Planning robust landmarks for sensor based motion

Michel Taïx[1,2] , Abed C. Malti[1,2] and Florent Lamiraux[1]

[1] LAAS-CNRS,
University of Toulouse, Toulouse, France
[2] University Paul Sabatier, Toulouse, France

Summary. Our work is focused on defining a generic approach for planning landmark based motion. In previous works we have introduced the formal basis for this and showed simulation results. In this paper we first demonstrate the relevance of our work with experiments on a real robot. Then, on the base of these results we introduce new strategy for planning selecting landmarks in order to improve the robustness of the navigation task in a cluttered environment.

1 Introduction

Path planning in a reference map for a robot produces a non-collision continuous path in the robot configuration space [7, 8, 2, 9]. However, the execution of this path in a real environment remains problematic for two main reasons. The first difficulty lies in the inaccuracy of the environment map used to plan the path, and the second is that the navigation task in cluttered environments requires precise localization. Many approaches have been proposed to solve these two problems :localizing the robot along the path with respect to local landmarks [15], reactive methods to avoid unexpected obstacles [12, 13, 1, 6], path planning with uncertain approaches [5, 4]. In our work, we aim to introduce sensor-landmark constraints along a geometric path to solve these problems. In a previous work [10, 11] we have introduced a generic approach to correct a planned geometric path. The idea was to plan sensor-landmark primitives to perform sensor based motion along a path. Instead of planning a path in a first stage and following it in a second stage, we will produce a sequence of sensor-landmark based motions, each sensor-landmark pairs are weighted. These weights distinguish the sensor-landmark pairs from the most to the least relevant. For example when passing through a door the most relevant landmarks are the two sides of the door. The goal is not to localize the robot but to give input for sensor-based motion controller.

In this paper we present the first validation conducted upon a real robot platform. These experiments lead us to propose several criteria in order to improve the selection process of the landmarks. These improvements involve the robustness of the localization process and the success of the matching process. In section 2 and 3 we give the definition of a landmark based motion and how we plan this motion. In section 4 we present a parking manoeuvre conducted on a mobile robot. In section 5 we describe the landmark selection that will allow to improve the robustness of the localization. In section 6 we introduce the landmark selection that takes into account the success of the matching process. Finally, in section 7, we give the navigation task experiment to show the relevance of our work.

2 Definition of a Landmark-Based Motion

Landmark: a landmark can be any geometric feature in the workspace. Let us denote by $\mathbb{L}$ the configuration space of a landmark L. We denote by l the configuration of L.

Sensor: a sensor S is a mobile device that maps one or several landmarks to a feature in the image space. Let us denote by $\mathbb{S} = SE(2)$ or $SE(3)$ the configuration space of sensor S. We will denote by s the configuration of S.

Sensing a landmark: the perception of landmark L by sensor S is a mapping between a pairs of configuration (sensor, landmark) and a feature in the image space $I_{S,L}$.

$$\begin{aligned} P_{S,L} : \mathbb{S} \times \mathbb{L} &\rightarrow \quad I_{S,L} \\ (s,l) &\rightarrow P_{S,L}(s,l) \end{aligned}$$

Localization: each pair (S_i, L_j) of sensors and landmarks gives rise to a *localization equation* where $im \in I_{S_i,L_j}$ is the image of L_j in S_i. l_j is supposed to be known and im is measured. The unknown of this equation is the configuration $\mathbf{q}$ of the robot:

$$P_{S_i,L_j}(s_i(\mathbf{q}), l_j) = im \tag{1}$$

The linearization of equation (1) around $\mathbf{q}_0$ leads to:

$$\frac{\partial P_{S_i,L_j}}{\partial s}(s_i(\mathbf{q}_0), l_j).\frac{\partial s_i}{\partial \mathbf{q}}(\mathbf{q}_0).(\mathbf{q} - \mathbf{q}_0) = im - im_0 \tag{2}$$

im_0 is the image of L_j in S_i at $\mathbf{q}_0$. This equation expresses the approximation of order 1 of the relation between a variation of configuration about $\mathbf{q}_0$ and the variation of the image of each landmark in the corresponding sensor.

Weighting localization: this equation can be written for all m sensor-landmark pairs. A weight ω can be associated to each sensor-landmark pair k :

$$w_k \frac{\partial P_{S_i,L_j}}{\partial s}(s_i(\mathbf{q}_0), l_j).\frac{\partial s_i}{\partial \mathbf{q}}(\mathbf{q}_0).(\mathbf{q} - \mathbf{q}_0) = w_k.im - w_k.im_0 \tag{3}$$

This weight expresses the importance of the sensor-landmark pair (collision and/or localization) along the path and is part of the motion control task specifications. Thus,from linear equations (3) we can build a linear system of equations by weighting each equation by a positive real number ω_j. We thus get the following linear system:

$$W(\mathbf{q} - \mathbf{q}_0) = IM - IM_0 \tag{4}$$

The least square solution of this system given by (W^+ is the pseudo-inverse of W):

$$\hat{\mathbf{q}} = \mathbf{q}_0 + W^+(IM - IM_0) \tag{5}$$

Landmark-Based Motion: given a mobile robot with n sensors S_i and an environment with p landmarks L_j, a *Landmark-Based Motion, LBM,* is defined by:

1. a reference collision-free path:

$$\begin{array}{rl} \gamma : [0, U] \rightarrow & \mathcal{C} \\ u \quad \rightarrow & \gamma(u) \end{array}$$

   where $[0, U]$ is an interval (U length of the free path),
2. m continuous positive real valued functions $w_1, ..., w_m$:

$$\begin{array}{rl} w_k : [0, U] \rightarrow & \mathbb{R}^+ \\ u \quad \rightarrow & w_k(u) \end{array}$$

such that w_k is associated to a pair (S_i, L_j).

The developments conducted in this section can be summarized as follows. Localizing a mobile robot using landmarks involves solving a system of equations that relate the configuration of the robot with the images of the landmarks in the sensors of the robot. If the system is over-constrained, localization involves finding a configuration that minimizes a weighted sum of residues. If the landmarks are at exactly the same position in the model map as in the real map, the choice of weights will have no effect on the result. However, if the map of landmarks is not exact, the choice of weights will have a big influence. That is why in our approach, we suggest using these weights as a tool for planning landmark-based motions.

3 Landmark based motion planning

At a first step, a geometric non-collision path $\gamma(u)$ is planned in the configuration space of the model map from an initial configuration to a final one. This path is computed by the probabilistic path planner Move3D which is developed in our laboratory [14]. Now it is necessary to calculate the weight of the sensor-landmark pairs during the path in the model map.

3.1 Weight Computing

In this work, the weight is an intrinsic specification of the robotic task associated to the execution of the geometric planned path. Thus for a given configuration along this path, the weight of any sensor-landmark pair has to represent its importance in relation to the environment and the path in order to avoid collisions and to satisfy the result of the geometric path planning stage. We define a weight of a sensor-landmark pair as a positive continuous function in the configuration space as :

$$\begin{aligned} w : \mathbb{S} \times \mathbb{L} \rightarrow & \quad \mathbb{R}^+ \\ (s,l) \mapsto & \; w(s,l) \end{aligned}$$

This function vanishes of the sensor range view. It represents :

- the visibility of the landmark (distance and orientation for example)
- the danger of collision with this landmark (collision distance)

3.2 The construction of a landmark based motion

For a static configuration we can draw comparisons between landmarks and depict the most relevant, so as to execute localization. Now it is necessary to plan the best N_L sensor-landmark pairs along the path. The landmarks which have good properties of localization or presenting a risk of collision with the path will have an important weight and will be selected thus automatically. We know that the minimum number of landmarks required to localize the robot is 2 in dimension two. In practice this number is too low because the equations of landmarks can be dependent and it is then necessary to consider a bigger number of landmarks. A number from 4 to 5 landmarks is sufficient in practice. It is thus enough to select in every $\mathbf{q}$, N_L sensor-landmark pairs having the best weight.

In general, the inputs of *LBM* algorithms are as follows :

1. a model map of the environment,
2. the set of sensors $\mathcal{S}$ and landmarks $\mathcal{L}_{env}$,
3. the non-collision geometric path $\gamma(u),\ u \in [0, U]$,
4. the number of maximum best landmarks N_L.

The output is a landmark based motion *LBM* composed of $\gamma(u)$ and a set of weighed sensor-landmark pairs. In basic terms, for a given sensor, this algorithm associates to every part of the path the best N_L landmarks that have the highest values of weights.

4 Parking Manoeuvre

We integrate the software *LBM* as a module in the generic architecture control of the Hilare 2 platform [3].

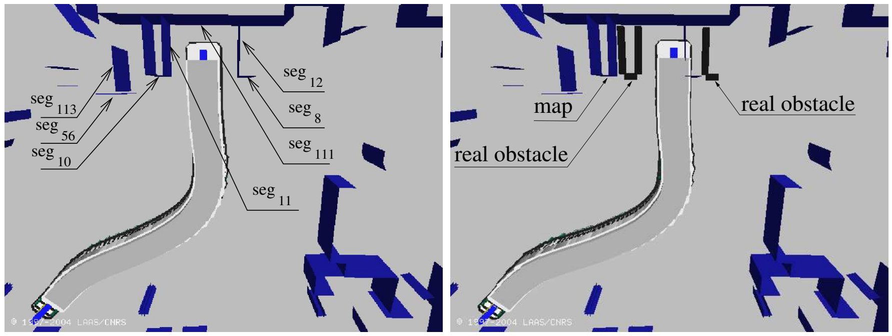

Fig. 1. The left figure shows the geometric free path planned to park the robot and the segments use to build the *LBM*. The right one shows the path executed to enter the shifted car park using the planned *LBM*.

At a first stage, a geometric non-collision path is planned in this map from an initial to a final configuration so that the robot will be able to enter the car park, as show in figure (1). The second stage involves planning the landmark based motion with the generic platform we developed. Along the geometric planned path, the four best sensor-landmark pairs are selected according to their weights. Before executing the landmark based motion, the car park is shifted right to modify the real environment in relation to the reference map. During the run of the movement, the robot corrects its path with regard to the new placement of the parking and then the task is led with success.

The scenario of this experiment shows the relevance of the formalism in a local area when the reference map is inaccurate. However, in a wider area, where the robot executes a navigation task, a landmark selection based on highest values of weights is restrictive for two main reasons :

1. It does not take into account the conditioning of the localization system (4).
2. It does not take into account the success of the matching process.

5 Landmark selection with regular matrix condition

In the previous experiment, when the robot starts the last stage of the parking it uses the three segments seg_{11}, seg_{12} and seg_{111}. If we remove the segment seg_{111}, the weighted localization matrix in (4) becomes ill-conditioned. Typically, this case happens when the robot navigates in a long corridor, segments seg_{11}, seg_{12} became the side of the corridor. So by taking into account solely both sides the weighted localization system is ill-conditioned. In this case, the localization process will produce a big jump in the value of the configuration of localization $\hat{\mathbf{q}}$ in relation to the current position. The following pseudo-code algorithmo avoids such undesirable situations :

Data: $\mathcal{L}_{env}$, $K_{d_{max}}$, *LBM*
Result: *LBM*
begin
  for *each piece of LBM along* γ **do**
    $\mathcal{L}_{select}$ ←get_landmark();
    K_d ←condition_number($\mathcal{L}_{select}$);
    if $K_d \geq K_{d_{max}}$ **then**
      $\mathcal{L}_{\mathcal{K}}$ ←improve_condition($\mathcal{L}_{env}$,$\mathcal{L}_{select}$,$K_{d_{max}}$);
    end
    LBM←add_landmark($\mathcal{L}_{\mathcal{K}}$);
  end
end

Algorithm 1: *LBM* with matrix condition.

The main input of this algorithm is the maximum condition number $K_{d_{max}}$ that is defined as the highest value we accept for the ratio between the highest and the smallest singular value of the localization matrix (this parameter is a good information for localization process). Thus, for each piece along a geometric path of a pre-constructed landmark based motion *LBM*, the condition number K_d is computed for the corresponding landmarks $\mathcal{L}_{select}$. If its value is higher than the maximum condition number, then the algorithm looks for visible landmarks that can improve the matrix conditioning and add them to the landmark based motion. The function *improve_condition* uses new landmark of $\mathcal{L}_{env}$ (not included in $\mathcal{L}_{select}$), $\mathcal{L}_{\mathcal{K}}$, to decrease K_d and include it in the *LBM*

We have no guarantee of convergence of the algorithm towards a solution but generally, the number of landmark-sensor pairs N_L used in the precalculation of *LBM* is less than the maximum number of useful sensor. In the case where there is no other landmarks, it is not possible to localize the robot with respect to landmarks, and therefore to realize the sensor based motion.

Finally, we obtain a set of landmarks that is less sensitive with respect to small errors between the reference map and the real environment.

6 Landmark selection with matching condition

The second important issue for successfully a landmark based motion is the ability to retrieve continuously the selected landmarks in the real environment. Recognizing those landmarks is absolutely required for mainly two reasons :

- to avoid big gaps in the computation of successive localization errors,
- those landmarks have to be taken into account during the achievement of the robotic task since they are considered as relevant.

Using the most relevant landmarks in the localization process is a good idea to give them more importance than other visible landmarks during the motion execution. However, this reasoning reduces the probability of retrieving them especially in the case where the environment is rich of landmarks and where the selected primitives do not constitute a recognizable shape in relation to the environment.

We give a generic pseudo-code algorithm 2 that allows to pick up those landmarks used to help the success of the matching process. This algorithm takes as inputs a landmark based motion, *LBM*, constructed as described in previous sections and the landmarks of the reference map $\mathcal{L}_{env}$. In a first stage, the algorithm introduces some perturbations on the configurations of the selected landmarks $\mathcal{L}_{select}$ so that it simulates errors in the reference map. In the second stage, it tries to match the perturbed landmarks $\mathcal{L}_{real}$ with those of the reference map by function *Matching*($\mathcal{L}_{real}$, $\mathcal{L}_{env}$). In the case where the matching is successfully then it concludes that the selected landmarks for the current piece of path constitutes a recognizable shape (same matching algorithm has been used to plan *LBM* than in real execution, only the data are different). In contrast to this situation, if one of the selected landmarks is not matched then the algorithm adds others visible landmarks. The function *Pick_visi_landmark* takes a new landmark-sensor pair include in $\mathcal{L}_{env}$ but not in $\mathcal{L}_{select}$ to try to construct a recognizable shape of selected landmarks. This operation is repeated while *Matching*($\mathcal{L}_{real}$, $\mathcal{L}_{env}$) fails or no other one exists. It is important to insist on the fact that the landmarks added by this algorithm are used to help the success of the matching process, but they are not used in the localization process.

```
Data: L_env, LBM
Result: LBM
L, L_real ← Null;
begin
    for each piece of LBM along γ do
        L_match ← Null;
        L_select ← Get_landmark();
        L_real ← Perturbation(L_select);
        while Matching(L_real, L_env) fails do
            L ← Pick_visi_landmark(L_env, L_select);
            L_match ← L_match ⊕ {L};
            L_real ← Perturbation(L_select ⊕ L_match);
            LBM ← Add_matching_landmark(LBM, L_match);
        end
    end
end
```

Algorithm 2: *LBM* with matching condition.

The improvement presented in the two last sections was integrated on our generic framework. Actually, the landmark based motion planner we are developing selects the landmarks that are the most relevant in relation to :

1. the danger of collision and their visibility,
2. the conditioning of the localization matrix,
3. the success of the matching process.

The selected landmarks according to the two first one criteria are used both in the matching process and in the localization process. The landmarks selected according to the last criterion are used solely in the matching process. This last version of our software has been tested and validated on the mobile robot Hilare 2 towing a trailer by realizing a navigation task. This experiment is detailed in the next section.

7 A navigation task in a cluttered environment

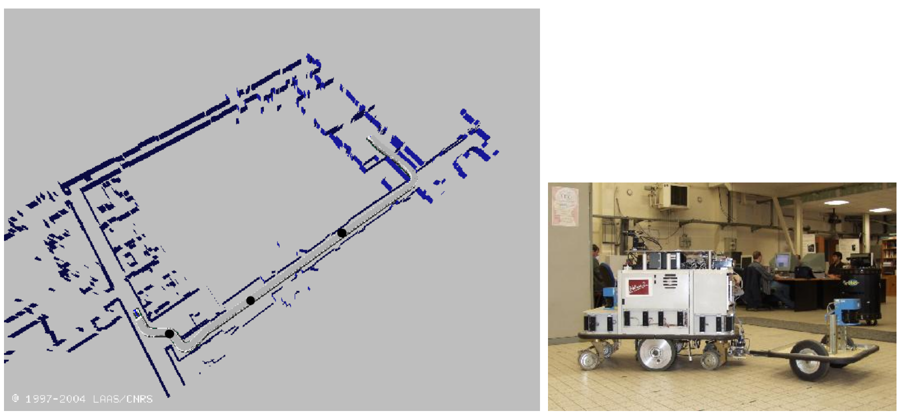

Fig. 2. The planned free path. Points represent three configurations for Hilare 2 robot

In the corridors of our laboratory we plan a geometric free path with Move3D (see figure 2). Thereafter, the produced path and the reference map of the environment are used by the landmark based motion planner to select the most relevant landmarks according to the three criteria presented above.

The navigation task we describe involves some difficulties that have to be raised :

- The reference map we have is not exact. Indeed, we pick up some errors in terms of distances between walls in the reference map and in the real

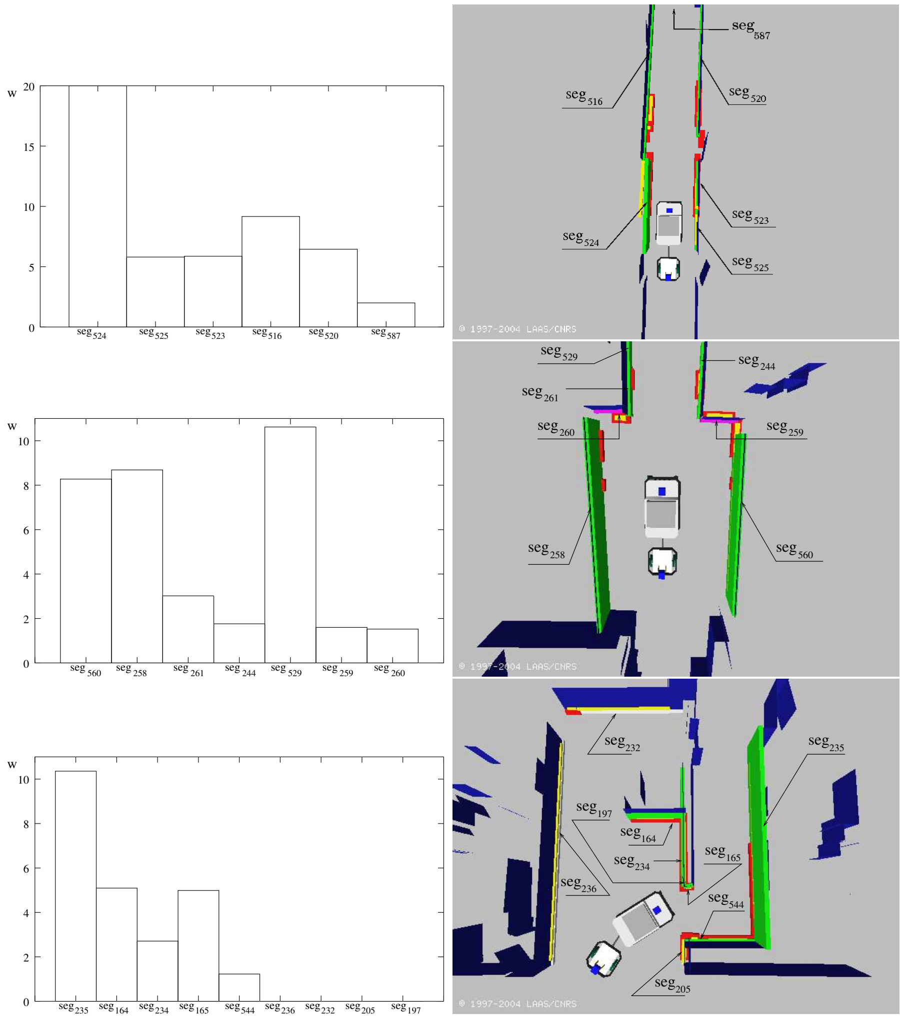

Fig. 3. The left figures shows the instantaneous weights associated to selected landmarks viewed by the front sensor for three configurations of the right figure. The green segments are selected by the initial LBM algorithm. The purple segments are selected by the criterion of algorithm 1 to improve the conditioning of the localization matrix. The white segments are selected by the criterion of algorithm 2 to help the matching process. The yellow color illustrate the segments matched with success.

environment (the difference is about an average value of some ten centimeters).

- The size of the robot in relation to the free space of the environment is a critic issue for the achievement of the navigation task. Indeed, unlike the parking manoeuvre (c.f. section 4) where the robot has a large free space, here the passages are narrowed and the manoeuvres are geometrically very constrained.

Although this difficulty constrained hardly the achievement of the navigation task, Hilare 2 drives with success the circuit using the landmark based motion to correct its path. Here after, we show and comment some of the critical passages. The figure (3) shows the details for the three configurations oto the left of the figure 2.

The bottom figure shows the first critical crossing. Even if the odometric error is not important at this step of the navigation, the errors of the map can generate failure in the experiment. The segments seg_{164}, seg_{165}, seg_{234}, seg_{235} and seg_{544}, are used for localization. The segments seg_{197}, seg_{205}, seg_{232} and seg_{236} are selected only to help the matching process (this is why their weights in left figure are zero). The passage being narrower in reality than in the map, the segments seg_{165} and seg_{544} have big values of weights to ensure a safe crossing.

The middle figure illustrates a passage where it is necessary to take into account landmarks that improve the conditioning of the localization matrix. The algorithm 1 allows to select segments seg_{259} and seg_{260} for this purpose.

The top figure represents the classical situation of a long corridor. Because of the limitation in the perception of the robot sensor's, the sole available landmarks are those of both sides of the corridor. To avoid jumps in the result of localization in such a situation, we correct the position only following the crosswise. Following the lengthwise, the robot continues its path without any correction until it senses the end of the corridor, the weight of seg_{587} (at the end of the corridor) is taken into account.

After the analysis of this experiment, the main issue that attracts our attention for future works concerns the incoherences between the reference map and the real environment. This study involves the formulation of such a problem in relation to a planned geometric free path in order to take a decision whether one has to correct this path in order to correct the map errors or to plan another one. Further works could be led about jumps in the localization caused by the unexpected appearance or disappearance too early or too late of some selected landmarks.

8 Conclusion

In this paper, we presented the first experiment we have conducted upon the mobile robot Hilare 2 towing a trailer. This experiment raises two main issues : the improvement of the conditioning of the localization matrix and the success of the matching process. Thereby, we have developed two algorithms that allowed to select further landmarks to overcome the lacks raised by these two issues. This improvements were integrated to our software and validated across a navigation task. The success of these experiments is very encouraging for future works on the link between path planning and real motion which requires procedures of localization and control.

References

1. K.O. Arras, J. Persson, N. Tomatis, and R. Siegwart. Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In *IEEE Int. Conf. on Robotics and Automation*, 2002.
2. Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
3. S. Fleury, M. Herrb, and R. Chatila. Genom: a tool for the specification and the implementation of operating modules in a distributed robot architecture. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1997.
4. N. Le Fort-Piat, I. Collin, and D. Meizel. Planning robust displacement missions by means of robot-tasks and local maps. *Robotics and Autonomous Systems*, 20:99–114, 1997.
5. M. Khatib, B. Bouilly, T. Siméon, and R. Chatila. Indoor navigation with uncertainty using sensor based motions. In *IEEE Int. Conf. on Robotics and Automation*, 1995.
6. F. Lamiraux and D. Bonnafous. Reactive trajectory deformation for nonholonomic systems: Application to mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, 2002.
7. J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
8. J.-P. Laumond, editor. *Robot Motion Planning and Control*. Lectures Notes in Control and Information Sciences 229. Springer, 1998.
9. S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
10. A.C. Malti, F. Lamiraux, and M. Taïx. Sensor landmark motion planning in mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.
11. A.C. Malti, F. Lamiraux, and M. Taïx. Sensor landmark succession for motion planning along a planned trajectory. In *IEEE Conf. on Mechatronics and Robotics*, Aachen, Germany, 2004.
12. J. Minguez, L. Montano, T. Siméon, and R. Alami. Global nearness diagram navigation (gnd). In *IEEE Int. Conf. on Robotics and Automation*, 2001.
13. S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *IEEE Int. Conf. on Robotics and Automation*, 1993.
14. T. Siméon, J-P. Laumond, and F. Lamiraux. Move3d : a generic platform for path planning. In *IEEE Int. Symp. on Assembly and Task Planning*, 2001.
15. A.C. Victorino and Patrick Rives. A relative motion estimation by combining laser measurement and sensor based control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.