

Robust path planning of a mobile robot in the space of sensor information

A. ATASSI, A. PRUSKI.

Laboratoire d'automatique des systèmes coopératifs (LASC),
Université de Metz,
Ile du Saulcy, BP80794, 57012 Metz,
FRANCE.

Abstract: This paper introduces a new approach to robust path planning for mobile robots. This approach is entirely based on ultrasonic sensors information, and avoids the use of odometry, which leads to the accumulation of errors resulting from the calculation of the robot position. We have proceeded as follows: For each segment of obstacle detected by the mobile robot sensors, we create a free space region. A node graph is used to represent the regions with their links. With this graph as a basis, we use a planning algorithm which chooses the required path. The final stage consists in finding for the robot motion a robust control, as regards the environment model errors (uncertainties). This approach could contribute in practise to a control system for indoor robot motion, which offers increased accuracy to an economical ultrasound device.

Key-Words: -path planning, sensor signal, robust control, environment model. Proc.pp..1941-1947

1 Introduction

Path planning is a fundamental domain of mobile robotics. Many results have been obtained in this field, most of which were focused on the search of collision free path between obstacles, that may connect the initial and final configuration of the robot. The methods used in this field require a full and accurate knowledge of the robot's environment, and a perfect control of the robot. Since this technique cannot avoid the accumulation of a large number of uncertainties, exteroceptive sensors (ultrasonic or laser) [1,2] are often used to minimize these uncertainties.

Several sensor based methods have already contributed to major improvements in path planning.

Among them, the Laugier approach [4,5], which is simple and has the distinctive feature of privileging contacts between the robot and its environment with the purpose of reducing the position uncertainty of the vehicle.

The Alami method [6], like Laugier's one, deals equally with the uncertainties at both the control level and the perceptual level. The motion control is defined in the Alami approach by means of four control primitives. It allows a simple connection (interface) between the planner and the execution control.

The sensory uncertainty field method of Takeda & Latombe [3] takes explicitly into account the

uncertainty problems at the robot's navigation stage. Given a model of the robot's environment, a Sensory Uncertainty Field (SUF) is precomputed over the collision-free subset of the robot's configuration space. The robot being at a given position, the SUF estimates, for every configuration of the subset, the distribution of possible errors (resulting from the environment modeling) in the robot configuration which has been computed by the sensors. The planner uses the SUF to generate paths that minimize expected errors, by crossing areas of the workspace where environment features are visible.

In [7,8], the authors use a task functions' planner for mobile robots. For each configuration of the free space, an uncertainty field CUP (Configuration Uncertainty Potential), and a list of environment's primitives are associated. Based on the CUP, the planner generates a path which realizes a compromise between the path length, the simplicity of the vehicle control, and the security conditions. Then a sequence of local roadmaps representing the hole set of primitives is introduced. At each local roadmap, the planner of task functions is associated with a nominal path, and with a mechanism of regulation (with feedback).

Our approach consists in planning the robot trajectory in a robust manner, as a direct result of sensors information. When the robot detects segments of obstacles, it receives a signal for each of these segments. For each segment of obstacles, we

associate a trapezium shaped region, within which the robot detects the corresponding segment. New regions are then created further to the intersections of the trapeziums. In each of these regions the robot detects a specific information from obstacles. A node graph is used to establish the different links between these regions. The links are realized according to the sensor information change from one region to another. To move the robot from an initial position to a final one, plenty of trajectories are considered. Depending on an optimality criteria, which take into account a given path's robustness, an algorithm will be used to determine the most adequate path. We have chosen the exhaustive algorithm of Dijkstra [1,9], which permits to resolve a path search problem by developing a set of nodes. We have chosen this method since, in contrast to the A* algorithm [1], it does not use heuristic functions. Finally, we have chosen a control in order to move the robot from one region to another.

2 Environment modeling

The environment modeling consists in creating, in free space, regions representing one or more sensor information. The environment is formed by a set of polygonal obstacles, and the robot has at its disposal a telemetric rotary sensor with infinite angles of measure. For simplicity, we will consider that the robot is reduced to point A on fig.1. The sensors information are represented in fig.1 by the angles between the positions of measure and axis x starting from the point of origin A.

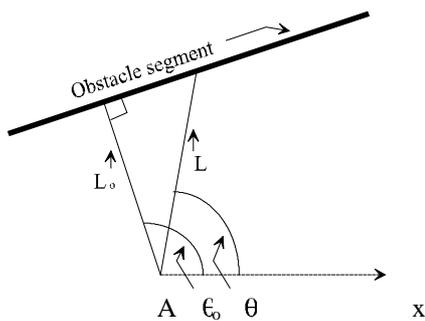


Fig.1 represents the distances between the robot and a segment of obstacle according to the positions of measure.

2.1 Representation of sensor signal

The relationship between a measure of the distance robot-obstacle, and between the orientation to which the measure is undertaken, is represented by the equation:

$$L = L_o \setminus (\cos(\theta - \theta_o)) \quad (1)$$

L_o being the position of measure, which is perpendicular to the segment of obstacle i. e. the nearest distance between the robot and the segment.

θ_o being the angle between the position of measure L_o and x axis of the reference frame.

We notice that function L tends towards infinity, when θ tends towards $\theta_o + \pi/2$ or towards $\theta_o - \pi/2$. To obtain a finite and bounded function, we take the inverse of distance L, and we obtain the following function :

$$d = 1 \setminus L = \cos(\theta - \theta_o) \setminus L_o \quad (2)$$

for $\theta_o - \pi/2 < \theta < \theta_o + \pi/2$
 Otherwise $d = 0$.

In the following figure, we have represented function $d(\theta)$ by a sinusoidal shaped signal.

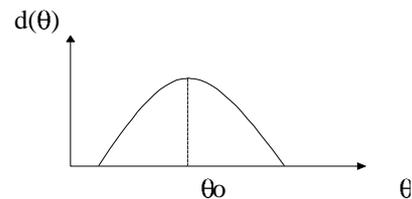


Fig. 2 Representation of signal $d(\theta)$ with a maximum for $\theta = \theta_o$.

We define by d_m the maximal distance at which the robot can identify a segment of obstacle. Starting from signal $d(\theta)$ and from d_m we can determine region t_r , within which the robot can detect this segment at any location.

If we are dealing with several segments, and by calling the segments detected by the robot s_1, s_2, \dots, s_n , we will obtain respectively for these segments the signals $d_1(\theta), d_2(\theta), \dots, d_n(\theta)$. If there is no intersection between segments, then the global function becomes :

$$d(\theta) = d_1(\theta) + d_2(\theta) + \dots + d_n(\theta) \quad (3)$$

In case 2 signals $d_1(\theta)$ and $d_2(\theta)$ interfere (intersection between 2 segments), then the resulting function will be :

$$d_r = \max(d_1(\theta), d_2(\theta)) \text{ for } \theta_1 < \theta < \theta_2 \quad (4)$$

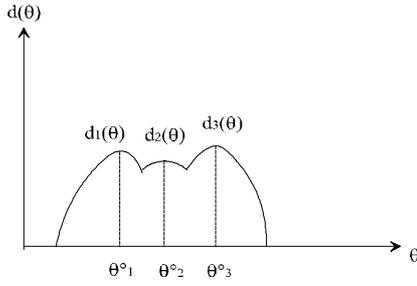


Fig.3 represents several signals with interference

The sensors information will be defined by the angles $\theta_{01}, \theta_{02}, \dots, \theta_{0n}$ that correspond to the maximum of the signals $d_1(\theta), d_2(\theta), \dots, d_n(\theta)$.

Before defining the various regions originated from the sensors information, it would be interesting to review the functioning method of two kinds of sensors: the ultrasonic and the laser range :

- The laser sends a beam v_s , that will be reflected against the segment of obstacle's plan. The beam will return to its transmitting source, provided the angle between the position of the transmitter and the segment of obstacle's plan, where the reflection occurs, doesn't exceed a threshold ϵ_s . Otherwise, the beam will diverge and will be completely lost by the sensor. In this case, we wouldn't obtain any information on the segment of obstacle.
- If an ultrasonic sensor is used, it will send a transmitting cone with an opening angle θ'_s . The ultrasonic sensor will then detect any segments of obstacles or parts of segments, which happen to be inside the transmitting cone.

We notice that both types of sensors acquire the same information when $\theta'_s = \theta_s$.

If we do not take into account the maximum reflection angle ϵ_s of the laser sensor's beam v_s , the region t_r will have a rectangular shape, which dimensions are $d_m.l_s$, l_s representing the length of the obstacle segment associated with this region. Otherwise, if $\epsilon_s \neq 0$, then the signal obtained for a segment will be truncated at both ends. (Refer to fig.4). We obtain the following relations:

$$d = l \cos(\theta - \theta_0) \quad (5)$$

for $\theta_0 - \theta_s < \theta < \theta_0 + \theta_s$,
 $d = 0$ otherwise.

$d(\theta)$

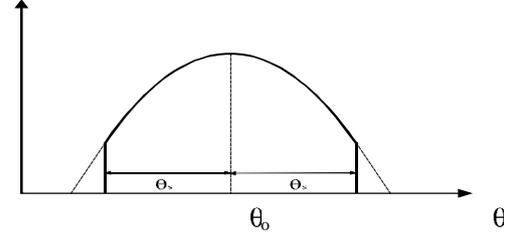


Fig. 4 Representation of signal $d(\theta)$ with a reflection θ_s .

2.2 Representation of the model regions

Taking into account angle θ_s , the region t_r will expand in the two opposite sides of the rectangles (along sides d_m), increasing by two angular sections, each with radius d_m and angle θ_s (as shown in fig.5).

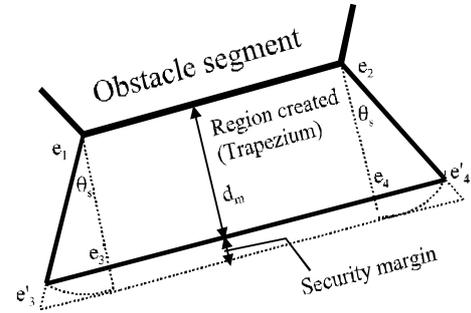


Fig. 5 Representation of a region where the robot detects an obstacle segment

The resulting zone is not polygonal, since it is curved in two locations ($e_3e'_3$ and $e_4e'_4$). To simplify matters, we prolong line e_3e_4 (which is parallel to obstacle segment e_1e_2) on both sides, until it intersects with lines $e_1e'_3$ and $e_2e'_4$. In this way, we obtain a new region t_r that has an exact trapezium shape.

In order to improve the robot detection, the distance d_m between robot and segment must be shortened, and a maximum security margin distance is required. Since we intend to keep distance d_m at a reduced size and the security margin high, the errors resulting from simplifying region t_r will remain negligible.

For each obstacle segment, we associate a trapezium which contains the information on its segment. We will have as many trapeziums as there are segments. When two trapeziums intersect, the robot will detect the two corresponding segments, in the intersecting area.

In order to obtain maximum robustness during the control application for planning, we will consider that the robot moves from one region to another by

changing one single sensor information, i. e. by adding or removing one single sensor information.

If we divide the union of two trapeziums $A_1 \cup A_2$ (\cup representing the union operator) in three sections, we have:

- $A_1 \setminus A_2$ (\setminus represents A_1 without A_2).
- $A_1 \cap A_2$ (\cap represents the intersection operator between the first and the second element).
- $A_2 \setminus A_1$ (A_2 without A_1).

Since region $A_1 \setminus A_2$ belongs to trapezium A_1 , the robot detects the same information as in A_1 , i. e. angle θ_{01} at the peak of signal $d_1(\theta)$.

In the same way, since $A_2 \setminus A_1$ is included in region A_2 , the robot detects the same information as in A_2 , i. e. angle θ_{02} at the peak of the signal $d_2(\theta)$.

Section $A_1 \cap A_2$ belongs to both regions A_1 and A_2 , so that the robot will detect the two information related respectively to regions A_1 and A_2 , which are angles θ_{01} and θ_{02} .

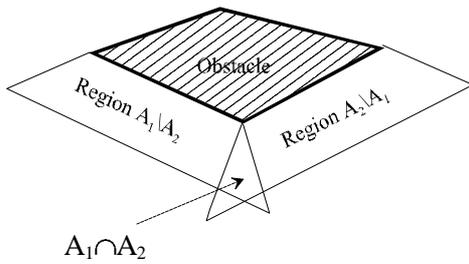


Fig. 6 Representation of the intersection between two regions A_1 and A_2

We notice that section $A_1 \cap A_2$ gives one additional information to the ones given by the two sections $A_1 \setminus A_2$ and $A_2 \setminus A_1$, which means that one information change has occurred between $A_1 \cap A_2$ and the two other sections. It follows that the robot can directly pass from region $A_1 \cap A_2$ to the two other regions and vice versa. Region $A_1 \cap A_2$ will serve as intermediary for the passage of the robot from region $A_1 \setminus A_2$ to region $A_2 \setminus A_1$, and inversely.

If we take into account all the regions issued from the intersections between the different trapeziums, it will become necessary to use a node graph to represent the various regions created and their links. This will be subject of next chapter.

3 Graph Computation

The node graph construction is based on the links between the regions. A link between two regions is

established when an information change between these two regions has occurred. We can eventually increase the number of links by changing (increasing) the length d_m of the trapezium. The graph construction will serve to find all possible paths between regions, and may also serve as a step for searching the best robot's trajectory.

Before defining the graph construction's algorithm, it will be interesting to determine the complexity of the number of region's, i.e. the maximum number of regions created according to the number of obstacle segments. The complexity is in fact a sum of combinations, defined by :

$$R_m = C_m^1 + C_m^2 + \dots + C_m^j + \dots + C_m^m \quad (6)$$

with $C_m^j = m! \setminus (m-j)! \setminus j!$

C_m^1 representing the number of regions that belong only to one trapezium, i.e. the regions which have only one information.

C_m^j representing the number of regions issued from the intersections of a number j of trapeziums, or that belong to j trapeziums. they have therefore j information.

C_m^m representing one single region issued from the intersections between all the trapeziums.

We designate a number m of trapeziums by A_1, A_2, \dots, A_m . We check whether an intersection between A_1 and A_2 exist. If this is the case, then region A_1 will be divided in two sections $A_1 \setminus A_2$ and $A_1 \cap A_2$, and we obtain in addition the part of section A_2 that does not belong to $A_1, A_2 \setminus A_1$. This translates into :

$$2 * 1(\text{two sections of } A_1) + 1(A_2 \setminus A_1) = 3 \text{ regions.}$$

If region A_3 intersects respectively with these three regions, then each of these will be divided again in two new subsections. A seventh region will be constituted by the portion of section A_3 that does not belong to A_1 and A_2 , which translates into :

$$2 * 3(\text{all } A_1 \cup A_2 \text{ subsections}) + 1(A_3 \setminus (A_1 \cup A_2)) = 7 \text{ regions.}$$

We notice that each time a trapezium intersects with a region, the number of new regions is multiplied by two and increased by one. If this statement is applied to a number m of trapeziums, we will have :

$$R_m = 2^m - 1 \quad (7)$$

We can see that the complexity increases very quickly (exponentially), depending on the number of segments. It is however proven in practice that matters are much less complex, which can be explained as follows:

- The probability that a region belongs simultaneously to m trapeziums decreases when m increases. In this case, this region will be the m^{th} 's subdivision of the set of trapeziums and will be sufficiently small to be neglected, if m represents a large figure.
- It is uncommon, in practice, that a region belongs to more than three trapeziums. A large number of regions do not have more than one or two intersections with other trapeziums, while some do not have any.

We conceive the node graph in light of these considerations. The graph is composed of a set of summits (nodes) and arcs. Summits represent regions, while arcs represent the links existing between regions. A link is established between two regions, each time the robot goes through a change of sensor information, i. e. when it crosses from one region to another.

Links between two regions are reciprocal : if the robot is able to cross from region A_1 to A_2 , it can also cross back from region A_2 to A_1 . The graph (and the arcs) is therefore not oriented. There is no difference between an arc's origin and its extremity, since the two summits, which are connected to this arc, point towards each other. Two coupled summits form a loop.

For the data-processing representation of the graphs, we have chosen the structure of lists' representation, since it is the most suitable to our study. The graph construction is realized by means of an algorithm that uses chained nodes (structures) lists.

Let ListTrap indicate the list representing trapeziums A_1 to A_m , and let I be a variable indicating the serial number of trapeziums under treatment. (I 's initial value is 2, and is incremented by 1 with each new treatment) We also initialize the $A_1 \setminus A_2$ and $A_1 \cap A_2$ links.

The algorithm consists in building regions in three steps:

- The first step consists in placing, in the graph GraphDiff, $A_1 \setminus A_{I-1}$, which is the exclusion between the A_I and A_{I-1} elements of ListTrap. We keep the same links as those of the $I-1^{\text{th}}$ treatment.
- The second step consists in placing, in the graph GraphAnd, $A_1 \cap A_{I-1}$, which is the intersection between the A_I and the A_{I-1} of ListTrap. Again, we keep the same links as those of the $I-1^{\text{th}}$ treatment.
- The third step consists in calculating an element of exclusion X , which has to be calculated separately from the two preceding lists. In other words, it

consists in calculating the exclusion between ListTrap's I^{th} element and the union of it's preceding elements (A_1 to A_{I-1}), i. e. $A_I \setminus (A_{I-1} \setminus (\dots \setminus A_1))$.

After each treatment (each time I is incremented by 1), the GraphAnd list is connected to the GraphDiff list, and the excluded element X is inserted at the end of GraphAnd.

The GraphDiff list is serial numbered from 1 to $(2^{I-1} - 1)$, the GraphAnd list from 2^{I-1} to $(2^I - 2)$ and the excluded element X will be to $2^I - 1$.

Let J and k represent variables indicating the respective serial numbers of the new regions created by trapeziums' intersections.

With each treatment, the following will be carried out :

- between GraphDiff and GraphAnd, for ($j=1$ to $j=2^{I-1} - 1$), a link between the J^{th} and the $(J + 2^{I-1} - 1)^{\text{th}}$ elements will be established.
- between GraphAnd and excluded element X , for ($k=1$ to $k=I-2$), links are established between element X and the $(2^{I-1} - 1 + 2^0)^{\text{th}}$ element, between X and the $(2^{I-1} - 1 + 2^0 + 2^1)^{\text{th}}$ element, ... ,and between X and the $(2^{I-1} - 1 + 2^0 + 2^1 + \dots + 2^k)^{\text{th}}$ element.

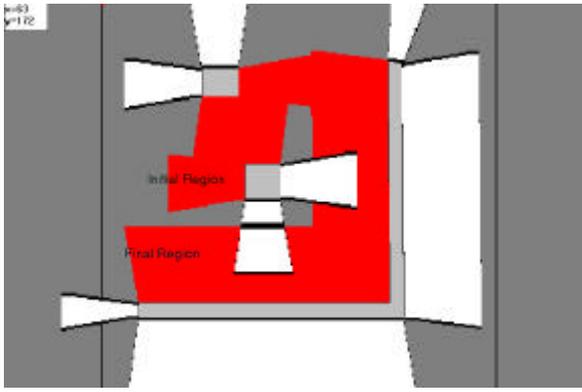
The same operation is carried out until ListTrap's last element. Regions with one single information will have at most $m-1$ elements, while regions with more information will have at most m links.

Once the graph construction has taken place, several potential paths between regions become available. We will have to select those responding best to the optimal criteria.

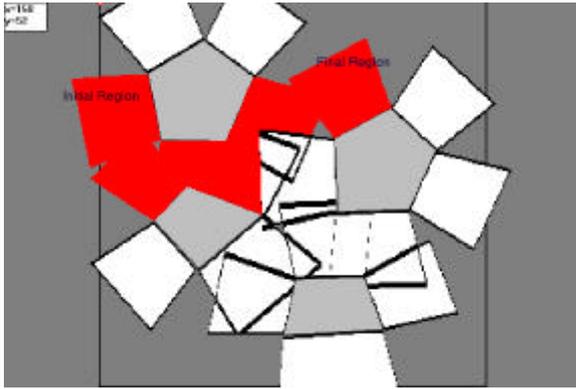
4 Optimality criteria

The selection of the most suitable path is attained by applying an optimal criteria, which is based on the search of the most robust path.

The most robust path is the one with the least number of sensor information changes. Since a change of sensor information occurs each time the robot crosses from one region to another, the optimal path will correspond to that involving the least number of regions crossed. The selected algorithm does not use heuristic functions, as is the case for A^* algorithm.



(a)



(b)

Fig.7 Example of path planning (grey) in the presence of rectangle (a) and polygonal (b) obstacles

5 Control

Once the graph is built and the optimal path determined, it becomes necessary to apply a control, in order to move the robot from one region to another.

The vehicle is maintained at a constant speed, and a control is applied on the vehicle's orientation. The control equation will be :

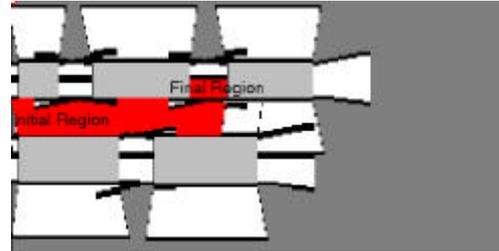
$$e = \int_{\dot{\alpha}}^{2\pi} (d(\theta) - d'(\theta - \theta_d)) \delta\theta \quad (8)$$

$d(\theta)$ represents the model (signal) seen on fig. 1 that we wish to obtain on the obstacle segment, and $d'(\theta - \theta_d)$ the signal obtained from the sensors on this segment. θ and θ_d correspond respectively to the orientation of the segment and the robot with respect to the global reference (x-axis reference). e represents the sum of differences between signals $d(\theta)$ and $d'(\theta - \theta_d)$ for $0 < \theta < 2\pi$. To obtain θ_d , we proceed with the calculation of the sum of products between $d(\theta)$ and $d'(\theta - \theta_d)$ such as θ_d gives :

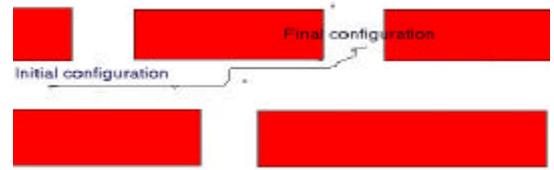
$$P_c = \text{Max} \left(\int_{\dot{\alpha}}^{2\pi} (d(\theta) \cdot d'(\theta - \theta_d)) \delta\theta \right) \quad (9)$$

To cross from one region to another, the robot might either move away from the segment and cross to another region when it stops detecting the segment, or move towards the segment, detect it and enter into a new region.

6 Application



(a)



(b)

Fig. 8 Path planning (a) and execution (b) for the passage of doors

Fig.8 (a) and (b) give an example of path planning and execution for the crossing of doors.

Fig.8(a) illustrates the path planning of the robot in the environment's model.

Fig.8(b) illustrates the path execution of the robot in the real environment.

The robot in the real environment crosses the second door to the left of the corridor (Fig.8b), as indicated by the model's grey colour path (Fig.8a). It shows the robustness of the control, as regards the errors of the model when, in the real environment, the corridor is longer than defined in the model of the environment.

7 Conclusion

In this article, we have proposed a new method for trajectory planning. The main interest of this method is in the proposed control, which is able to determine the robot orientation at each moment. This allows the robot to locate itself (or identify its own position) with respect to a local referential (the region in which he is located). Since the robot orientation is permanently calculated, out of the measures obtained by the ultrasound sensor, the orientation calculation errors do not accumulate. since we are not using odometry,

i.e. not calculating the position of the robot, we are avoiding the uncertainties (errors) deriving from the robot position calculations, and thus increasing the control accuracy.

The practical application of this method is also offering several advantages: Ultra sound detectors are less accurate than camera and laser sensors, (although ultrasound detectors have a greater scanning field than lasers), but are more economical and of easier utilisation. Our method, contributes in increasing the economical advantage of ultrasound detection, since odometry would be no longer needed.

References

- [1] J. C. Latombe, Robot Motion Planning. Boston, MA : Kluwer Academic Publishers, 1991.
- [2] J. C. Latombe, A. Lazanas and S. Shekhar, « Robot motion planning with uncertainty in control and sensing, » Artificial Intell. J., vol. 52, no. 1, pp. 1-47, 1991.
- [3] H. Takeda, C. Facchinetti and J. C. Latombe, « Planning the Motions of a Mobile Robot in a Sensory Uncertainty Field, » IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, no. 10, October 1994.
- [4] Christian Laugier. Planning Fine Motion Strategies by Reasoning in the Contact Space. In IEEE International Conference on Robotics and Automation, pages 653-659, 1989.
- [5] F. De La Rosa, J. Najera and C. Laugier, Planning robot motion strategies under geometric constraints, in : Proc. Int. Symp. On Intelligent Robot Systems (SIRS'94) pages 37-44,1994.
- [6] R. Alami and T. Simeon. Planning Robust Motion Strategies for a Mobile Robot. In the IEEE International Conference on Robotics and Automation, 2 :1312-1318, San Diego, USA - May 1994.
- [7] I. Collin, D. Meizel, N. L. Fort and G. Govaert. Local Map Design and Task Function Planning for Mobile Robots, Proceedings of the International Symposium on Intelligent Robotic Systems IRS'94, pp. 22-29, Grenoble, 1994.
- [8] 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, pages 99-114, April 1997.
- [9] Alain Pruski, Robotique Mobile, Hermès France 1996.