



HAL
open science

Indoor Navigation with Uncertainty using Sensor-Based Motions

Maher Khatib, Bertrand Bouilly, Thierry Simeon, Raja Chatila

► **To cite this version:**

Maher Khatib, Bertrand Bouilly, Thierry Simeon, Raja Chatila. Indoor Navigation with Uncertainty using Sensor-Based Motions. IEEE International Conference on Robotics and Automation (ICRA 1997), Apr 1997, Albuquerque, United States. 10.1109/ROBOT.1997.606804 . hal-04295423

HAL Id: hal-04295423

<https://hal.science/hal-04295423>

Submitted on 20 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Indoor Navigation with Uncertainty using Sensor-Based Motions

M. Khatib

B. Bouilly

T. Siméon

R. Chatila

LAAS-CNRS

7, avenue du Colonel-Roche
31077 Toulouse Cedex - France

Abstract: *This paper presents an operational framework to bridge the gap between planning with uncertainty and real-time sensor-based motion control. The environment being known, a planner produces a plan composed of free space and sensor-based motion commands. The representations of uncertainty and its evolution, environment landmarks, and actions generated at the planning level are discussed. Sensor-based actions and command definitions for a nonholonomic mobile robot based on a Task-Potential field approach are developed. These various elements are integrated in a system that actually generates the motions of the **Hilare2** mobile robot.*

1 Introduction

Motion planning and control are two important aspects required for mobile robot autonomous navigation. However, they are frequently considered as two separate phases. Many motion planning techniques have been proposed (see [7] for a survey) to produce a collision-free and feasible path avoiding the obstacles of the workspace. Motion control then consists in making the robot follow this path, possibly using a potential-field based method [12, 6] for reacting to unexpected obstacles sensed during the execution. Such an approach suffers from a lack of robustness when experimented in real world settings since the execution of the planned trajectory is conditioned by several sources of uncertainties.

Control and sensing errors are particularly significant in the context of mobile robot navigation. Indeed, mobile robots are not generally equipped with an absolute positioning system and are subject to cumulative dead-reckoning errors. Inaccuracies in localizing the robot with respect to its environment may lead to failures during the execution of the motions eventually preventing from reaching the goal.

To overcome the uncertainty accumulated during

the motions, the robot has to be equipped with environment sensors that can provide additional information by identifying appropriate features of the workspace. Localization techniques (eg. [4, 9]) can be applied to reduce the uncertainty of the estimated configuration. One may also use sensor-based motion commands which are more tolerant to errors than classical position-controlled primitives. Sensor-based control has been so far limited to specific implementations - e.g., wall-following using ultrasonic sensors [11], vision [13]-, without connection with a planning component. It is however preferable to reason during the planning process on the uncertainty that will be accumulated during the motions and on the capacities of the available sensing functions in order to produce robust motions plans that will reliably guide the robot towards its goal.

Among the contributions to motion planning under uncertainty [7] which addressed this issue, only a few described navigation experiments performed with real robots: the landmark-based planner of [8], applies the preimage-backchaining approach [10] for navigating a mobile robot subject to control errors through circular landmarks where it can exactly determine its position. The SUF-based planner of [15] allows to take into account more sophisticated models of sensing uncertainty for generating a trajectory that minimizes the expected localization errors.

In this paper, we consider the navigation problem for a mobile robot operating, in presence of *control and sensing* errors, in an a priori known office-like environment. We present the algorithms that have been developed and integrated onto our **Hilare2** robot for achieving robust motion planning and control. The planner, based on the approach presented in [3], considers a set of motion commands which may accumulate uncertainty and plan the corrective sensor-based motions needed to reduce it. It is able, to combine several sensing modalities, and also to navigate, when-

ever possible, without relocalizing the robot when the task does not impose it. The sensor-based motions are implemented based on a general Task-potential field approach [11, 5] which allows to directly express the potential function from a space parametrized in terms of the sensor output variables. This system actually runs in our laboratory and enables **Hilare2** to achieve motion plans robustly.

2 The motion primitives

We first describe the available motion primitives defined both at planning and execution level. These primitives can be classified into two categories: **Open-Loop Free Space Motions**¹ and **Sensor-Based Motions**.

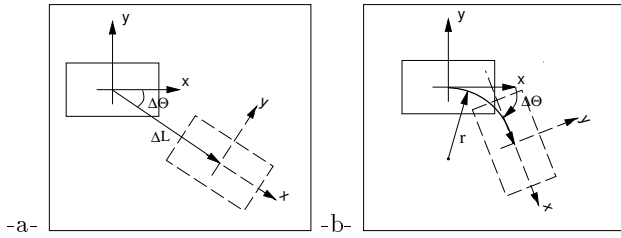


Figure 1: Two examples of the **MOVE** command: with parameters $(\Delta\Theta, \Delta L, 0)$ (-a-) and $(\Delta\Theta, 0, r)$ (-b-)

MOVE $(\Delta\Theta, \Delta L, r)$: is a composed free-space motion command controlled by odometry. It corresponds to a straight-line motion of length ΔL executed after a circular arc of angle $\Delta\Theta$ and radius r (Fig. 1.a).

MOVE_LANDMARK $(\Delta\Theta, \Delta L, r)$: has a similar definition as the previous command. However this command uses the measure provided by an external sensor to perform an absolute relocalization of the robot during the motion.

SBC_MOVE_TO_CONTACT (d_c, L_{max}) : is a guarded motion command which executes a straight line motion stopped when an object is detected at a frontal distance d_c (Fig. 2-a). A maximum motion length is defined by the parameter L_{max} .

SBC_PARALLELIZE $(\Theta_{max}, side)$: is a sensor-based command executed to align the robot to a wall (Fig. 2-b). Θ_{max} is the maximum allowed rotation and $side$ determines the direction where the wall has to be searched.

SBC_WALL_FOLLOW $(walls_{list})$: is a composed sensor-based command for following a succession of walls (Fig. 2-c). Several events (vertex, traveled distance, obstacles, ...) are checked during its execution. Each wall is defined by a structure where:

¹ie. using no exteroceptive feedback

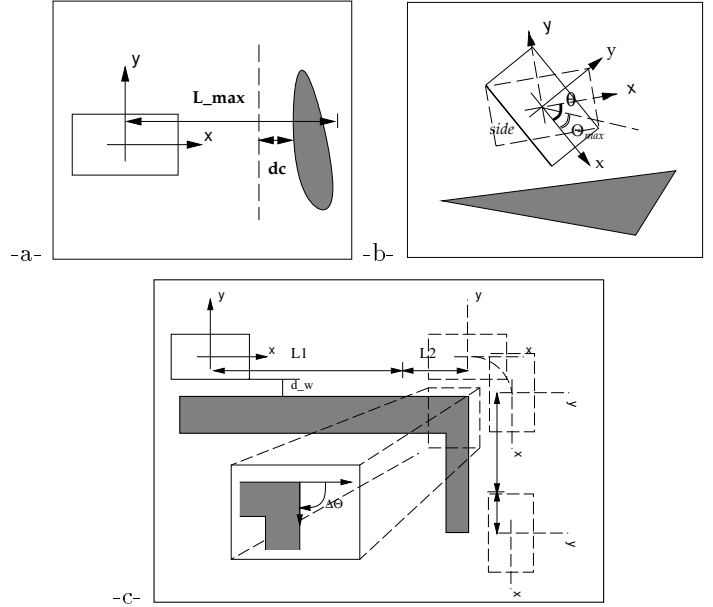


Figure 2: Sensor-based motions commands.

- $side \in \{\text{left}, \text{right}\}$ is the wall relative location,
- d_w is the distance to the wall,
- L_1 is the length of the motion executed without activating the vertex detection system ,
- L_2 is the maximum motion length allowed for detecting and matching a vertex defined by its type ($type \in \{\text{Convex}, \text{Concave}\}$) and by the relative orientation $\Delta\Theta$ of the next wall in $walls_{list}$.

3 Planning robust strategies

3.1 Problem statement

The planner [3, 2] considers a polygonal description of the obstacles and also a set of landmark regions where the robot can be relocalized in the environment. It maintains a model of the uncertainty in order to produce robust motion strategies composed of a sequence of the primitives defined in section 2. The strategy guarantees that the robot, starting from an uncertain initial position, can reliably reach its goal with an error lower than a pre-specified value, despite the errors on its position accumulated during its motion. The uncertainty can be reduced either by entering in the landmark regions (absolute relocalization) or by detecting some edges or vertices of the known polygonal obstacles.

3.2 Dealing with uncertainty

For the planner, motion primitives are defined by the initial situation where they can be applied, by the

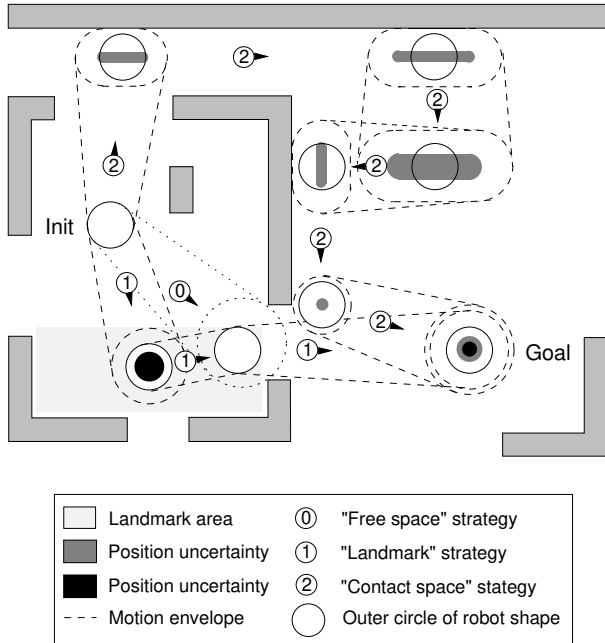


Figure 3: Motion strategies possibly generated by the planner

set of final situations they can reach, and by the trajectory envelopes they induce. A distinction is made between free-space motions controlled by odometry only and sensor-based motions controlled by both odometry and proximity sensors, or by the external sensor used in the localization areas.

- **MOVE** primitives induce cumulative errors due to odometry. The position uncertainty is then defined as a function linearly dependent on the distance covered by the robot.
- After a **SBC_MOVE_TO_CONTACT**, uncertainty is set to the proximity sensor accuracy along the direction orthogonal to the edge. For the **SBC_WALL_FOLLOW** command, the error linearly grows only in the direction of the followed edge because of the use of the odometric sensor to determine the covered distance.
- **MOVE_LANDMARK** uses an absolute localization of the robot which resets the position uncertainty to the sensor accuracy modelled by a disc of fixed radius.

Figure 3 shows the uncertainty evolution for different strategies that could be produced by the planner for going from "Init" to "Goal" position. Free space motions only are not possible since the accumulated error would be too important to safely pass

the doorway. Strategy 1 solves the problem by traversing a landmark region (within which the uncertainty remains constant) whereas strategy 2 illustrates the evolution of uncertainty using wall-following motions.

3.3 Planning approach

The algorithm is based on the propagation of a numerical potential modeling the uncertainty over a grid superposed on the workspace.

During the propagation of the potential, collisions are checked between the obstacles and the robot shape grown by the current uncertainty. The robot is approximated by its enclosing disk during the free-space motions. The exact shape is only considered during the wall-following commands for which the robot has to be aligned with the wall. This allows to solve constrained problems (eg. crossing of narrow doorways) where the circular approximation would be too conservative, while only considering a two-dimensional configuration space.

When the propagation reaches a position which may collide with a given edge because of a too large uncertainty, a geometric visibility test is performed to check if a **SBC_MOVE_TO_CONTACT** is guaranteed to end onto this edge. When this test succeeds, the algorithm starts from this position a propagation adapted to the uncertainty model of the **SBC_WALL_FOLLOW** command. Similarly, a propagation adapted to the **MOVE_LANDMARK** command is started at positions where the robot grown by its current uncertainty is included in a landmark region.

Finally, once the goal position is reached with an admissible uncertainty, the motion plan is deduced by backchaining configurations from the goal to the initial position. The strategies returned by the planner either minimize the distance traveled by the robot (ignoring in this case relocalizations when the task does not impose it) or the final uncertainty.

4 Sensor-based actions

The sensor-based motion primitives are defined at the execution level by specific functions called *Task-Potential* functions. The artificial potential field method [12] is the basis of a *Task-potential* concept developed in [11, 5]. This approach can be viewed as increasing the low level control capabilities. Here, we present the basic idea of this concept and describe how to implement low-level sensor-based actions.

Task-potential Formulation

Let \mathbf{q} denotes the robot configuration. If $\mathcal{F}_i(t)$, $t \in [0, T]$, $i \in [0, \dots, n]$ are positive and monotonous

scalar functions of time, and $\mathcal{P}_i(\mathbf{q})$ are positive, differentiable and convex potential functions, the potential function $\mathcal{P}(\mathbf{q}, t)$ obtained from a linear application

$$\mathcal{P}(\mathbf{q}, t) = \sum_i \mathcal{F}_i(t) \mathcal{P}_i(\mathbf{q}), \quad \forall t \in [0, T], \quad (1)$$

is a convex potential function where $\frac{\partial \mathcal{P}}{\partial \mathbf{q}}|_{\mathbf{q}_m, t=T} = 0$, and has \mathbf{q}_m as a unique solution. $\mathcal{F}_i(t)$ is called a *control function*.

Proposition 4.1 *Executing and controlling a robot sensor-based task can be expressed by the gradient descent of a Task-Potential function $\mathcal{P}(\mathbf{q}, t)$, $t \in [0, T]$, within a linear composition of several **convex** potentials.*

Proof and details are developed in [5]. Based on this proposition, we give, in the following subsections, the implementation ideas for sensor-based actions needed by our system.

4.1 Move To Contact

The contact is defined by a distance d_c between the robot and the object (Fig. 2). If d is the current distance to the wall, the robot has to generate a motion which minimizes the term $\|d - d_c\|$. The robot displacement is limited by the length L_{max} (see definition in section 2). Using the Task-potential formulation (eq. 1), this can be achieved by the following convex potential

$$\mathcal{P}_c(d, t) = f(t) P_c(d),$$

where $f(t) \in [0, 1]$, $t \in [0, T]$ is an onto monotonous decreasing function of the current displacement value $l(t) \in [0, L_{max}]$. In fact, $f(t)$ insures a null potential value for $l(t) \geq L_{max}$. $P_c(d)$ is a user-defined convex potential having its global minimum at $d = d_c$ (an example of such potential and control functions are proposed in [5]). Here, the distance measures and the force are totally projected on the robot x-axis thus generating, a pure linear acceleration. The asymptotic stability, at the global minimum, is insured by adding a dumping term to the total force giving

$$\mathbf{F}_c = -\nabla \mathcal{P}_c - k_v \dot{\mathbf{q}}, \quad (2)$$

where k_v is a positive gain and $\dot{\mathbf{q}}$ denotes the robot velocity.

4.2 Parallelize

The aim of this command is to position the robot in a valid configuration to start a wall following command. More precisely, the goal is to minimize the angle θ between the wall segment and the robot x-axis

(Fig. 2.b). This action formulation is completely similar to the previous one. But here, the force derived from the potential represents a pure torque around the robot rotation center. This is why the dumping term depends on the angular velocity (eq. 2).

4.3 Wall Following

We assume here that the robot is initially placed at a position such that the wall is detectable by the sensors. We desire it to follow the wall keeping a given nominal distance d_w from it. This task is maintained until a certain condition is disabled (e.g. end of the wall reached, traveled distance value, etc.).

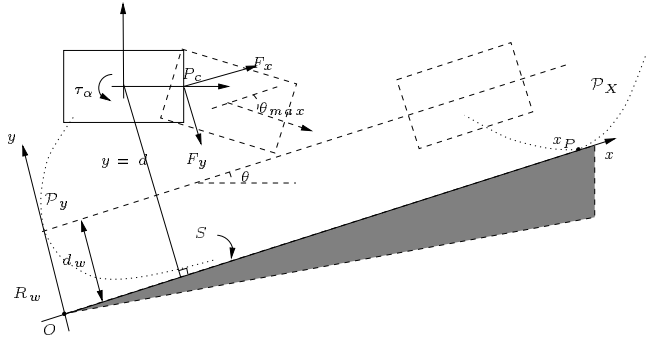


Figure 4: *Wall Following idea.*

The wall is represented by a segment S lying on the x-axis of a Fresnel frame R_w associated to it (Fig. 4). The robot configuration $\mathbf{q} = (x, y, \theta)^T$ is expressed in this frame. In order to achieve this action, we imagined an attractive potential \mathcal{P}_x along the wall x-axis that attracts the robot to the end of the wall at the position x_p . This potential will guide the robot along this direction until the global minimum at $x = x_p$. Similarly, to maintain a d_w distance, we attribute to the wall y-axis a potential \mathcal{P}_y , function of the robot y position whose minimum is at $y = d_w$. In addition, in order to keep the wall visible by the sensors, we have to achieve an incidence angle θ_{max} to the wall. Thus, we introduce a rotational potential \mathcal{P}_θ , function of the robot orientation whose global minimum is at $\theta = 0$. \mathcal{P}_θ will aim to maintain the parallelism with the wall. The task potential is thus:

$$\mathcal{P}_{wf} = \mathcal{P}_x + \mathcal{P}_y + \mathcal{P}_\theta.$$

In order to insure a collision-free motion, a repulsive potential, function of the shortest distance to the robot, can also be added to keep it away from unexpected obstacles occurring within a certain distance.

All force components derived from task potentials are applied to a *control point* P_c defined on the robot

x-axis at a certain non-null distance from its center. P_c is considered as a holonomic point [14, 1] for it can move in all directions whereas the robot is non-holonomic. This is a key issue of these actions stability proof developed in [5]. Execution control, however, can be insured using some suitable control functions (e.g. if $l(t)$ represents the current displacement, the potential value has to be null for $l(t) \geq L_1 + L_2$, see section 2).

The angular variation $\Delta\theta$ and the type of vertex, allow to achieve the wall switching using the following potential function of vertex position (x_v, y_v)

$$\mathcal{P}_{sw} = \frac{((x - x_v)^2 + (y - y_v)^2 - r^2)^2}{(x - x_v)^2 + (y - y_v)^2}$$

$r = d_w + \frac{E}{2}$, where E is the robot width. Transition between the wall task potential \mathcal{P}_{wf} and the switch wall potential \mathcal{P}_{sw} can be done by a pure convex composition such as

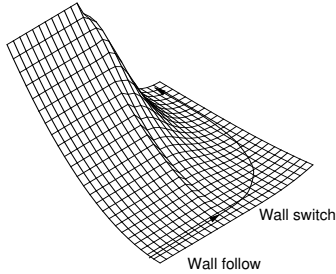


Figure 5: Potential distribution for switching walls.

$$\mathcal{P} = (1 - f_{wf}(t)) \mathcal{P}_{wf} + f_{wf}(t) \mathcal{P}_{sw}$$

where $f_{wf}(t) \in [0, 1]$ is a monotonous increasing control function, depending on the robot abscissa in the frame R_w . Figure 5 shows the distribution of the potential \mathcal{P} . Switching to the next wall task potential can also be achieved by such a control function which depends, in this case, on the executed angular variation with respect to the given value $\Delta\theta$ describing this vertex.

5 Experimental work

The **Hilare2** robot is equipped with a belt of 32 sonars and a laser range finder. Sonar data are used for the wall-following, move to contact and parallelize actions. The laser is used for edge detection and matching during a wall-following. A VME rack supporting 6 CPU boards of the Motorola 680x0 family is mounted

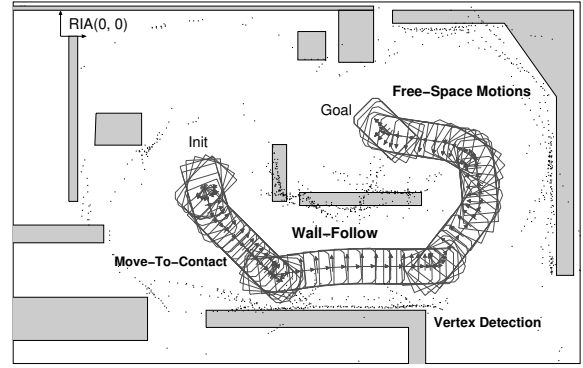


Figure 6: Experimentation 1

on the robot running under the VxWorks real-time system.

Figure 6 shows one of the experimentations. **Hilare2** has to move from *Init* to *Goal* position in a known environment. Given the important uncertainty value of the initial position, the goal is not reachable using a direct open-loop free-space motion. That is why, the planner strategy was to generate several sensor-based actions (move to contact, follow wall until vertex) to reduce the uncertainty, then the goal could be reached with open-loop actions.

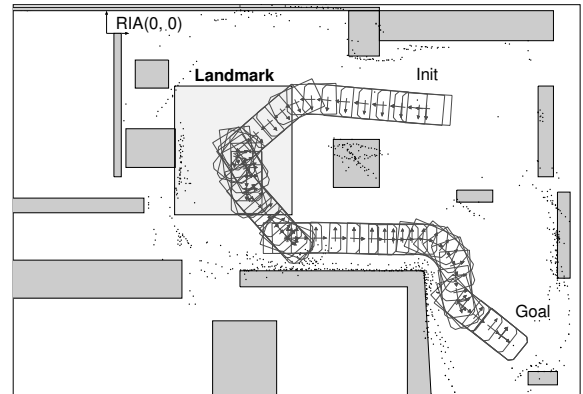


Figure 7: Experimentation 2

The experiment of figure 7 corresponds to a case where the wall cannot be reliably reached with open-loop actions, because of their uncertainty evolution. The motion plan first reduces the position error by using the landmark zone before seeking the contact with the wall. Dynamic robot relocalization in the landmark zones has not been implemented yet; several **MOVE_LANDMARK** actions are sent to the execution level to achieve this task.

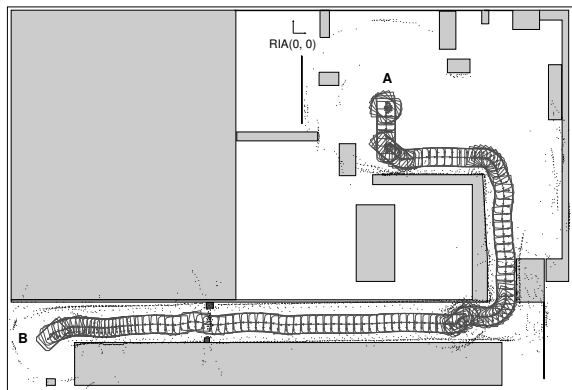


Figure 8: Experimentation 3

A more significant task is shown in figure 8. Indeed, this mission is not feasible using open-loop actions in such a constrained space. We can note here the important traveled distance by the sensor-based commands. This task was successful almost at every execution, the robot runs more than (60m) at a rate of (.4m/sec) while following the walls. The robustness of the system lies not only in uncertainty reduction, but also in the ability of coping with environment variations (e.g., unexpected obstacles) thanks to the sensor-based motions. This is exemplified in the experiment shown in figure 8 where the doorway almost in the middle of the corridor was not modelled, and the wall following action can comply with this during execution.

6 Conclusion

Robot displacement uncertainties make its motion by open-loop actions unreliable, preventing it from reaching an assigned goal. We have presented here an operational framework linking planning with uncertainty and sensor-based motion, thus providing more robust execution. Hence, the planner generates an execution plan composed of both open-loop and sensor-based motions. At the planner level, robot sensing (for localization) and motion constraints are taken into account to produce a plan in terms of sensor-based actions. These actions are defined as gradient descents in adequate potential fields called Task-Potentials. Several experimentations with real data in an office environment show the robustness of the system. On-going developments consider more realistic sensor properties in terms of uncertainties and field of view.

References

[1] G. Campion B. d'Andréa Novel and G. Bastin. Control of nonholonomic wheeled mobile robots by state

feedback linearization. *The International Journal of Robotics Research*, 1995.

- [2] B. Bouilly. *Planification de stratégie de mouvement robuste pour robot mobile*. PhD thesis, LAAS-CNRS, Toulouse, France, December 1996. To be published.
- [3] B. Bouilly, T. Siméon, and R. Alami. A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty. In *IEEE International Conference on Robotics and Automation, Nagoya (Japan)*, 1995.
- [4] J.L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *IEEE International Conference on Robotics and Automation, Scottsdale (USA)*, 1989.
- [5] M. Khatib. *Contrôle du mouvement d'un robot mobile par retour sensoriel*. PhD thesis, LAAS-CNRS, Toulouse, France, December 1996. To be published.
- [6] B.H. Krogh and C.E. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. In *IEEE International Conference on Robotics and Automation, San Francisco (USA)*, 1986.
- [7] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [8] A. Lazanas and J.C. Latombe. Motion planning with uncertainty: a landmark approach. *Artificial Intelligence*, 76:287–317, 1995.
- [9] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.
- [10] T. Lozano-Perez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.
- [11] M. Khatib and R. Chatila. An extended potentiel field approach for mobile robot sensor-based motions. *Intelligent Autonomous Systems (IAS'4), Karlsruhe (Germany)*, 1995.
- [12] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
- [13] R. Pissard-Gibollet P. Rives and K. Kappellos. Development of a reactive mobile robot using real time vision. In *International Symposium on Experimental Robotics, Kyoto (Japan)*, 1993.
- [14] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. In *IEEE International Workshop On Intelligent Robots and Systems, Osaka, (Japan)*, 1991.
- [15] H. Takeda, C. Facchinetti, and J.C. Latombe. Planning the motions of a mobile robot in a sensory uncertainty field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(10):1002-1017, 1994.