Topological Autonomous Navigation for Mobile Robots in Indoor Environments using ANN and FSM

Daniel Oliva Sales, Fernando Santos Osório, Denis Fernando Wolf

USP – University of São Paulo / ICMC – SSC – LRM (Mobile Robots Lab.)
São Carlos, SP - Brazil
{dsales, fosorio, denis} @ icmc.usp.br

***Abstract*—In this paper, we present an autonomous navigation system based on a finite state machine (FSM) combined with an artificial neural network (ANN) in an indoor patrolling robot. In the first step, the ANN is trained to recognize the different specific environment configurations, identifying the different robot situations (states) based on laser detections. Then, a program generates the expected sequence of states and actions for a specific route defined by the user, configuring a path in a topological like map. So, the robot becomes able to autonomously navigate through this environment, reaching the destination after going through a sequence of specific environment places, each place being identified by its local properties, as for example, straight path, path turning to left, path turning to right, bifurcations and path intersections. The experiments were performed with a Pioneer P3-AT robot equipped with a 180º Sick Lidar and a in a Patrolling task in order to validate and evaluate this approach. The proposed method demonstrated to be a promising approach to autonomous mobile robots navigation.**

I. Introduction

THE application of Artificial Intelligence techniques to Autonomous Mobile Robots and Intelligent Vehicles have an important role in the international scientific robotics community [3][4][5]. One of the most desirable features in a mobile robot is the autonomous navigation capability. There are many important and well known works in this domain, as for example the Darpa Challenge (2004 and 2005 Grand Challenges at desert and 2007 Urban Challenge) [14][15] and the annual ELROB initiative [6][7], two of the most visible projects in this field of research.

Autonomous mobile robots usually execute three main tasks: localization, mapping, and navigation [8]. The localization task is related to estimating the robot's position in a known environment, using its sensors data. Mapping is responsible for creating a model to represent the environment based on robot's localization and sensors data. Navigation is the robot's capability to obtain information about the environment through its sensors, process it, and act, moving safely through this environment.

In order to develop an Intelligent Autonomous Robot, capable of navigating into structured environments composed by lobbies and rooms, one can assume that the robot must know its approximate localization, the environment map and the path to be followed (origin/destination). Navigation in this environment consists basically to follow a well-defined path, based on a previously well-defined map of the environment, and also considering its sensors data [1].

In this paper we focus on the topological navigation task, following a path in an indoor environment executing a patrolling task. The typical application of the proposed system is an indoor service robot which can autonomously move around, monitoring and detecting abnormal situations (e.g. intrusions, spots of fire). Indoor patrolling tasks are an important research challenge among the main working groups (WG1) of the Brazilian National Institute of Science and Technology on Embedded Critical Systems (INCT-SEC) [Ref-site-inct]. In order to develop such applications, the system should be easy to configure and use, with a quick setup of the environment map and patrolling task, and also, it can be robust in order to allow the robot to to move around and to detect abnormalities.

Our approach does not require a well-defined metric map of the environment, only a "sketch" representing the main components/elements describing a rough view of the environment. Moreover, our approach does not require knowing precisely the robot's position in the environment. Our main goal was to make the robot autonomously navigate through an indoor environment, executing a standard patrolling task. The robot is able to recognize some key-points in the environment and decide when/how to proceed in order to go straight, turn left or right, even when all these possibilities are detected simultaneously (e.g. intersections).

Our topological navigation approach uses an Artificial Neural Network [17] to classify the data obtained from sensors, and a FSM to represent the sequence of steps according to the chosen path. The ANN learns all possible states, and a FSM generator converts a single path into a sequence of these states. So, the system combines this deliberative topological navigation with a simple reactive control allowing the robot to safely navigate through the environment, and simultaneously patrol.

The next topics of this papers are organized as follows: Section 2 presents a review of some important related works; Section 3 presents the techniques and features used to identify the current state and actions, used to move the robot through the environment; Section 4 shows the experimental results obtained from tests in the indoor environment; Section V presents the conclusion and future works.

II. Related Works

Many different approaches were developed for navigation, using different types of sensors (e.g. laser, sonar, GPS, IMU, compass, cameras), individually or grouped [3][8][9]. If the environment map and/or robot's localization is unknown, usually only a reactive navigation is possible.

In order to implement our autonomous navigation system, a simple reactive system was not adequate, since the immediate reaction to the information provided by the sensorial system is not enough to guarantee the correct mobile robot control when following a complex path. A more robust control system should be implemented providing the sense of sequence and context which is absent in purely reactive models.

In Robotics, Finite State Machine (FSM) [10] based approaches are often used [11][12], as for example, the "Situated Automata" and the "Reactive Deliberation Architecture". FSMs are useful because the system can be easily described as a sequence of states (context changes), taking into account: inputs (sensors) that allows changing from one state (situation) to another one, and also defining for each state a specific action (motor action) associated to it. So for each state and state change, the robot is able to react properly. We chose to implement our control system based on this main idea that the mobile robot control system can be described by a FSM, using as inputs the route detection information obtained from sensors.

The use of a machine learning method such Artificial Neural Networks (ANNs) showed to be an interesting way to process the sensors data, identifying and classifying the states (current and transitions), and determining which actions must be taken. [2]

ANNs are tolerant to noise and input data imprecision, and are also able to identify the states and transitions between the states. ANNS are also very efficient in generalizing its knowledge and adjusting its outputs to many inputs, even when some inputs were not explicitly taught to the net (generalization capability). This way, ANNs are a very useful tool for path features classification and state detection.

The association of ANN and FSM is not an entirely new proposal, since this type of approach have being discussed and studied since the 90s [21][22][23][24], when the ANN models were developed and matured, occupying an important place in Artificial Intelligence and Machine Learning researches, but few works applied this concept to solve robotic problems.

Recent researches were developed combining FSM and ANN in robotic problems (autonomous vehicles) [18]. These works, however, were focused in a specific application: autonomous vehicle parking control. Little was done to evaluate the possibilities of extending this work to other applications.

On the other hand, some other recent works were developed focusing in road detection and classification using image processing and computer vision algorithms, allowing robotic systems (autonomous vehicles) to identify the route pavement and also to detect straight paths and turning paths [19][20]. The problem with this approach is that sometimes we need a higher level of decision making, as for example in a road intersection, where we need to consider the road detection information combined with a more sophisticated navigation plan.

In a previous work [1] we proposed a static pre-defined FSM combined with a vision-based sensorial method for an indoor robot's navigation in a structured environment, but only a reactive control was implemented and the FSM was defined by hand and hard coded into the system. The FSM was used to allow the system to previously detect the 90º turns, and act even when the camera was not detecting the track anymore.

III. System Overview

The developed system is composed by three main stages. The first one is to train the system to recognize all possible states (situations) using data collected previously from the environment. The second one is to generate the FSM for a specific planned path, and the last one is to autonomously navigate combining this deliberative control (topological path plan) with a reactive control to keep the robot in the center of the track. Figure 1 illustrates the sequence of steps performed by the system before the navigation.

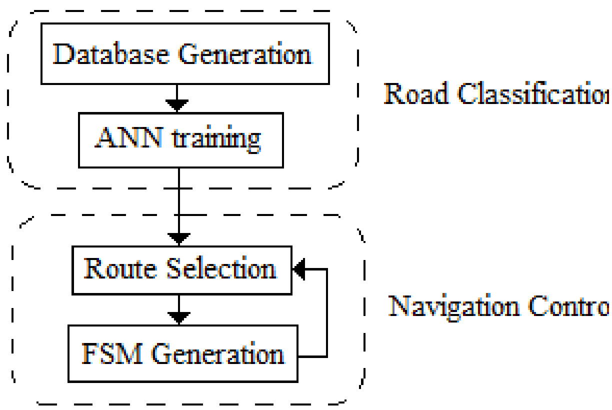


Fig. 1. System Setup Overview

A. *Path Classification*

This step consists in training the ANN to accurately recognize all possible situations (5 states). The inputs of the ANN are the data received from the sensors, and the outputs are the classified road states, classified into straight path, road turn to left, road turn to right, road intersection and bifurcation. The database is generated saving the log of collected data in a run through the track. As the network must have a supervised learning, a specialist must classify this data before the ANN training. The developed ANN is a multilayer feed-forward net with 3 layers. It has 20 neurons on input layer, 20 neurons on hidden layer, and 3 neurons on output layer, as shown of Figure 2.

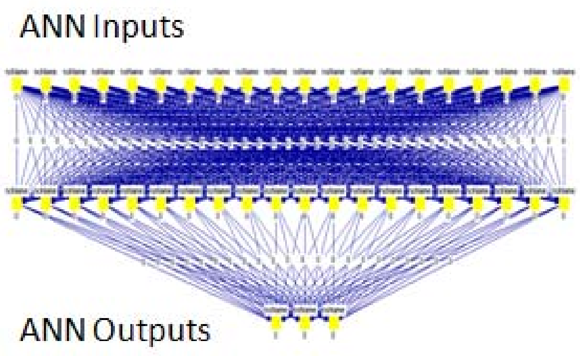


Fig. 2. Designed ANN Topology

Each input neuron corresponds to a different laser beam, in the range between 0 and 180º (Figure 3). The output is a binary representation of each road recognized state. The ANN must be trained just once and works for every possible path in the map.

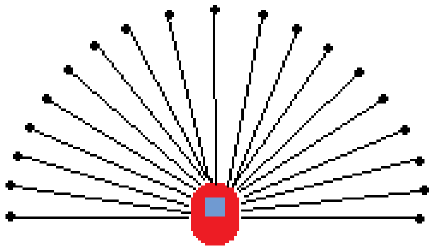


Fig. 3. Laser beams taken as inputs for the ANN

B. *FSM Generation*

Once the topological map is available for an indoor environment, it is easy to determine a route between two points of this map.

Every route can be seen as a sequence of steps (states), so it is trivial to generate a FSM for a defined path. A program converts any possible path of the map into a sequence of states and expected actions (it considers that one state can lead to more than one action, as shown next). So, the system select one desired path, and the FSM is saved to be read and used by the control unit.

C. *Navigation Control*

The hybrid control developed combines the deliberative control obtained from the FSM approach for topological navigation, with a reactive control in order to avoid the walls and to alert for human presence, performing its patrolling task.

The topological navigation allows the robot to follow the path and know its approximate location on it, but does not control the navigation inside every state. For example, when the robot is following a straight path, a reactive control is used to maintain the robot in the center of the road. That is why the hybrid control model was adopted, allowing the integration of a topological navigation (deliberative) with a reactive control.

The sensor processed data is used to know if the robot is still inside the current state (road segment) or if a context change is needed. A state change only occurs when the detected data is compatible with the next expected state.

There are five possible states, and nine possible actions, as described on Table 1. Every state has at least one associated action (intersection and bifurcation states has more than one), once the robot can choose one of them among the different possible paths connected with these situations.

TABLE I
POSSIBLE ACTIONS FOR EACH STATE

State	Related Actions
Straight	Go forward with reactive control
Right	Turn Right
Left	Turn Left
Intersection	Go forward / Turn right / Turn left
Bifurcation	Turn Right / Turn Left

While the robot is performing 90º turns, the sensorial system must be momentarily turned off, in order to avoid wrong data processing and classification during the operation. If the robot detects human presence, the navigation control must be momentarily paused, until the path becomes free. Figure 4 illustrates the navigation control flowchart.

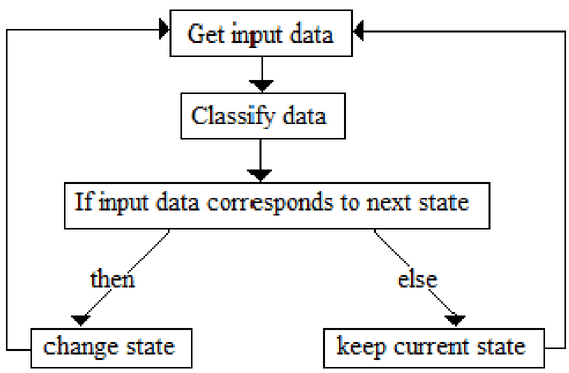


Fig. 4. Navigation control flowchart

IV. EXPERIMENTS AND RESULTS

Initial experiments were realized on Player/Stage environment, with a map which simulates a standard indoor situation, with 90º turns, straight paths, and intersections. This map required only four states and seven possible actions, so the initial ANN had two neurons on output layer. Fig. 5 shows the simulated robot performing the autonomous navigation in this environment, and a video of the state detection through the simulated path is available at [http://www.youtube.com/watch?v=J1utmdcDZXw or http://bit.ly/eLXx2f]

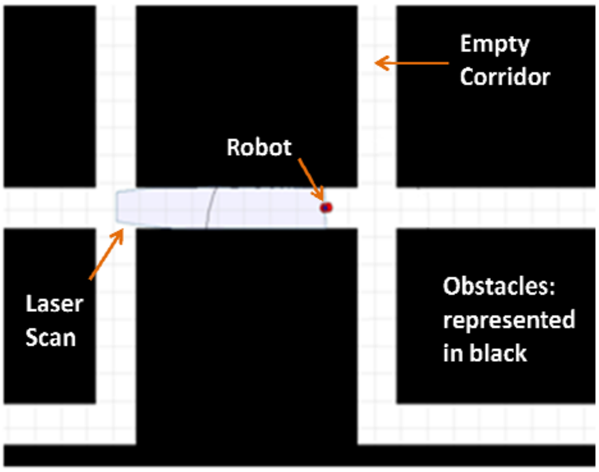


Fig. 5. Simulated robot navigating through the environment

Fig. 6 presents the laser scan data obtained in two different situations, and Fig.7 presents two actions being performed by the simulated robot for these situations.

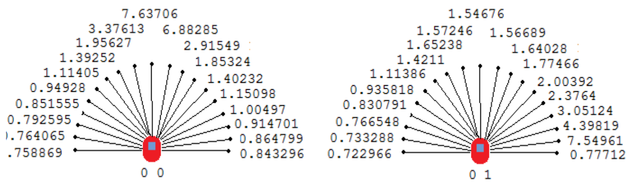
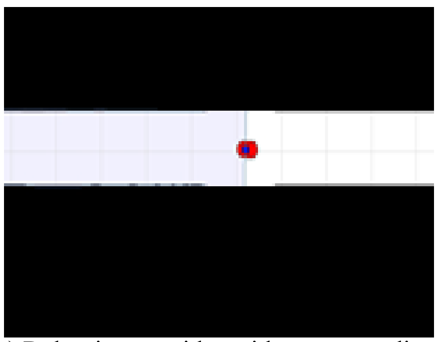


Fig. 6 – Laser data taken for two different states. Straight (0 0) and Right (0 1)


a) Robot in a corridor with corresponding laser scan area (in blue)

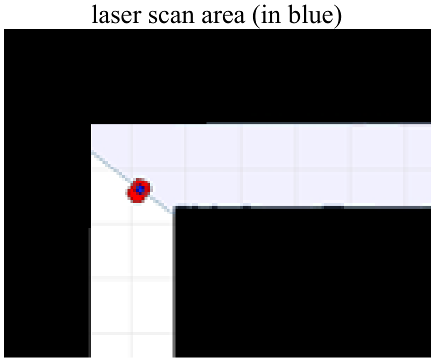

b) Robot in a corner with corresponding laser scan area (in blue)

Fig. 7 – Robot in straight path and right turn states

After this initial step, we validated the approach performing some experiments using a Pioneer P3-AT robot equipped with a 180º Sick Lidar (Fig 8), in a patrolling task through an indoor environment (a lobby with many adjacent rooms). The robot must autonomously navigate through this lobby, reaching specific pre-determined points, avoiding obstacles, with a FSM as specified on Table 1.

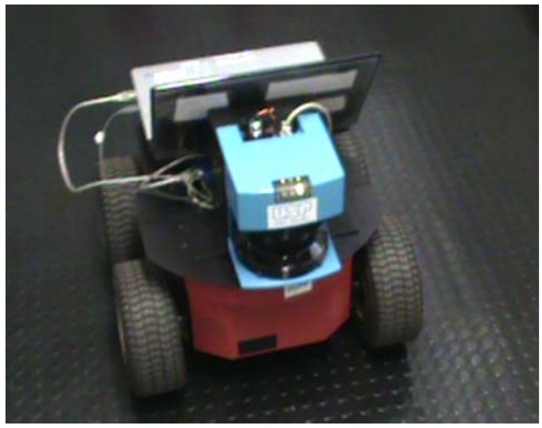


Fig. 8. Pioneer P3-AT robot with Sick Lidar used in experiments

In both experiments the ANN was implemented using Stuttgart Neural Network Simulator (SNNS) [13], so the trained ANN was converted to C language, using SNSS2C tool, and integrated with the robot control program.

The collected data to generate the database for ANN training was taken after manually control the robot through the map in many different angles and positions.

The ANN architecture is defined as follows: feed-forward multilayer perceptron network with twenty input neurons, three hidden layers with ten neurons each and a 3 neuron output layer (2 neurons only on first experiment). All activation functions were defined as Act_Logistic function of SNNS [13] that applies sigmoid logistic.

The log with laser data was saved with current state codification after every laser scanning. At this step, we generated 5429 input/output pairs for ANN training (supervised learning), and 250 pairs for validation.

This input/output pairs were necessary because training algorithm chosen for this network was Resilient Back-Propagation (Rprop)[16], which is a supervised learning algorithm that needs pairs with inputs/desired outputs.

Training parameters were defined as: $\delta 0 = 0.1$, $\delta Max = 50$, $\alpha = 4.0$ and number of epochs = 500. The error per output was very close to 0 (Error < 0.02). Figure 9 contains the error graph of the ANN training.

This algorithm have been achieving good results for feed-forward networks for many applications comparing to other training algorithms in convergence and training time. The main difference of this algorithm is that it takes into account only the sign of the partial derivative over all patterns (not the magnitude), and acts independently on each "weight".

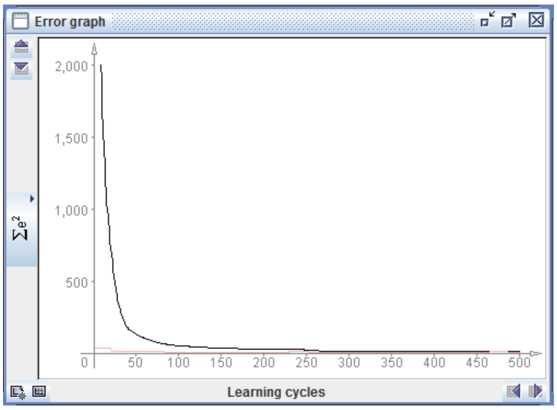


Fig. 9. ANN training error graph

Many different routes were tested, using different sequences of states and actions, and the robot performed all of them as expected, following correctly the pre-specified path (topological route).

The initial position and topological map of the environment are always known, and the actual position, estimated with laser data (current state). The exact position is not necessary, because a reactive control is responsible for the navigation inside every state.

Fig 10, 11, 12 and 13 illustrates the robot performing four different actions. A video of some interesting parts of the performed paths is available at [http://www.youtube.com/watch?v=faCEN9-SIao or http://bit.ly/fe05SK].

In order to execute the patrolling task, a parallel system was developed. It uses a FLIR thermal camera for human detection. The frames captured by the camera are processed using OpenCV in order to determine if a human is at the lobby. If a human is detected, the alert state is activated.

Fig 14. contains the original frame taken from FLIR camera in this situation, and Fig 15 contains the processed frame, with the alert state active.

A video of this system working is available at: [http://www.youtube.com/watch?v=1MNXZkoUKuM or http://bit.ly/gpjJvP].

Fig. 10. Robot Before a Turn Right State

Fig. 11. Robot in a Intersection State

Fig. 12. Robot in a Turn Right State

Fig. 13. Robot in a Straight Path State

Fig. 14. Original frame taken from FLIR camera

Fig. 15. Processed frame with alert state indicated

This system is also useful for the navigation task, because human presence could affect the laser data classification, so when a human is detected the robot momentarily stops the navigation task while the alert state is active.

V. Conclusion and Future Works

The implemented method obtained very good results, with 100% accuracy on classifying the track and driving through the expected route in all experiments carried out, showing that the association of ANN and FSM is a very convenient approach for mobile robotic navigation.

The system can be retrained to recognize more situations and also it can use and combine many other sensors, allowing its implementation on mobile robots both in indoor and outdoor environments. The FSM can be used not only to patrolling tasks but also in many other applications, such driving in urban environments for example, so the proposed method also demonstrated to be flexible in order to be easily adapted to other situations.

On the other hand, a poor ANN training or a high level of noisy inputs could lead to wrong classification and inadequate state changes, so we expect to solve this problem using a more robust ANN, and combining more sensors in order to increase the context detection and avoid fast state changes.

Acknowledgment

This paper presents the different contributions and results achieved by our research group which are directly related to the Working Group 1 (WG1) of the INCT-SEC. The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9

References

[1] SALES, Daniel Oliva ; SHINZATO, Patrick ; PESSIN, Gustavo ; OSÓRIO, Fernando S. ; WOLF, Denis F. . Vision-Based Autonomous Navigation System Using ANN and FSM Control. In: Proceedings of the IEEE LARS/EnRI 2010, 2010, São Bernardo do Campo, SP. IEEE Society Press, 2010. v.1. p. 85-90.

[2] Marino, A.; Parker, L.; Antonelli, G. and Caccavale, F. Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach. In: ICRA, 2009.

[3] Buehler, Martin; Iagnemma, Karl; Singh, Sanjiv (Editors). The 2005 DARPA Grand Challenge: The Great Robot Race (Springer Tracts in Advanced Robotics). Springer; 1st. edition (October, 2007).

[4] Bishop, R.; (2000) "Intelligent vehicle applications worldwide". IEEE Intelligent Systems - Intelligent Systems and their Applications, Volume: 15, Issue: 1. pp.78-81.

[5] Bishop, Richard. (2000). "A Survey of Intelligent Vehicle Applications Worldwide". Proceedings of the IEEE intelligentVehicles Symposium 2000. pp.25-30.

[6] Kuhnert, K.-D. (2008). "Software architecture of the Autonomous Mobile Outdoor Robot AMOR". Proceedings of the IEEE Intelligent Vehicles Symposium, 2008. pp. 889-894.

[7] Schilling, Klaus. (2008) "Assistance Systems for the Control of Rovers". SICE Annual Conference, Tokyo, Oct. 2008.

[8] Wolf, Denis F.; Osório, Fernando S.; Simões, Eduardo; Trindade Jr., Onofre. Robótica Inteligente: Da Simulação às Aplicações no Mundo Real. [Tutorial] In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). JAI: Jornada de Atualização em Informática da SBC. Rio de Janeiro: SBC - Editora da PUC. RJ, 2009, v. 1, p. 279-330.

[9] Goebl, M.; Althoff, M.; Buss, M.; Farber, G.; Hecker, F.; Heissing, B.; Kraus, S.; Nagel, R.; Leon, F.P.; Rattei, F.; Russ, M.; Schweitzer, M.; Thuy, M.; Cheng Wang; Wuensche, H.J.; (2008) "Design and capabilities of the Munich Cognitive Automobile". IEEE Intelligent Vehicles Symposium, 2008. Page(s): 1101 – 1107

[10] Hopcroft, J.E., Ullman, J.D. (1979) "Introduction to Automata Theory, Languages and Computation". Addison - Wesley, 1979.

[11] Medeiros, Adelardo A. D.. "A survey of control architectures for autonomous mobile robots". Journal of the Brazilian Computer Society - JBCS. 1998, vol.4, n.3.

[12] Sahota, Michael K. "Reactive Deliberation: An Architecture for Real-Time Intelligent Control in Dynamic Environments". Proceedings of the AAAI-94, p. 1303–1308.

[13] Stuttgart Neural Network Simulator, http://www.ra.cs.unituebingen.de/SNNS/, Accessed in Aug, 2010.

[14] Thrun, S. et al. (2006) "Stanley: The Robot that Won the DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, No. 9, June 2006, p.661-692. http://robots.stanford.edu/papers.html (Visited 08/02/2009).

[15] Urmson, Chris et al. (2008). "Autonomous driving in urban environments: Boss and the Urban Challenge". In: Journal of Field Robotics. Vol. 25 , Issue 8 (August 2008). Special Issue on the 2007 DARPA Urban Challenge, Part I. Pages 425-466.

[16] Rprop-Description and Implementation Details. Technical Report Januray 1994. Martin Riedmiller, University of Karlsruhe.

[17] Haykin, S. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[18] Heinen, M. R.; Osório, F. S.; Heinen, F. J.; Kelber, C. SEVA3D: Using Artificial Neural Networks to Autonomous Vehicle Parking Control. In WCCI-IJCNN. 2006

[19] Shinzato, P. Y. ; Fernandes, L. C. ; Osório, F. S. ; Wolf, D. F.. Path Recognition for Outdoor Navigation Using Artificial Neural Networks: Case Study. In: 2010 IEEE-ICIT International Conference on Industrial Technology, Viña del Mar, 2010.

[20] Shinzato, P. Y. ; Wolf, D. F. . Features Image Analysis for Road Following Algorithm Using Neural Networks. In: 7th IFAC Symposium on Intelligent Autonomous Vehicles, Lecce. 2010.

[21] Giles, C. Lee; Horne, Bill G.; LIN, Tsungnan. Learning a class of large finite state machines with a recurrent neural network. Neural Networks 8(9): 1359-1365. 1995.

[22] Omlin, Christian W.; Giles, C. Lee. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. Jornal of the ACM 43(6): 937-972. 1996

[23] Frasconi, Paolo; Gori, Marco; Maggini, Marco; Soda, Giovanni. Representation of finite state automata in Recurrent Radial Basis Function networks. Machine Learning 23:1, 5-32 1996.

[24] Cleeremans, Axel; Servan-Schreiber, David; MCCLELLAND, James L. Finite State Automata and Simple Recurrent Networks. Neural Computation, Vol. 1, No. 3, Pages 372- 381. 1989.