**Sciences & Technologie B – N°27**, Juin (2008), 71-80.

# MODULAR NEURAL NETWORKS ARCHITECTURE FOR NAVIGATING MOBILE ROBOT IN CHANGING ENVIRONMENTS

*Reçu le 29/10/2006 – Accepté le 18/06/2008*

**Résumé**

Cet article traite le problème de la navigation des robots mobiles dans des environnements d'intérieur inconnus. La plupart des travaux se basant sur les réseaux de neurones (NN) pour traiter ce problème utilise un seul réseau qui reçoit et analyse toute l'information disponible, ce qui engendre en général plusieurs inconvénients : temps d'apprentissage relativement long, exemples d'apprentissage souvent contradictoires, minimums locaux, et une pauvre capacité de généralisation finale. Le travail présenté dans cet article évite ces problèmes par l'utilisation d'une architecture de contrôle modulaire (stratégie « diviser pour régner ») combinant un module pour la classification d'environnements avec plusieurs comportements pour la navigation réactive. Les comportements sont appris par les réseaux de neurones modulaires (MNN). La coordination entre les divers comportements se fait à la fois d'une manière coopérative et concurrentielle.

Pour vérifier la validité de notre approche, une interface graphique est mise au point. Elle nous a permis de tester l'architecture proposée dans plusieurs situations différentes, et qui se rapprochent de la réalité. Dans tous les cas les résultats obtenus sont très encourageants, et illustrent l'efficacité de cette architecture.

***Mots clés*** *: Robots mobiles intelligents, réseaux de neurones modulaires (MNN), apprentissage, navigation réactive, classification d'environnements.*

**Abstract**

This paper addresses the navigation problem of a mobile robot in unknown indoor environments. Most neural network (NN) approaches to this problem focus on a monolithic system, i.e., a system with only one neural network that receives and analyses all available information, resulting in conflicting training patterns, long training times and poor generalization. The work presented in this article circumvents these problems by the use of modular architecture ("divide and conquer" strategy) combining behavior based environment classification and several behaviors based reactive navigation. The behaviors are learned by modular neural networks (MNN), coordination between these various behaviors is done at the same time in a cooperative and competitive way.

To check the validity of our approach, a graphic interface is developed. It enabled us to test the proposed architecture in several different situations which approach reality. In all the cases, the results obtained are very encouraging, and illustrate the effectiveness of this architecture.

***Keywords*** *: Intelligent Mobile robots, Modular Neurons Networks, Learning, Reactive navigation, environment classification.*

**[1]F. HENDEL,**
**[1]N. BERRACHED**
**[2]G. BOUVIER**
[1]Research Laboratory in Intelligent Systems
Electronic Department
University of Sciences and Technology of Oran, Algeria.
[2]Laboratoire Images et Signaux de Grenoble (LIS) UMR 8053
France

**ملخص**

نتطرق في هذه المقالة إلى إشكالية في مجال الآلي المتحرك في وسط داخلي متغير.الحلول المقترحة في هذا المجال من طرف الباحثين و المعتمدة على الشبكات العصبية تستعمل غالبا شبكة واحدة، مما يؤدي إلى عدة نقائص منها: زمن التدريب، أمثلة التدريب متناقضة و خاصية التعميم فقيرة.

الحل الذي نقترحه يتجنب كل هذه المشاكل باستعمال هندسة التحكم للآلي المتحرك متجزئة مكونة من عدة شبكات العصبية "إستراتيجية فرق تسود" هذه الشبكات تمكن الآلي أولا من اكتشاف و تخصيص وسطه القريب ؛ ثم من التحرك في اتجاه هدفه مع تجنب كل العراقيل الممكن إيجادها في طريقه. جربت الهندسة المقترحة على الكمبيوتر في عدة أمثلة متنوعة و متغيرة من الأوساط؛ النتائج المتحصل عليها في كل الحالات مشجعة وتؤكد على فعالية هذا الحل.

***الكلمات المفتاحية:*** آلي متحرك ذكي، شبكات العصبية، التدريب، تخصيص الوسط

There has been an increasing interest in the development of ***intelligent*** mobile robots, i.e., robots that are able to ***learn*** to navigate and act in complex, possibly unknown, environments [1, 16]. This interest grew with the realization that a mobile robot's application environments are usually dynamic, leading to the search for new solutions for mobile robot navigation.

There are several approaches to the navigation problem [7, 13], and they can be broadly divided into two groups:

Traditional methods referred to as ***Model based approaches*** or ***global navigation***, mainly used when the robot environment is completely known.

***Sensor based approaches or reactive navigation*** [6], generally used in unknown environment, and are completely based on sensory information to determine the robot's path online.

Significant results on the navigation problems of a robot have been obtained in the past decades. However, the problem of reactive navigation in uncertain dynamic environments has not been fully investigated.

Until now, progresses have been made in applying intelligent control and machine learning methods to reactive navigation systems of mobile robots. As an important class of machine learning, artificial neural networks have attracted interests in the literature [3, 4, 5, 10, 13, 14].

Indeed, in recent years, neural networks (NN), with their strong learning capability, have proven to be a suitable tool for complex nonlinear dynamic systems such **as** mobile robots. Neural networks are used to process data from many sensors for the real-time control of mobile robots and to provide the necessary learning and adaptive capabilities for responding to the environmental changes in real time.

However, traditional methods use monolithic neural networks. Monolithic systems are composed of just one (NN) that receives all data, learning the solution mapping. Monolithic (NN) present some problems, due to the usually conflicting tasks that exist in mobile robot navigation.

These can be handled by following a modular strategy, applying the ***divide and conquer*** principle and using ***functional task division*. This** approach leads to (NN) that are known as Modular Neural Networks (MNN) [2, 4]. A (MNN) consists of a multiplicity of (NN) organized in a way that improves both the systems overall performance, and the effectiveness of the training and architecture determination.

Monolithic (NN) training is normally a tedious procedure, and it is usually difficult to justify the obtained parameters. One of the most serious criticisms of (NN) is the fact that one does not know what is happening inside it. In other words, an (NN) behaves like a black box. A considerable benefit that can emerge from (MNN) is an interpretable and relevant neural representation of the systems behavior [4].

This article presents a new MNN architecture for reactive navigation of mobile robots in unknown environments. Figure 1 shows its global scheme. This architecture is based on modularity, which can be viewed as a manifestation of the principle of divide and conquer, which allows us to solve complex problems (navigation), by diving them into smaller sub-problems (modules), easier to solve and combining their individual solutions to achieve the final solution.

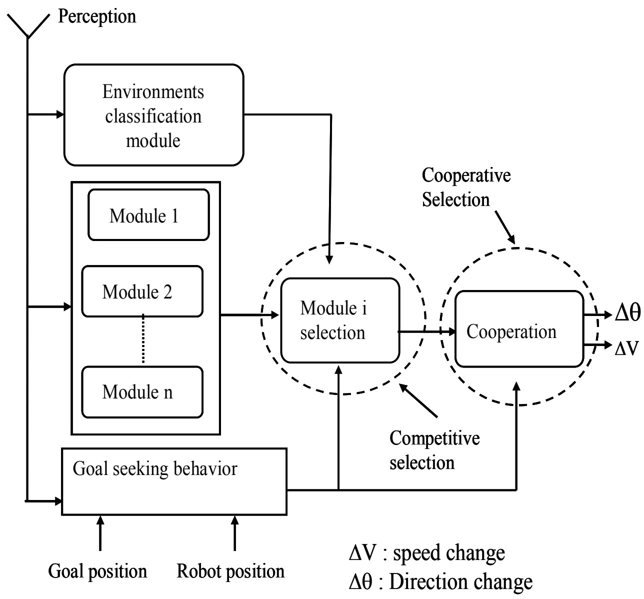



**Figure 1 :** Proposed structure

The principal modules developed in this architecture are:

- First, a Neural Network (NN) in charge of environment classification has been developed based on 11 prototypes of topological maps roughly describing various local navigation environments. This environment classification eliminates the requirement for prior detailed modeling of the environment since it needs to memorize only "rough" information on local environments encountered along the way that might be sufficient for navigation. This module is carried out using a

Bayesian classifier based on GBF (Gaussian Basic Functions) neural networks [8, 17, 18].

- Next, three elementary modules of reactive navigation directly associating a simple action to the perception without passing by an explicit model of the environment, are then trained to learn the rules of navigation deduced from the human expertise in each of the 11 prototype local environments: a module to turn right, a module to turn left, and a module to advance straight. These modules are carried out using MLP (Multi Layers Perceptron) trained by the back propagation gradient algorithm [4, 8].
- The choice of the module which takes the control of the robot navigation is done in a ***competitive*** [2, 13, 15] way using the environment classifier. Once selected, the local navigator works in ***co-operation*** with a module of attraction towards the goal to determine the shifting of speed and orientation of the robot. This simple and intuitive strategy is deduced from the human expertise.

The proposed architecture is adaptive to dynamic environments, robust against sensor noise; it avoids local minimum traps as well as solves the problems of poor obstacle clearance or oscillation. It is also amenable to easy addition of new behaviors due to its modularity.

In this study we assume the following conditions:

- Robot moves from a starting point towards a target point in a structured unknown environment without any preliminary knowledge neither on the form nor on the position of the obstacles which it can meet at any time of its mission.
- It is supposed that the robot has non-holonomic characteristic, it moves without slip on a plane ground. It is equipped with a dead-reckoning system for keeping track of its orientation and position. The dead-reckoning system determines the robot's present position from a previous one with information regarding the path and velocity taken between the two positions.
- The robot can move forward with varying speed $(\nabla V)$ and turn right or left with variable number of degrees $(\nabla \theta)$.
- The robot has 16 ultrasonic sensors for observing the surrounding environment and measure the distances separating the robot from the walls of the obstacles and the environment. Ultrasonic sensors are evenly distributed around the robot, yielding a 22.5 degree angle between any two adjacent sensors. However, for navigation task, the robot in general does not need "to see" what there is behind it; we thus considered only the nine sensors of front, and for safety reasons we add a sensor on both sides; we obtain thus a vector of perception containing eleven distances (figure 2**).**
- Robot sails step by step. At each step, takes telemetric measures, recognizes its local environment, and moves towards its goal without entering in collision with the obstacles, under the control of the one of the four developed reactive modules of navigation.
- All these sub-tasks are acquired by learning by (MNN). It is what we explain in what follows :

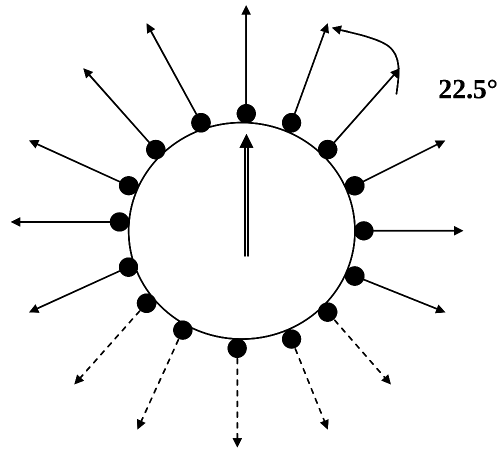



**Figure 2 :** Robot sensors localization. The dashed sensors are not used

This paper is organized as follows: the section 2 describes the neural network based environment classification. Section 3 presents various developed reactive navigation behaviors, and then it explains the action selection scheme.

The section 4 reports and discusses some experimental results. A conclusion and possible extensions concerning this work are given in section 5.

## 1. ENVIRONMENT CLASSIFICATION MODULE

This module is used to determine the necessary behaviors at the high level. It takes as inputs the telemetry measurements, identifies and classifies in real time the environment surrounding the robot as one of 11 prototype environments defined in figure 3, that any mobile robot could encounter during navigation. This environment list is of course non-exhaustive but sufficient enough to validate the proposed architectures.

This module has been designed with a hierarchical classifier for both an increase of speed and a higher recognition ratio; it splits the set of environments into different classes by an upper level network (figure 4).

The outputs of this level are the class 1 (corridor class), class 2 (crossing class), class 3 (corner class) and class 4 for environments not belonging to the previous ones as well as for future developments of the network. The current environment is then classified by the networks of the second level which holds three networks working in parallel for giving the posteriori probability of each class [4, 10, 11].

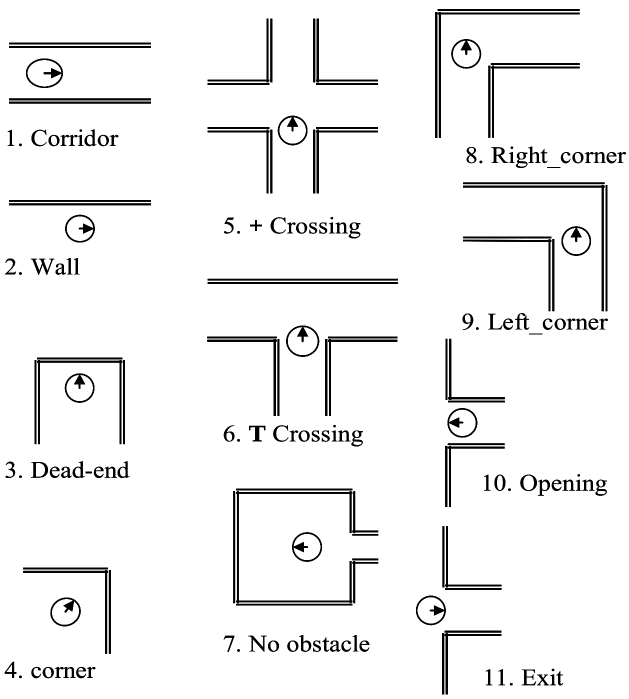



**Figure 3** : Eleven prototypes of local environments

Much research show that GBF network is able of very powerful calculations; it has more recognition precision and faster learning rate than back propagation network, and it can improve the veracity of the classified information.

Each network has been implemented with a Gaussian Basis Function (GBF) [9, 11, 17, 18] (figure 5).

Several designs related to the network architecture and training have been made to construct the GBF network; and generally there are three learning phases in the GBF network.

- The number of units in the hidden layer (M)
- Centers of the radial basis function ($C_i$)
- Weighted links for the hidden layer to the output units (W).

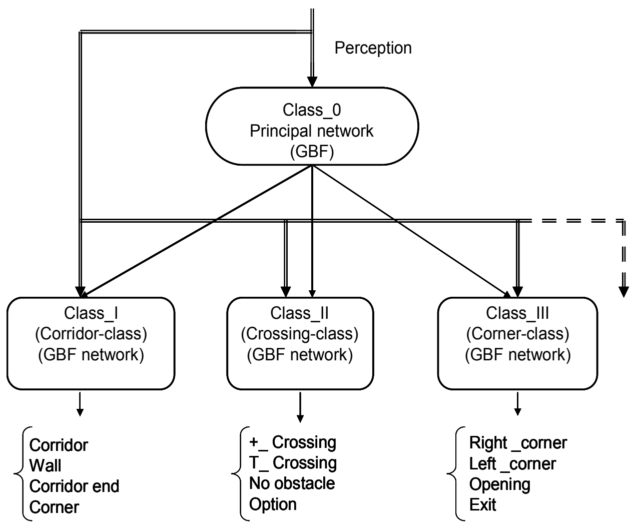



**Figure 4** : Hierarchical classifier

The key for the GBF network is to choose the center Ci of the Gaussian basis function. In some practices, centers are random samples selected from the input space and they may have approximate linear correlation leading to poor performance of the network.

The training of the hidden layer center can be done using clustering techniques (K-means, Kohonen. [8]). Here the orthogonal least squares algorithm (OLS) [11, 13] is adopted to resolve the key problem of GBF network construction. The network is trained by training samples based on OLS algorithm to get the parameters of the GBF network : M, Ci and W.

The OLS algorithm is better than the random selection method and K-means method to train the GBF network considering the performances of the approximation time and ultimate effect of the network.

When the training phase is achieved, the generalization takes place by means of the examples of the generalization database. This base is set up by :

- Randomly choosing the environments.
- Changing the dimensions of the environments.
- Locating the robot at different places in the environments with different orientation.
- Introducing noise in the measures.

These examples are given to the network for checking its ability to work in real environments.

The results were encouraging since in all cases, a recognition rate exceeding 94% was obtained.

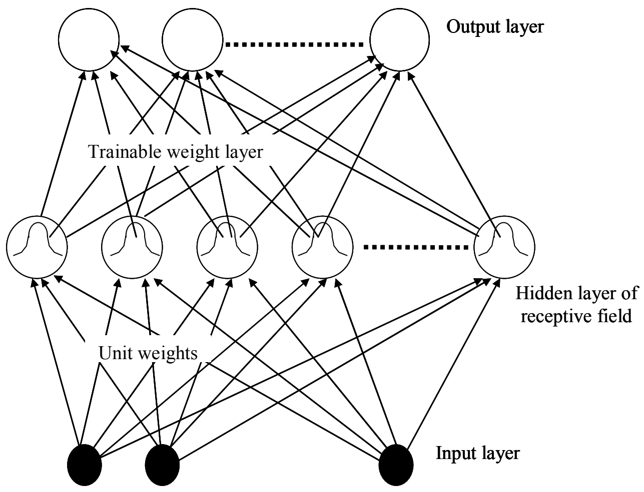



**Figure 5 :** Neural network structure

The output of the environment classification network will determine the high level behavior in conjunction with the goal seeking module, and further, may even be used to construct a topological map of the global environment. This topological map will memorize only the features of the environments essential for navigation and thus lends itself to real time control by not storing any surplus information referring to unnecessary details.

## 2. REALIZATION OF BEHAVIORS

After an environment has been classified, the needed reactive behavior for that class of local environment has to be determined in order to control the robot navigation**.**

### 2.1. Reactive navigation behaviors

Three basic behaviors: *Right turn*, *Left turn*, and *Straight-going*, have been implemented using a modular multi layers neural network (figure 6).

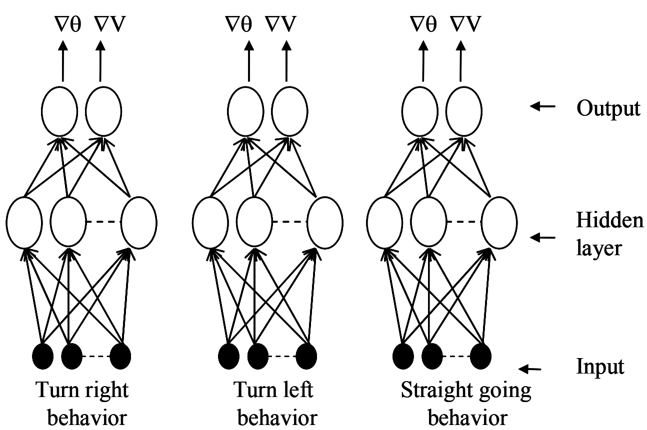



**Figure 6 :** Structure of Reactive navigation behaviors

This network will look at the sensory situations and generate a suitable change of direction and speed of the robot.

The modular neural net controller has several advantages: (1) being resistant to noise that exists in real sensors (2) smooth control performance, (3) being able to generalize their ability in new situations, and (4) elimination of blocking and oscillation that may arise when using only one neural network due to contradictory training examples [4].

Each controller is trained only under feasible environments listed in Table 1. The robot is placed in several different positions, with a random initial orientation in each predefined environment; the desired commands are then generated from a human supervisor.

This way, the neural network may even capture a human personality of driving.

**Table 1 :** Training environments for each behavior

| Behavior | Training environments |
|---|---|
| Straight going behavior | 1,4, 5, 7, 10, 11 |
| Turn right behavior | 2, 4, 5, 6, 7, 8, 10, 11 |
| Turn left behavior | 2, 4, 5, 6, 7, 9, 10, 11 |

### 2.2. Special modules

One of the advantages of the modular architecture is that it easily allows the insertion of new behaviors. We thus exploited this property to add two special modules: one to deal with the navigation of the robot in the U-shaped environments; the other, to allow the robot to reach its target.

#### 2.2.1. Backing up behavior

The U-shaped environment can be recognized as one of the eleven prototypes defined in section 3. This environment, in general poses problems of blocking and oscillations during navigation [11, 15] (figure 7a).

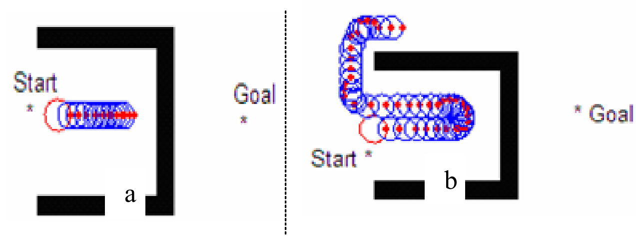



**Figure 7 :** U shaped environment
(a) Local minimum
(b) Escape from U-shaped local minimum by the baking up behavior

To circumvent these problems and to be able to consider a great diversity of environments, we preferred to carry out special reactive navigation behavior for this type of environment. The methodology of its development is the same one as that adopted for the other behaviors (section 2.1).

When the dead end environment is recognized by the environment classifier, the robot slows down, changes gradually direction with a minimal speed until forming an angle of 180°, circumvents the obstacle and it then points towards the target (figure 7 b).

### 2.2.2. Goal seeking module

The task of navigation of the mobile robots is in general divided into two important sub-tasks which are:

- Obstacle detection and avoidance
- Attraction towards the goal

This division is easily justifiable. Indeed, to go towards a destination, the human being must firstly detect and avoid all the obstacles, and secondly move towards the target. It is as obvious as these two tasks can be contradictory, i.e. that one can carry out the robot in a direction different from the other; this justifies and reinforces the application of the modular concept to the problem, to avoid the blocking and oscillation situations that can occur when a problem has contradictory objectives.

The module of attraction towards the goal determines constantly the speed and the change of direction which brings more closer the robot to its objective.

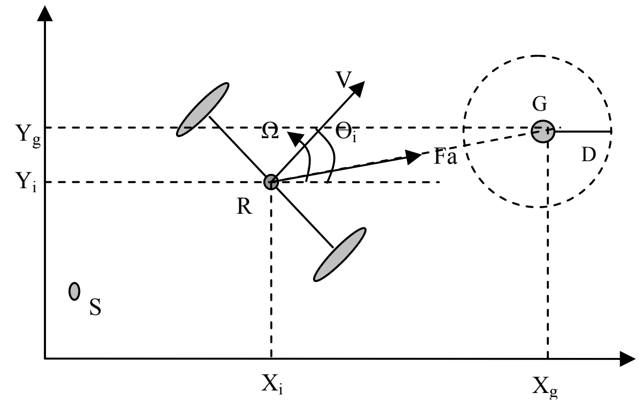



**Figure 8 :** Attractive force produced by the goal

The entries of the module are the position $(X_g, Y_g)$ of the target to be reached; and the current position and orientation $(X_i, Y_i, \theta_i)$ of the robot. Its operation is based on the concept of the potential fields [10, 13] (figure 8).

The goal G produces an attractive force $F_a$ that guides the robot to its destination. The actions $(\nabla V)$ and $(\nabla \theta)$ generated by this force are modulated by the inverse of the distance *RG* between the center of the robot and the goal. *D* is the distance of influence of the goal. It is supposed that no obstacle exists in the circle of diameter D.

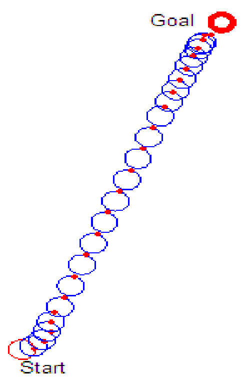



**Figure 9 :** Robot attraction towards the goal

The robot always points in direction of the goal with a step of ± Π/8 with minimal speed, and when it is enough faraway from goal (*RG* > *D*), it advances with a maximum speed. As soon as the robot reaches the zone of influence of the goal, it starts to slow down until it completely stops in front of the objective (figure 9).

### 2.3. Coordination of behaviors

The choice of the reactive behavior which deals with the control of the robot (figure 10) is done in a competitive way, according to the environments classifier output and the goal position.

- If the environment surrounding the robot is recognized as being a U shape, then the backing up behavior controls the robot to escape the dead end.
- Else, the action behavior is selected first according to the type of the local environment recognized (table.1), and then according to the target position compared to the robot (figure 14).

- If the target is in zone 1 (figure 11) of the vision field of the robot, the ***straight going behavior*** is then selected to control the robot.
- If the target is in zone 2 (figure 11) of the vision field of the robot, the ***right turn behavior*** is then selected to control the robot.

- And finally if the target is in zone 3of the vision field of the robot, it is the ***turn left behavior*** which is selected to control the robot.

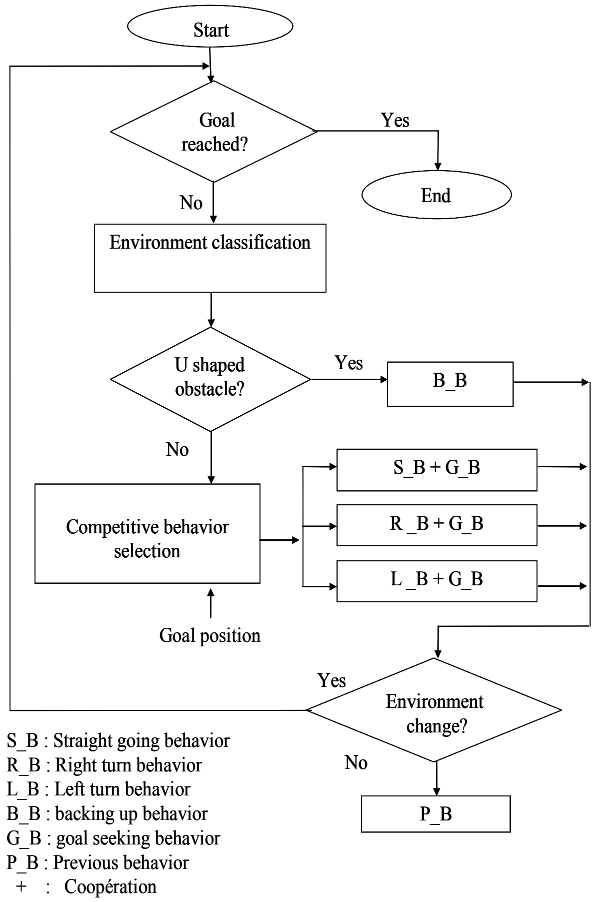



**Figure 10 :** The flowchart of coordination of modules

Once that one of the three reactive behaviors is chosen, it must work in a cooperative way [6, 15] with the goal seeking module to control the robot.

The linear and angular velocity to apply to the robot, are the result of the following linear combinations:

$$\nabla V = \alpha \nabla V_1 + \beta \nabla V_2 \qquad (1)$$

$$\nabla \theta = \alpha \nabla \theta_1 + \beta \nabla \theta_2 \qquad (2)$$

Where:

$\nabla V_1 , \nabla \theta_1$: are the linear and the angular velocity given by the reactive behavior.

$\nabla V2 , \nabla \theta 2$: are the linear and the angular velocity given by the goal seeking module.

$\alpha$: posteriori probability of the recognized environment [11, 19] (section 3).

$\beta$: posteriori probability of the class no obstacle (class 7 see figure 3).

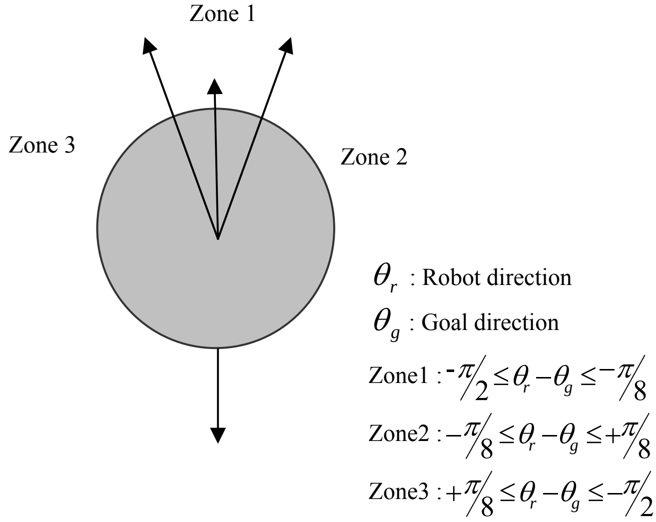



**Figure 11.** Goal position compared to the robot

This simple rule of fusion makes it possible to give the priority to the seeking goal module, each time the zone (goal – robot) is free obstacles and thus to optimize the trajectory.

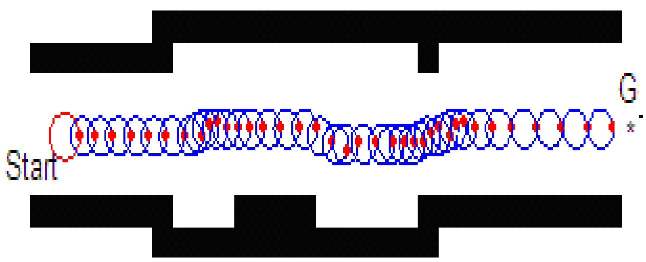



**Figure 12 :** Follow corridor under the control of the *straight-going* behavior

Once, a reactive behavior is selected, it continues to control the robot navigation as long as the environment did not change.

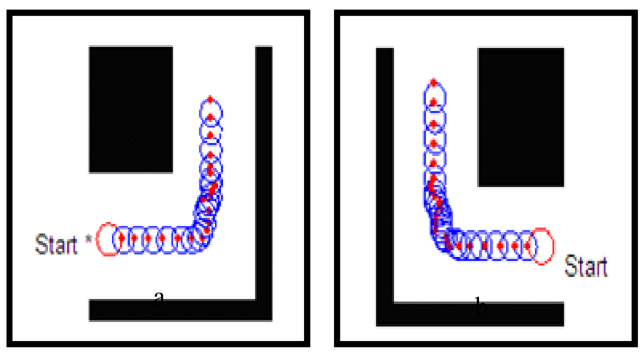



**Figure 13 :** Navigation of the robot in a right turn and left turn under the control of the ***turn right*** (a) and the ***turn left*** (b) behaviors

## 3. RESULTS

To evaluate the performances of the suggested architecture, we first tested each behavior separately in test environments different from those used in training phase.

Figure 12 shows the robot navigation in a corridor under the control of the ***straight going reactive behavior.*** We note that the robot follows the center of free space, and it reduces its speed each time that it meets an obstacle.

This test enabled us to evaluate the adaptation capacity of the module to the abrupt changes in environment dimensions and to avoid certain obstacles with a suitable speed.

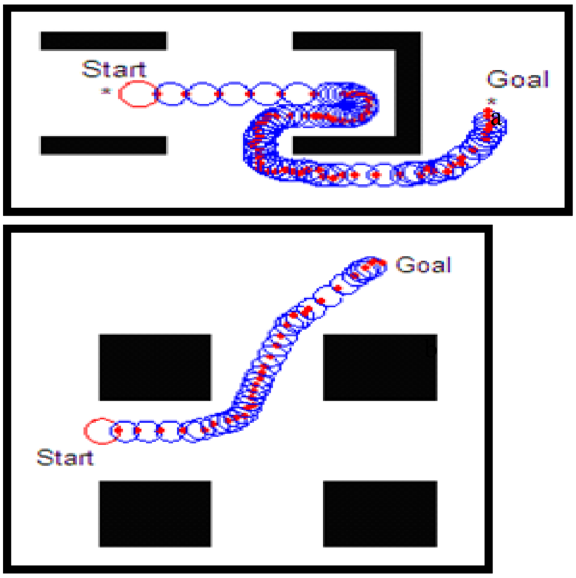


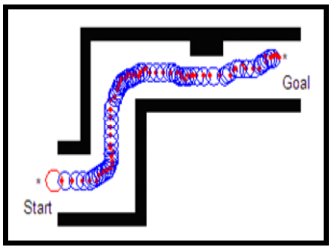

**Figure 14 :** Co-operation between the reactive navigation behaviors and *goal seeking* behavior

In the same way, the figure 13 shows that the robot turns on the right under control of the ***turn right behavior*** (a), and turns on the left under the control of the ***left turn behavior*** (b) successfully. It follows always the centre of free space and reduces its speed during the direction changes (concepts introduced during the training).

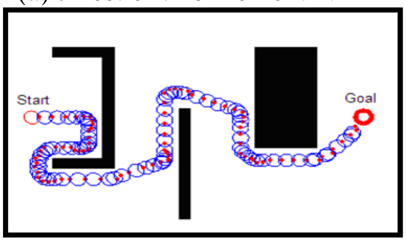
The environments of figure 14 allow us to test and to evaluate the capacity of the suggested architecture, to cooperate the various reactive navigation agents, with the ***goal seeking*** behavior.

On the environment (a) of figure 14, at the crossing, the robot has several choices: turn right, turn left or advance; it chooses the ***straight-going*** behavior since the target is in front; arrived at the dead end, the special ***backup behavior*** takes control and makes the mobile back up to escape the U-shaped dead end. And then, the ***straight-going*** behavior takes again control and works in co-operation with *goal seeking* module so that the robot approaches more the target with a suitable speed.

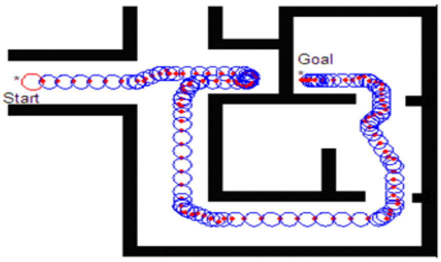
In the same way on the environment (b) of figure 14, we note a perfect attraction towards the goal wile avoiding all obstacles.




(a) : Test environnement N° 1




(b) : Test environnement N° 2




(c) : Test environnement N° 3 (b)

**Figure 15 :** Robot navigation in unknown environments, under the developed architecture.

Figure 15 shows some examples of robot navigation under the control of the whole architecture using all the developed reactive agents.

These results are obtained in more complex environments composed of the various assemblies of situations increasingly distant from those of the training base. In all cases, one note that the robot carries out a route without difficulty, adapts very well to the dimensions changes, carries out the crossing of the doors successfully, avoids the obstacles, and always follows a smooth trajectory in the center of free space with a perfect attraction towards the goal.

## CONCLUSION

A new modular architecture has been developed in this paper which achieves safe and robust navigation to a given goal in an arbitrary environment in real time.

Our contributions are as follows:

- First, a Modular Neural Network (MNN) in charge of environment classification has been developed based on 11 prototypes of topological maps roughly describing various local navigation environments. This environment classification (MNN) not only enables the navigator to avoid local minimum points but also eliminates the requirement for prior detailed modeling of the environment since it needs to memorize only “rough” information on local environments encountered along the way that might be sufficient for navigation task.
- Next, a set of reactive behaviors controller have been trained to learn human steering commands for each of the 11 prototype local environments.
- Third, an objective direction module is used to select a particular reactive behavior in conjunction with the classification (MNN).
- Finally, a modular control architecture integrating all these concepts was developed.

The proposed architecture avoids local minimum traps as well as solves the problems of poor obstacle clearance or oscillation. It is robust against sensor noise due to the use of NN) and adaptive to dynamic environments. This architecture is also amenable to easy addition of new behaviors due to its modularity.

The results obtained with this modular architecture show its effectiveness and robustness, for navigating a mobile robot around obstacles, without knowledge of the environment.

Future work will include the following :

- The study of online learning as a way to improve the system performance. If it is found that the robot has made a mistake, it could be possible to identify the module responsible and eventually re-train or re-construct it.
- The environment classification network may be used as a useful component of map building based on the robot’s navigation experience. And, after a map has been built, an optimization of the path based on the prior map may be further investigated.
- Ultrasonic and vision sensors may be fused for more robust and accurate landmark representation.
- Another line of future work is the research into potential improvements introduced by adding static environmental information, such as artificial landmarks
- The ultimate goal will be global navigation based on a map data consisting of the chain of landmarks like we humans do in our daily maneuvers.

## REFERENCES


**[1]** Acosta L., Marichal G. N., Moreno L., Mendez J. A. and Rodrigo J. J.; “ Obstacle Avoidance Using the Human Operator Experience for a Mobile Robot”; Journal of Intelligent and Robotic Systems, 27, pp. 305–319, 2000.

**[2]** Althaus P. and Christensen H. “Behavior coordination in structured environments”, Advanced Robotics, Vol. 17, N°. 7, pp. 657–674, 2003.

**[3]** Berlanga A., Sanchis A., Isasi P., And Molina J. “Generalization Capabilities of Co-evolution in Learning Robot Behavior” Journal of Robotic Systems 19(10), pp.455–467, 2002.

**[4]** Bishop C.M., ″Neural network for pattern recognition”, Oxford University Press, 1995.

**[5]** S. Branko “An integrated learning approach to environment modeling in mobile robot navigation”, Neurocomputing, N° 57, pp. 215-238, 2004.

**[6]** Brooks R.A., “Intelligence without representation”, Artificial Intelligence, N° 47, 1991, pp. 139-159.

**[7]** Carme T., “Robot adaptivity”, Robotics and Autonomous Systems, N°.15, pp. 11-23, 1995.

**[8]** Freeman J. and Saad D., "Radial basis function networks: generalization in over realizable and unrealizable scenarios ", Neural Network, N°. 9, pp. 1521- 1529, 1996.

**[9]** F. Hendel, L. Mesbahi, et N. Berrached, "Application des réseaux de Kohonen à la localisation des robots mobiles ", Conférence Maghrébine sur le génie électrique de Constantine le 05 et 06 novembre 2001

**[10]** Hendel F. et Berrached N., ″Navigation des robots mobiles par les réseaux de neurones″, Conf. Int. CSS'95, Rabat, Maroc, Oct. 95 (in French).

**[11]** Hendel F., Mesbahi L, et Berrached N. «Application des réseaux RBF à la reconnaissance d’environnements d’un robot mobile autonome ", Conf. Inter. CIP, CDTA, Algérie, pp. 392 – 398, 9 - 11 juin 2001 (in French).

**[12]** Meng M., ″Mobile robot navigation using neural network and non metrical environment″, IEEE Trans. On Control Systems, Vol.13, No.5, pp. 30-39, 1993.

**[13]** Mesbahi L, Hendel F., Berrached N. et Benyettou A., « Implementing Temporal RBF for reactive navigation of robot mobile » Inter. Arab Journal of Information Technology, Vol. 2, N°. 4, pp. 275 - 280, 2005.

**[14]** Pérez José P., Simo G.B. E., and Blanes M. M. F., “Differentiating walls from corners using the amplitude of ultrasonic echoes”, Robotics and Autonomous Systems N°.50, pp. 13–25, 2005.

**[15]** Rylatt M., Carnekhi C. and Routten T., “Connectionist Learning in Behaviour Based Mobile Robots: A Survey”, Artificial Intelligence Review 12, pp. 445–468, 1998.

**[16]** Simon Yang X, and Meng M., “Neural Network Approaches to Dynamic Collision-Free Trajectory Generation,”, IEEE Transactions on Systems, Man, and Cybernetics —Part B:, Vol. 31, N°. 3, June 2001.

**[17]** Sueira J. C., "Cost functions to estimate a posteriori probabilities in multiclass problems" IEEE transactions on Neural Network, vol. 10, N°. 3, pp. 645-6551999.

**[18]** Tresp V., Hollatz J., and Subutai A., “Representing Probabilistic Rules with Networks of Gaussian Basis Functions” Machine Learning, 27, pp. 173–200, 1997.

**[19]** Yong-Kyun Na. and Se-Young Oh., “Hybrid Control for Autonomous Mobile Robot Navigation Using Neural Network Based Behavior Modules and Environment Classification”, Autonomous Robots N°15, pp.193–206, 2003.