Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Intelligent System for Robotic Navigation Using ANFIS and ACOr

Malika Lazreg & Nacéra Benamrane

To cite this article: Malika Lazreg & Nacéra Benamrane (2019) Intelligent System for Robotic Navigation Using ANFIS and ACOr, Applied Artificial Intelligence, 33:5, 399-419, DOI: 10.1080/08839514.2019.1577012

To link to this article: https://doi.org/10.1080/08839514.2019.1577012

Published online: 01 Mar 2019.

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 5, 399–419
https://doi.org/10.1080/08839514.2019.1577012

Intelligent System for Robotic Navigation Using ANFIS and ACOr

Malika Lazreg[a,b] and Nacéra Benamrane[b]

[a]Institut de Maintenance et de Sécurité Industrielle, Département de Génie Industriel, Univ Oran 2 Mohammed Ben Ahmed, Oran, Algérie; [b]Laboratoire SIMPA Département d'Informatique, Faculté des Mathématiques et Informatique, Université des sciences et de la technologie Oran « Mohamed BOUDIAF », Oran, Algérie

ABSTRACT

In this article we propose an intelligent system for mobile robot navigation in different environments, using ANFIS and ACOr. This system is capable of ensuring to mobile robot to navigate by reacting to the various situations encountered in different environments. In a first step, we use the ANFIS controller (Adaptive network-based fuzzy inference system) in which the contribution of the fuzzy logic of TAKAJI-SUGENO is added to that of the neural networks in a suitable way. In the second step, the ant colony method in a continuous environment ACOr (Ant colony optimization for continuous domains) is grafted into the second layer of the ANFIS network for hybridization. Simulations of the movements of the robot and the graphic interfaces are realized under the C ++ language.

Introduction

In mobile robotics, the importance is given to the optimization of the robot's trajectory towards its target in an unstructured environment. In planning his trajectory the robot must take into account its current position and the various obstacles that may hinder it during its course. The algorithm to complete his trajectory perfectly must be able to give a robot:

- The information about his location and environment.
- The best reaction in the situation encountered, to avoid the obstacles thanks to the sensors embedded on its front structure.
- To plan the best path that guides him to his goal.

Several effective techniques have been developed by researchers in the motion planning of mobile robots. (Kundu, Parhi, and Deepak 2012) Propose hybrid fuzzy controller with a multi-layered neural network that

CONTACT Malika Lazreg malikalazreg@hotmail.com; malikalazreg@univ-usto.dz Institut de maintenance et sécurité industrielle, Univ Oran, Ahmed Ben Ahmed

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaal.

allows the robot to adopt changes by moving towards its target. In this paper, three types of membership functions (Trapezoidal, Triangular and Gaussian) are hybridized in a single controller for each input and output variables. (Mon 2013) Use a camera, instead of sensors. The pixels of the image from VGA camera can be processed to get edge detection data. The horizontal edge numbers (HEN) and vertical edge numbers (VEN) are feed into controller ANFIS to train the fuzzy rules to control the right and left wheels of the mobile robot to avoid obstacles. (Castillo et al. 2013), Present a new approach in ant colony optimization (ACO). ACO algorithm is highly dependent on its parameters that have an effect on its convergence. One variant proposed approach is a convergence fuzzy logic controller with the objective of maintaining diversity at some level to avoid premature convergence. The optimization of membership functions for mobile robot trajectory control is presented with the proposed method. (Li, Rong, and Li 2014) Present AKELM (An Improved Kernel-Based Extreme Learning Machine), to predict internal failures of robots. In the learning, it is necessary to find the optimal values by the PSO algorithm. The algorithm provides a solution for the cameras calibration, in the space and robot localization at the same time. (Ansari et al. 2015) Suggest an analysis to aims to minimize the cost from source to destination. This algorithm is a hybrid of backtracking and a new technique (modified 8-neighbor approach); with reducing the number of iterations. The cost of each node reached is equivalent to the number of iterations and it is stored. This optimized hybrid approach requires shortest path computation. Malathi, Ramar, and Paramasivam (2016) Propose, the watershed algorithm (WSA) where the value of each pixel in the digital image of the basin represents an elevation at this point. These points represent rigid lines which will path lines for mobile robot movement. This algorithm is enriched along with particle swarm optimization (PSO).

Related Works

In recent decades, a lot of work has been proposed in Mobile robotics, namely intelligent controllers. Our approach aims at designing a controller whose algorithm is the hybridization of the ANFIS method and the ACOr method. Where the algorithm ANFIS is a contribution of the fuzzy logic of TAKAJI-SUGENO added to that of the neural networks in a suitable way; and the method, ACOr reproduced from the social behavior of ants in the search for food, has inspired researchers in the work of optimizing solutions. Several researchers propose simple and hybrid methods in order to optimize the results related to obstacle avoidance navigation in any environment. (Vaidhehi 2014) Chooses the ANFIS system with grid partitioning. With a limited number of membership functions, it operates with precision and obtains low error values. It is suitable that the inputs are less than 6. The selection of data sets for ANFIS training is important to its performance. (Mohanty, Dayal, and Parhi 2014) Develop two adaptive fuzzy controllers whose the inputs are front obstacle distance, left obstacle distance, right obstacle distance from the robot, and angle robot target. The two outputs correspond to the two wheels velocities of the mobile robot. This algorithm used performed better results for mobile robot navigation (Charabaruk, Manning, and Nokleby 2014). In this article the Max-Min ACO algorithm, with modifications of the heuristic factor is based on to the goal to be achieved. The ant must strengthen the shortest path by the pheromone deposit once borrowed. The amount of pheromone will not be allowed to decrease below a minimum level, and then the algorithm ACO Max-Min performs iterations for another ant population and will continue to do so until an optimal solution is reached. (Al-Mayyahi, Wang, and Birch 2014) suggest four controllers which will be combined through a switch block for choosing which controller will be activated. The vehicle navigates in an environment where the obstacles have different sizes, different shapes, and different number to make a more complex environment. The obstacle avoidance ANFIS controllers are activated frequently in order to avoid obstacles. After each avoiding, the switch active the ANFIS controller, to reach the target.

(Parhi and Mohanty 2016) Prove that by developing a controller with hybridization of the two algorithms, namely invasive weed optimization (IWO) used for training the premise parameters and the least square estimation (LSE) used for training the consequent part of the adaptive neuro-fuzzy inference system (ANFIS). They ensure the navigation of the robot in any complex environments. (Cao 2016) Develops an improved algorithm of artificial ant colonies for the planning the robot path. The goal is to dynamically adjust the evaporation rate of pheromones to improve global search capability and convergence speed, and to change the heuristic function to improve the state transition probabilities in order to find the optimal solution as quickly as possible. And finally, change the pheromone update strategy by reinforcing it on the optimal path and limiting its level. (Han, Park, and Seol (2016)) Present the ACOIC (Ant Colony Optimization with Critical Obstacle Influence) method, which uses the influence values propagated by critical obstacles such as initial pheromones, and the initial transition probabilities in ACO. This approach enhances traditional ACO by guiding ants towards the optimal path without wandering unnecessarily. Jianjun Ni and al (2016) Propose a controller based on the bio-inspired intelligent algorithm (BIA) to solve the problems of planning the trajectories of mobile robots. The central idea of this controller consists of a collection of vertebral fields responsible for generating behaviors according to their activation levels. This method improves the intelligence and autonomy of the mobile robot.

(Yang et al. 2017) Taking advantage of ACOR, they propose an AM-ACO, for multimodal optimization. Furthermore, to accelerate the convergence speed, a basic DE mutation operator is incorporated into AM-ACO. An adaptive local search technique is further absorbed into the algorithm, to enhance exploitation. At last, a random-based niche size setting strategy is developed for AM-ACO to deal with the dilemma that the niche size is a problem. (Lee 2017) Attempt to develop a global path planner that can directly find an optimal and smoother path without post-processing. They propose a heterogeneous-ants-based path planner (HAB-PP). The main objective of the GPP approach is to find a feasible and optimal path to the goal. The HAB-PP create by modifying and optimizing the global path planning procedure from the ant colony optimization (ACO) with three changes: modified transition probability function for moving ants, modified pheromone update rule, and heterogeneous ants. (Deepa and Senthilkumar 2016) they tell works on the evolution of swarm intelligence from Natural to Artificial Systems; particularly Ant Colony Optimization ACO. He exposes methods and shows the advantages and disadvantages when applying the ACO algorithm.

Robot System

We opted for a circular platform of the robot, allowing it to turn on itself without hitting the obstacles. The robot has a displacement device consisting of two driving wheels independent and a free wheel for balancing. Two infrared sensors are embedded on the lateral sides of the robot, and another on its front part so as to extract all the information on the scanned to ensure the system a perception covering the front half plane. Figure 1. In its

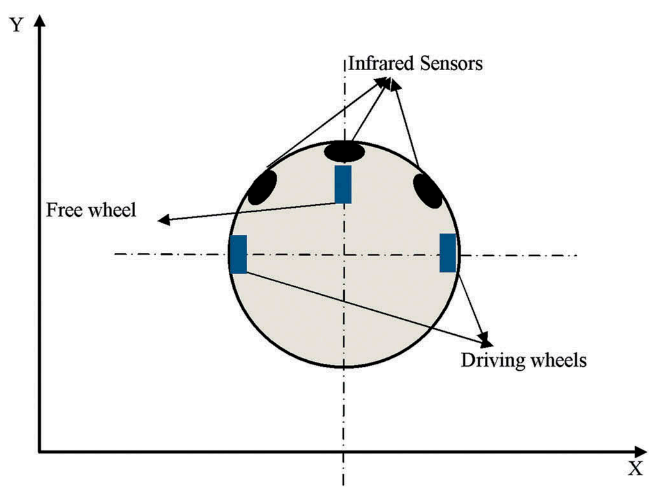

Figure 1. Mobile robot representation.

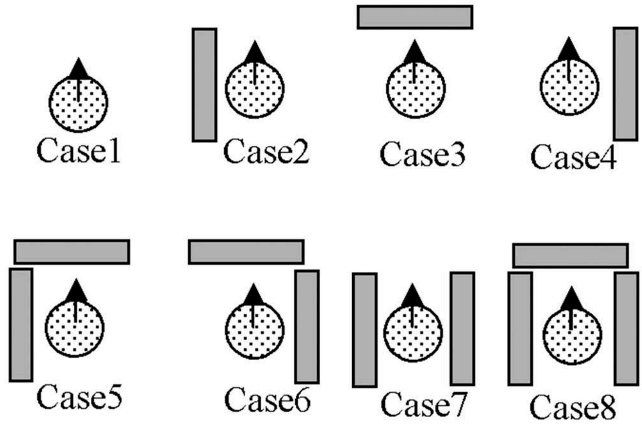

Figure 2. Eight basic situations.

navigation to reach its goal, the robot can be confronted with eight situations where obstacles can occupy the positions represented by Figure 2 (Lazreg, Meghdir, and Kies 2014).

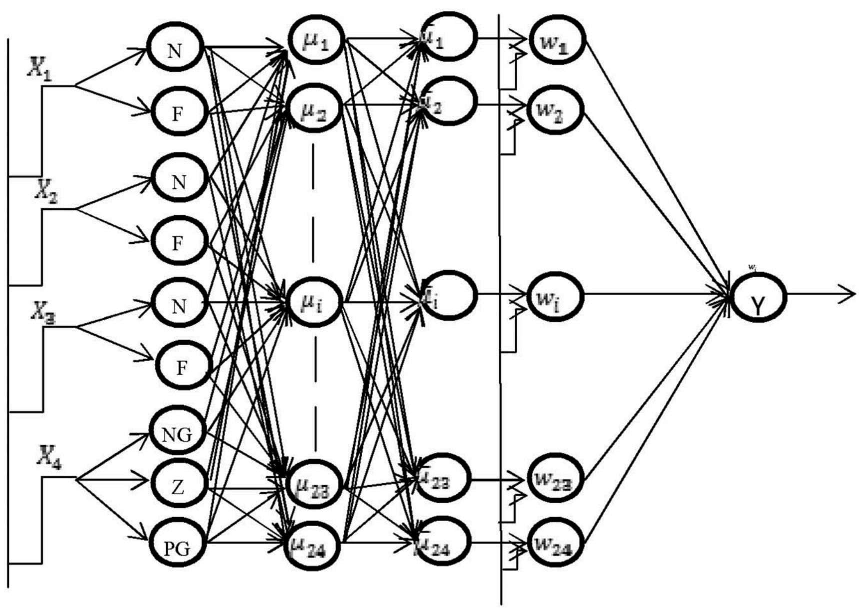

Figure 3. ANFIS Architecture.

Proposed Methods

ANFIS Controller

ANFIS is one of the hybrid controllers whose neural structure is composed of five layers each of which is a step of the fuzzy system of SUGENO-TAKAJI, see Figure 3. In our case, the inputs are the information collected by the three sensors on board the robot and which indicate to him the different distances compared to the possible obstacles which can be on his trajectory, and the angle of orientation which informs him about his situation in relation to its purpose.

The different steps of the ANFIS controller are shown in "Figure 3." where the five stages of the controller represented by the layers of the network are detailed by the following:

-**First layer**: this adaptive layer (fuzzification layer) contains nine neurons that transform the digital data of the inputs measured by the sensors into linguistic interpretations. Each neuron calculates its activations which are equal to the degrees of membership of the inputs (x1, x2, x3, x4) in the fuzzy subsets represented by Gaussian functions described by:

$$\mu_n = exp\left(\frac{-0.5(x-c)^2}{\sigma^2}\right) \tag{1}$$

where c is the center of the Gaussian and σ the standard deviation

–**Second layer**: This layer called rule layer, where each neuron corresponds to a fuzzy rule. This layer calculate, respectively, their activation by a simple product to give the value of truth

$$\mu_i = \mu_{1j}(x_1).\mu_{2j}(x_2).\ \mu_{3j}(x_3).\ \mu_{4l}(x_4) \tag{2}$$

As j = near or far, and l = NG (large negative) or Z (zero) or PG (large positive)

This represents the degree of truth of the i^{th} rule with i = 1 … .k the number of fuzzy rules (24 for our case) according to the method of Sugeno-Takagi.

-**Third layer**: This is the normalization layer. It calculates the effective weight of each rule by calculating its probability by:

$$\bar{\mu}_i = \frac{\mu_i}{\sum_{i=1}^{k} \mu_i} \tag{3}$$

-**Fourth layer**: Adaptive layer or defuzzification layer. The role of this step is important since it must evaluate the values of the consistent weights of the given rules to obtain optimized results for each rule. Each node of this layer receives from the layer preceding the corresponding normalized value and the initial entries, and the defuzzification is given by:

$$w_i = \bar{\mu}_i.f_i = \bar{\mu}_i[q_{i1}(x_1) + q_{i2}(x_2) + q_{i3}(x_3) + q_{i4}(x_4) + q_{i5}] \tag{4}$$

With q_{in} $n = 1 \ldots 5$ all the consequent parameters that contribute to give the desired position to the robot to reach its target.

-**Fifth layer**: This layer is representing by a single (Node) denoted by Σ, and determine the variable corresponding to the steering angle of the robot

$$y = \sum_{i=1}^{k} \bar{\mu}_i . f_i = \frac{\sum_{i=1}^{k} \mu_i . f_i}{\sum_{i=1}^{k} \mu_i} \tag{5}$$

-**Update**: the learning combines the least squares method and the method of backpropagation. The consequent parameters q_{ij} are computed as follows:

$$Y_d = A.q_{ij} \text{ such as } Y_d = \sum_{j=1}^{k} y_{dj} \tag{6}$$

With Y_d the vector of the desired values of the steering angles y_{dj} with j = 1 … k, and k the number of solutions or steering angles, A is the matrix composed of the elements of the ANFIS. The least square method calculates the consequent parameters by:

$$q_{ij} = \left(A^T.A\right)^{-1}.Y_d.A^T \tag{7}$$

Once the consequent parameters q_{ij}obtained, and the computation corresponding to the steering angles valued we calculate the error

$$E_p = \sum_{j=1}^{l} \left(y_{dj} - y_j\right)^2 \tag{8}$$

The back propagation learning procedure is the derivation from the output layer and backtracking layer by layer until the input layer is reached:

$$\frac{\partial E_p}{\partial y_j} = -2\left(y_{dj} - y_j\right) \tag{9}$$

Continuing layer-by-layer back propagation, results in:

$$\frac{\partial E_p}{\partial \alpha} = \frac{\partial E_p}{\partial y_j} * \frac{\partial y_{y_j}}{\partial w_i} * \ldots\ldots * \frac{\partial \mu_n}{\partial \alpha} \tag{10}$$

With $\alpha = c \ et \ \alpha = \sigma$

So the update of these two variables is expressed by:

$$\Delta\alpha = -\eta \frac{\partial E_p}{\partial \alpha} \tag{11}$$

Where η is a learning rate.

To obtain the values that served to the ANFIS controller to allow the robot to move from its initial position to its target avoiding all obstacles and taking the optimal path are summarized in Table 1.

Table 1. Specifications of ANFIS parameters.

PARAMETERS	VALUES
Rules	24
Near	6
Far	24
Angle Robot target	−60° 0 + 60°
Negative Great, Zero, Positive Great	
NG Z PG	
Iterations	1000
Number of points	100

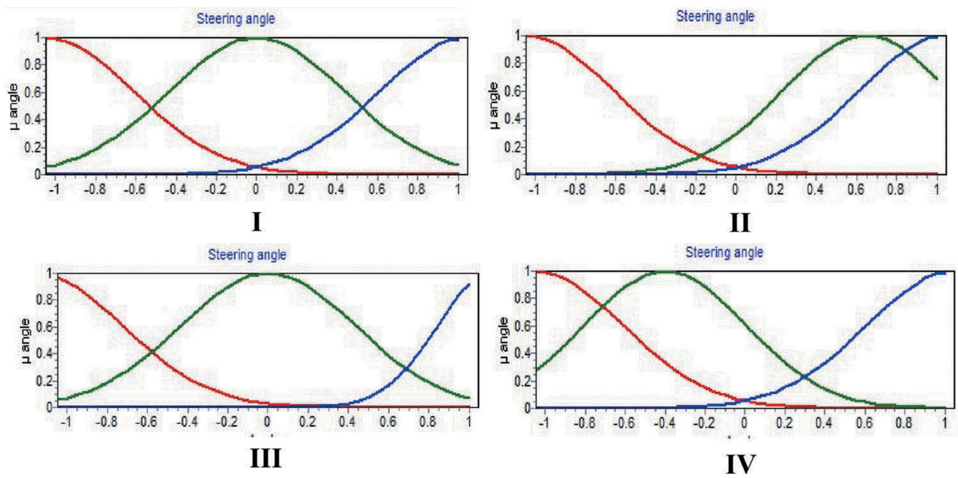

Figure 4. I Input membership function, -II, III, IV: Output membership functions after 1000 iterations of points 56, 70, 91.

The following figures in Figure 4 represent the membership functions characterizing the orientation angle with respect to the target. Figure 4. I show the Gaussian representation of the target robot angle before updating. Figures 4. II, III, IV shows the tuning of centers and standard deviations of the Gaussians describing the target robot orientation angle after 1000 iterations.

The simulations are performed in different environments with different numbers, and different forms of obstacles to evaluate the reaction of the robot. At every moment we could determine the coordinates of its position in the chosen environment, and the angle of orientation in relation to its purpose, and its reactions to avoid. Reach the target is indicated by a message read on the interface, and the distance traveled is then indicated. For the simulations we have chosen eight environments, see Figure 5 in which we have either used obstacles of the same nature and of the same size, either non-similar obstacles or environments describing different labyrinths.

The following figures of Figure 5 represent the motion of the robot in different environments.

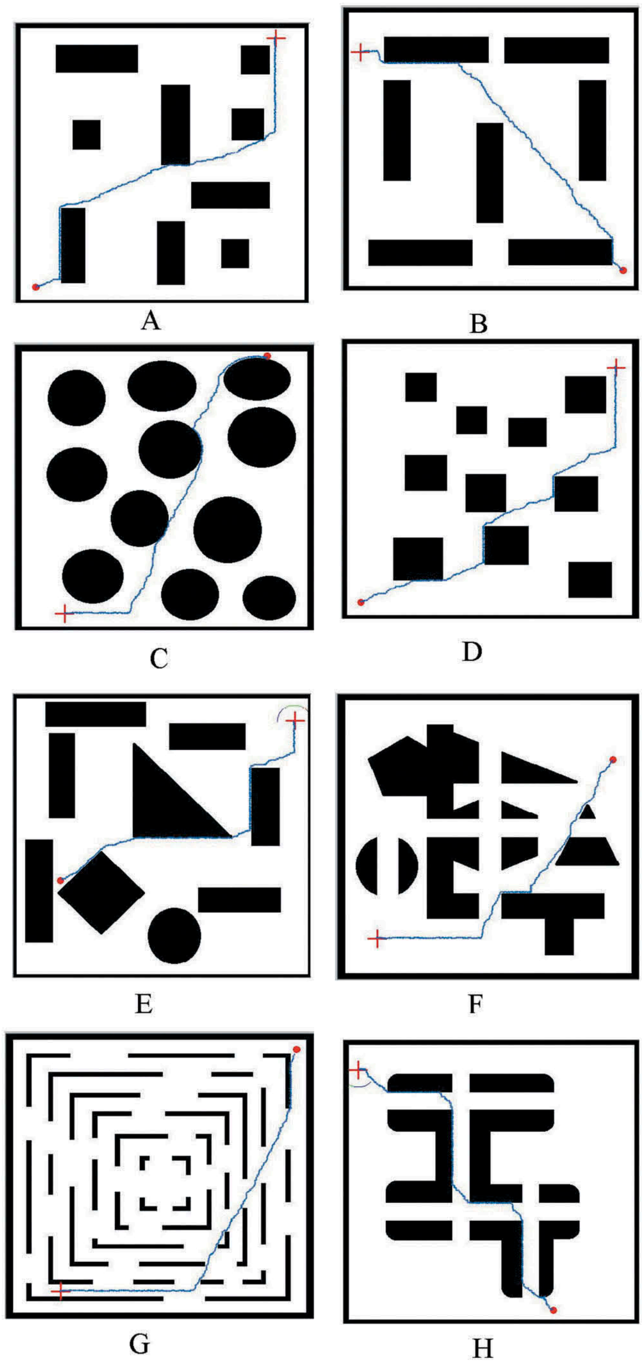

Figure 5. Simulations in different environments by ANFIS controller.

Ant Colony Optimization

The meta-heuristic Ant colony optimization consists of a colony of artificial ants with the characteristics to search good solutions to discrete optimization problems. Path planning for mobile robot represents a large part of the field of application of the ACO algorithm. In real life, many problems cannot easily be represented in a discrete space. The biological inspiration for ACO is not set in a discrete space, but rather a continuous one. The approach proposed by (Socha and Dorigo 2008) is Ant Colony Optimization for continuous domains ACOr which uses a continuous probability distribution, specifically a Gaussian probability density function. The probability with which the candidate component will be chosen in the following steps is evaluated by quantities of pheromone reported. A list is drawing up to keep only the best candidates to store in the solution called solution archive.

The Gaussian that represents the probability of distribution is expressed by:

$$g_i = \frac{1}{\sigma^i \sqrt{2\pi}} \exp\left(\frac{x - \mu^i}{2\sigma^i}\right)^2 \tag{13}$$

Where μ^i the vector of Gaussian centers, and σ^i is the vector of standard deviations.

Hybrid Controller ANFIS/ACOr

In this section we present a proposed approach based on hybridization of the Adaptive Network-based Fuzzy Inference System (ANFIS) method and ant colony optimization for continuous spaces method (ACOr), where all stages are represented by the organizational chart detailed by Figure 6.

The hybrid controller that we will develop has the same architecture as the ANFIS controller. Except that from the fuzzification layer which is identical, we graft in the following layers the algorithm of the ant colonies for the continuous domains, while respecting the nature of the architecture.

- **First layer**: The first layer is an adaptive layer (fuzzification layer), that transform the digital data of the inputs into linguistic interpretations. It has nine neurons that can be interpreted as nests from which a number of ants disperse for foraging. Each neuron calculates its activations which are equal to the degrees of membership of the inputs.

-**Second layer**: This second layer is called compared with the layer in the ANFIS controller the sub-colonies layer. Each neuron receives a subgroup of four ants from the first layer and distributed as follows (Liu, Dai, and Gao 2014):

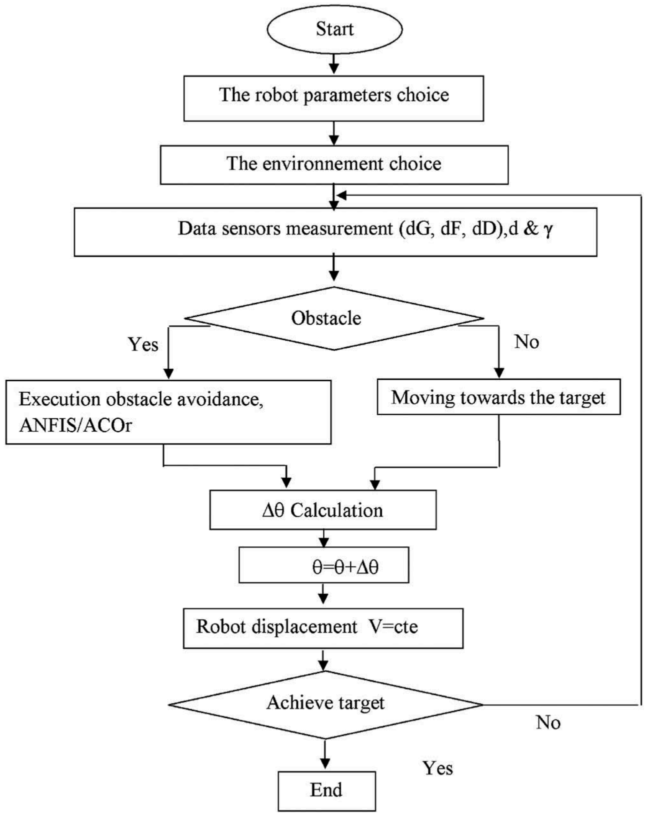

Figure 6. Proposed system robotic navigation.

$$\begin{bmatrix} \text{Layer N}^\circ 2 \\ \mu_{11}\ \mu_{12}\ \mu_{13}\ \ \mu_{14} \\ \mu_{21}\ \mu_{22}\ \mu_{23}\ \mu_{24} \\ \vdots \\ \mu_{24,1}\ \mu_{24,2}\ \mu_{24,3}\ \ \mu_{24,4} \end{bmatrix} \tag{14}$$

Who will be evaluated to elect the best subgroup considered as a parent by the expression

$$eval(i) = \frac{1}{1 + \exp\left[\frac{f(\mu_{i1}\ \mu_{i2}\ \mu_{i3}\ \ \mu_{i4})}{T}\right]} \tag{15}$$

Where: $f(\mu_{i1}\ \mu_{i2}\ \mu_{i3}\ \ \mu_{i4}) = \mu_{i1}\ * \mu_{i2}\ * \mu_{i3}\ * \mu_{i4}$ is the value of the function of the subgroup i in ANFIS and T a positive coefficient to adjust the intensity of the selection.

Third layer: To respect the structure of the ANFIS controller, this is a normalized layer; it calculates the probability of selection of each subgroup by the following formula:

$$p_i = \frac{eval(i)}{\sum_{j=1}^{k} eval(j)} \tag{16}$$

- **Fourth layer**: This layer is an adaptive layer each neuron is connected to the corresponding normalized neuron of the previous layer, and also receives generated subgroups C, as indicated by the following formula:

$$s = \begin{cases} argmax\ (eval(i))_{j=1..24} & for\ q \leq q_0 \\ C & for\ q \geq q_0 \end{cases} \tag{17}$$

where q_0 is a parameter given for $0 \leq\ q_0 \leq 1$;

q Is a random variable distributed uniformly in the interval $[0\ 1]$.

C is a random variable generated according to the selection formula of layer four.

-**Fifth layer**: The only neuron in this layer receives the sum of the solutions of the previous layer

$$\mathrm{S} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{s_i} \tag{18}$$

-**Update**: In ACOr, there are no apparent pheromone representation and updating strategies. The weight of a solution decreases exponentially with its rank (Liao et al. 2014) and, this weight determines the probability of the solution chosen by ants. Thus, the weight operates as the pheromone. (Yang et al. 2017). In the ACOr algorithm, the solutions are pheromone distributions expressed by Gaussian probability density functions. ACOr initializes the solution archive with *k* solutions that are generated uniformly at random, keeps track and stocks them in the solution called archive solution (Socha and Dorigo 2008). The best solutions receive the highest weights. The k solutions of the archive are kept sorted according to their quality (from best to worst).

For each solution S_j of a problem with n dimensions ACOr stores the n variables and the

Objective function f (S_j) in T as shown below (16)

$$\begin{bmatrix} s_1^1 & s_1^2 & & s_1^{n-1} & s_1^n \\ s_2^1 & s_2^2 & \cdots & s_2^{n-1} & s_2^n \\ \vdots & & s_j^i & & \vdots \\ s_{m-1}^1 & s_{m-1}^2 & & s_{m-1}^{n-1} & s_{m-1}^n \\ s_m^1 & s_m^2 & \cdots & s_m^{n-1} & s_m^n \end{bmatrix} \begin{bmatrix} f(S_1) \\ f(S_2) \\ \vdots \\ f(S_{m-1}) \\ f(S_m) \end{bmatrix} \tag{19}$$

The ultimate goal is to look for the best solutions. For a solution S_j, the ant chooses the j value of the vector $(s_j^1,\ s_j^2, \ldots,\ s_{j,}^{n-1}\ s_j^n)$ with, j = 1...m. And therefore a Gaussian distribution in its perception radius represented by the pheromone rate given by the following expression:

$$g_j^i(s) = \frac{1}{\sigma_j^i\sqrt{2\pi}} exp - \frac{0.5 * \left(s - \mu_j^i\right)^2}{\left(\sigma_j^i\right)^2} \text{ Knowing that } s_j^i = \mu_j^i \tag{20}$$

Each ant chooses a direction in the search space at each step of the construction process; by performing n steps of solutionS_j.

At each step of the construction of the solution, the Gaussian functions representing the value of the pheromone deposited by the ant j differs by the variation of the standard deviation between the solutions of the archive and that of the ant j the value is:

$$\sigma_j^i = \xi \sum_{r=1}^{k} \frac{\left(s_r^i - s_j^i\right)}{k-1} \text{ Avec } \xi > 0 \tag{21}$$

The ant performs well in the n stages of the construction of the solution; by storing in the archive the n solutions represented by the Gaussians form. A weighted sum of several one-dimensional Gaussian functions will form the Gaussian kernel that will guide the ants in their search.

This kernel is defined by the following formula:

$$G^i(x) = \sum_{j=1}^{m} \omega_j g_j^i(x) = \sum_{j=1}^{m} \omega_j \frac{1}{\sigma_j^i\sqrt{2\pi}} \exp\left(-\frac{x - \mu_j^i}{2\sigma_j^i}\right)^2 \tag{22}$$

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} \exp\left(-\frac{(rang(j) - 1)}{2q^2k^2}\right)^2 \tag{23}$$

With the probability of choosing the jth Gaussian given by:

$$p_j = \frac{\omega_j}{\sum_{r=0}^{k} \omega_r} \tag{24}$$

qk Represent a standard deflection and q is algorithm parameters. The influence of this parameter on ACOr is similar to adjusting the balance between the best iterations and the best pheromone updates so far used in ACO. At the beginning of the algorithm, the archive solution is initiated by generating solutions with uniform random samples. The pheromone Update is accomplished by adding new solutions to the archive solution while removing the worst ones.

The values used for the algorithm are shown in Table 2.

The representations in Figure 7 Shows the update cycles. The black color curves plot the weighted sum of several one-dimensional Gaussian functions in the

Table 2. Specifications of ANFIS/ACOr parameters.

PARAMETERS	VALUES
Number of subgroups	24
Number of ants per subgroups	4
Number of solutions in archive	24
q	0.5
adjustment coefficient T	2.5
Number of cycles	100

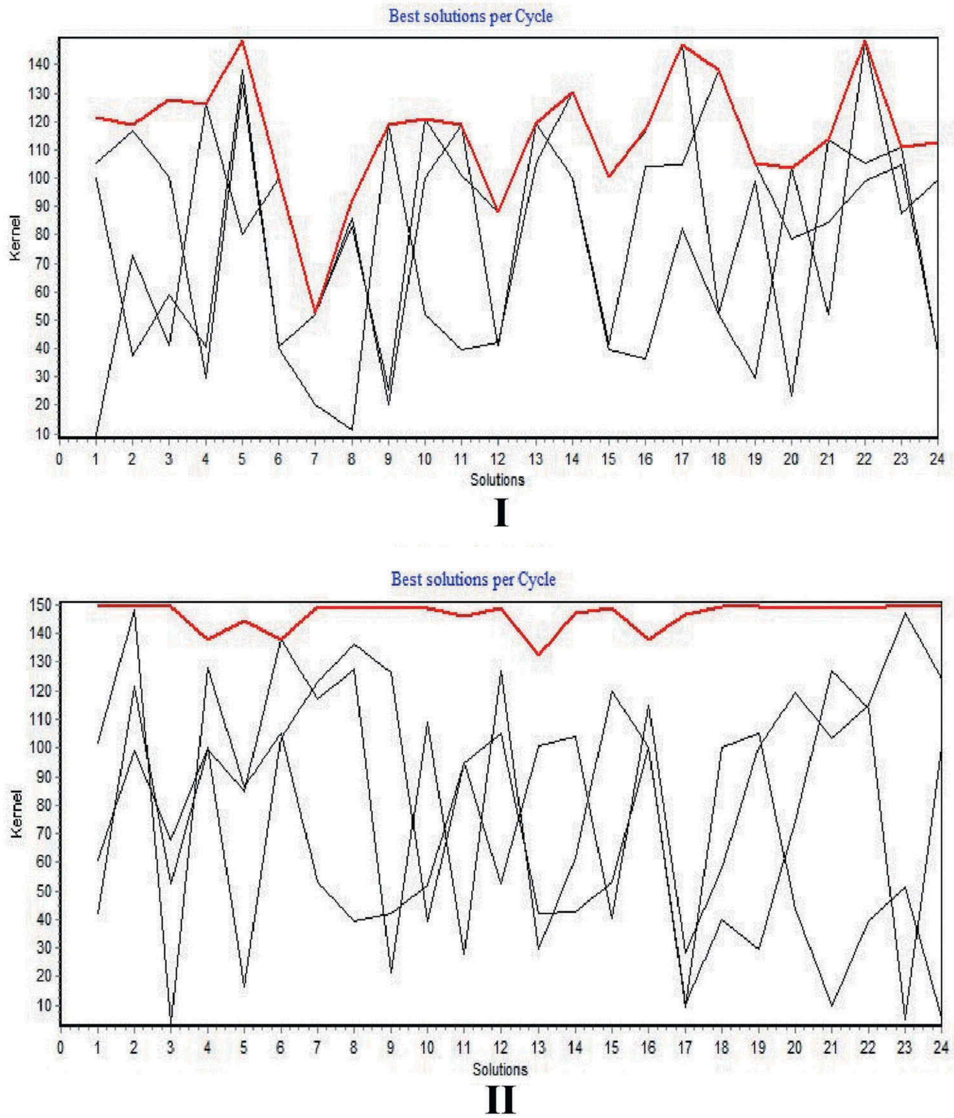

Figure 7. I Representation of solutions with five cycles; II Representation of solutions with one hundred cycles.

archive solution. The solutions are ordered in the archives according to their quality, from the worst to the best. The Gaussian kernel PDF (Probability Density Function) G^i is built using only the ith coordinates of all k solutions in the archive. The red color curve represents the best solution at the end of the cycles.

In Figure 7.II we draw a hundred diagrams representing a hundred cycles but to avoid cluttering we just let appear the four firsts in black and the hundredth in red.

The simulation with the hybrid controller ANFIS/ACOr is done with the same environments and the same coordinates of the target and the robot as those applied by the ANFIS controller. Figure 8 represents the motion of the robot in different environments with method ANFIS/ACOr

Results and Discussions

The coordinates in Table 3 are used to simulate distances traveled by the robot in the eight environments to reach its goal in Figures 5 and 8. calculated by the two methods.

The mobile robot is put into action in eight environments that we have developed to test its capabilities when it is submitted to the ANFIS controller and the ANFIS/ACOr hybrid controller. The simulations of the two methods applied display the distances traveled by the robot. During navigation, the robot reacts to obstacles by avoiding them in changing direction.

For that he consults his database developed by the calculations of ANFIS method and hybrid (ANFIS/ACOr) method, and chooses the value of the steering angle corresponding to the desired deflection; to continue the trajectory who leads him to his goal. After the tests of the table above we made a comparison to show the efficiency of each method. We chose an environment; we placed the robot and the target in positions defined by their coordinates. The

robot thus subjected to both methods crosses the same distances in the same direction and the same direction to reach the target. Then, we switched the positions of the robot and the target which changes the direction of the robot's navigation.

The comparison is made on the length of the distance crossed and the elapsed time that the robot is supposed to be made. This is shown by the realization of a graphical interface on which are displayed the coordinates and the distance in pixels as indicated in Figure 9.

Among the studies done on the planning of the trajectories of the mobile robots with the ANFIS controller we chose an environment of the article (Mohanty, Dayal, and Parhi 2014) and applied the results of our ANFIS controller and our hybrid ANFIS/ACOr controller for comparison. We notice in the same way that the hybrid controller is more robust as shown in Figure 10.

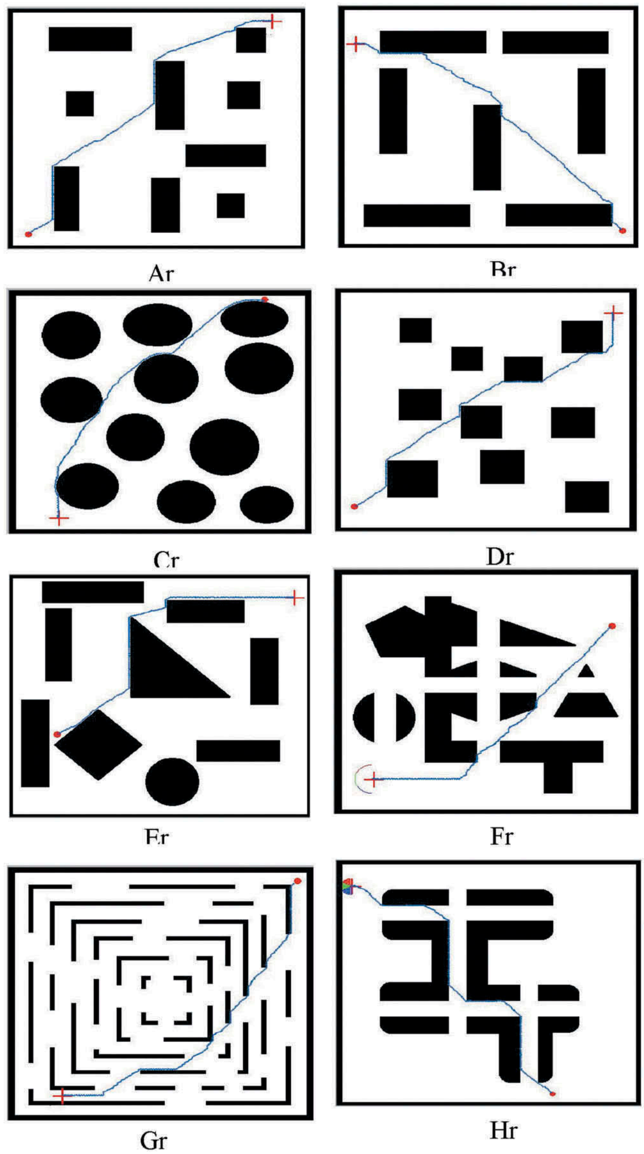

Figure 8. Simulations in different environments by ANFIS/ACOr controller.

From the results of the simulation, we can say that both methods ensure the robot to achieve its goal by avoiding obstacles encountered during navigation. However, we note that for the same distance traveled, the ANFIS/ACOr hybrid controller provides the robot with better results. The robot is faster than when it is submitted to the ANFIS controller. We think that the purpose of the hybrid algorithm is to plot and calculate the shortest path between the robot and its target (in reference to the ants that start from the nest towards the place of food, taking the shortest path). The

Table 3. Comparative simulations of method ANFIS and ANFIS/ACOr.

Environment	Taget coodinates	Robot coodinates	Distance(pixels) traveled:ANFIS	Distance(pixels) traveled: ANFIS/ACOr
Env_1	Xc = 444	Xr = 34	815	446
	Yc = 26	Yr = 469		
Env_2	Xc = 28	Xr = 474	447	375
	Yc = 78	Yr = 463		
Env_3	Xc = 82	Xr = 422	487	411
	Yc = 466	Yr = 19		
Env_4	Xc = 461	Xr = 31	707	402
	Yc = 50	Yr = 448		
Env_5	Xc = 474	Xr = 79	586	470
	Yc = 46	Yr = 328		
Env_6	Xc = 64	Xr = 457	463	305
	Yc = 427	Yr = 114		
Env_7	Xc = 88	Xr = 472	471	356
	Yc = 450	Yr = 27		
Env_8	Xc = 24	Xr = 355	638	440
	Yc = 52	Yr = 475		

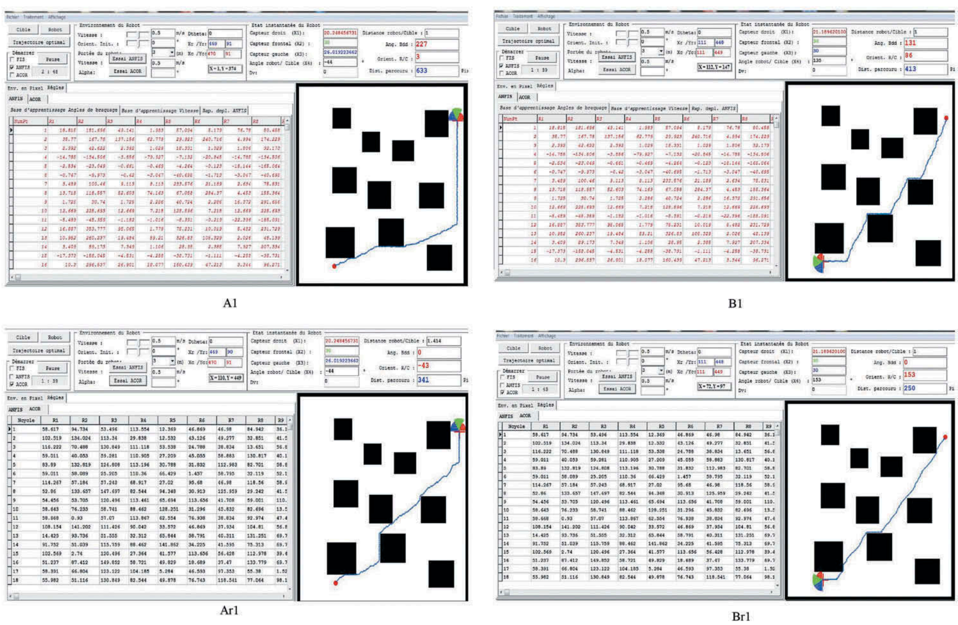

Figure 9. Same distance traveled with switched coordinates.

construction of the appropriate solution is a dynamically created different probability distributions. This distribution depends on the previous construction steps.

The variables of the solution are stored to form archive solution. Each iteration, this protocol continues to update this distribution of probabilities, keeping the best solutions and eliminating the worst ones, while respecting the number of solutions. The solutions are ordered in the archives according

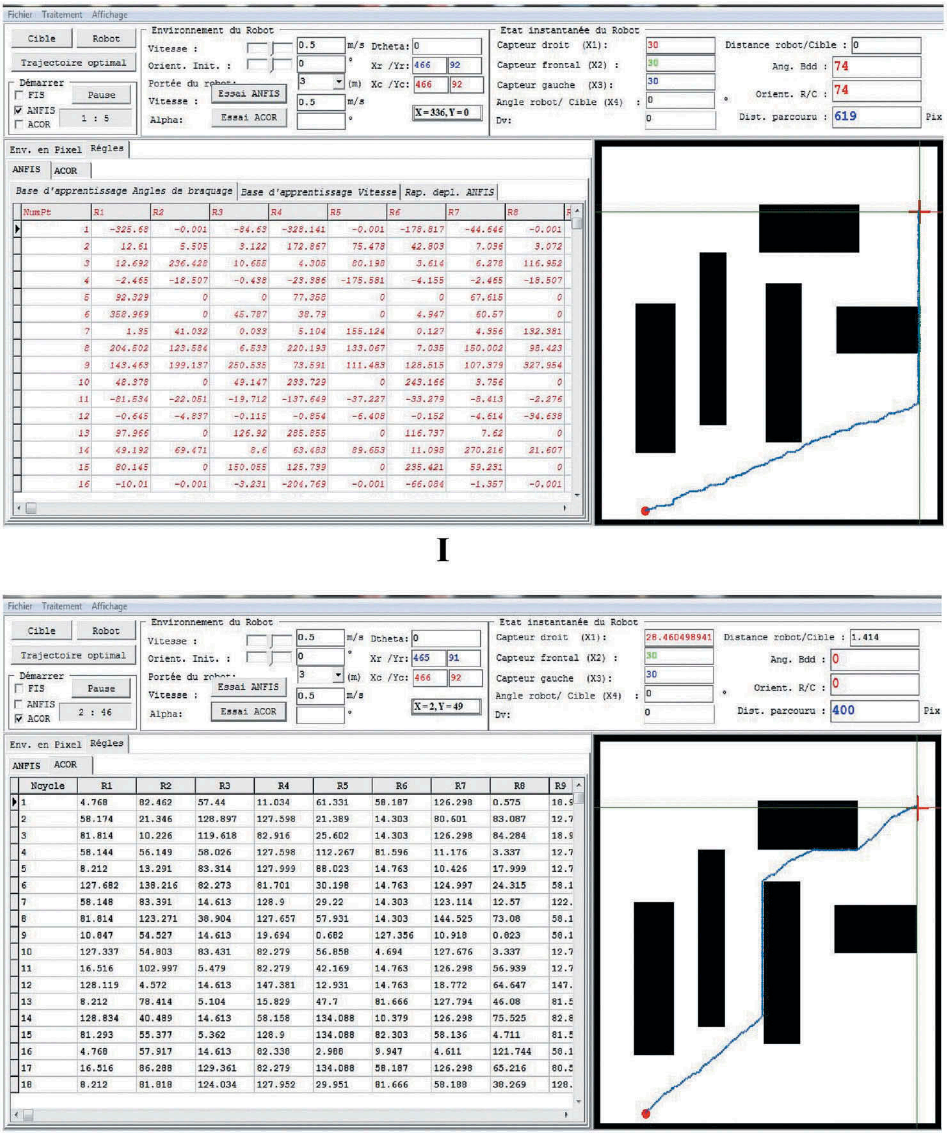

Figure 10. I distance traveled by using ANFIS controller, II distance traveled by ANFIS/ACOr controller.

to their quality, from the worst to the best (Figure 6). This way of updating increases the robustness of the hybrid algorithm, to build the shortest path.

On the other hand, the ANFIS controller adopts the algorithm to optimize a path between the robot and the target. After estimating the consequent least squares parameters that determine the results, the algorithm adopts the learning technique used by the back-propagation, to reduce the error, between the calculated results and the desired results, and to make it as small as possible and, if necessary, within the required limits. At each iteration a new independent solution without taking into account the previous ones is generated; unlike the hybrid algorithm. This process does not eliminate the bad solutions, which can lead to a decrease in robustness compared to ANFIS/ACOr.

Conclusion

In this article we presented two methods of artificial intelligence, to test them on a mobile robot. Both methods must allow the robot to react to possible obstacles and avoid them; therefore a reactive navigation strategy. Reactive navigation uses reflex actions to guide the robot based on some type of perception to activate the action. It is, therefore, a sequence of actions associated with each of the places crossed and thus defines a path that allows reaching the goal. The ANFIS controller is defined by a five-layer neural structure. Each neuron is represented by a Gaussian distribution describing the different steps of the TAKAGI-SUGENO fuzzy logic method. In the ANFIS controller algorithm updating is defined by the combination of the least squares method and that of the back-propagation to improve the value of the parameters (centers, standard deviations) of the Gaussian curves. The strategy of the adjustment of the parameters of the Gaussian curves (centers and standard deviations) is to refine the value of the error to be the calculated results and the desired results. By comparison, the hybrid ANFIS/ACOr controller presented in our work is designed with the same neural structure and the same Gaussian distribution dedicated to each neuron but whose network steps are described by the ACO for the continuous domains algorithm proposed by (Socha and Dorigo 2008), while respecting the nature of each layer.

The pheromones update is done by adding all newly generated solutions to the solution archive, then removing the same number of the worst solutions, so that the total size of the archive does not change. This process ensures that only the best solutions are kept in the archive so that they effectively guide ants through the search process. The components of the solutions are used directly to modify the pheromone table, in the case they are used to dynamically generate PDF (Probability Density Functions). The solutions in the archive are used to calculate parameter values (centers and standard deviations of Gaussian functions and their weight), and thus form the Gaussian kernel PDF who is composed of a number of regular Gaussian functions. This number is equal to the size of the solution archive. This Gaussian PDF is used to guide the ants in their search. There is a great similarity in the search for optimization by both methods, seeking to improve the values of the centers and standard deviations of the Gaussian functions.

The ANFIS and ANFIS/ACOr controllers both allow the robot to trace its course by reacting and avoiding the obstacles encountered whatever their size, shape or number. We can say that the simulations performed by the hybrid controller give better results, thanks to its robustness.

References

Al-Mayyahi, A., W. Wang, and P. Birch. 2014 Adaptive neuro-fuzzy technique for autonomous. *Ground Vehicle Navigation Robotics* 3:349–70. doi:10.3390/robotics3040349. ISSN 2218–6581 November 2014.

Ansari, A., M. A. Sayyed, K. Ratlamwala, and P. Shaikh. 2015 March. An optimized hybrid approach for path finding. *International Journal in Foundations of Computer Science & Technology (IJFCST)* 5 (2):47-58.

Cao, J. 2016. Robot global path planning based on an improved ant colony algorithm 4:11-19. *Journal of Computer and Communications*. Scientific Research Publishing Inc.

Castillo, O., H. Neyoy, J. Soria, M. García, and F. Valdez. 2013. Al [dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot]. *International Journal of Advanced Robotic Systems* 10:51. doi:10.5772/54883.

Charabaruk, N., M. Manning, and S. Nokleby. Dynamlc robot path planning using improved MAX/MIN ant colony optimization. Prosseding of the Internationnal conference of control, dynamic systems and robotics Ottawa. Ontario, paper No.49 Canada May 15–16 2014.

Deepa, O., and D. A. Senthilkumar. 2016 March. Swarm Intelligence From Natural To Artificial Systems Ant Colony Optimization, International journal on applications of graph theory in wireless. *Ad Hoc Networks and Sensor Networks(GRAPH-Hoc)* 8 (1):9-17.

Han, J., H. Park, and Y. H. Seo. 2016 September. Path planning for a mobile robot using ant colony optimization and the influence of critical obstacle. Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management, Detroit, Michigan, USA, September, 178–186.

Kundu, S., D. R. Parhi, and B. B. V. L. Deepak. 2012. Fuzzy-neuro based navigational strategy for mobile robot. *International Journal of Scientific & Engineering Research* 3 (6):1–6. ISSN 2229–5518.

Lazreg, M., A. Meghdir, and K. Kies. 2014. Implementation of fuzzy controller for navigationof a mobile robot in different environments. *Proceedings in Manufacturing Systems* 9 (3):151–56. 2067–9238 2014.

Lee, J. 2017. Automation and Systems, *International Journal of Control* 15 (4):1754–69. ISSN:1598-6446 eISSN:2005-4092 2017. doi:10.1007/s12555-016-0443-6.

Li, B., X. Rong, and Y. Li. 2014. An improved kernel based extreme learning machine for robot execution failures. *Hindawi Publishing Corporation Scientific World Journal* 2014:1–7. Article ID 906546:7.

Liao, T., M. A. Krzysztof Socha, M. de Oca, and T. Stutzle. 2014 August. Senior member, IEEE, and Marco Dorigo, Fellow, IEEE "Ant colony optimization for mixed-variable optimization problems. *IEEE Transactions on Evolutionary Computation* 18 (4). doi:10.1109/TEVC.2013.2281531.

Liu, L., Y. Dai, and J. Gao. 2014. Based on position distribution model of ant colony foraging ant colony optimization algorithm for continuous domains. *Hindawi Publishing Corporation the Scientific World Journal* 2014:1–9. Article ID 428539: 9.

Malathi, M., K. Ramar, and C. Paramasivam. 2016. Optimal path planning for mobile robots using particle swarm optimization and dijkstra algorithm with performance comparison). *Middle-East Journal of Scientific Research* 24 (S1):312–20. ISSN 1990–9233.

Mohanty, P. K., R. Dayal, and A. Parhi. 2014 September. New intelligent motion planning for mobile robot navigation using multiple adaptive neuro-fuzzy inference system. *Applied Mathematics & Information Sciences An International Journal* 8:2527–35. doi:10.12785/amis/080551.

Mon, Y.-J. 2013. Vision-based obstacle avoidance controller design for mobile robot by using single camera. *IJCSI International Journal of Computer Science* 10 (2):248–250. Issues, ISSN (Print): 1694–0814 | ISSN (Online).

Ni, J., L. Wu, X. Fan, and S. X. Yang. 2016. Bioinspired intelligent algorithm and its applications for mobile robot control, survey. *Computational Intelligence and Neuroscience Journal*:16. Article ID 3810903.

Parhi, D. R., and P. K. Mohanty. 2016 April. IWO-based adaptive neuro-fuzzy controller for mobile robot navigation in cluttered environments. *International Journal of Advanced Manufacturing Technology* 83:1607–25. doi: 10.1007/s00170-015-7512-5.

Socha, K., and M. Dorigo. 2008. Ant colony optimization for continuous domains. *European Journal of Operational Research* 185:1155–73. doi:10.1016/j.ejor.2006.06.046.

Vaidhehi, V. 2014. The role of dataset in training ANFIS system for course advisor. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 1 (6):249-253. ISSN: 2349–2163.

Yang, Q., W.-N. Chen, Z. Yu, G. Tianlong, L. Yun, H. Zhang, J. Zhang et al. 2017 April. Adaptive multimodal continuous ant colony optimization. *Transactions on Evolutionary Computation* 21 (2). doi:10.1109/TEVC.2016.2591064.