DEVELOPMENT OF A HYBRID ARCHITECTURE FOR THE AUTONOMOUS CONTROL OF MOBILE ROBOTS

Hugo Silva, hugodaluz@unb.br
Alberto Álvares, alvares@alvarestech.com
GIAI (Group of Innovation in Industrial Automation), Department of Mechanical Engineering and Mechatronics, University of Brasilia - CEP 70910-900, Brasília, DF, Brasil.

Abstract. *One of the most significant challenges in mobile robotics are found in the autonomous navigation area, where the robot must be able to feel and act in an uncertain and geometrically restricted environment, without suffering any type of external interference. The objective of this work is the development of a hybrid architecture (coordinating techniques of planning and reaction) for the autonomous navigation, allowing a mobile robot to navigate in a known and structured environment. This architecture is being developed in a modular and incremental way allowing the incorporation of several techniques of mapping, localization and path-planning independently of the reactive strategy used by the robot. The proposed architecture aims to integrate a module of path-planning using Voronoi diagram and a module of reactive behavior using fuzzy controllers for the control of direction and speed through a module of coordination. The path-planning shall be implemented using a road map created a priori and the Dijkstra's Algorithm will be used to search the best path. After the generation of the path to be followed, the module of coordination will be responsible for sending actions to control the robot using the sensory information to determine if the robot will follow the path planned or will execute reactive actions. In addition, the module of coordination monitors the execution of the defined path, using a fuzzy controller, deciding if the robot achieved the goal or whether it is necessary to re-planning the path. The proposed architecture is being validated in a robot Nomad XR4000. The first results of this work are: the graphic interface developed to integrate the modules and a prototype of the planning module.*

Keywords: *Fuzzy Logic, Hybrid Architecture, Mobile Robotic, Path-Planning, Voronoi Diagram.*

1. INTRODUCTION

The mobile robots are used as tools in several fields of application such as medical services, tour guides, space exploration or robotic soccer. Such versatility has contributed to the development of a mathematics and scientific basis and through the research theoretical and fundamental questions. An autonomous robot can be defined in an abstract way as the mapping of a sequence of sensorial input for an appropriate action in response to these perceptions (Duffy 2000). According to Siegwart and Nourbakhsh (2004), the robot's performance in the physical world is subject to the geometric restrictions and uncertainties, thus, it is necessary the combination of researches in perception, localization, mapping and path planning for the autonomous navigation has success.

This work is part of a research in process of the Group of Innovation and Industrial Automation (GIAI) of the University of Brasilia and it has the objective of create a hybrid architecture for the autonomous control of mobile robots. It is important to perceive that this research is addressed to physical robots located in a real world that influences the behavior of the system directly. The expected result to the end of the research is an architecture to allow the robot behaves autonomous, coordinating path planning with avoiding collisions behavior to unexpected obstacles. In this architecture the planning and the reaction are complementary, because each of the tasks is essential for the other one's success.

Although several architectures exist proposed in publications (Fernandez-Leon et al., 2008; Grassi, 2006; Barberá, 2001; Arkin, 1998; Brooks, 1986), most of them consider negligible the inherent uncertainties of the navigation task in real environments or they are limited to applications in simulated environments with ideal conditions or in a behavioral-based approach in which the robot is armed with a simple set of behaviors (wall following, obstacle avoidance). The experimental results indicate the utility of these approaches as well as their counterpart algorithms, which provide a good base for complete planners. However, those approaches not always guarantee proofs of correctness that a path can be found.

The article is organized in the following way, the section 2 presents the types of architectures for robot control, the section 3 presents the proposed architecture, in the section 4 it is introduced the mobile robot used in the validation, Nomad XR4000, the section 5 brings the current state of the development of the architecture and finally, the section 6 brings the conclusions of the work in process.

2. ARCHITECTURES FOR ROBOT CONTROL

An architecture does that the perceptions of the available sensors are supplied to a program, generating actions to be executed by the actuators. Many researchers have been presenting proposal of architectures along the time, each one adapted to the tasks to be executed in the research and to the characteristics of the used robot. The architectures may be classified with respect to the level of deliberation and reactivity in the system (Arkin, 1998):

- Reactive: The reactive robots (also known as behavior-based) actuate through the activation of a collection of primitive behaviors of low level (Arkin, 1998), providing faster and flexible answers and the environment's representation is unnecessary.
- Deliberative: The deliberative paradigm, according to Duffy (2000), derives from the classic artificial intelligence, which tries to represent the planning, learning and decision making of human beings in a robot, but it is naturally slow and limited. Such systems are naturally slow, serial, based on knowledge and with limited resources.
- Hybrid: The hybrid paradigm connects deliberation and reaction reducing the response time of the robot to environmental changes and performing plans with a high computational cost. This paradigm is considered ideal for robots that need to travel in known environment.

The development of an architecture that allows the autonomous navigation arises as a unique combination of issues such as: perception, localization, mapping and path-planning.

2.1 Perception

The perception is the process for which the robots map the sensor's measurements in internal representations of the environment, but it is a difficult task because in general the sensors are noisy and the environment is partially observable, unpredictable and usually dynamic. The internal representations should have three properties mentioned by Russel and Norvig (2003): they should contain enough information for the robot to make the correct decisions, they should be well structured to be updated of efficient form, and they should be natural and have internal variables whose values correspond to the natural state of their counterparts in the physical world.

2.2 Localization

The localization is one of the major problems in robotics, because it is the main point in any success physical interaction. Robots must know where they are in order to find their way to goal locations. In this task, the use of odometer's data has the main disadvantage that error grows indefinitely, unless an independent reference is used to reduce it (Feng et al., 1996).

2.3 Mapping

Mapping is a technique in which the robot uses its sensors to create a map of its local environment (Feng et al., 1996). The maps generated by automated methods have advantages as the environment's characteristics to be recognized by sensors and the level of detailing offered (Choset et al., 2005) and the construction can be made before or during the navigation (Elfes, 1987).

2.4 Path-planning

According Choset et al. (2005), the problem of the path-planning can be described as: given a robot with an initial configuration, a goal configuration and your shape, find a path free of collisions from the initial configuration to the goal take into account a number of obstacles located in space. Methods of motion planning can be used to determine which motions the robot should accomplish from way to reach a position or configuration wanted in the environment free from collisions with obstacles, taking in account aspects of their dynamics and their motion restrictions (Grassi, 2006).

There are several methods to deal with the problem of path-planning, a systematic discussion about these methods can be found at (Choset et al., 2005; Siegwart and Nourbakhsh, 2004; Russel and Norvig, 2003; Latombe, 1991). In this work a method of roadmap called Voronoi diagram (figure 1) will be used. This is a complete method of roadmaps which tends to maximize the distance between the robot and the obstacles on map. Algorithms which find paths in Voronoi's roadmaps are complete because the existence of a path in free space implies the existence of the same path in the Voronoi diagram. The Fortune's algorithm, commonly known as the plane sweep algorithm, was used to create the Voronoi diagram. This algorithm sweeps a horizontal line (called the sweep line) from top to bottom over the plane. While the sweep line moves downwards the information does not change, except at the event points (Berg et al, 2008).

The advantage of this technique is that a robot with sensors of ultra-sound may follow the Voronoi's extremity using the same rules as the creation of diagram (Choset *et al.*, 2005). Thus, the movement based in Voronoi can minimize the sensor's inaccuracy. The disadvantage is the production of long paths when the configuration space is wide open (Russel and Norvig, 2003).

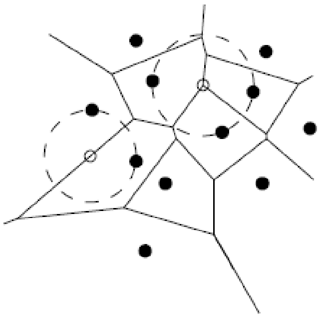

Figure 1: Voronoi Diagram of a set of points. The vertex is a point next to three or more sites. And the edge is a point next to two sites (Berg et al, 2008).

3. PROPOSED ARCHITECTURE

The proposed architecture in this work aims to supply autonomous behavior in known environment considering the uncertainties of the robot's sensors and mainly the possibility of existence of mobile or stationary obstacles which are not expected in the navigation plan. The architecture will allow that the robot:

- Formulate, execute and modify the navigation plans in agreement with the world's state.
- Perceive and react to the changes in the environment, navigating without collisions with mobile or stationary obstacles.

The proposed modules that compose the architecture are: Sensing, Mapping, Localization, Planning, Coordination and Action. These modules will be executed in an independent way, but they will share information during the autonomous navigation. The interaction among the modules can be seen in figure 2 and it will be described during the next sections.

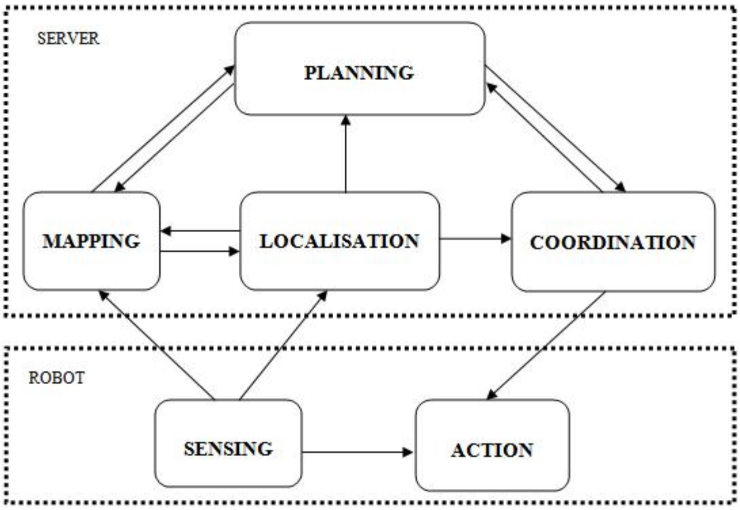

Figure 2: The interaction among the modules of the architecture, where the arrows represent flow of information.

3.1 Sensing

The module of Sensing is responsible for acquisition and storage of data from the robot's sensors. Besides storing the sensor's data, this module should treat the uncertainty in the environment readings, as well as in the robot's positioning reading by applying filtering techniques to discard invalid data. This information will be used by the modules of Localization, Mapping and Action.

3.2 Localization

The Localization module is responsible for defining the robot's position in the environment as accurate as possible. The proposed architecture provides routines for corrections in the positioning through the combination of information of the Mapping, Sensing and Location modules, even though previous works consider negligible this kind of error when the robot travels small distances.

3.3 Mapping

The module of Mapping is the responsible for creating, storing, updating and supplying access to the maps and to the global and local environment's information. This module will allow the construction of maps identifying the characteristics and the obstacle's position in the environment (with an associated uncertainty).

This module will have two behaviors, in an off-line way it will allow the construction of the environment's maps through the sensor of distance and during the navigation it will create the local maps and send to the module of Localization which will identify the areas of similarity between the local and the global maps and will calculate the robot's current position. The figure 3 describes with details the involved modules and the dynamic of the position's calculation through the sensorial information and the correspondence among the local and global maps.

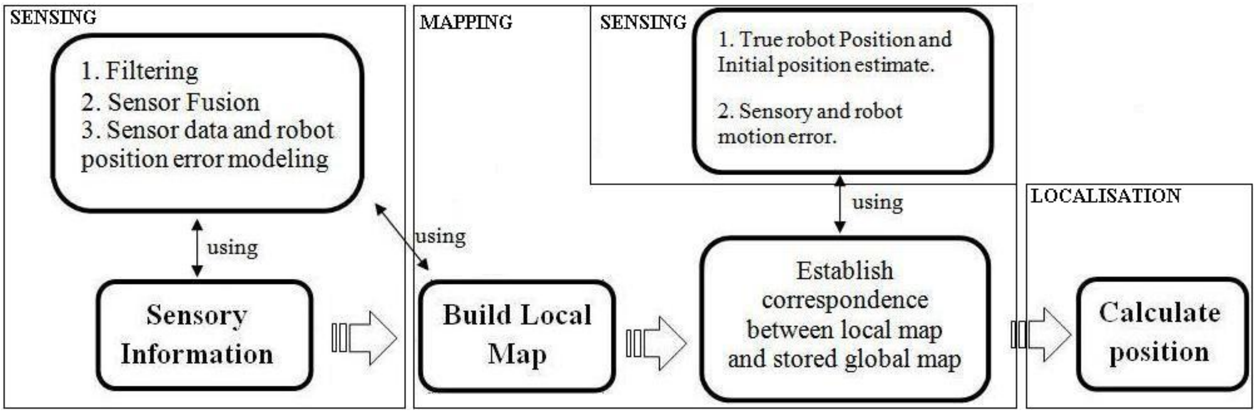

Figure 3: Interaction between the modules for determination of the robot's position. Adapted from Feng et al., 1996.

3.4 Planning

In the proposed architecture, the module of Planning is the responsible for generating the path for the reach of the goal using the environment's representation and the robot's location, supplied by the modules of Mapping and Location respectively. The Voronoi diagram will be calculated before the navigation using a global environment's map provided a priori by the module of Mapping. After the creation, it happens the "pruning" to eliminate the invalid roads (where is not possible the robot's motion). For to accomplish the search in the Voronoi graph, it was chosen the Dijkstra's algorithm, used to compute the shortest path between the start and the goal, and that requires which each edge in graph is associated to a weight. In this way, the original path-planning problem is reduced to finding a path on the Voronoi diagram, which is search problem in a discreet graph (Choset *et al.*, 2005).

3.5 Coordination

The Coordination module is responsible to coordinate the interaction between the planning and reactivity, to manage the execution of the path planned and to monitor the robot performance during this task. This module should generate a set of actions using the planned path, send them to the module of Action in the robot and monitor its execution in real time, requesting a new path planning whenever necessary.

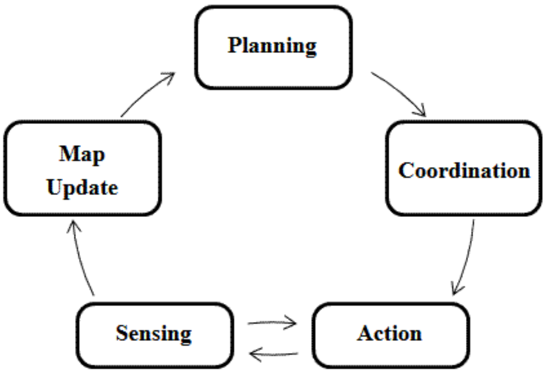

Figure 4: The cycle which involves sensing, map update, planning, coordination and action should be repeated during the autonomous navigation.

If an error occurs because a new mobile or stationary obstacle appeared, this module should interrupt the execution of the current path and if it is identified the need of a re-planning, the detected obstacle should be included in the map by the Mapping module and a new path should be calculated. If the geometry of the obstacle cannot identified completely by the robot, the map will be updated partially and the planned path can be modified when the robot possesses more details about the shape of the obstacle. The figure 4 shows the planning cycle that can happen during the navigation.

To accomplish the monitoring of the path followed by the robot, it will be created a fuzzy controller which determines if the robot is following the path correctly or if exist the need of creation of a new path between the current position and the goal position. This fuzzy controller will use the robot’s position information, the defined path by the planning module and a set of fuzzy rules.

3.6 Action

The Action module is responsible for the generation of the robot’s motion using the path planned and reactive behaviors according to the sensor’s perceptions. This module consists of fuzzy controllers responsible for adjustments in the direction of the robot, speed control and a set of classes that control the robot's motion. In this way, the robot can avoid collisions during the navigation while new planning action is not executed.

The fuzzy controller for navigation must control the robot's obstacle avoidance in a reactive way. A set of rules will be created to relate the sensorial information (infrared and ultrasound) and signs of acting to the robot’s current direction. The proposed rules will be created with the objective of maintaining the robot in the direction of the goal and at the same time to avoid obstacles, so that there is not great path deviation and that the robot can identify, even partially, the obstacle that is impeding the navigation. The tasks of obstacle avoidance and navigation to goal can be competitive and the rules should prioritize the robot’s safety.

The fuzzy controller for adjustment of speed will allow the control of the robot's speed allowing safe adjustments in the direction using the sensorial information, accelerating when away of obstacles and slowing down when close.

4. NOMAD XR4000

The Nomad XR4000 is a complex mobile system that incorporates state of the art drive, control, networking, power management, sensing, communication and software technologies. Nowadays, the XR4000 (figure 5) offers capabilities for research in mobile manipulation, machine learning, and sensor based navigation (Nomadic, 1999a). Utilizing distributed control techniques and a robust modular mechanical design, the XR4000 is a holonomic drive system that offers three full degrees of freedom (x, y, and theta). It has caster wheels that have independently powered steering and translation axes. It uses four caster wheels, resulting in an eight-axis, underconstrained system (Nomadic, 1999a).

Figure 5: The robot Nomad XR4000 (Nomadic, 1999a).

The Nomad XR4000 has three standard sensor systems: tactile, ultrasonic, and infrared, providing information about distance of objects up to 7 meters. The robot has control software supplied by the manufacturer, called XRDev. This software is composed of three main processes: Nrobot, responsible for the communication with the robot’s hardware; Ngui, which provides a graphic interface of control and the user’s processes, responsible for the communication with Nrobot for execution of defined tasks for the user (Nomadic, 1999b).

5. IMPLEMENTATION AND RESULTS

The development of the proposed architecture started from a graphic interface for the integration of the modules for communication with the robot. This interface allows the choice of the robot's shape for its representation and the map of the environment that will be used. In the figure 6 it is possible to visualize the prototype of the graphic interface.

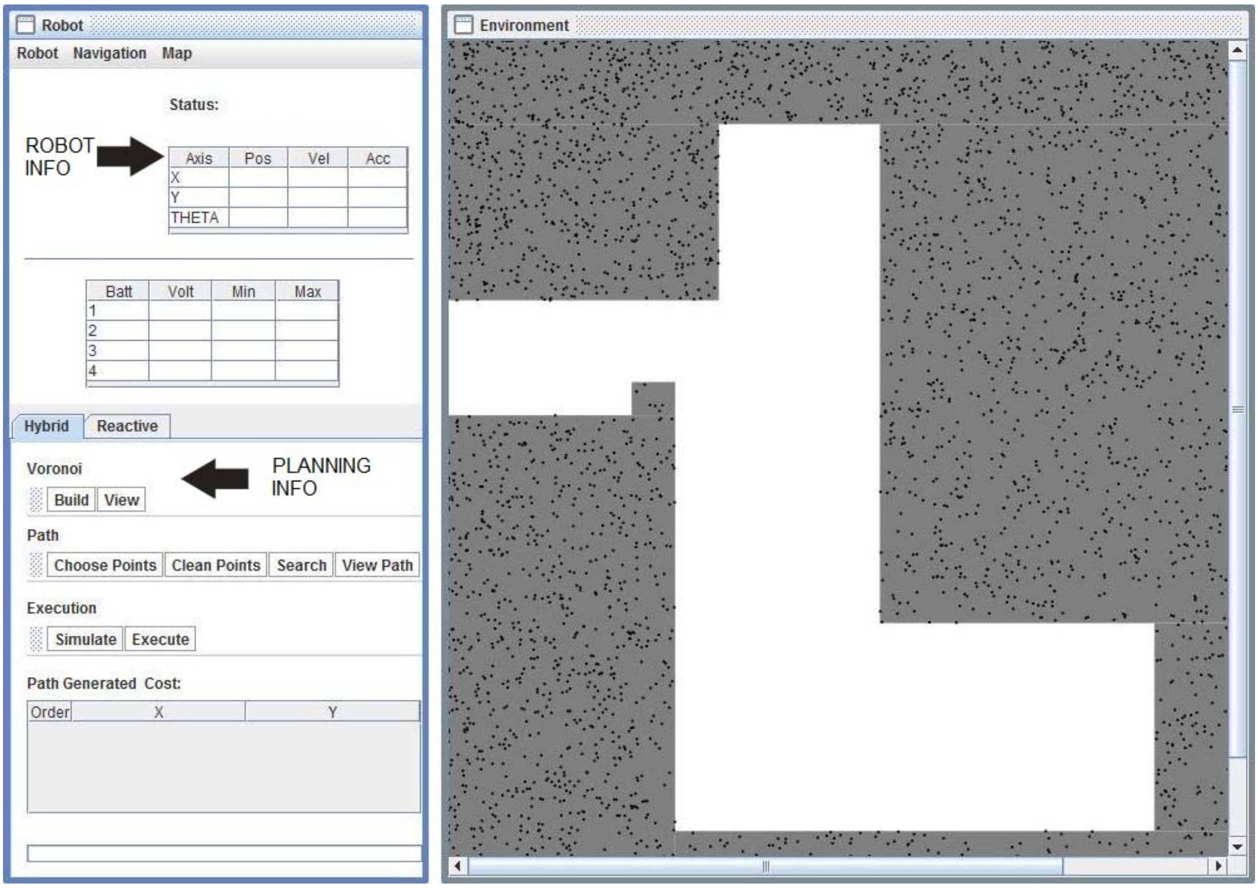

Figure 6: Graphic Interface of the prototype developed.

Using a single map was possible to generate the Voronoi Diagram. The result is a roadmap that allows the robot's safe movement. Starting from the definition of the points of start and goal, the Dijkstra's algorithm was used to search for the best path inside of the Voronoi graph, resulting in the robot's path. Finally, the possible collisions are counted and informed to the user before the execution of the planned path.

The figure 7 display the sequence of execution of the planning:

- The Voronoi Diagram created.
- The start and goal points' chosen.
- The chosen path by Dijkstra's algorithm.

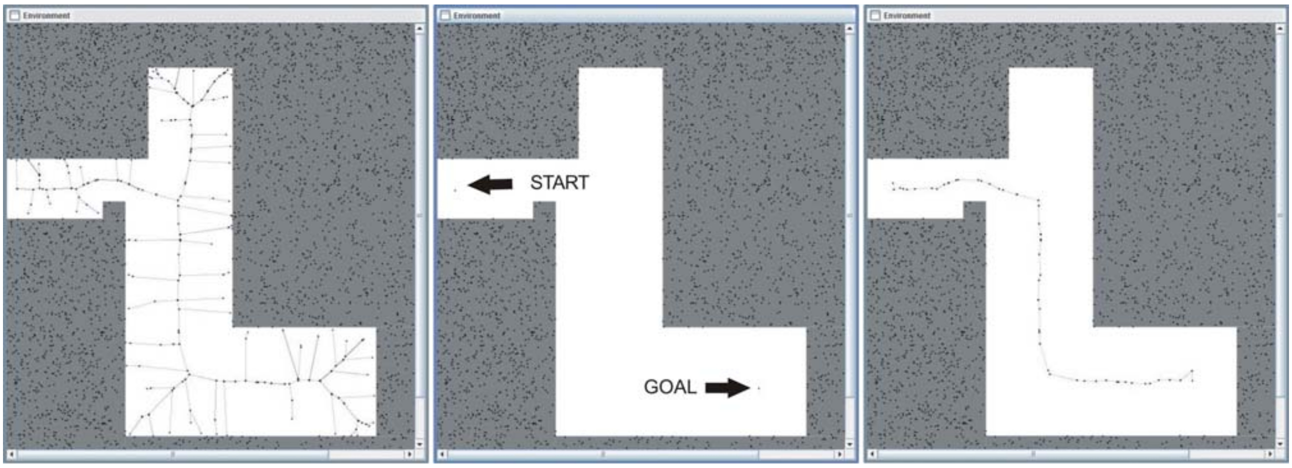

Figure 7: Sequence of actions: The Voronoi Diagram created, the start and goal points' and the best path.

The figure 8 shows the simulation of execution of the path. This functionality allows the analysis of the viability of the planned path. The viability is defined in agreement with the quantity and the severity of the collisions. The severity is high when the robot suffers several collisions in a same point of the path. In the simulation, the Nomad robot is represented by a white sphere and the green area represents the uncertainty region of the position. This region changes for the red color when a collision occurs. The figure 8 presents two different moments in the simulation of the execution of a same path: when a collision occurs and when the robot reaches the goal.

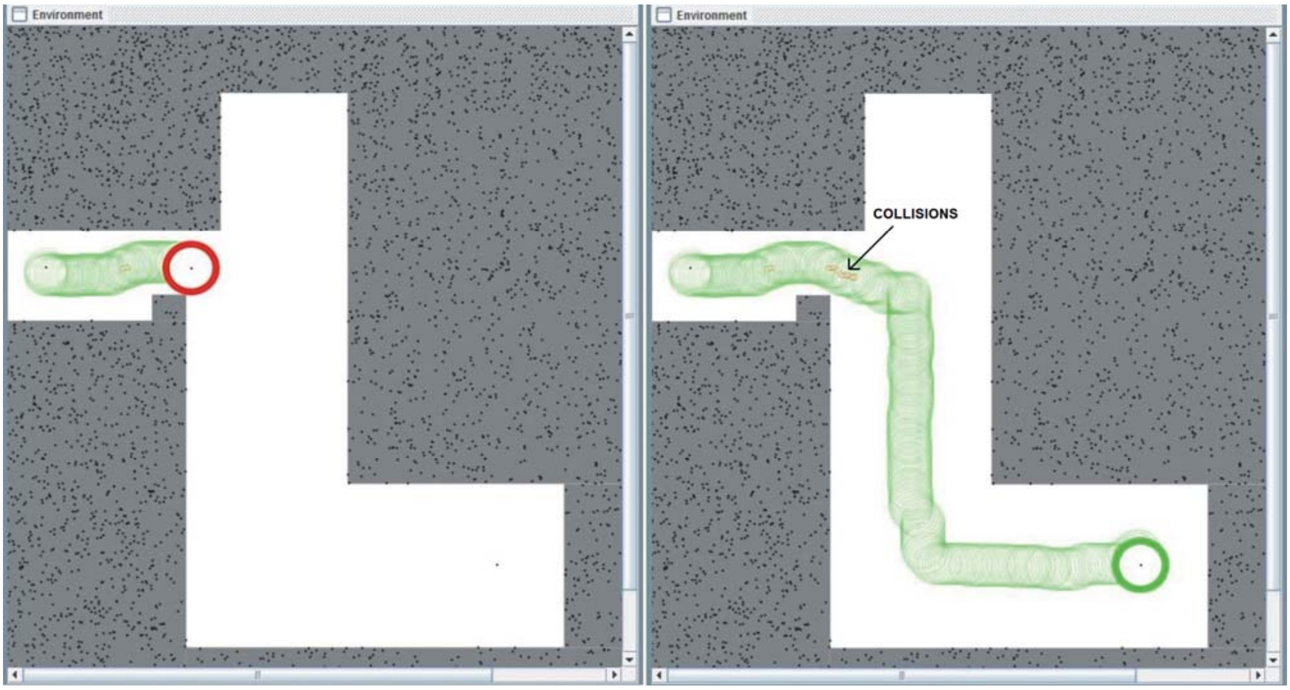

Figure 8: Uncertainty representation changes to red when occurs a collision.

In the figure 9, two Voronoi diagrams created for different environments, the first without obstacles and the second with several obstacles distributed in the free space. The comparison among the two images indicates that the configuration of the diagram just differs in the points where new obstacles exist.

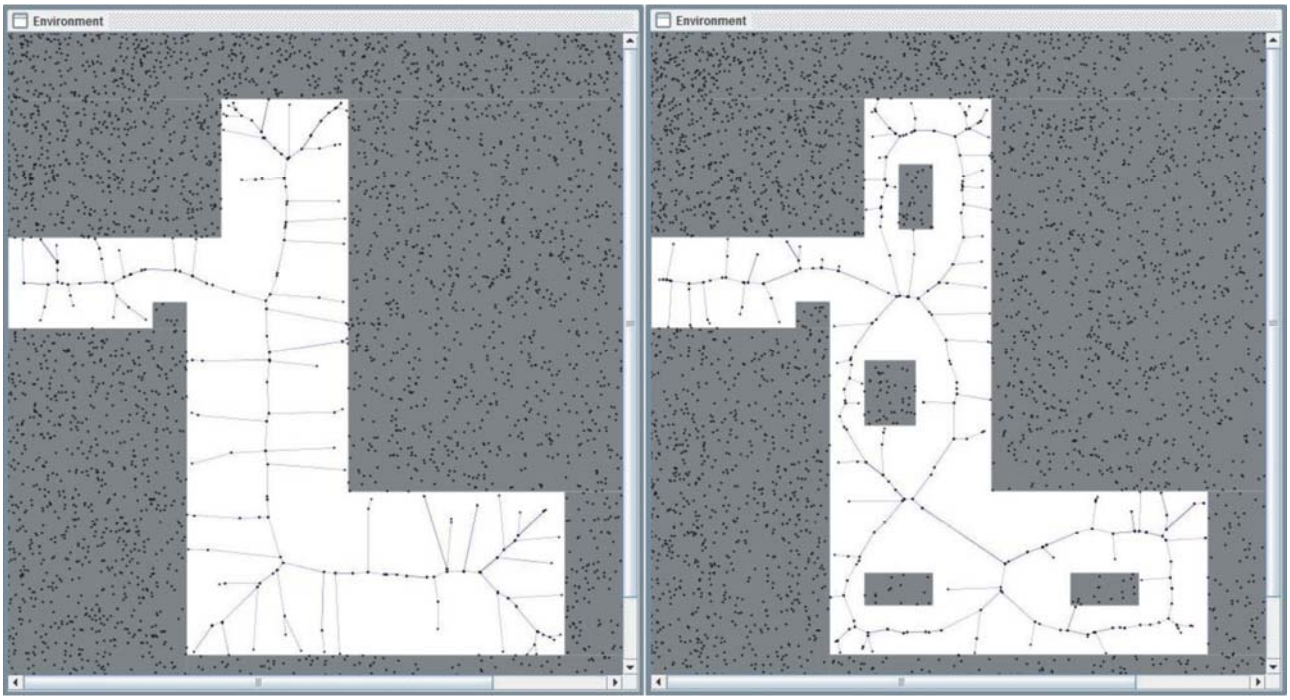

Figure 9: Two Voronoi diagrams created for different environments.

The figure 10 presents the followed paths by the robot leaving of the same origin to reach the same goal in the two environments presented in the figure 9. The followed path is the most distant possible of the obstacles in the map in both cases.

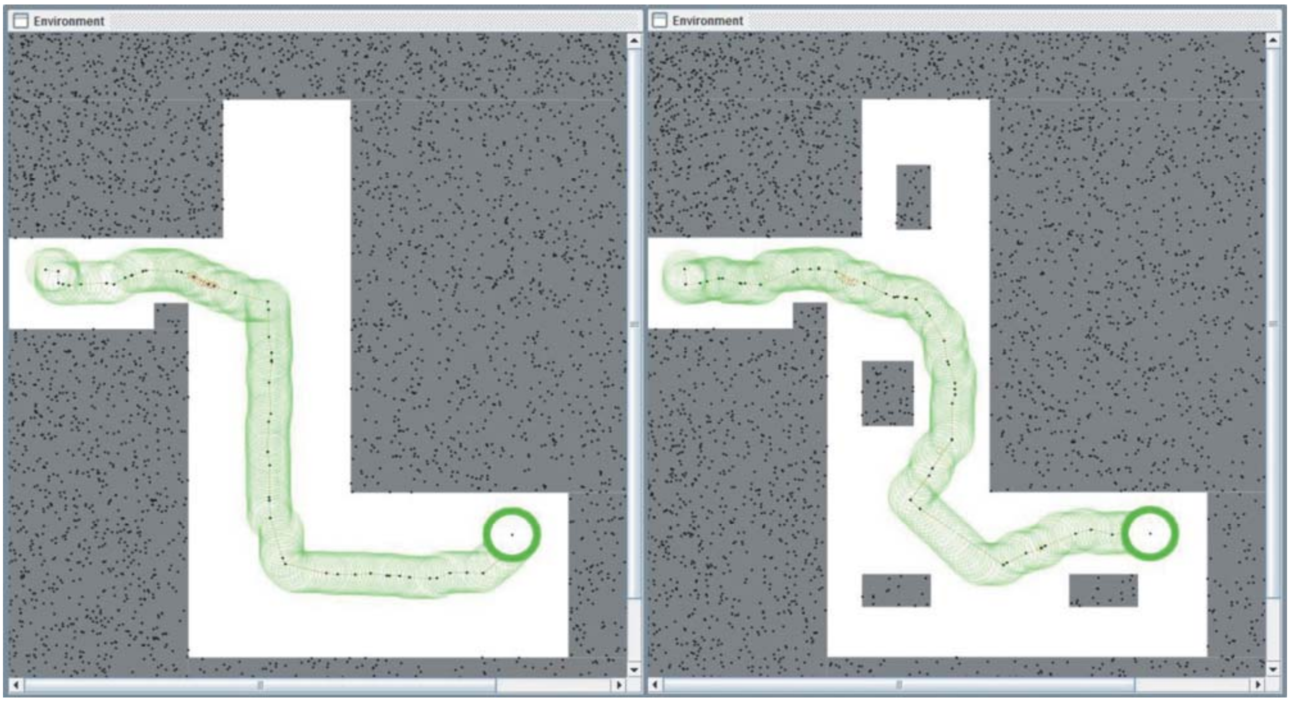

Figure 10: The followed path in the two environments presented in the figure 9.

6. CONCLUSIONS

The research around the autonomous navigation of mobile robots has been presented, as well as several works with a great variety of different approaches. The proposal of this work involves the development of a hybrid architecture coordinating planning and reactivity. This architecture should work in unexpected situations, adapting the robot's path in real time, avoiding collisions with mobile or stationary obstacles, sensing and reacting to the environmental changes.

In spite of being a work still in course is already possible to evaluate the viability of the architecture proposed with developed prototype. The planning module is already ready to generate a navigation plan for the robot using a created map. Even though the used algorithm still not to consider the uncertainty associated to the obstacles in the map, in subsequent implementations this characteristic will be incorporate to increase its robustness. In future works, the development and integration of the other modules of the proposed architecture will be implemented, ensuring, the complete autonomous navigation.

7. REFERENCES

Arkin, R. C., 1998, “Reactive robotic systems”, In The Handbook of Brain theory and Neural Networks. Ed. MIT Press, Cambridge, MA, pages 793-796.

Barberá, H. M., 2001, “A Distributed Architecture for Intelligent Control in Autonomous Mobile Robots - An Applied Approach to the Development of the Quaky-Ant Platform”, PhD thesis, Dept. of Communications and Information Engineering, University of Murcia.

Berg, M.; Cheong, O.; van Kreveld, M. and Overmars, M., 2008, Computational Geometry: Algorithms and Applications, Springer, Berlin.

Brooks, R., 1986, “A robust layered control system for a mobile robot”, Robotics and Automation, IEEE Journal of 2(1), 14-23.

Choset, H.; Lynch, K. M.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. E. & Thrun, S., 2005, “Principles of Robot Motion: Theory, Algorithms, and Implementations” (Intelligent Robotics and Autonomous Agents), The MIT Press.

Duffy, B., 2000, “The Social Robot, Ph.D Thesis”, PhD thesis, Department of Computer Science. University College Dublin, Ireland.

Elfes, A., 1990, “Sonar-based real-world mapping and navigation”, 233--249.

Feng, L., Borenstein, J., Everett, H. R., Lee, W. and Byrn, R. H., 1996, “Where am I? Sensors and Methods for Mobile Robot Positioning”. University of Michigan, Oak Ridge Lab.
Fernandez-Leon, J. A.; Acosta, G. G. and Mayosky, M. A., 2008, “Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation”, Robotic Autonomous Systems 57(4), 411--419.
Grassi, V., 2006, “Hybrid Architecture for Mobile Robots Based on Navigational functions with human interaction”, PhD thesis, Polytechnic School, USP.
Latombe, J.-C., 1990, “Robot Motion Planning” (The Springer International Series in Engineering and Computer Science), Springer.
Nomadic, 1999a, “Nomad XR4000 Hardware Manual Release 1.0”, Nomadic Technologies, California, USA.
Nomadic, 1999b, “Nomad XRDEV Software Manual Release 1.0”, Nomadic Technologies, California. USA.
Russell, S. J. & Norvig, P., 2003, “Artificial Intelligence: A Modern Approach”, Pearson Education.
Siegwart, R. & Nourbakhsh, I. R., 2004, “Introduction to Autonomous Mobile Robots”, Bradford Book.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.