A Learning method of a Mobile Robot using ACO and a Fuzzy Navigator

Hamid Boubertakh[#1], Salim Labiod[#2], Mohamed Tadjine[*3], Pierre-Yves Glorennec[**4]

[#] *LAMEL, University of Jijel,*
BP. 98, Ouled Aissa, 18000, Jijel, Algeria.
[1]boubert_hamid@yahoo.com
[2]labiod_salim@yahoo.fr
**LCP, ENP d'Alger*
10 av. Hassen Badi, BP. 182, El Harrach, Algiers, Algeria.
[3]*tadjine@yahoo.fr*
[**] *INSA de Rennes,*
20 Av. des Buttes de Coësmes, 35700, Rennes, France.
[4]glorenne@irisa.fr

***Abstract*—This paper, proposes an ant colony optimization (ACO)-based learning method for a fuzzy navigator of a mobile robot evolving in unknown environment. The navigator is a collection of IF THEN rules translating the human reasoning. The robot decision is made using fuzzy logic. The learning method consists in the online searching of the best fuzzy rules conclusions when the robot executes its predefined task. Finally some simulation results are presented which show the performance of the proposed method.**

***Keywords*— Mobile robot, obstacle avoidance, Fuzzy logic, fuzzy systems, Ant colony Optimization (ACO).**

I. INTRODUCTION

Path planning is an important topic in robotics, mainly for its practical applications, and can be classified on two types; the global path planning and the local path planning. The first one is done in off-line manner, and since the path is designed, the user can decide about the control method to use to follow the path by the robot. Many methods are developed in the literature; A* algorithm [1], potential field method [2, 3], cell decomposition method [3] and recently, the ant colony optimization (ACO) methods [4, 5]. The second one also named obstacle avoidance or navigation is done in on-line reactive manner; it imposes that the robot must be equipped by sensors to have a vision of its neighbourhood, and after, takes the adequate action to achieve its a priori defined task. We can find in the literature a large number of publications dedicated to this purpose where several methods are used; the force field [6], the reinforcement learning [7], and the one which has known a great success in the last years is the fuzzy logic [1, 8-14]. The success of fuzzy logic in mobile robots navigation is due to its capability to represent the human reasoning and therefore the robot doesn't need an exact vision of its environment. Recently many research woks in vehicles navigation using fuzzy logic are published.

In the present work, we propose a new ACO-learning method for a fuzzy navigator of mobile robot in unknown environment. We use the fuzzy rule base proposed in [15] where both obstacles avoidance and goal seeking behaviours are merged

This paper is organized as follows. Section II describes the considered robot model, in section III; the fuzzy navigator is presented. The section IV presents the proposed ACO learning method. Section V presents the simulation results and finally, section VI, concludes the paper.

II. ROBOT MODEL

We use a differential drive cylindrical mobile robot model with a radius of 20 cm. The robot is equipped with 24 ultrasonic sensors evenly distributed in a ring as depicted in Fig.1 (a). Each sensor, Si for i=1,…,24, covers an angular view of 15° and gives the distance to the obstacle Li in its field of view. To reduce the number of inputs for the navigator, sensors in the front of the robot are arranged into tree sensor groups; the left group SL consists of the 6 neighbouring sensors Si (i=1,…,6), the face group SF consists of the 6 neighbouring sensors Si (i=7,…,12), and the right group SR consists of the 6 neighbouring sensors Si (i=13,…,18). The distances measured by the tree groups SL, SF and SR denoted respectively by dL, dF and dR are expressed as follows:

$$\begin{cases} dR = R + min_{i=1,\dots,6}(L_i) \\ dF = R + min_{i=7,\dots,12}(L_i) \\ dL = R + min_{i=13,\dots,18}(L_i) \end{cases} \quad (1)$$

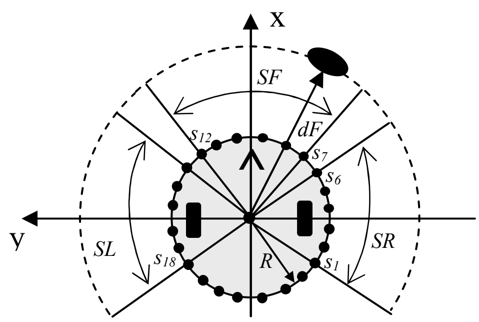

Fig. 1 Mobile robot and the sensor arrangement

We use two coordinate systems; the world coordinate system *XOY* and the mobile robot coordinate system *xoy* where *o* is in the center of the robot and the *x* axis goes in the middle between the two sensors S_6 and S_7 (see Fig. 2). The robot actions are the change of the heading angle $\Delta\Phi$ and the linear velocity *v* of the robot. For a goal seeking behaviour, the robot knows the position of its goal and θ defined as the angle between the orientation axis and the line connecting the centre of the robot to the goal.

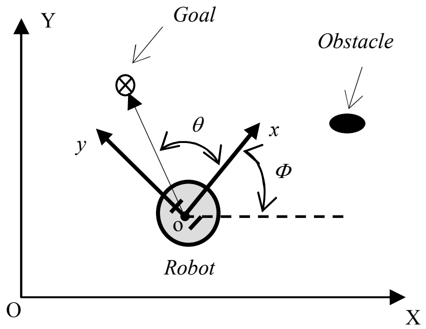

Fig. 2 The system coordinates

III. FUZZY NAVIGATOR

The fuzzy navigator inputs are distances in the three directions; left dL, face dF and right dR, and the outputs are the speed *v* of the robot centre and the steering angle $\Delta\Phi$.

A. Fuzzification variables

The measured three distances to obstacles are evaluated by two fuzzy labels; Near (N) and Far (F) represented by the membership functions shown on Fig. 2, where d_m is the minimum permitted distance to an obstacle, and d_s is the safety distance beyond which the robot can move at high speed.

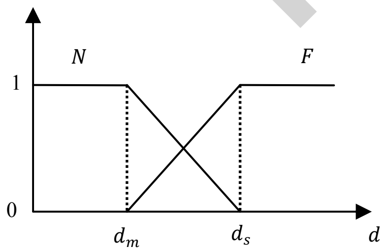

Fig. 3 Inputs membership functions

The rules conclusions are singletons expressed linguistically by: PB (*Positive Big*), PM (*Positive Medium*), PS (*Positive Small*), ZR (*Zero*), NS (*Negative Small*), NM (*Negative Medium*), and NB (*Negative Big*).

B. Fuzzy rule base

We use the fuzzy rule base proposed in [15] which combines the wall following and the goal seeking tasks. This rule base is constructed based on the human reasoning. It can be interpreted by the following meta-rules:

- If the robot is far from obstacles in the three directions, then the robot steers to the goal and goes with its maximum speed;
- If the goal is not in the front of the robot and there exist obstacles, then the robot follows the nearest obstacle on its right or left, according to the smallest distance to the obstacle.
- If both the goal and obstacles are in the front of the robot, then the robot tries to steer to the goal and follows the nearest obstacle on its right or left, according to the smallest distance to the obstacle.

The navigation mode is selected using two parameters p and q; if $p = 1 (p = 0)$ then the right (left) wall following is selected, and if $q = 1$ the goal seeking behaviour is activated else the wall following is activated. The two parameters are expressed by:

$$p = \begin{cases} 0 & if\ dL \leq dR\ and\ dL < d_M \\ 1 & else \end{cases} \quad (2)$$

$$q = \begin{cases} 1 & if\ 90° < \theta < 270° \\ 0 & else \end{cases} \quad (3)$$

d_M is the maximum distance that can be detected by the sensors.

We group the three distances in a triplet (dL, dF, dR) which define the configuration of the robot in its environment against the obstacles. Then, the fuzzy rule base is expressed as:

$R1: IF\ (N, N, N)\ THEN\ v_1\ is\ ZR\ and\ \Delta\Phi_1\ is\ pPB + (1-p)NB$
$R2: IF\ (N, N, F)\ THEN\ v_2\ is\ ZR\ and\ \Delta\Phi_2\ is\ NB$
$R3: IF\ (N, F, N)\ THEN\ v_3\ is\ C \times V_{max}\ and\ \Delta\Phi_3\ is\ ZR$
$R4: IF\ (N, F, F)\ THEN\ v_4\ is\ V_{max}\ and\ \Delta\Phi_4\ is\ pNS + (1-p)PS$
$R5: IF\ (F, N, N)\ THEN\ v_5\ is\ ZR\ and\ \Delta\Phi_5\ is\ PB$
$R6: IF\ (F, N, F)\ THEN\ v_6\ is\ ZR\ and\ \Delta\Phi_6\ is\ pPB + (1-p)NB$
$R7: IF\ (F, F, N)\ THEN\ v_7\ is\ V_{max}\ and\ \Delta\Phi_7\ is\ pNS + (1-p)PS$
$R8: IF\ \ (F, F, F)\ THEN\ v_8\ is\ V_{max}\ and \Delta\Phi_8\ is\ q\theta + (1-q)(pNM + (1-p)PM$

$\Delta\Phi_i$ and v_i are respectively the steering angle and the velocity of the robot, V_{max} is the maximum velocity of the robot and C is the speed decrease coefficient.

Here, because of the geometrical symmetry of the robot, we take: $NB = -PB$, $NM = -PM$ and $NS = -PS$. Hence, the number of parameters is reduced and the rule base becomes:

$R1: IF\ (N, N, N)\ THEN\ v_1\ is\ ZR\ and\ \Delta\Phi_1\ is\ (2p-1)PB$
$R2: IF\ (N, N, F)\ THEN\ v_2\ is\ ZR\ and\ \Delta\Phi_2\ is\ NB$
$R3: IF\ (N, F, N)\ THEN\ v_3\ is\ C \times V_{max}\ and\ \Delta\Phi_3\ is\ ZR$
$R4: IF\ (N, F, F)\ THEN\ v_4\ is\ V_{max}\ and\ \Delta\Phi_4\ is\ (2p-1)NS$
$R5: IF\ (F, N, N)\ THEN\ v_5\ is\ ZR\ and\ \Delta\Phi_5\ is\ PB$
$R6: IF\ (F, N, F)\ THEN\ v_6\ is\ ZR\ and\ \Delta\Phi_6\ is\ (2p-1)NB$
$R7: IF\ (F, F, N)\ THEN\ v_7\ is\ V_{max}\ and\ \Delta\Phi_7\ is\ (2p-1)NS$
$R8: IF\ (F, F, F) THEN\ v_8\ is\ V_{max} and\ \Delta\Phi_8\ is\ q\theta + (1-q)(2p-1)PM$

C. Defuzzification

In order to determine control actions; the steering angle ΔΦ and the robot linear speed v, we use the Sugeno fuzzy inference method [15].

$$\Delta\Phi = \frac{\sum_{i=1}^{8} \alpha_i \Delta\Phi_i}{\sum_{i=1}^{N} \alpha_i} \tag{4}$$

$$v = \frac{\sum_{i=1}^{8} \alpha_i v_i}{\sum_{i=1}^{N} \alpha_i} \tag{5}$$

α_i: is the truth value of the rule i calculated by the algebraic product method.

IV. ACO LEARNING

In this section, we propose the ACO-based method for tuning the conclusions of the fuzzy navigator controllers presented in section 2 such as to learn the robot to navigate reactively and safety without collusion with obstacles.

A. Overview of ant colony optimization

The SI techniques study collective behaviour in decentralized systems. Its development was based on mimicking the social behaviour of animals or insects swarms in an effort to find the optima in the problem space. In SI, a population of simple agents interacts locally with one other and with their environment.

The ACO has been inspired by the foraging behaviour of real ant colonies. In this algorithm, computational resources are allocated to a set of artificial ants that exploit a form of indirect communication mediated by the environment to find the shortest path from the ant nest to a set target. Ants can follow through to a food source because, while walking, they deposit pheromone on the ground, and they have a probabilistic preference for paths with larger amount of pheromone. For optimization problems, artificial ant colonies cooperate in finding good solutions. The performance measure is based on a quality function.

The ACO has been applied successfully to several optimization problems [16-21].

B. The learning problem

The learning parameters' vector is composed of the conclusions of the fuzzy navigator:

$$\mathrm{C} = [\mathrm{C}_1\ \mathrm{C}_2\ \mathrm{C}_3\ \mathrm{C}_4\ \mathrm{C}_5\ \mathrm{C}_6\ \mathrm{C}_7\ \mathrm{C}_8]^{\mathrm{T}} \tag{6}$$

To each parameter are associated several competing candidates. To simplify, for each parameter we need the minimum and the maximum possible values. The proposed values are equally distributed between these bounds. Lets P_min^i and P_max^i be the bounds of i^th element of the parameters' victor P. Therefore we can take

$$C_{i1} = C_{min}^{i}, C_{i2} = C_{i1} + \frac{P_{max}^{i} - P_{min}^{i}}{J-1}, \dots, C_{iN} = C_{max}^{i}$$

Thus:

$$C_i \in \{\, C_{i1}, \quad C_{i2}, \dots, \quad C_{iN}\}$$

The tuning problem is to find the best conclusions of the fuzzy rules which allows to achieve its defined task . It is a combinational optimization problem with a complexity equal to N^8.

The Fig.2 gives a graphical representation of the problem, the eight sets of parameters P_i,$(i = 1, \dots, 8)$ are arranged in eight column lists where each candidate value is represented by a node.

The tour of an ant consists of a combination of the fuzzy navigator parameters. Starting from its nest N, an ant moves through the columns C_i,$(i = 1, \dots, 8)$. Finally, the ant reaches the food source F which is added here just to match the real world.

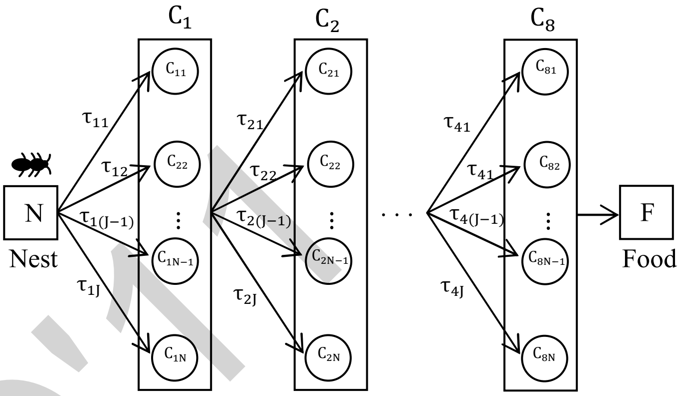

Fig. 4 The relation between the tour of ant and the conclusions of the fuzzy navigator

C. The rule of selecting parameters by ants

For each set of parameters, the node visited by the ant is selected as the value of the parameter. Selection of a parameter value is based on pheromone trails between parameter vectors. The size of pheromone matrix τ is $8 \times N$ and each element of the matrix is denoted by τ_{ij}, $i = 1,2, \dots, 8$ and $j = 1,2, \dots, N$. As shown in figure 6, when the ant arrives at vector P_x, selection of the next parameter value P_{x+1j}among the candidate list P_{x+1} is done using the state transition rule (7) which depends on pheromone trails τ_{x+1j}, $j = 1,2, \dots, N$, the cost function and some heuristic.

$$j = \begin{cases} \arg\max_{y\in J}\{\tau_{iy}(t)\} & if\ q \le q_0 \\ j_1 & if\ q_0 < q < q_1 \\ j_2 & if\ q \ge q_1 \end{cases} \tag{7}$$

$$Pr_{ij_1}(t) = \frac{\tau_{ij_1}(t)}{\sum_{l=1}^{N} \tau_{il}(t)} \tag{8}$$

where q is a random variable uniformly distributed over $[0, 1]$, q_0 and q_1 $(q_0 < q_1)$ are two tunable parameters in the interval $[0, 1]$, j_1 belongs to the candidate list, $j_1 \in \{1,2, \dots, N\}$ which is selected based on the above probabilistic rule (8) and j_2 is selected based on a uniform probabilistic rule over candidate list $\{1,2, \dots, N\}$.

D. The proposed rule of updating pheromone trails

Each ant modifies the environment by the pheromone deposit. Assume that at initial time $t = 0$ all the trails have the same pheromone concentration τ_0 , that is, $\tau_{ij}(0) = \tau_0$ $(i = 1,2, \dots ,8$ and $j = 1,2, \dots , N)$.
When an ant passes through a node P_{ij} , the pheromone concentration of this node will be updated immediately using the following pheromone updating rule:

$$\tau_{ij}(t + 1) = \tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (9)$$

$\Delta\tau_{ij}$ is the pheromone update value which is presented on the Fig. 5 and expressed by:

$$\Delta\tau_{ij} = max\left\{\frac{d-d_m}{d_s-d_m}, 1\right\} \quad (10)$$

ρ is a fixed constant which define the amount portion of the pheromone update.

That is the best tour will receive the highest pheromone concentration i.e. more I is small more the pheromone concentration update is big.

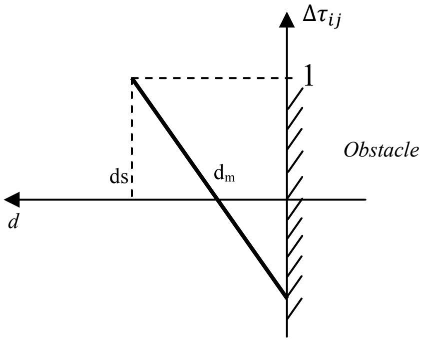

Fig. 5 The pheromone update-value

The ants have to explore and test several conclusions values. The exploration phase is often long. Though, as fuzzy rules are interpretable and tuning parameters have physical meaning, this phase can be drastically reduced by introducing knowledge in the initial fuzzy navigator.

After a sufficient number of learning tours the algorithm can be stopped and the optimized conclusions are those corresponding to maximum pheromone trails. For each parameter C_i, the optimal value C_{ij_0}, is selected according to the following greedy low:

$$j_0 = arg\, max_{y\in J}\{\tau_{iy}(t)\} \quad (11)$$

V. SIMULATION RESULTS

The problem consists of learning the mobile robot the wall following and goal seeking without collusion with obstacles when it evolves in unknown environment. For this purpose we use the mobile robot model presented in section II, we assume that the effective range of the ultrasonic sensors is 10 cm - 250 cm. We use the proposed fuzzy navigator presented in section III, and to learn the robot, we use ACO algorithm proposed in section IV. In the simulation we take $V_{max} = 1\, m/sec$ and $C = 0.1$.

The number of potential candidate conclusions is $J = 5$. And bounds of each fuzzy rule conclusion singleton are given by Table 4.

Table 1 Fuzzy rule conclusions for $\Delta\Phi$

	NS	NM	NB	ZR	PB
MIN	-25°	-5°	-90°	0°	30°
MAX	-5°	-30	-30°	0°	90°

In the learning phase,only the wall right following behaviour is selected (p=1 and q=1), the robot is kept in unknown environment. The simulation results at the beginning of learning phase are presented on Fig. 6. The Fig. 7 presents the simulation after 1000 steps. After 5000 steps of time the learning is stopped and the learned values of conclusions are donated by table 2.

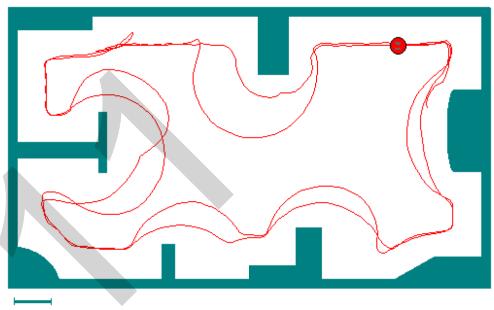

Fig. 6 Beginning of the learning phase

Table 2 The learned conclusions values

Rule	R1	R2	R3	R4	R5	R6	R7	R8
Symbolic Conclusion	PB	NB	ZR	NS	PB	PB	NS	NM
Learned Value								

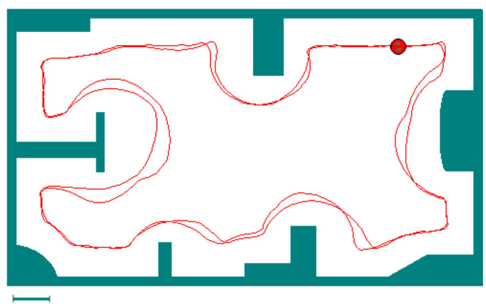

Fig. 7 After a period time of learning

VI. CONCLUSIONS

In this paper, we have proposed a learning method for mobile robots navigation using fuzzy logic and ACO. The used navigator is a collection of fuzzy rules inspired from human reasoning. The ACO is used to fine tune the conclusions of the fuzzy rules in order to learn the robot a predefined task. The proposed method was tested with the wall following and goal seeking navigation tasks for a mobile robot evolving in unknown environment. The proposed method is easy to implement for industrial applications. The simulation results demonstrate the efficiency of the proposed method.

REFERENCES

[1] Maaref, H., Barret, C.: Sensor Based Navigation of an Autonomous Mobile Robot in an Indoor Environment. In: Control Engineering Practice, Vol. 8, pp. 757-768, 2000.

[2] Latombe, J-C., Robot Motion Planning. Kluwer Academic Publishers (1991).

[3] Koren, Y., Borenstein J.: Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In: the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California (1991).

[4] Tan, G.-Z., He, H., and Sloman A.: Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots. In: Acta Automatica, Vol. 33, No. 3, (2007).

[5] Liu, G., Li ,T., Peng, Y., Hou ,X. In: The ant algorithm for solving robot path planning problem. In: the third IEEE International Conference on Information Technology and Applications. Vol.2, pp. 25 – 27, Sydney (2005).

[6] Brooks, R. A.: A robust Layered Control System for a mobile robot. In: IEEE Trans. On Robotics and Automation, Vol. RA-2, No. 1, pp. 14-23, Mar. (1986).

[7] Suwimonteerabuth, D., Chongstvatana, P.: Online robot learning by reward and punishment for a mobile robot. In: the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Swizerland, Oct. (2002).

[8] Beom, H. B., Cho, H. S.: A Sensor Based Navigation for a Mobile Robot using Fuzzy Logic and Reinforcement Learning. In: IEEE Trans. On Systems, Man, and Cybernetics, Vol. 25, No. 3, Mar. (1995).

[9] [Ye C., Nelson, H.C.Y.: A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance. In: IEEE Trans. On Systems, Man, and Cybernetics, Vol. 33, No. 1, Feb.(2003).

[10] Zavlangas, P. G., Tzafestas, S. G.: Motion control for Mobile Robot Obstacle Avoidance and Navigation: A Fuzzy Logic-Based Approach. In: Systems Analysis Modelling Simulation. Vol. 43, No. 12, pp. 1625-1637 , Dec.(2003)

[11] Dahmani, Y., Benyettou, A.: Fuzzy Reinforcement Rectilinear Trajectory Learning. In: Journal of Applied Science, Vol. 4, No. 3, pp.388-392, (2004).

[12] Kermiche, S., Saidi, M. L., Abbassi, H.A.: Gradient Descent Adjusting Takagi-Sugeno Controller for a Navigation of Robot Manipulator. In: Journal of Engineering and Applied Science, Vol. 1, No 1, pp.24-29,(2006).

[13] Glorennec, P. Y., Jouffe, L.: A Reinforcement Learning Method for Autonomous Robot. In: the Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, Sept. (1996).

[14] Gu, D., Hu H. : Accuracy Based Fuzzy Q-Learning for Robot Behaviors, Proceedings of the IEEE International Conference on Fuzzy Systems (IEEE-FUZZY2004), Budapest, 25-29 July (2004).

[15] H. Boubertakh, M. Tadjine and P. Y. Glorennec, “A New Mobile Robot Navigation Method using Fuzzy Logic and a Modified Q-learning Algorithm”, Journal of Intelligent & Fuzzy Systems, Vol. 21, No. 1-2, 2010.

[16] Dorigo M., and Gambardella L. M., “Ant colony system: A cooperative learning approach to the travelling salesman problem,” IEEE Trans. on Evolutionary Computation, vol. 1, 1 (1997) 53-66.

[17] Dorigo M., Maniezzo V. and Colomi A. , “Ant system: Optimization by a colony of cooperating agents,” IEEE Transactions on Systems, Man, and Cybernetics-Part B, vol. 26,1(1996). 29-41.

[18] Clerc M., and Kennedy J., “The particle swarm - explosion, stability, and convergence in a multidimentional complex space,” IEEE Trans. Evolutionary Computation, vol. 6, 1(2002) 58-73.

[19] Mendes R., Cortez P., Rocha M., , and Neves J., “Particle swarms for feedforward neural network training,” In Proc. of Int. Joint Conf. on Neural Networks (2002)1895-1899.

[20] Sim K. M., and Sun W. H., “Ant colony optimization for routing and load-balancing: survey and new directions,” IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 33, 5 (2003) 560-572.

[21] H. Boubertakh, , M. Tadjine, , P.Y. Glorennec and S., Labiod , “Tuning Fuzzy PID controllers using ant colony optimization,” Mediterranean Conference of Control and Automation (MED’09), Thessaloniki(Greece), 13-18 (2009).