15th Triennial World Congress, Barcelona, Spain

ROBOT NAVIGATION IN VERY COMPLEX, DENSE, AND CLUTTERED INDOOR/OUTDOOR ENVIRONMENTS

Javier Mínguez Luis Montano

Computer Science and Systems Engineering, CPS,
University of Zaragoza, Spain

Abstract: This paper addresses the reactive collision-free motion generation for indoor/outdoor robots which have geometric, kinematic, and dynamic constraints. Most of the current mobile robots are designed exhibiting some of these constraints: (1) typically they are circular, square or rectangular robots and (2) they are differential-driven robots, car-like robots, tri-cycle robots, etc. On the other hand, many navigation methods do not take into account the specific shape or robot's kinematics and dynamics. In this case, these methods relax some constraints or they rely on approximations. It is clear that this is a gap in research that needs to be closed, by devising mechanisms to generalize navigation methods to be applied over a wide range of mobile platforms. This paper focuses on the generalization of a reactive method - *Nearness Diagram Navigation* - to work over a fleet of geometric, kinematic, and dynamic constrained indoor/outdoor mobile robots. This framework has been extensively tested using four indoor and one outdoor robots equipped with different sensors. To validate the method, we report experiments in unknown, non-predictable, unstructured, cluttered, dense and complex environments.

Keywords: Autonomous mobile robots, obstacle avoidance, robot navigation.

1. INTRODUCTION

This paper addresses robot navigation. In typical mobile robotic missions the indoor/outdoor environment is unknown and non-predictable (e.g. offices, museums, planetary surfaces). From the navigation point of view, the different nature of these environments also imposes a wide range of difficulties: They are unstructured, non-predictable, dense, cluttered, etc. In addition, the robots that are designed to accomplish mobile tasks in these environments usually exhibit different shapes and specific kinematics and dynamics.

To design the safe motion generation task, there are at least three concepts to address: the method of navigation, the specific robot shape, and the robot's kinematics and dynamics.

The **navigation technique** that we use is the *reactive collision avoidance approach* (O.Khatib, 1986), (J.Borenstein and Y.Koren, 1991*b*), (I.Ulrich and J.Borenstein, 1998), (D.Fox et al., 1997), (J.Mínguez and L.Montano, 2000), (S.Quinlan and O.Khatib, 1993) among others. Based on a perception-action process, they generate collision-free motion commands in a goal-directed fashion. The main advantage is that they require a very low computational load. This provides the system with the possibility of a high-rate environmental feedback (which is well-suited in unknown and non-predictable environments). The drawbacks of these approaches are that they only produce sub-optimal solutions, and they cannot guarantee to reach the goal location.

Currently, most of the robots designed exhibit **kinematic** and **dynamic constraints**. For these robots, the navigation method has to take into account these constraints. Otherwise, the robot approximately exe-

---

[1] This work was partially supported by Spanish CICYT project DPI2000-1272.

(O.Khatib, 1986), (J.Borenstein and Y.Koren, 1991*b*), (J.Mínguez and L.Montano, 2000), (S.Quinlan and O.Khatib, 1993).

The robot's **geometric constraint**, that is the specific robot shape, is a much harder problem in reactive navigation. Classically, the robot shape is taken into account by translating the problem into the configuration space (J.C.Latombe, 1991), where a point represents the robot. To map the obstacles onto the configuration space is a difficult and very time-consuming task. Some reactive navigation methods avoid the configuration space computation by assuming circular geometry and checking collisions in the workspace (J.Borenstein and Y.Koren, 1991*b*), (J.Mínguez and L.Montano, 2000), (S.Quinlan and O.Khatib, 1993). If these methods are used over non-circular bases, the robot shape needs to be approximated under constraining the method solution.

This paper presents the generalization of a reactive navigation method - *Nearness Diagram Navigation* - to take into account the specific robot's shape, kinematics and dynamics.

2. THE FRAMEWORK

Our approach is based on breaking down the motion generation process into three sub-problems related with: (1) collision avoidance, (2) the kinematic and dynamic constraints, and (3) the geometric constraints:

(1) Assuming a circular robot free of motion constraints, the reactive collision avoidance approach - *Nearness Diagram Navigation* - is used to calculate the most promising motion direction and the desired velocity.
(2) The *Motion Generator* uses the information provided by the *Nearness Diagram Navigation* to calculate a motion command that takes into account the kinematic and dynamic constraints.
(3) If the robot is non-circular, the *Shape Corrector* modifies the pre-calculated motion command to take into account the robot's shape.

Fig. 1 illustrates the complete process. Seen as a whole, this framework generates the collision-free motion commands to drive a kinematic, dynamic, and geometric constrained mobile platform towards a given goal location.

We next analyze each of these modules. Let us start by the reactive method used - *Nearness Diagram Navigation.*

2.1 Nearness Diagram Navigation

The *Nearness Diagram Navigation* (ND) (J.Mínguez and L.Montano, 2000) is a reactive collision avoidance approach. Given depth information obtained from perceptions of the environment and a goal location, the method calculates a motion command (most promising motion direction and the velocity) to drive a holonomic and circular robot towards the goal location whilst avoiding collisions with the environment.

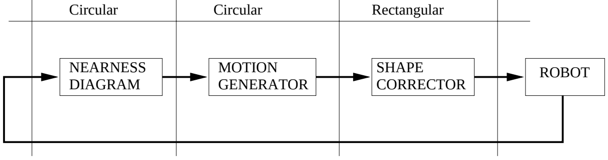


Fig. 1. The reactive navigation problem breaken down in subproblems.

The ND is designed using the situated-activity paradigm of behavioral design (R.Arkin, 1998). First, a set of situations is defined to fully describe the relative state of the robot, obstacle distribution and goal location. Subsequently, one action is designed for each situation. In real time, the perception is used to identify the current situation, and the associated action is executed generating the motion commands. The ND computes a velocity vector (velocity module, v_{ND}, and velocity direction, θ_{ND}) for each sample period.

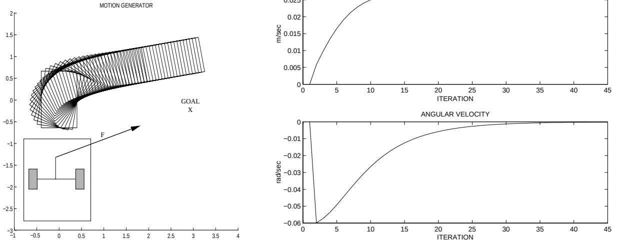
Good results in very cluttered, complex and dense environments have been reported using the ND. This is the main motivation to select the ND off-the-shelf for our framework. The ND's limitation is that it does not take the kinematic constraints, dynamic constraints, and non-circular shapes into account. This paper addresses a generalization of this reactive method to take these constraints into account.

2.2 The Kinematic and Dynamic Constraints

Once the ND has calculated the most promising motion direction and the desired velocity, we still need to convert this information into motion commands for the non-holonomic mobile base. To accomplish this we use the *Motion Generator*.

The Motion Generator (MG) (J.R.Asensio and L.Montano, 2002) is a dynamic model-based robot controller for differential-drive robots. Each sample period, the MG computes the velocity commands from a virtual force applied to a point on the robot. The MG has significant advantages:

(1) The model implicitly takes into account the kinematic and dynamic constraints, thus the controller provides feasible commands for a real robot.
(2) The model's parameters have a clear physical sense. This allows an easy tuning of the parameters in order to obtain the desired dynamic behavior (non overshooting during turns, limits in velocities and accelerations).
(3) The model filters the sudden changes in direction of commanded motions, which are usually produced by many reactive navigation methods.



(a) (b)

Fig. 2. a) Robot's trajectory. b) Translational and rotational velocities (motion command) obtained with the MG.

Fig. 2a illustrates the robot moving between two locations. The input of the MG is a force, $\overrightarrow{\mathbf{F}}$, recalculated at each sampling period, which "pulls" the model towards the goal location. The output of the MG is the motion command given to the differential-driven robot $\mathbf{v} = (v, w)$. See Fig. 2b.

We still need to connect the *Nearness Diagram Navigation* and the *Motion Generator*, Fig. 1. The most promising motion direction, θ_{ND}, and the velocity, v_{ND}, obtained with the ND are transformed into a force $\overrightarrow{\mathbf{F}} = (|F|, \theta_F)$, the input of the MG.

$$|F| = F_{max} \cdot \frac{v_{ND}}{v_{max}} \; , \quad \theta_F = \theta_{ND} \qquad (1)$$

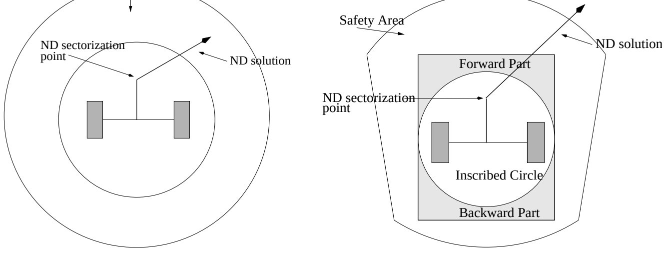
This framework calculates the collision-free motion commands to drive a circular and differential-driven robot towards the goal location.

2.3 The Geometric Constraint

At this point, we have used the *Nearness Diagram Navigation* and the *Motion Generator* to calculate the collision-free motion commands for a circular and differential-driven platform. So far, the reactive method only takes into account a circular geometry, other robot shapes are not considered. To extend this framework to take into account other geometries, we use the *Shape Corrector* (See Fig. 1). For now on we concentrate on rectangular shapes (the square shape is a particular case).

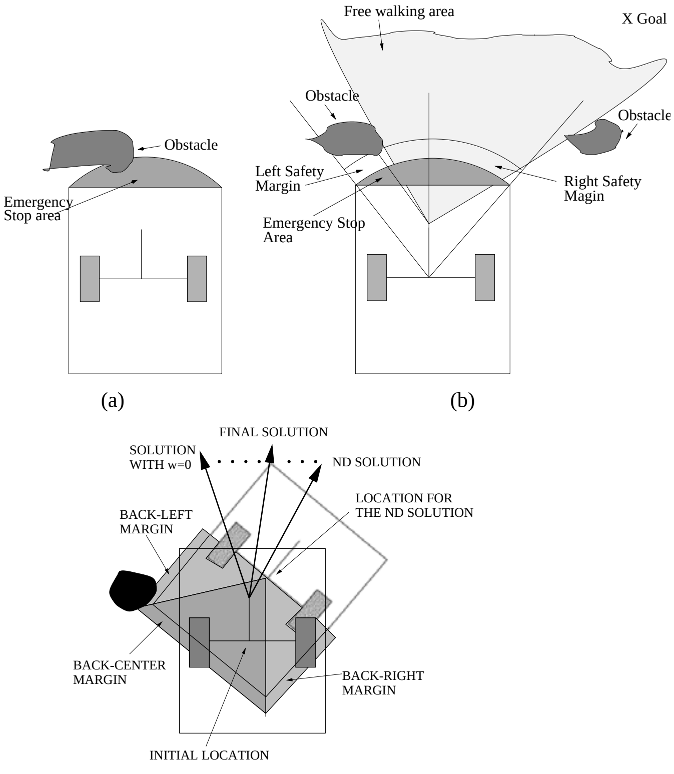
In a first step, the - ND + MG - framework is used to calculate the motion command approximating the robot shape by the inscribed circle (see Fig. 3a). (It is a non-conservative selection motivated by the type of environment: Very dense, complex and cluttered, where the circumscribed circle is a coarse approximation). The role of the *Shape Corrector* is to modify the pre-calculated motion command to protect the two parts of the robot not covered by the inscribed circle, (the forward and backwards part), see Fig. 3b.

The - ND + MG - is a framework that calculates motion commands to produce instantaneous forward motion $v >= 0$. (The reactive method is constrained to calculate instantaneous forward directions of motion). Then, the *Shape corrector* is based on three situations that exploit this constraint:



(a) (b)

Fig. 3. a) Circular robot. b) Rectangular robot.



(a) (b)

(c)

Fig. 4. a) Inminent collision.. b) Backwards collision danger. c) Forward collision danger.

(1) **Imminent collision**: The robot is in this situation when the collision cannot be avoided with any sequence of forward motions. The motion command stops the robot ($v = 0, w = 0$), because the collision is imminent. A flag is then fired to launch a higher-level module to remove the robot from this situation (e.g. a motion planner). Fig. 4a illustrates this situation: When there are obstacles within the *Emergency Stop Area* (calculated from the circumscribed circle to the robot shape), no motion with $v >= 0$ (forward motion) or set of them can avoid the collision.

(2) **Forward collision danger**: The robot is in this situation when there is a potential risk of entering into an *Imminent collision* situation. To detect it, we define a *Safety Margin* (divided into *Left* and *Right Safety Margins*) to enclose the *Emergency Stop Area*, see Fig. 4c. When there are obstacles within the *Safety Margin* the robot is

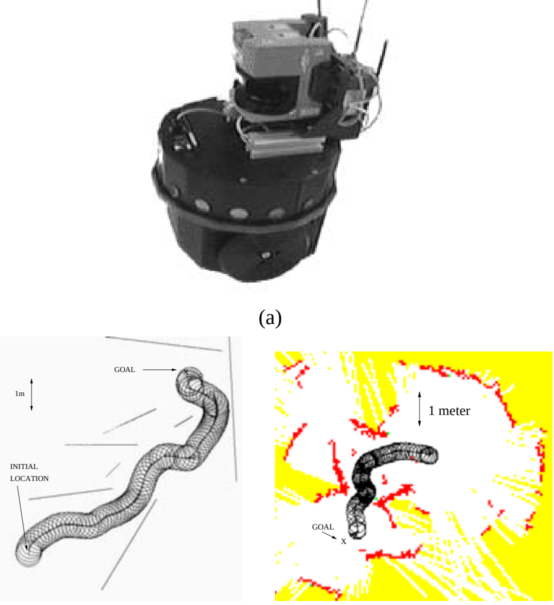
turn the robot in place ($v = 0, w >< 0$) to clear the *Safety Margin* of obstacles. The selection of the turn direction depends on: (1) the part of the *Safety Margin* that contains obstacles, and (2) an internal piece of information of the *Nearness Diagram Navigation* called - *free walking area*. Deeper details are out of the scope of the paper, but it is important to note that the result clears the *Safety Margin* of obstacles, implicitly avoiding the *Imminent collision* situation. Fig. 4b illustrates an example of this situation, in which the robot stops and turns right in place.

(3) **Backward collision danger**: The robot is in this situation when the pre-calculated motion command would produce a collision with the back part of the robot. To detect it, we analytically calculate the trajectory, which would be obtained by the execution of the pre-calculated motion command (following (D.Fox et al., 1997)). Then, the robot motion is simulated over the trajectory to check for collisions. For safety reasons, the back part of the robot is extended with a *Safety Margin* (divided into three parts *Back-Left*, *Back-Right and Back-Center Safety Margins*), see Fig. 4c. Then we distinguish three cases:

- A collision is detected with the *Back-Left Margin*. The angle between the ND solution and the direction that would produce $w = 0$ is discretized in sub-directions. The MG calculates a new motion command by using one of those sub-directions. The new motion command is tested for collisions, repeating the procedure until a collision-free motion command is found. Fig. 4b illustrates an example where the pre-calculated motion command produces a collision with the *Back-Left Margin*. Some intermediate directions are tested until a collision-free one is found.
- A collision is detected with the *Back-Right Margin*. The above procedure is repeated but towards the right-hand direction.
- A collision is detected with *Back-Center Margin*, or simultaneously with *Back-Left and Back-Right Margins*. The motion command is a forward motion without rotation ($v >< 0, w = 0$).

3. EXPERIMENTAL RESULTS

This framework has been tested in four indoor and one outdoor robots. The main characteristics of the robots are: (1) they are differential-driven robots except the outdoor robot that can be set to work in differential-driven mode. (2) The robots are circular, square and rectangular. (3) The available on-board sensors are ultrasounds, 2D and 3D laser rangefinders, and a stereo pair of cameras.



(a)

(b)

(c)

Fig. 5. a) Nomadic Scout robot. b) Simulated experiments with laser. c) Real experiment with ultrasounds.

To adapt the general framework to each robot, the main adjustments were: (1) An special safety area (for each robot geometry) was designed for the *Nearness Diagram Navigation* (see Fig. 3a and Fig. 3b). (2) The *Motion Generator* parameters were tuned to have the desired dynamic behavior compatible with each platform constraints (due to the different motion capabilities of each robot). (3) Only some geometric parameters needed changes in the *Shape Corrector*.

The experiments reported have a common objective: *To show that this framework is able to safely drive a kinematic, dynamic, and geometric constrained mobile platform in very dense, complex and cluttered environments*. This is the type of environment where other approaches are susceptible to failure for the following reasons: (1) the reactive method itself. (2) Approximations in the motion generated. (3) Approximations in the robot shape.

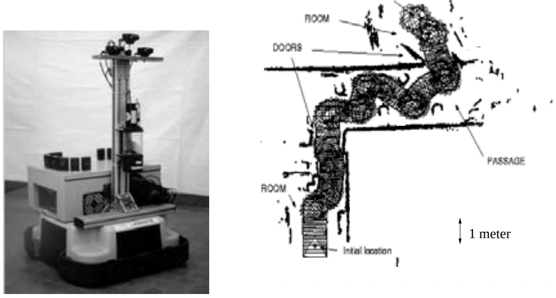
In all the experiments the environment was completely unknown, and only the goal location was given in advance to the robot.

3.1 Circular robots

For circular robots we use the *Nearness Diagram Navigation + Motion Generator* framework.

Scout

This framework has been implemented in an Scout (Nomadic technologies) at the Instituto Superior Técnico de Lisboa, Portugal. The Scout is a circular and differential-driven robot with two active wheels, that moves up to $v_{max} = 1m/sec$. This base has a ring of 16 Polaroid ultrasounds, and it is equipped with a SICK planar laser rangefinder with a field of view of 180°, a range of $32m$, and an accuracy of up to 1cm. See Fig. 5a.



(a) (b)

Fig. 6. a) *Otilio* platform. b) Real Experiment.

Fig. 3a illustrates the robot model and the point used to apply the ND. The sampling time selected was $T = 0.25msec$ and the maximum speed $v = 0.3m/sec$.

We tested the framework in a simulator and in the real robot. In the simulator we used a simulated SICK laser. In the real robot we used the ultrasounds as main sensors. We processed the ultrasound measurements to build depth maps following (J.Borenstein and Y.Koren, 1991*a*). The experiments showed good results in dense, complex and cluttered environments. See Fig. 5b,c.

3.2 Rectangular and square robots

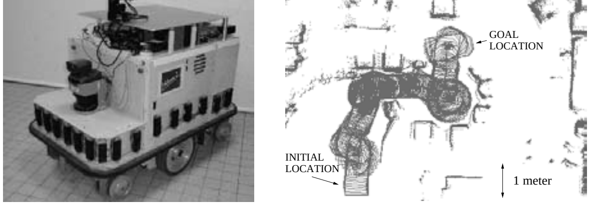
For square and rectangular robots we use the *Nearness Diagram Navigation + Motion Generator + Shape Corrector* framework.

Otilio

This framework has been implemented and tested on the *Otilio* platform at the Universidad de Zaragoza (Spain). *Otilio* is a differential-driven robot, which has two driving wheels and its maximum speed is $1m/sec$. The robot is square ($0.8m \times 0.8m$). It is equipped with a 3D laser rangefinder that scans the environment with a maximum range of $6.5m$ and an accuracy up to $2.5cm$. See Fig. 6a.

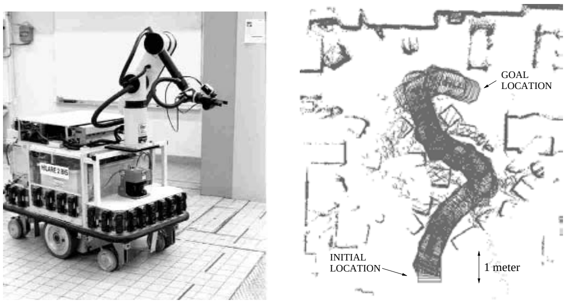
For the experiments, we used the last 20 laser measurements, projected onto the floor and corrected to the actual robot location (short-time memory). The sampling period was set to $T = 0.4sec$ and the maximum speed to $0.3m/sec$.

Fig. 6b. shows an experiment in a typical indoor environment. The robot passed through two doors (the last one was half-open) and a corridor full of obstacles. The goal location was successfully reached without any collisions with the environment.



(a) (b)

Fig. 7. a) *Hilare2* platform. b) Real Experiment.



(a) (b)

Fig. 8. a) *Hilare2Bis*. b) Real Experiment.

Hilare2 and Hilare2Bis

This framework has been implemented and tested on the *Hilare2* and *Hilare2Bis* platforms at LAAS (CNRS), France. Both platforms have two driving wheels and their maximum speed is $1.5m/sec$. The robots are rectangular ($1.3m \times 0.8m$) and they are equipped with a SICK laser rangefinder. The main difference between both robots is that *Hilare2Bis* is equipped with a GT6A arm with 6 degrees of freedom. The arm weighs about 60kg and it is located in the forward part of the robot. This produces a high inertia when the robot turns. See Fig. 7a and Fig. 8a.

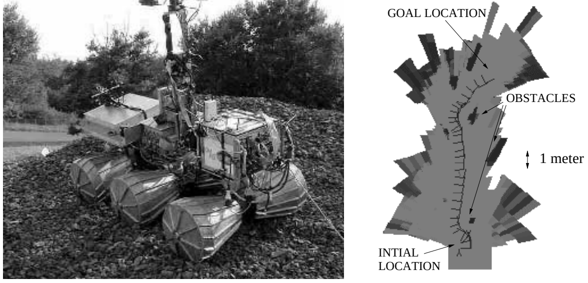
For the experiments, we used the last 40 laser measurements corrected with the robot odometry (short-time memory). The sampling period was set to $T = 0.4sec$ and the maximum speed to $0.3m/sec$. To take the inertia into account in *Hilare2Bis*, we have tuned some parameters of the *Motion Generator* to have an adequate dynamic response.

Fig. 7b shows an experiment with *Hilare2* in a very dense, complex and cluttered environment. The robot passed through a very narrow corridor ($< 10cm$ at sides) and it maneuvered reactively in a constrained space (central part of the experiment) to find the exit. The goal location was successfully reached without collisions with the environment.

Fig. 8b shows an experiment with *Hilare2Bis* in a very complex and dense environment built while the robot was moving (this creates the environment's non-predictability component). The robot passed through the asymmetric corridor and in the last part of the experiment it maneuvered to the right to reach the goal location. The robot successfully arrived at the goal location without collisions with the environment.

Lama

This framework has been implemented and tested on the outdoor *Lama* platform at LAAS (CNRS), France. *Lama* has 6 driving wheels that can be programmed to work in differential-drive mode. The robot is rectangular ($1.85m \times 1.2m$) and its maximum speed is



(a) (b)

Fig. 9. a) *Lama* platform. b) Real Experiment.

$0.17 m/sec$. For the collision avoidance task we used a pair of B/W cameras. See Fig. 9.

The obstacle information is obtained with a particular probabilistic obstacle detection procedure (H.Haddad et al., 1998). The perceived area is described by a set of polygonal cells. By means of a Bayesian classifier, the cells are labeled with a probability that an obstacle will occupy them. The image-processing period is about four seconds. The sampling time was set to $T = 0.4 sec$ and the maximum speed $10 cm/sec$.

The experiment was conducted in a typical outdoor environment (see Fig. 9b). The framework drove the robot towards the goal location avoiding collisions with the obstacles. The complete run was about $20 m$.

4. CONCLUSIONS

We have presented a framework to generalize a reactive navigation method - *Nearness Diagram Navigation* - to work on robots with kinematic, dynamic and geometric constraints. This framework generates reactive motion commands to drive a mobile platform towards a goal location whilst avoiding collisions with the environment.

We have shown experiments in very dense, complex and cluttered environments, where: (1) it is necessary to use a reactive method able to deal with these type of environments. (2) The robot's kinematic and dynamic constraints limit the motion capabilities, and this has to be taken into account. (3) The geometric constraint of the robot, that is the robot shape, also reduces the possible collision-free motions and this has to be taken into account.

To validate the proposed approach, and demonstrate the easy portability among different platforms, the framework has been tested in five different robots at three different laboratories. We also tested the sensor influence by using ultrasounds, 2D lasers, 3D lasers and vision. The results were very satisfactory, and we were able to safely move the mobile platforms during hours of tests.

From our point of view the main limitation of the framework is the division of the navigation problem into subproblems. This is an underconstrainted solution since the kinematic, dynamic, and geometric reactive method stage. The reactive collision avoidance approach calculates solutions that separately are converted into feasible motion commands according with the platform constraints. The results are remarkable with this framework because the typical mobile platforms have quick dynamic and the kinematic is not very significant with the small sample periods used. On the other hand the result would be degraded.

5. ACKNOWLEDGEMENT

We thank R. Alami and T. Simeon of LAAS/CNRS (France) for accepting J.Mínguez into their working group. Moreover, we thank S. Fleury of LAAS/CNRS (France), for her help with algorithm implementation on the *Hilare2*, *Hilare2Bis* and *Lama* platforms. We thank J.Santos-Victor of ISR/IST of Lisbone (Portugal) for accepting J.Mínguez into his working group. Moreover, we thank D.Ortín of University of Zaragoza (Spain) and C.Marques of ISR/IST of Lisbone (Portugal) for her help with the algorithm implementation on the *Scout* platform.

REFERENCES

D.Fox, W.Burgard and S.Thrun (1997), 'The dynamic window approach to collision avoidance', *IEEE Transactions on Robotics and Automation* **4:1**.

H.Haddad, M.Khatib, S.Lacroix and R.Chatila (1998), 'Reactive navigation in outdoor environments using potential fields', *IEEE/RSJ International Conference on Robotics and Automation (ICRA 1998)* .

I.Ulrich and J.Borenstein (1998), 'Vfh+: Reliable obstacle avoidance for fast mobile robots', *IEEE International Conference on Robotics and Automation* p. 1572.

J.Borenstein and Y.Koren (1991*a*), 'Histogramic in-motion mapping for mobile robot obstacle avoidance', *IEEE Journal of Robotics and Automation* **7(4)**, 535–539.

J.Borenstein and Y.Koren (1991*b*), 'The vector field histogram (vfh)-fast obstacle avoidance for mobile robots', *IEEE Transactions on Robotics and Automation* **7(3)**.

J.C.Latombe (1991), *Robot Motion Planning*, Kluwer Academic Plubishers.

J.Mínguez and L.Montano (2000), 'Nearness diagram navigation (nd): A new real time collision avoidance approach', *IEEE/RSJ International Conference on Intelligent Robots and Systems* .

J.R.Asensio and L.Montano (2002), 'A kinematic and dynamic model-based motion controller for mobile robots', *International Federation of Automatic Control, IFAC 2002* .

O.Khatib (1986), 'Real-time obstacle avoidance for manipulators and mobile robots', *The International Journal of Robotics Research* **5(1)**.

R.Arkin (1998), *Behavior-based robotics*, MIT Press.

S.Quinlan and O.Khatib (1993), 'Elastic bands: Connecting path planning and control', *IEEE International Conference on Robotics and Automation* **2**, 802–807.