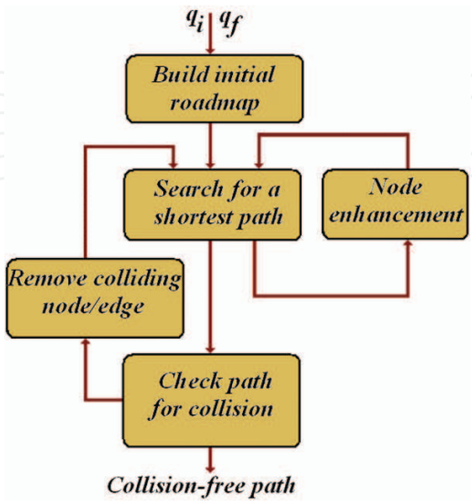


Figure 3. High-level description of the lazy PRM approach

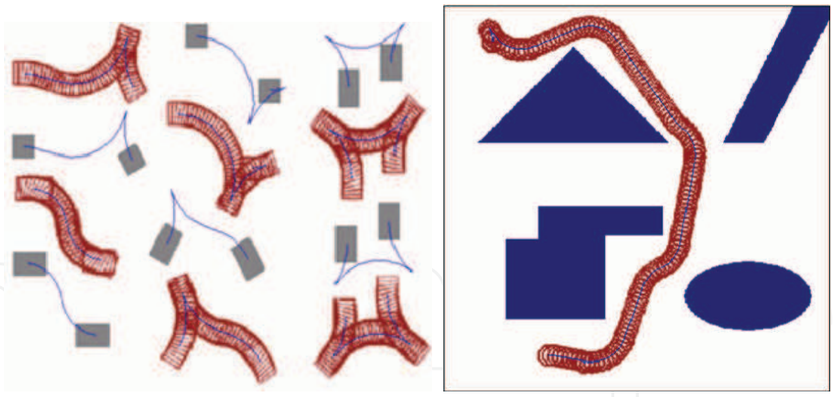


Figure 4. Shortest paths for car-like robots and a computed path in a polygonal environment

3. Reactive Lazy PRM

This section describes the proposed approach, which 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 is 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 force the planner to go back to the previous state.

The algorithm can finish of 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 5 shows a high-level description of the proposed approach.

The lazy PRM planner for non-holonomic mobile robots is detailed in (Sanchez et al., 2000). We consider that the other components of this approach are more important and they will be detailed in the next subsections.

3.1 Reactive control by DVZ

By using the equations discussed in the section II, and the next equation. We can use the next DVZ form (see figure 6).

$$d_{hi} = K_1 V_1^2 \cos^2(\beta_i + K_2 \dot{\theta}) + d_i^{\sec} \tag{14}$$

where K_1 and K_2 are constants, V_1 and $\dot{\theta}$ are the robot's velocities (see equation 14), β 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.

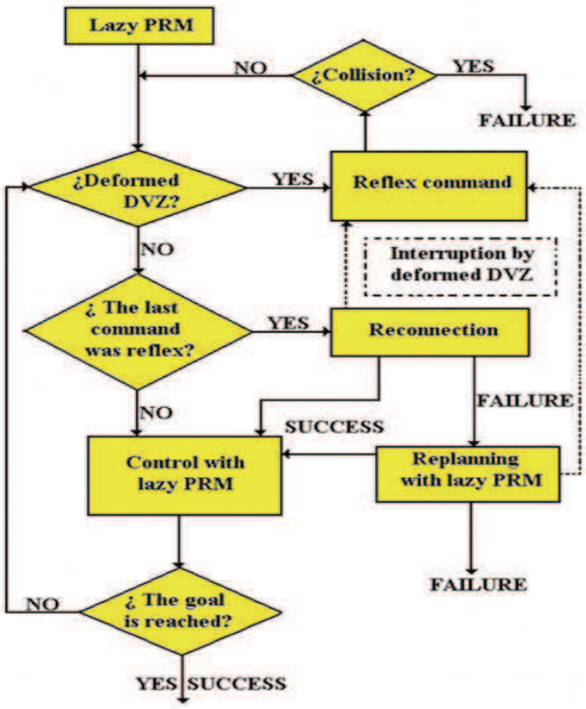


Figure 5. High-level description of our proposed approach

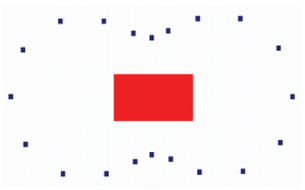


Figure 6. The obtained form of the DVZ using 20 simulated sensors

For the first case in equation (7), ($d_i > d_{hi}$), the DVZ is not deformed by the environment, the control is performed by the lazy PRM method and the reflex actions are not required. For the second case, when ($d_i \leq d_{hi}$), a re flex action is necessary, the executed path by the lazy PRM method is suspended and the robot control is taken by the DVZ method.

3.2 Generation of reflex commands

When the DVZ takes the control, it has the task of taking the robot to a free state of deformations, indicating the kinematics attitudes that should continuously have the robot. These attitudes constitute the vector π described as follows:

$$\pi = \begin{bmatrix} V_1 \\ \dot{\theta} \end{bmatrix} \tag{15}$$

We do not use the equation (12) to implement the control, 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

$$f_i[n] = \begin{cases} \Delta_i[n] - \Delta_i[n-1] & if\ \Delta_i[n] - \Delta_i[n-1] > 0 \\ 0 & if\ \Delta_i[n] - \Delta_i[n-1] \le 0 \end{cases} \tag{16}$$

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

$$F[n] = \sum_{i=1}^{c} f_i[n] \tag{17}$$

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

$$\pi[n] = \begin{cases} V_1[n] = V_1[n-1] + K_v * \|F[n]\| * sign(\cos(\hat{F}[n])) \\ \dot{\theta}[n] = \dot{\theta}[n-1] + K_t * (\sin(\hat{F}[n])) \end{cases} \tag{18}$$

3.3 Kinematics of the robot

The robot learns the kinematics attitudes that it should constantly adopt through the path computed by the lazy PRM method and the reflex actions that should be taken. These attitudes are V_1 and $\dot{\theta}$ are fixed in every Δt interval. We consider a model of the car-like robot as follows:

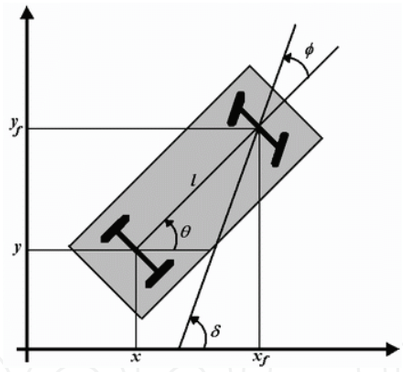


Figure 7. A car-like robot

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ (\tan\phi)/l \\ 0 \end{bmatrix} V_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} V_2 \tag{19}$$

Because the change of x, y and θ is constant foe every interval, the steering angle ϕ in the front wheels stays fixed in the interval (V_2=0), describing a circular path.
The last observation is useful to avoid the integration operation that may be required, otherwise, to determine the x, y and θ values for the next interval of time. Instead, it is enough to use the analytic geometry properties of the circumference.

The next equation shows how to obtain a new configuration for the robot, after the application of a specific impulse.

$$\begin{aligned} x_2 &= x_1 + r(\cos\gamma_2 - \cos\gamma_1) \\ y_2 &= y_1 + r(\sin\gamma_2 - \sin\gamma_1) \\ \theta_2 &= \theta_1 + q * (V_1 / r) * \Delta t \end{aligned} \tag{20}$$

where

$$\begin{aligned} r &= \max(abs(V_1 / \dot{\theta}), R_{\min}) \\ q &= sign(V_1 / \dot{\theta}) \\ \gamma_1 &= \theta_1 - q(\pi / 2) \\ \gamma_2 &= \theta_2 - q(\pi / 2) \end{aligned}$$

$R_{\min}$ is the trajectory's radius that describes the car-like robot when its front wheels are in its maximum steering angle. The figure illustrates the relation between the variables in equation (20).

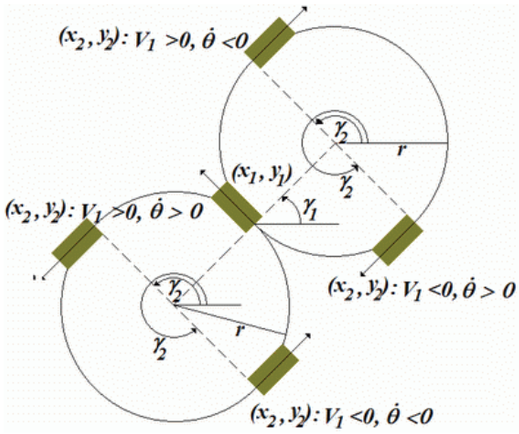


Figure 8. The relation between the variables of equation (20)

3.4 Reconnection

After a successful reflex action, the mobile robot recovers the intact state of its DVZ, but the initial planned path is lost (Fig. 9b), and the lazy PRM method needs to have a path to push the mobile robot to the goal. For this reason it is necessary to provide a path for such aim.

Since the computational cost of a complete re-planning is high, it is avoided as far as possible by executing a process that consists of a reconnection with the planned path by using a single collision-free Reeds & Shepp curve (Fig. 9c).

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. 9d).

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.

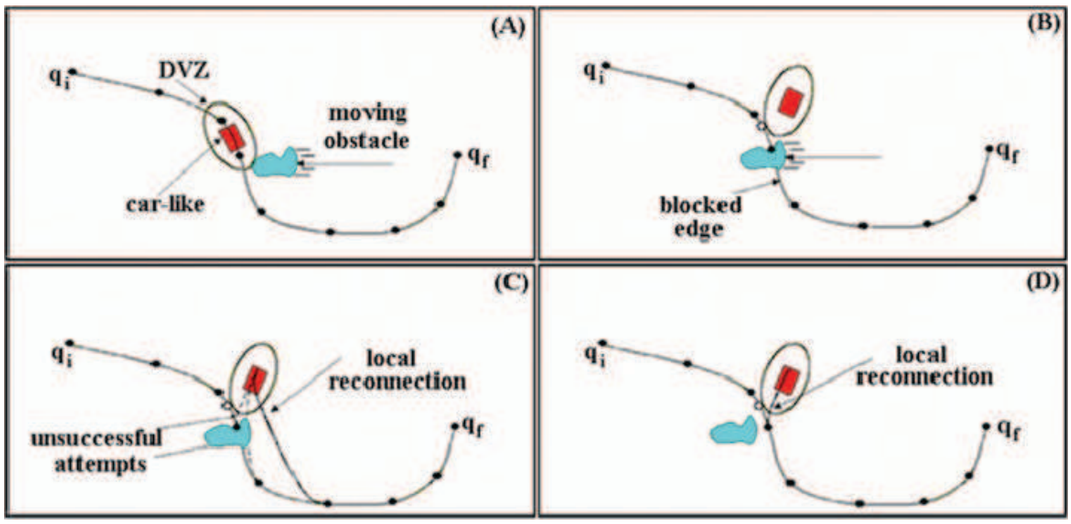


Figure 9. Cases of the reconnection process: a) to avoid a dynamic obstacle, b) after a reflex action, c) after many previous attempts, d) a successful reconnection

3.5 Re-planning

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.

In the case that the mobile robot is developing in a static environment (or partially static), the planned path is enough to avoid a collision. Under this assumption, there is not need to generate any reflex action when a fixed obstacle enters the DVZ.

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 re-planning 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. Simulation results

Some simulation results are presented in this section. The planner was implemented in Builder C++ and the tests were performed on an Intel © Pentium IV 2.4 GHz processor and 512 MB memory. 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 tests 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 in their movement a new random direction. Figure 10 shows an environment composed of narrow passages and dynamic obstacles moving randomly at the same velocity than the mobile robot.

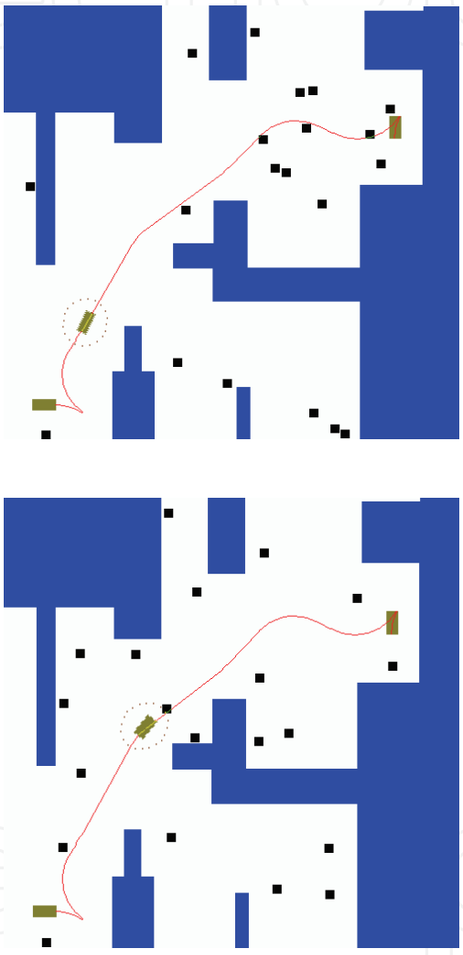


Figure 10. An example of a query and the path solution in an environment with 20 moving obstacles. The robot starts moving under the permanent protection of its DVZ

In order to evaluate the performance of the planner, we performed tests on the environment of Figure 11 for several roadmap sizes and different number of moving obstacles. The different settings are summarized in the tables 1, 2 and 3. In our case, due to the strategy of node addition, the time for the roadmap's construction is proportional to the number of nodes. The number of nodes at the beginning is a critical parameter that affects the lazy PRM's performance (Sánchez et al., 2002). To show the methodology proposed, we performed 30 trials.

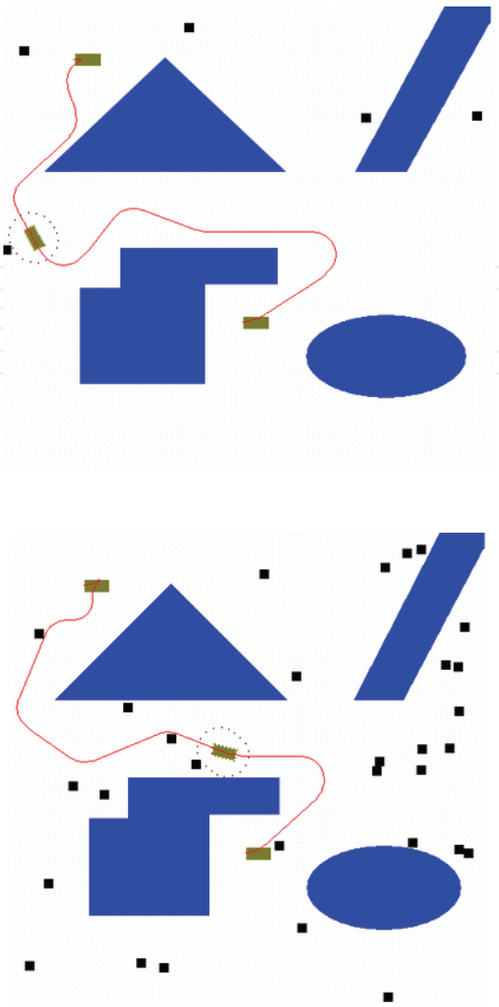


Figure 11. Trajectory execution control by the proposed planner. The environment constains 5 and 30 moving obstacles

Settings	50 nodes	100 nodes	50 nodes	100 nodes
Steering angle	45	45	70	70
Graph building	0.007	0.016	0.008	0.028
Graph searching	0.005	0.016	0.005	0.015
Coll. checking	980	1299	1093	1712
Total time (secs)	0.071	0.109	0.074	0.147

Table 1. Performance data for Lazy PRM

Reconnections	Time for reconnection	Replanning	Time for replanning	Collision	Success
43	0.027	0	0	No	Ok
12	0.025	0	0	No	Ok
3	0.025	0	0	No	Ok
65	0.030	0	0	No	Ok
97	0.027	0	0	No	Ok
84	0.031	0	0	No	Ok
12	0.026	0	0	No	Ok
16	0.026	0	0	No	Ok

Table 2. Performance data with 5 moving obstacles

Reconnections	Time for reconnection	Replanning	Time for replanning	Collision	Success
36	0.029	0	0	No	Ok
90	0.029	1	0.019	No	Ok
3	0.027	0	0	Ok	No
5	0.026	0	0	Ok	No
81	0.032	1	0.115	No	Ok
27	0.032	1	0	No	Ok
4	0.026	0	0	No	Ok
9	0.026	0	0	No	Ok

Table 3. Performance data with 10 moving obstacles

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, and a process of re-planning that is executed if the Lazy PRM and DVZ's parameters are appropriate. As mentioned in earlier sections, it can be considered that the methodology proposed here, includes these characteristics.

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 re-planning processes if necessary, without incurring in bigger problems.

If the execution's parameters for the Lazy PRM and DVZ methods are adapted, the re-planning process will not be called very often and will be successful in the absence of narrow passages. Figure 12 presents a case where the reflex actions were not sufficient. The presence of narrow passages is an important problem to being considered.

5. Experimental results

We have implemented the approach on the Pionner-3 robot from the ActivMedia Robotics. This robot is driven by two independent wheels , it is an agile, versatile intelligent mobile robotic platform updated to carry loads more robustly and to traverse sills more surely with high-performance current management to provide power when it's needed. It has a ring of 8 forward sonar and 8 rear sonar ring. 3-DX's powerful motors and 19cm wheels can reach speeds of 1.6 meters per second and carry a payload of up to 23 kg. In order to maintain accurate dead reckoning data at these speeds, the Pioneer uses 500 tick encoders. Its sensing moves far beyond the ordinary with laser-based navigation options, bumpers, gripper, vision, stereo rangefinders, compass and a rapidly growing suite of other options.

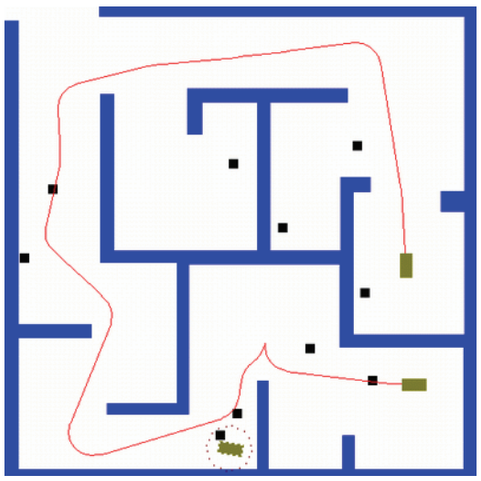


Figure 12. The reflex actions were not sufficient, the mobile robot collides with a moving obstacle

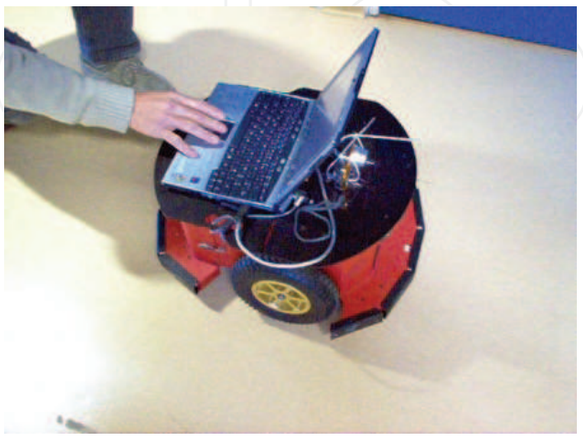


Figure 13. The mobile robot used in the experimental part



Figure 14. The mobile robot avoids an obstacle

The experimental part was done considering that the robot is able to follow a geometric trajectory previously calculated by a Lazy PRM planner, we considered a model of the environment on scale. In the absence of obstacles, the robot follows the trajectory until arriving at the goal region, if there are unknown obstacles, the robot executes reactive controls to avoid them and to return to its trajectory.
Figure 14 illustrates this single experiment, where the robot avoids an unknown obstacle. One can see that robot clearly avoids the obstacle and returns to the nominal path.

6. Conclusion

The motion planning for non-holonomic robots in moving environments is a complex problem. The results obtained in the evaluation of the reactive lazy PRM method, proposed in this work, show the importance of finding a solution for this problem.

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, and a process of re-planning 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 re-planning processes if necessary, without incurring in bigger problems.

If the execution's parameters for the Lazy PRM and DVZ methods are adapted, the re-planning process will not be called very often and will be successful in the absence of narrow passages.

A reactive lazy PRM planner for dynamically changing environments is presented in this chapter. 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 these cases.

7. References

Bohlin, R. & Kavraki, L. (2000). Path planning using lazy PRM, *Proceedings of the IEEE Robotics and Automation Conference*, pp. 521-528, ISBN 0-7803-5889-9, San Francisco, CA, USA, April 24-28

Brock, O. & Kavraki, L. (2001). Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces, *Proceedings of the IEEE Robotics and Automation Conference*, pp. 1469-1474, ISBN 0-7803-6578-X, Seoul, Korea, May 21-26

Cacitti, A. & Zapata R. (2001). Reactive behaviours of mobile manipulators based on the DVZ approach, *Proceedings of the IEEE Robotics and Automation Conference*, pp. 680-685, ISBN 0-7803-6578-X, Seoul, Korea, May 21-26

Jaillet, L. & Siméon, T. (2004). A PRM-based motion planner for dynamically changing environments, *Proceedings of the IEEE Intelligent Robots and Systems Conference*, pp. 1606-1611, ISBN 0-7803-8464-4, Sendai, Japan, September 28 – October 2

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research*, Vol. 5, No. 1, (Spring 1986) pages 90-98, ISSN: 0278-3649

Kavraki, L., Svetska, P., Latombe, J. C., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (August 1996) pages 566-580

Jaouni, H., Khatib, M., & Laumond, J. P. (1998) Elastic bands for non-holonomic car-like robots: Algorithms and combinatorial issues, *Proceedings of WAFR 98,* Houston, Texas, USA, March 5-7

Laumond, J. P (Ed.). (1998). Robot motion planning and control, Springer-Verlag, ISBN 3-540-76219-1

Quinlan, S. & Khatib. O. (1993). Elastic bands: Connecting path planning and control, *Proceedings of the IEEE Robotics and Automation Conference*, pp. 802-807, ISBN 0-8186-3450-2, Atlanta, Georgia, USA, May 93

Leven, P. & Hutchinson, S. (2000). Toward real-time path planning in changing environments, *Proceedings of WAFR 2000,* Hanover, NH, USA, March 16-18

Reeds, J. A. & Shepp, R. A. (1990). Optimal paths for a car that goes both forward and backwards, *Pacific Journal of Mathematics*, Vol. 145, No. 2, pages 367-393

Sánchez, G. & Latombe, J. C. (2002). On delaying collision checking in PRM planning: Application to multi-robot coordination, *International Journal of Robotics Research*, Vol. 21, No. 1, pages 5-26

Sánchez, L. A., Zapata, R., & Arenas B. A. (2002) Motion planning for car-like robots using a lazy probabilistic roadmap method, *Advances in Artificial Intelligence*, Springer-Verlag, pp. 1-10

Sánchez, L. A., Zapata R., & Lanzoni, C. (2003) On the use of low-discrepancy sequences in nonholonomic motion planning, *Proceedings of the IEEE Robotics and Automation Conference*, pp. 3764-3769, Taipei, Taiwan, September 14-19

Sánchez L. A., Cuautle P. R., Zapata, R., & Osorio L. M. A. (2006) A reactive lazy PRM approach for non-holonomic motion planning, *LNAI 4140 Springer-Verlag*, pp. 542-551

Svestka, P. & Overmars, M. (1997) Motion planning for car-like using a probabilistic learning approach, *International Journal of Robotics Research*, Vol. 16, No. 2, pages 119-143

Zapata, R. (1991) Quelques aspects topologiques de la planification de mouvements et des actions réflexes en robotique mobile, Thèse d'Etat (Phd Thesis in French), University of Montpellier II

Zapata, R., Lépinay, P., & Thompson, P. (1994) Reactive behaviors of fast mobile robots, *Journal of Robotic Systems*, Vol. 11, No. 1, pages 13-20



Motion Planning

Edited by Xing-Jian Jing

ISBN 978-953-7619-01-5

Hard cover, 598 pages

Publisher InTech

Published online 01, June, 2008

Published in print edition June, 2008

In this book, new results or developments from different research backgrounds and application fields are put together to provide a wide and useful viewpoint on these headed research problems mentioned above, focused on the motion planning problem of mobile ro-bots. These results cover a large range of the problems that are frequently encountered in the motion planning of mobile robots both in theoretical methods and practical applications including obstacle avoidance methods, navigation and localization techniques, environmental modelling or map building methods, and vision signal processing etc. Different methods such as potential fields, reactive behaviours, neural-fuzzy based methods, motion control methods and so on are studied. Through this book and its references, the reader will definitely be able to get a thorough overview on the current research results for this specific topic in robotics. The book is intended for the readers who are interested and active in the field of robotics and especially for those who want to study and develop their own methods in motion/path planning or control for an intelligent robotic system.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Abraham Sánchez, Rodrigo Cuautle, Maria A. Osorio and René Zapata (2008). Reactive Motion Planning for Mobile Robots, Motion Planning, Xing-Jian Jing (Ed.), ISBN: 978-953-7619-01-5, InTech, Available from: http://www.intechopen.com/books/motion_planning/reactive_motion_planning_for_mobile_robots

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen