A Hierarchical Self-Organizing Controller for Navigation of Mobile Robots*

Rodrigo Calvo and Roseli Ap. Francelin Romero
ICMC-SCE- University of Sao Paulo
13560-970 – Sao Carlos – Brazil
{rcalvo,rafrance}@icmc.usp.br

***Abstract* - In this work an autonomous navigation system based in a modular neuro-fuzzy network for controlling mobile robots is proposed. Based on this system the robot is able to reach goals avoiding collisions against obstacles in an unknown environment. The system architecture belongs to the reactive paradigm. A reinforcement learning mechanism balanced with two innate behaviors, which are to avoid obstacles and seek to goals, guides the robot from an initial point to the goal. The validation of the proposal system has been done by using the Saphira simulator. The results obtained in the tests performed on Saphira simulator and on the Pioneer robot show the efficiency and learning capabilities of this system.**

I. INTRODUCTION

An important issue that has gained a lot of attention in intelligent robotic is robot autonomous navigation. Autonomous navigation problem consists in a development of getting of decisions mechanisms for only robot and multiple robots navigating in known environment. In this context, autonomous navigation systems must be able to guide robots without any external interference (independent systems) and defining actions to the robots.

In the literature, several works use the computational intelligence approach to obtain robust and efficient systems, such as target seeking, to avoid collisions against obstacles, environments exploration [1] [2]. Several applications can be find in this issue, such as, environment cleaning, object transporting, vigilance systems and high risk task for human [3].

Several intelligent autonomous systems proposed show interesting results considering this navigation problem. Neural networks are adopted to design an autonomous system that learns to control double navigation control variables: speed and steering angle [4]. The system is provided with two classes of sensorial fields: direction of the target and distance from the obstacles, respectively. A more complex navigation controller is based on an evolutionary technique (classifier system). Simulation results show that the system learns simultaneously to avoid obstacles, to reach targets, and to coordinate these behaviors when they are conflicting [5].

In [6], innovate neuron models provide the navigation system the capacity for learning spatial concepts. Target seeking and collision avoidance behaviors are incrementally associated with specific features of objects (e.g. color) during the environmental interactions. A reactive system using neural-fuzzy networks that coordinate innate behaviors of target seeking and collision avoidance (basic behaviors) becomes a mobile robot able to reach a goal point in unknown environments. Reinforcement learning strategy updates parameters of networks. Results show an efficient navigation for reaching goal points [7].

Navigation systems are proposed for mobile robots with features of mobility and autonomy through interacting with several models of environment generating new applications [8]. In [9] a multi-layer network is designed to control a robot in a Shop Floor environment. Probabilistic methods of mapping and localization are used to navigation of Pioneer I mobile robot. They become the robot able to transport documents from specified point to other [10]. A remote system for controlling robots is presented in [11]. In this system, via web interface, a mobile robot is controlled from anywhere of world. In [12], a computational vision system based in multi-layer neural networks guides a mobile robot to an object of a specific color and form avoiding collisions against obstacles.

This current work extends the proposal presented in [7]. Some changes allow the system to operate in a real robot. A laser sensor is attached to system and its signals are responsible for interacting robot with an environment. Initially, this system is tested in Saphira simulator, software that simulates exactly the actions of Pioneer I mobile robot. The simulation in Saphira is the previous stage of tests performed in the real robot. A comparative analysis between the system presented in [7] and the extended system proposed here is presented to evaluate the efficiency and learning capabilities of the system proposed.

This paper is organized as follows. In section 2, the robot model used in [7] is described. The navigation system is presented in Section 3. In section 4 and 5, are briefly described the Pioneer I mobile robot and Saphira simulator system, used in real experiments, respectively. The simulation results are shown in Section 6, besides a

* This work is supported by CAPES, Coordination for the Improvement of Graduated Personal, and FAPESP, Foundation for the Promotion of Science of the State of Sao Paulo, on this research and publication.

comparative analysis to evaluate the performance of the system proposed. A brief discussion about the results and future work possibilities are presented in Section 7.

II. ROBOT MODEL

The robot model does not present internal dynamics (Fig. 1). The robot adjusts its steering direction and moves (a move equals one distance unit and is constant). The steering direction adjustment may assume values from -15^{o} to 15^{o}.

Different sensor types capture signals from the environment. There is a set of target direction sensors. The output of every one is proportional to the angle ϕ between the sensor direction and the target direction ($\phi < 180^{o}$). A set of target distance sensors provides information about the distance from the target to the robot. All target distances are mapped to [0, 1]. Each sensor is associated with a specific distance. The closer the specific distance is to the real distance, the greater is the sensor output. Another type of sensor, the obstacle sensor, the distance measures between the robot and the obstacle situated in front of the sensor. They capture an obstacle landscape for the navigation system. There are 50 of each type of sensor, and they are uniformly distributed. One hundred target sensors are all around the robot, but the obstacle sensors are limited to the front (from -90^{o} to 90^{o}).

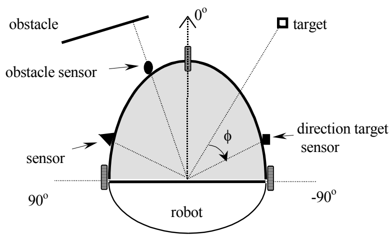

Fig. 1: A sketch of the robot model.

There are two sets of sensors to detect any collision against obstacles or target captures, respectively. The outputs of these sensors (collision and capture) are binary (1 if an event occurs, 0 otherwise).

III. AUTONOMOUS NAVIGATION SYSTEM

The autonomous navigation system (autonomous controller) consists of three main neural modules connected to an output neuron. Two of them, Obstacle Avoidance (OA) and Target Seeking (TS) modules, generate instinctive behaviors. A coordination module (CM) establishes (after learning) suitable weights for the behaviors generated by OA and TS modules. The weighed behaviors are combined in the output neuron (Fig. 2).

A. Neural Modules OA and TS

The OA and TS modules are neural networks with a priori knowledge about the respective behavior (they do not learn). They are innate neural networks. If they operate independently the robot is able to do specific tasks.

The OA module generates the obstacle avoidance behaviors. The inputs stem from the obstacle sensors and the outputs correspond to the adjustments for the steering angle. If only the OA module guides the robot, it does not collide. Unfortunately, it does not reach targets. The TS module generates the target seeking behaviors. It receives inputs from the direction target sensors. If the TS module guides the robot, it is able to reach targets. However, if there is an obstacle between the robot and the target, a collision occurs. These modules are innate, so, to keep the analogy with biological systems, the respective neural networks are configured according to an evolutionary approach [13].

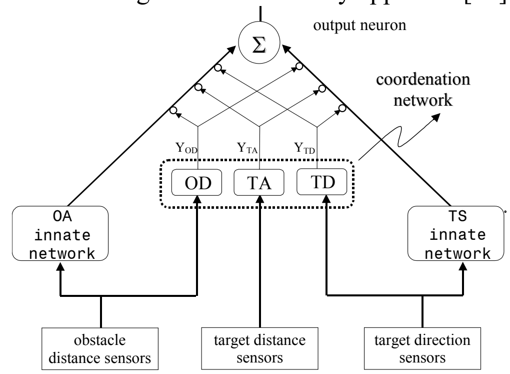

Fig. 2: Autonomous navigation system.

B. Coordination Neural Module

If the navigation system only consists of the innate modules, and they operate together without coordination, then there will be many conflicting situations and the navigation performance will certainly be poor. The function of the coordination module is to coordinate the instinctive behaviors generated by OA and TS innate modules.

The coordination module consists of three neural fuzzy networks: the Obstacle Distance (OD), Target Direction (TA) and Target Distance (TD) networks. They are connected to different sensorial fields: Obstacle Distance, Target Direction (Angle) and Target Distance, respectively (Fig. 2).

After a learning period, the coordination module learns the output weights of the innate modules, balancing each behavior according to the situation presented to the robot. Its outputs establish the weights of the output neuron. The reinforcement learning strategy is adopted for every three networks.

The architecture of neural networks consists of two layers of fuzzy neurons (neurons modeled according to Fuzzy Theory [14]). The first layer is constructive, that is, as the network learns some neurons are added to the layer. Each neuron connects to every sensor in the obstacle distance, target direction and target distance sensorial field (OD, TA and TD network, respectively). There is only one neuron in the second layer and it is connected to every neuron in the first layer (Fig. 3).

The OD network influences the behavior of robot by balancing the strength of each instinctive behavior (target seeking and obstacle avoidance) based on the obstacle landscape (vector of obstacle distance inputs received from the obstacle distance sensor field). It learns to associate obstacle landscape classes to their respective collision risk degrees. At every collision events (t_c moments) the learning process is triggered. In addition to adjust of synaptic weights of neurons are adjusted, a new neuron may also be inserted in first layer of network.

The TA network learns to associate classes of target direction signals with a suitable target seeking robot behavior. The learning process is triggered at every moment t_a when the robot captures a target (target capture sensors detect this kind of event). In the analogous way (considering the OD network) a neuron may be inserted to the TA network architecture at every target capture moments.

The TA network learns to associate classes of target direction signals with a suitable target seeking robot behavior. The learning process is triggered at every moment t_a when the robot captures a target (target capture sensors detect this kind of event). Following the same procedure as in both the OD and TA networks, a neuron may be inserted to the TD network architecture at every target capture moments.

The details of architecture (neuron models, layers and connections) and neural processing are presented in [7].

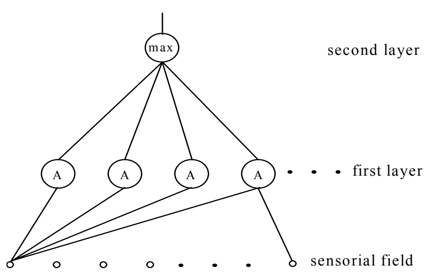

Fig. 3: Architecture of coordination networks.

IV. PIONEER I MOBILE ROBOT

Pioneer I mobile robot (Fig. 4) was designed by Dr. Kurt Konolige, researcher of Laboratory of Artificial Intelligence of University of Stanford. It is manufactured by Real World Interface and ActiveMedia. It has seven sonars for detection of environment information. Five of them are placed in front of robot. The remainder of this sensor is placed on side of robot. Three wheels are responsible for locomotion of robot: two immovable wheels are placed in front and one mobile dorsal that allows the robot to realize rotational movements. There are three ways to the robot communicates with a computer that controls it, such as transmission via radio, TCP/IP connection and via serial connection.

Software called Saphira keeps up the robot for supporting the development of control system for Pioneer robot. It is described in Section 4.

Because the detection of signals by sonars has high probability of presenting noises and low accuracy, the robot receives a sensor called PLS (Proximity Laser Scanner). It has some features that can improves the performance navigation, such as, range as 180º, angular resolution as 0.5º, maximum distance as 50 meter and error as 131 millimeters.

The robot and sensor PLS are connected to a computer that controls the robot through serial ports. A DC-DC converser to increment the voltage of robot battery that is 12V to 24V, so sensor PLS operates in this tension level.

Fig. 4: Pioneer I mobile robot of LABIC-ICMC-USP.

V. SAPHIRA SYSTEM

Saphira system is software that supports the development of applications specifically to Pioneer mobile robot control. Thus, this system is like architecture for controlling mobile robots. It was designed by Stanford Research Institute (SRI) International's Artificial Intelligence Center directed by Dr. Kurt Konolige.

Saphira operates in a client/server environment. The Saphira library is a set of routines for building clients. These routines perform most of the thankless work of communications and housekeeping for the robot server. Saphira library integrates a number of useful (Fig. 5) functions for sending commands to the server, gathering information from the robot's sensors, and packaging them for display in a graphical window-based user interface . In addition, Saphira supports higher-level functions for robot control and sensor interpretation, including fuzzy-control behavior and reactive planning systems, and a map-based navigation and registration system.

The Saphira client expects that the robot has the basic components for the robotics sensing and navigation, including drive motors and wheels, position encoders, and sensors. Saphira also expects that the robot support some, albeit little, onboard intelligence to handle the low-level details of robot sensor and drive management, and to be able to send that information and respond to Saphira commands - act as a server - through a special communications packet protocol.

Saphira system is composed by two architectures. The first, System Architecture, is a set of routines for communication and robot controlling to define applicatives. The second, Robot Control Architecture, controls navigation problems as motors and sensors control and planning and objects recognition. Both the architectures are open. Users are allowed to rewrite and replace the existents routines or add news functions.

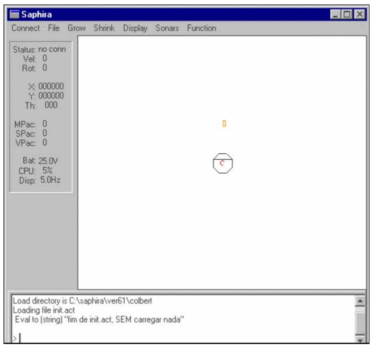

Fig. 5: Saphira simulation environment.

VI. RESULTS

In this section, a performance comparison between the navigation system described in [7] and the extended navigation system proposed in this paper is shown. They are called RS and SS from now on, respectively. In RS system, the environment is ideal and the sensors are simulated, hence there are not noises like ones found in real environment. Saphira system provides two devices to detect the environment signals, such as SICK laser and sonars. They are used to detect distance from obstacles. Therefore, they correspond to the sensors simulated in RS. Both the devices are compared to classes of sensors in RS. Since in SS the resolution of SICK laser as 0.5º (as mentioned in Section IV), then it has been considered that RS system should have 360 obstacle distance sensors. In similar way, as SS provides only 7 sonars, the experiments have been accomplished, in RS system, using 7 obstacle distance sensors.

A mobile robot simulator is conceived to evaluate the approach presented in [7], called RS simulator. There are not difficulties in a simulator that are found in real environment. On that score RS is implanted in environments where there are difficulties of real world. Then it is accomplished in SS simulator. The simulation in Saphira is the previous stage of tests performed in real time with Pioneer robot.

Some changes are done in signals as they are detected by sensors, because the differences between the sensor in RS and SS simulators are too significant. The model of robot in SS simulator just has sonar and Sick laser sensors. However, target direction and target distance sensor are similar to robot model in RS.

In Fig. 6 are shown the environments models in RS and RS simulators. Obstacles are represented by dark and clear (RS and SS simulators, respectively) rectangles. Targets are represented by dark ellipses. The robot is represented by a dark triangle.

The experiments are organized as follow. First, the innate modules experiments are presented. After, the coordinate neural module is validated. Each experiment is divided in four phases. They identify what the simulator and the sensor field are used during them. Experiments realized in mobile robot simulator and Saphira simulator are referenced as RS and SS, respectively. Simulations in RS that use 360 obstacle sensors and in SS simulator use the Sick laser is mentioned as Sick laser. In the same way, the simulations in RS and SS simulators using a small amount of obstacle sensors (7 sensors) and sonar device is mentioned as sonars.

Initially, OA and TS modules experiments are presented. In OA module, the robot guides avoiding collision, but it is not able to reach target. OA module simulations in RS and SS, (Fig. 6a and 6b, respectively), use Sick laser as obstacle distance sensor. In Fig. 6c and 6d is presented OA module simulations in RS and SS, respectively, using sonar sensors as obstacle distance sensor. During the simulation, the robot explores environments avoiding collisions against obstacles. This is observed in Table 1.

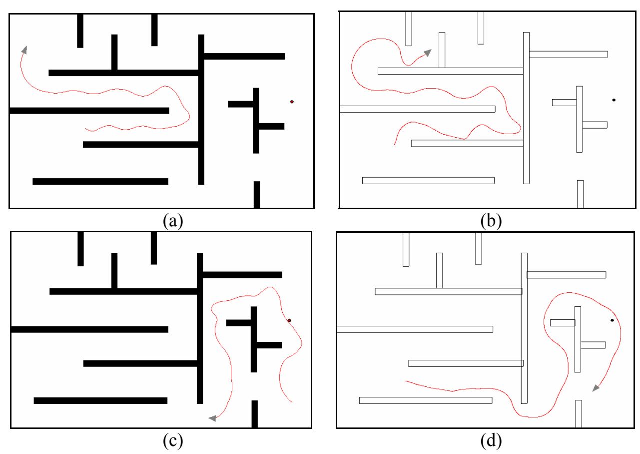

Fig. 6: OA module simulations: (a) RS simulator with Sick laser, (b) SS simulator with Sick laser, (c) RS simulator with sonars and (d) SS simulator with sonars.

TABLE I
OA INNATE MODULE PERFORMANCE.

Simulator environments	**Collisions**
RS / Sick laser	0
SS / Sick laser	0
RS / sonars	0
SS / sonars	0

In the second experiment is realized TS module simulations. In this case, the only robot task is target seeking. The obstacles are not considered, then a collision occurs if there is an obstacle between the robot and the target.

Simulations in the TS module are shown in Fig. 7. Fig. 7a and Fig. 7b present TS module simulations in RS and SS, respectively, using Sick laser as obstacle distance sensor. TS module simulations in RS and SS are seen in Fig. 7c and Fig 7d, respectively, using sonar sensors as obstacle distance sensor. The target seeking task fails, because there is an obstacle between the robot and the target. Therefore successive collisions occur as is observed in Table 2.

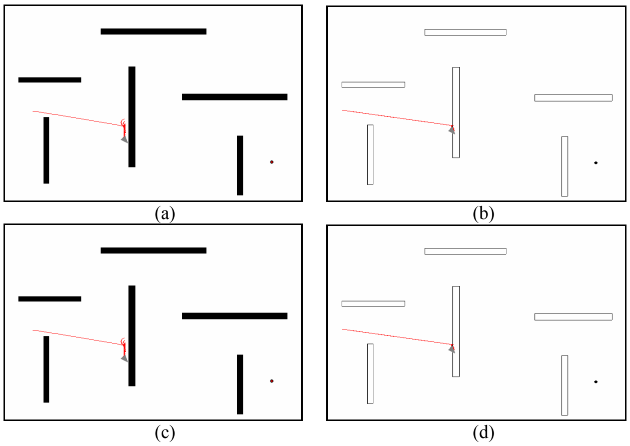

Fig. 7: TS module simulations: (a) RS simulator with Sick laser, (b) SS simulator with Sick laser, (c) RS simulator with sonars and (d) SS simulator with sonars.

TABLE II

OA INNATE MODULE PERFORMANCE.

Simulator environments	Captures
RS / Sick laser	0
SS / Sick laser	0
RS / sonars	0
SS / sonars	0

The third component of autonomous navigation system (Fig. 2) to validate is the coordination neural module. This module is assigned for balancing two innate behaviors: obstacle avoidance and target seeking. As the navigation proceeds, environment interactions provide the basis for a reinforcement learning strategy (described in [7]). After learning process period, the robot is able to reach targets without collision. Fig. 8 illustrates the environments where the coordination neural module simulations are accomplished. There are fifteen targets in each environment. During simulation, there is only one target in the environment. At capture moments, the target (reached) is eliminated and other one is inserted in the environment.

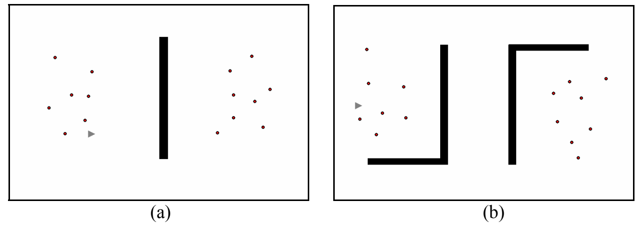

Fig. 8: Architecture of coordination networks.

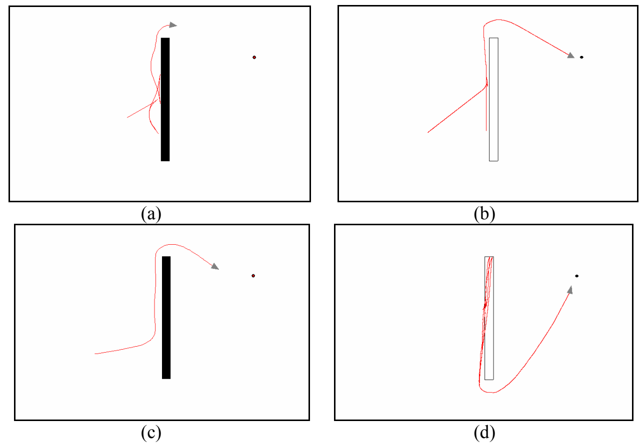

Fig. 9: Coordination neural module simulations: (a) RS simulator with Sick laser, (b) SS simulator with Sick laser, (c) RS simulator with sonars and (d) SS simulator with sonars.

TABLE III

COORDINATION NEURAL MODULE PERFORMANCE.

Simulator environments	Captures	Collisions
RS / Sick laser	15	9
SS / Sick laser	15	11
RS / sonars	15	10
SS / sonars	15	12

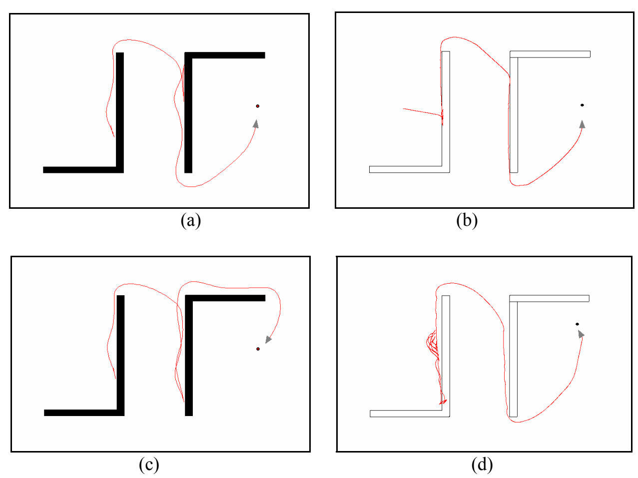

Fig. 10: Coordination neural module simulations: (a) RS simulator with Sick laser, (b) SS simulator with Sick laser, (c) RS simulator with sonars and (d) SS simulator with sonars.

TABLE IV
COORDINATION NEURAL MODULE PERFORMANCE.

Simulator environments	Captures	Collisions
RS / Sick laser	15	12
SS / Sick laser	15	12
RS / sonars	15	11
SS / sonars	15	12

In the validation experiments of coordination neural module shown in the Fig. 9 and Fig 10, both the RS and SS simulators have navigation system performance similar (Table 3 and 4). During environment adaptation, the robot suffers several collisions and reaching targets stimulates its seeking targets behavior. This is fundamental to validate the learning strategy adopted in [7]. It means that navigation system in [7] is robust and it is ready to be implanted in real time system as Pioneer I mobile robot.

During experiments, there are some difficulties to treat sonars signals because they present no accuracy and much noise. Repeating the sonars reading process instead of once is a solution for this problem. After that, the final reading is obtained through the reading arithmetic average. This process does not prejudice the extended navigation system (proposed in this paper) performance. It also improves sonars accuracy.

Attaching Sick laser to robot provides great performance, because this, there are, in the extended system navigation, 360 reading signals for distance obstacle sensor. The high amount of sensorial signals allows real environment to be transferred to system navigation.

This is beginning of simulation results. However, a reduction of number of collision when Sick laser is attached to robot is expected, so the laser signals have high accuracy.

VII. CONCLUSIONS AND FUTURE WORKS

The expectation is very intense for mobile robot applications where the environment is unknown. In these cases, autonomy is an essential characteristic for robot navigation systems. This work describes an autonomous navigation system based on a modular neural-fuzzy network. The system has two innate behaviors, namely, target seeking and obstacle avoidance, but initially it does not have ability to balance them. It learns, adopting a reinforcement strategy, to coordinate these behaviors from a continuous interaction with unknown environment. Computer simulations show that as the robot experiences some collisions and captures, the system improves its navigation strategy and efficiently guides the robot to target. This system is applied to Saphira simulator using sonar and Sick laser sensors to treat distance obstacle signals. Simulation results show that the performance of extended navigation system is similar to system in [7]. As innate modules presented satisfactory results in SS simulator, the learning strategy adopted in [7] is the responsible by the good performance obtained by coordination neural module. The simulations realized in Saphira simulator is the previous stage of tests performed in Pioneer robot. It consolidates the approach proposed in [7]. Therefore, as future work, the extended navigation system will accomplish in Pioneer mobile robot.

REFERENCES

[1] J. Borenstein, and Y. Koren, “The vector field histogram - fast obstacle avoidance for mobile robots”, IEEE Transactions on Robotics and Automation, vol. 7, (3), pp. 278–288., 1991.

[2] S. Thrun, “Bayesian landmark learning for mobile robot localization”, Machine Learning, 33:41, 1998.

[3] R. Rao and O. Fuentes, “Learning Navigational Behaviors Using a Predictive Sparse Distributed Memory”, In From Animals to Animats 4, Maes, P. et. al. (Eds.), Publisher: MIT Press, Cambridge MA, USA, pp. 382-390, 1996.

[4] P. R. Crestani, M. Figueiredo and F. Von Zuben, “A hierarchical neuro-fuzzy approach to autonomous navigation”, Proceedings of 2002 International Joint Conference on Neural Networks, (cd-rom), Honolulu, USA, 2002.

[5] R. Cazangi and M. Figueiredo, “Simultaneous emergence of conflicting basic behaviors and their coordination in an evolutionary autonomous navigation system”, Proceedings of 2002 IEEE Congress on Evolutionary Computation, (cd-rom), Honolulu, EUA, 2002.

[6] E. A. Antonelo, M. Figueiredo, A. Baerveldt and R. Calvo, “Intelligent Autonomous Navigation for Mobile Robots: Spatial Concept Acquisition and Object Discrimination”. 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation, Helsinki, Finland, 2005.

[7] R. Calvo and M. Figueiredo, “Reinforcement learning for hierarchical and modular neural network in autonomous robot navigation”, Proceedings of 2003 International Joint Conference on Neural Networks – IJCNN, Oregon, USA, 2003b.

[8] A. C. O. S. Aragão e E. Marques, “FPGA Technology”, Technical Report nº 60, Institute of Mathematics and Computer Science, University of Sao Paulo, Sao Carlos, Sao Paulo, Brazil, 2000. (in Portuguese)

[9] D. M. Medeiros e R. A. F. Romero, “Neural Networks Applied To Mobile Robot Navigation in a Shop Floor”, RIC – Scientific Initial Magazine, no. 6, pp. 45-51, 2004. (in Portuguese)

[10] R. E. Bianchi and R. A. F. Romero, “Messenger robot based in probabilistic navigation”, In VI Brazilian Simposium on Intelligent Automation, Bauru, Sao Paulo, Brazil 2003. (in Portuguese)

[11] A. W. Barbosa and R. A. F. Romero, “Control of Mobile Robots Via Internet”, Proceedings of COBEM –

18th International Congress of Mechanical Engineering, novovember, (cd-rom), Ouro Preto, Minas Gerais, Brazil, 2005.

[12] M. G. Quiles and R. A. F. Romero, “A Computer Vision System based on Multi-Layer Perceptrons for Controlling Mobile Robots”, Proceedings of COBEM – 18th International Congress of Mechanical Engineering, November, (cd-rom), Ouro Preto, Minas Gerais, Brazil, 2005.

[13] M. Figueiredo. and F. Gomide,. “Evolving neurofuzzy networks for basic behaviors and a recategorization approach for their coordination,” In: Genetic Algorithms and soft Computing (Herrera, F, and Verdegay, J. (Eds)), pp 533-552, Springer Verlag, USA, 1996.

[14] F. Gomide and W. Pedrycz “An Introduction to Fuzzy Sets: Analysis and Design”. The MIT Press, Cambridge, 1998.