International Journal of Control and Automation
Vol.7, No.5 (2014), pp.323-342
http://dx.doi.org/10.14257/ijca.2014.7.5.33

Modeling and OnLine Control of Nonlinear Systems using Neuro-Fuzzy Learning tuned by Metaheuristic Algorithms

[1]Mourad Turki, [1]Sana Bouzaida, [1]Anis Sakly and [1]Faouzi M'Sahli

[1] Research Unit: Etude des Systèmes Industriels et Energies Renouvelables, The National School of Engineers of Monastir, Tunisia

mouradessturki@yahoo.fr, bouzaida_sana@hotmail.fr, sakly_anis@yahoo.fr, faouzi.msahli@enim.rnu.tn

Abstract

This paper proposes a hybrid learning algorithm for tuning an Adaptive Network based Fuzzy Inference System ANFIS. The proposed scheme of adapting parameters in ANFIS employs evolutionary techniques PSO and GA to adjust the antecedent parameters. The least-squares (LSE) algorithm is used to adjust consequent parameters. The number of fuzzy rules is fixed and given by using the Xie Beni's index. This new approach is applied to identify and control nonlinear systems with an on-line strategy. The obtained results are compared to similar ANFIS using gradient descent method GD as antecedent parameters of training algorithm and other methods applied in the same problems.

Keywords: *Neuro-fuzzy, ANFIS, metaheuristic algorithms, hybrid learning algorithms, identification, nonlinear systems, online control*

1. Introduction

Fuzzy model saw an increasing interest in last decade, which have been emerged from the fusion of neural networks and fuzzy inference systems. They benefited from the advantages of the previous two techniques. On the one hand fuzzy logic allows fast specifications spots to be achieved by a human expertise. But, the parameters of fuzzy logic are difficult to optimize. On the other hand, the neural networks do not need knowledge of the system. But, they make its regulations possible according to an algorithm of training the precise behaviour of the system. These techniques have advantages of excellent capability to deal with complex systems and can be used in modelling and controlling [1-3].

Many different structures for neuro-fuzzy networks have been proposed, but the most popular types are the Mamdani type and the Takagi Sugeno one [4].

Among them, ANFIS is a neural network based on fuzzy approach. It implements a TS model where the output is a linear combination of inputs and has been applied widely because of its capability to approximate a wide range of nonlinear systems [5].

ANFIS uses the least-squares to optimise the consequent parameters and backpropagation to train the antecedent parameters or membership function of inputs [6].

The fuzzy systems are optimized by adapting the antecedent and consequent parameters. Several learning algorithms of fuzzy models have been proposed [7]. The backpropagation algorithm, which is known as a powerful training technique, is widely used for training fuzzy models such as ANFIS by means of error propagation via variation calculus. However, the

ISSN: 2005-4297 IJCA

backpropagation learning algorithm may reach the local minima due to the steepest descent, also known as Gradient Descent (GD), optimization technique is used in the backpropagation learning algorithm to minimize the error function. In addition, the performance of the algorithm is dependent on both the initial values of the parameters which will be optimized as well as the heaviness which is due to the derivation of the functions of each layer.

Those problems lead to a non optimal solution. To tackle these disadvantages, several Evolutionary Algorithms (EA) such as: the Genetic Algorithm (GA) [8], the genetic programming [9], the evolutionary programming [10], the evolutionary strategies [11] and the Particle Swarm Optimization (PSO) [12], have been proposed to optimize fuzzy models and to give more efficient robust approaches to deal with these complex problems. These heuristic and stochastic methods make them difficult to fall on a local minimum, since they use the trained populations of the individuals.

The GA is largely used to optimize fuzzy models. It makes it possible to overcome the backpropagation problem thanks to a phase diversification of research [13-16]. Optimization is related to the membership functions or to fuzzy rules or both at the same time [17].

In this field of research, Karr [13] uses GA to generate membership functions for fuzzy systems. The method is based on fixing the number of linguistic terms a priori then GA is used to code the functions of membership in a binary chromosome [18], where the membership functions and the rules are in co-relation. Herrera *et al.* [19, 20] proposed the use of real coding to codify the membership functions. For the sake of the improvement capacity of GA in searching the optimal solution, Juang [21] proposes the use of PSO [12, 22].

Many researchers applied hybridization of evolutionary techniques for optimization, such as: hybridization of GA and PSO [21], Ant Colony Optimization (ACO) and PSO [23].

Hybridization of PSO and GA is proposed for the optimization of the neuro-fuzzy systems such as in [24] where PSO is used to determine the optimal elements in the phase of elitism and the parameters to be optimized are the number of rules and the parameters of the network recurring TS neuro-fuzzy. Whereas, the algorithm HGAPSO [25] which works with a fixed rules number in advance and PSO is used in the phase of reproduction by elitism of GA and tested on a network of recurring neurons and a network TSK neuro-fuzzy system. In [26], a new method of crossing is presented with the use of PSO in the phase of change of GA and proved an optimal time of convergence.

Some papers [27, 28], have been written on the hybridization between classic and evolutionary method for identification problems. Oliveira et al. [27] used an implemented Neuro-fuzzy function in Matlab to monitor sensor degradation using PSO algorithm to learn parameters of ANFIS.

In this paper, we will use the meta-heuristic methods, GA and PSO, to train the antecedant parameters of ANFIS system which uses GD to train theme and the LSE method that is used to learn consequent parameters and compare the obtained results of these methods applied in identification and on-line control of nonlinear systems with others methods in literature applied in the same problems.

This paper is organised as follows: in Section II the ANFIS system is presented with the training algorithm used in this work. Section III introduces the meta-heuristic methods GA and PSO. In Section IV, the hybrid algorithms for training TS neuro-fuzzy systems was explained. Section V presents the simulations of identification and control examples tested with the proposed algorithms. Conclusions are given in the last section.

2. The Concept of ANFIS

2.1. ANFIS structure

The ANFIS proposed by Jang *et al.* [6] implements a TS fuzzy inference system and has five layers in which each node performs a particular function. The structure of ANFIS is given in Figure 1.

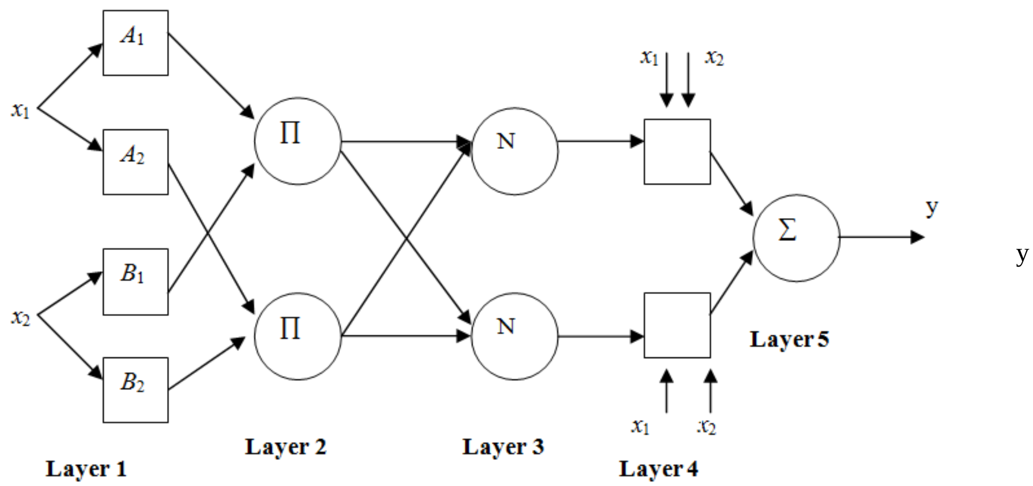


Figure 1. Structure of ANFIS

Layer 1
The neurons in this layer represent the membership functions used as the antecedents rules. This layer represents the fuzzification stage and the output of each neuron is given by:

$$y_i^1 = \mu_{A_i}(x_1) \quad i=1,2 \tag{1}$$

$$y_i^1 = \mu_{B_i}(x_2) \quad i=3,4$$

where x_1 (or x_2) is the input to the node and A_i (or B_i) is the membership function.

Layer 2
This layer contains nodes that generate the firing strengths by multiplying the incoming signals using the following equation:

$$y_i^2 = w_i = \mu_{A_i}(x_1).\mu_{B_i}(x_2) \quad i=1,2 \tag{2}$$

Layer 3
This layer contains nodes that give the normalized firing strengths of each rule:

$$y_i^3 = \frac{y_i^2}{\sum_{j=1}^{n} y_j^2} \quad i=1,2 \tag{3}$$

where n is the rules number.

Layer 4

This layer calculates the first order TS rules for each fuzzy rule based on the consequent parameters (a_i, b_i, c_i):

$$y_i^4 = y_i^3.(a_i.x_1 + b_i.x_2 + c_i)\ i=1,2 \tag{4}$$

Layer 5

This last layer calculates the global output of the system which is the sum of the outputs of layer 4.

$$y^5 = \sum_{i=1}^{n} y_i^4 \tag{5}$$

2.2. Learning algorithm

A neuro-fuzzy system is optimized by learning the antecedent parameters, which are the membership function parameters, and consequent parameters, which are the polynomial coefficients of the consequent part. The famous training algorithm for ANFIS is that uses GD method. This method based on backpropagation algorithm to update antecedent parameters, which are the centers c_i and variances σ_i, and the least-squares method to update conclusion part parameters. Jang *et al.* [6] has introduced four methods to update the parameters of ANFIS structure. Those methods are listed below according to their computation complexities:

1. Gradient decent only: all parameters are updated by the GD.

2. Gradient decent only and one pass of LSE: the LSE is applied only once at the very beginning to get the initial values of the consequent parameters and then the gradient decent takes over to update all parameters.

3. Gradient decent only and LSE: this is the hybrid learning.

4. Sequential LSE: using extended Kalman filter to update all parameters.

A step in the learning procedure is composed of two passes: forward pass and backward pass.

These methods, characterized by high complexity, update antecedent parameters by using GD or Kalman filtering. The ANFIS, which is presented in the toolbox Fuzzy of Matlab, uses the hybrid learning. We have constructed our program in Matlab by imitating the same structure of ANFIS and the same hybrid algorithm described in the next lines. The aim is to optimise the parameters by using other methods due to the limitation of ANFIS of Matlab used as black box and the difficulty to access to its code.

- **Forward pass:**

In the forward pass, the optimal consequent parameters are estimated by LSE-method while the antecedent parameters are assumed to be fixed. The output data can be expressed as a linear combination of the consequent parameters:

$$\begin{cases} y_d(1) = \bar{w}_1 . f_1(1) + \bar{w}_2 . f_2(1) + \ldots + \bar{w}_r . f_r(1) \\ y_d(2) = \bar{w}_1 . f_1(2) + \bar{w}_2 . f_2(2) + \ldots + \bar{w}_r . f_r(2) \\ \vdots \\ y_d(p) = \bar{w}_1 . f_1(p) + \bar{w}_2 . f_2(p) + \ldots + \bar{w}_r . f_r(p) \\ \vdots \\ y_d(n) = \bar{w}_1 . f_1(n) + \bar{w}_2 . f_2(n) + \ldots + \bar{w}_r . f_r(n) \end{cases} \tag{6}$$

where:

$$f_i(p) = a_i . x_1(p) + b_i . x_2(p) + c_i \tag{7}$$

y_d: desired output.
n: number of data.
r: number of rules.

We can translate this form into the following matrix form

$$y_d = A.k \tag{8}$$

$$A = \begin{bmatrix} \bar{w}_1 . x_1(1) & \ldots & \bar{w}_r . x_1(1) & \ldots & \bar{w}_1 . x_2(1) & \ldots & \bar{w}_r . x_2(1) & \ldots & \bar{w}_r \\ \bar{w}_1 . x_1(2) & \ldots & \bar{w}_r . x_1(2) & \ldots & \bar{w}_1 . x_2(2) & \ldots & \bar{w}_r . x_2(2) & \ldots & \bar{w}_r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{w}_1 . x_1(n) & \ldots & \bar{w}_r . x_1(n) & \ldots & \bar{w}_1 . x_2(n) & \ldots & \bar{w}_r . x_2(n) & \ldots & \bar{w}_r \end{bmatrix} \tag{9}$$

k represents the vector of consequent parameters. It can be resolved by the LSE-method

$$k = (A^t A)^{-1} A^t y_d \tag{10}$$

- **Backward pass:**

In the second pass, backpropagation is used to modify the antecedent parameters while the consequent parameters remain fixed.

The antecedent parameters are updated by GD according to the equations:

$$c_{ij}(t) = c_{ij}(t-1) - \alpha(\partial E(t) / \partial c_{ij}) \tag{11}$$

$$\sigma_{ij}(t) = \sigma_{ij}(t-1) - \alpha(\partial E(t) / \partial \sigma_{ij})$$

where:

$$E(t) = \frac{1}{2}(y_d(t) - y(t))^2 \tag{12}$$

c_{ij} and σ_{ij} are respectively the center and the standard deviation values of each input x_j onto the ith membership function.
α : learning rate for antecedent parameters.
E (t): instantaneous square error.

3. Meta-heuristic Optimization Methods

Meta-heuristic optimization algorithms have attracted many researchers in the last decade. They are using learning strategies in order to find efficient near-optimal solutions.

3.1. Genetic Algorithm

Genetic Algorithm GA is a random search technique that imitates natural evolution with Darwinian survival of the fittest approach. It can handle any kind of objective functions and constraints without much mathematical requirements about the optimization problems.

The population strategy enables GA to search the near optimal solutions from various parts and directions simultaneously within a search space [20]. GA uses random choice and probabilistic decision to guide the research, where the population improves towards near-optimal points from generation to generation. In GA, the problem's variables are represented as genes in a chromosome, and the chromosomes are evaluated according to their fitness values. The chromosomes with better fitness are found through the three basic operations of GA: selection, crossover and mutation.

The genetic operators alter the composition of genes to create new chromosomes called offspring, and with the selection operator chromosomes with better fitness have higher probabilities of being selected in the next generation.
GA is described as follows:

1. Initialization of initial population,
2. Evaluation of each element in the population using the fitness function,
3. Selection of the chromosomes,
4. The use of crossover and mutation operations with the chromosomes selected in the aim to generate new chromosomes,
5. If the stopping criterion is validated then the parameters are stored else return to step 2.

3.2. Particle Swarm Optimization

The particle swarm optimization PSO is a parallel evolutionary computation developed in 1995 by Kennedy and Eberhant [20]. It imitates the movement of birds flocking or fish schooling looking for food [16]. The population is referred to as a swarm and the individual in the swarm is called as particle in PSO.

The PSO algorithm is initialized with a population of random solutions. Each particle has a position and a velocity representing the potential solution to the problem. The particle adjusts the velocity and position according to the location of the best fitness achieved by the particle itself pbest and the location of the best fitness achieved across the whole population g_{best}.

The PSO use the following equations to update its velocity and position:

$$v(k+1) = wv(k) + c_1 r_1 (p_{best} - x(k)) + c_2 r_2 (g_{best} - x(k)) \tag{13}$$

$$x(k+1) = x(k) + v(k+1) \tag{14}$$

Where x and v represent the position and the velocity of the particle, respectively. c_1 and c_2 are positive constants referred to the acceleration constants. r_1 and r_2 are random numbers between 0 and 1. pbest refers to the best position found by the particle and gbest refers to the global best position. w is the inertia weight.

The main steps of PSO optimization are as follows:

1. Initiation : initialize the particle's position with a random vector of the search space,
2. Fitness based position update : the particle adjusts the velocity and position according to the location of the best fitness achieved by the particle itself pbest and the location of the best fitness achieved across the whole population gbest,
3. Convergence: the process is repeated until the reach of the termination condition.

4. Proposed Hybrid Algorithms

In the hybrid algorithms, we use evolutionary techniques or meta-heuristic methods to adjust the antecedent parameters as well as the LSE method to adjust the consequent parameters.

The structure of a particle or chromosome in each evolutionary technique is shown in Figure 2. It represents the parameters of Gaussian membership function σ_{ij} and c_{ij}, respectively, of the i^{th} dimension and the j^{th} membership function.

c_{11}	σ_{11}	…	c_{ij}	σ_{ij}	…

Figure 2. Structure of particle or chromosome

4.1. Hybrid PSO - LSE training algorithm

Unlike ANFIS which uses the backpropagation method to update the antecedent parameters and the LSE to update consequent parameters, we propose PSO in the place of backpropagation method in order to solve the inconveniences of LSE method and exploit the major advantages of PSO like its capacity which gives global solution with a few execution time.

The flowchart of the proposed hybrid learning algorithm PSO-LSE is shown in Figure 3.

4.2. Hybrid GA - LSE training algorithm

In this part, we employ GA method to adjust the parameters of the membership functions or antecedent parameters. A chromosome has the same structure of a particle. It represents the parameters of Gaussian membership function σij and cij. Like the hybridization of PSO and LSE, GA replaces the backpropagation method of ANFIS to optimize the antecedent parameters and LSE for updating the consequent parameters.

The flowchart of the proposed hybrid learning algorithm GA-LSE is shown in Figure 4.

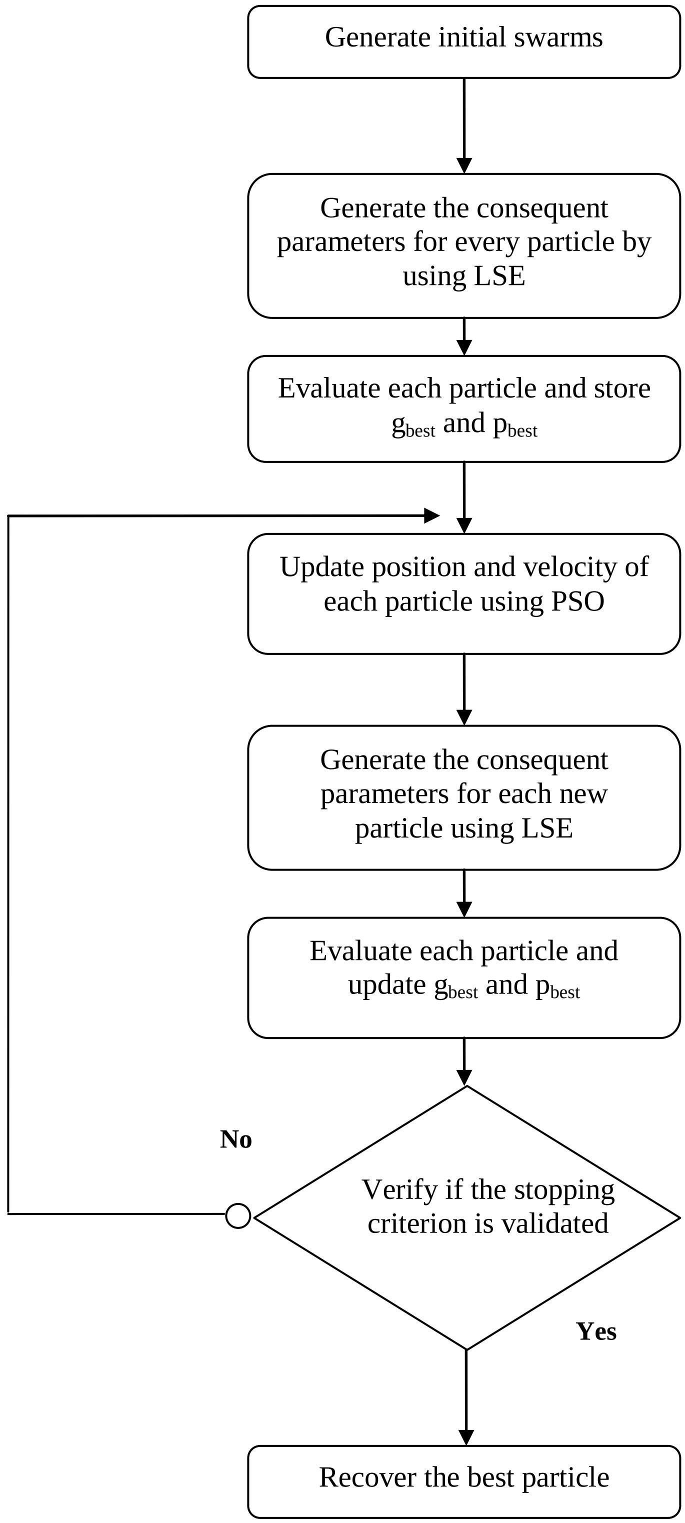


Figure 3. Flowchart of the hybrid algorithm PSO-LSE

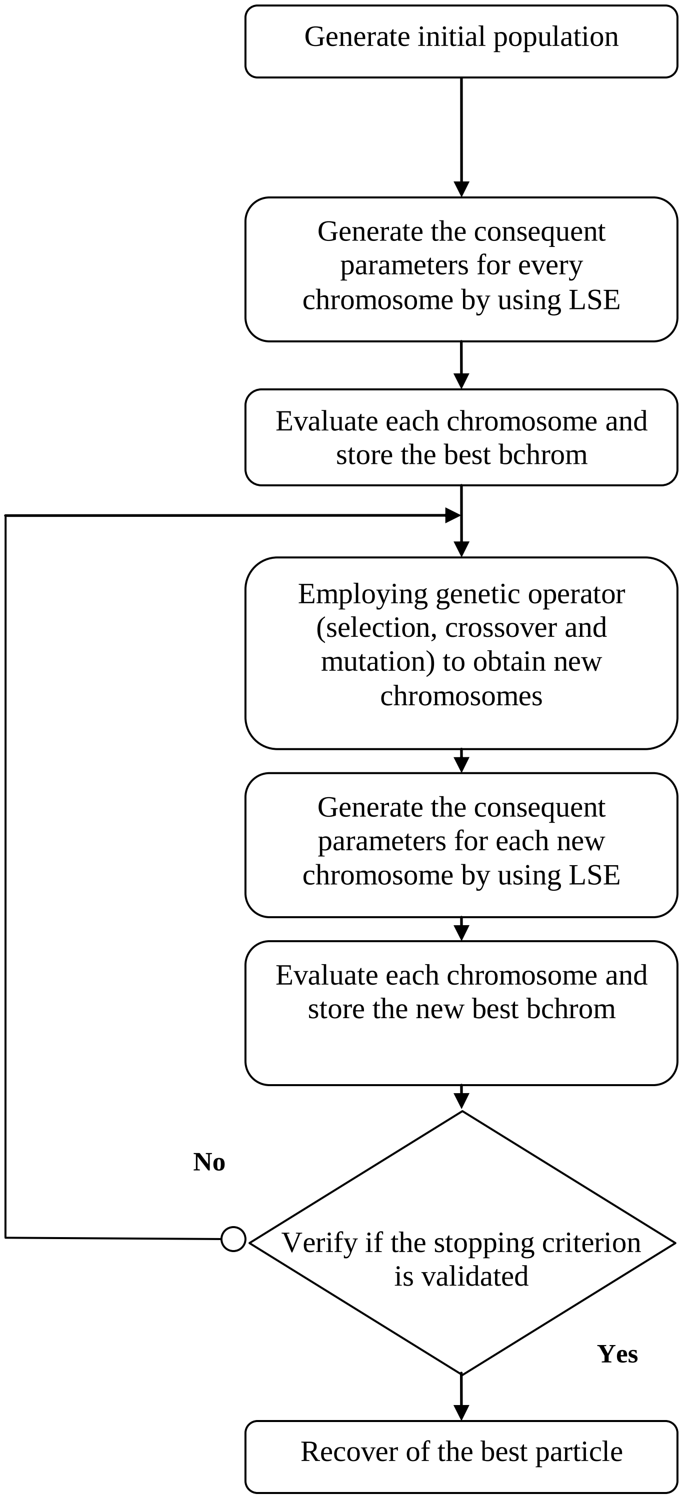


Figure 4. Flowchart of the hybrid algorithm GA-LSE

5. Simulations

In this section, we use two different examples for the simulations of the proposed methods. The first one used the example of a nonlinear dynamic system identification given by Narendra and Parthasarathy [29]. The second one was used to control the water bath temperature system [30].

The ANFIS created is configured to have two inputs x_1 and x_2 and one output y. It is trained by three learning algorithm, GD-LSE, PSO-LSE, and GA-LSE and applied for the predefined examples.

For the two examples, the initial parameters for each method of hybrid algorithms are given in Table 1 and Table 2 before training.

Table 1. The initial parameters before training for GA-LSE method

Parameters	Values
Population size	20
Crossover rate	1
Mutation rate	0.1

Table 2. The initial parameters before training for PSO-LSE method

Parameters	Values
Population size	20
b_1	2
b_2	2
$w_{\max}$	0.9
w_{min}	0.4

To evaluate each particle or chromosome, we use the RMSE that is defined as follows:

$$E(y, y_d) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_{di})^2} \tag{15}$$

where n represents the number of input data; y and y_d represent respectively the model output and the desired output.

Example 1: Identification of nonlinear system

The first example used for identification is described by the next difference equation:

$$y(k+1) = \frac{y(k)}{(1+y^2(k))} + u^3(k) \tag{16}$$

$$u(k) = \sin(\frac{2.\pi.k}{25}) \tag{17}$$

The output of this equation depends nonlinearly on both its past values and the input. The 200 training input patterns are randomly generated in the interval [-1 1] by using Eq 17. Evolution progressed for 50 generations.

To fix the number of rules, we have used the method described by Xie Beni [31]. This optimal number is obtained by the minimization of Xie Beni's index given by the next function:

$$XB = \frac{\sum_{i=1}^{r}\sum_{j=1}^{n} w_{ij} \left\| x_j - c_i \right\|^2}{n.\min_{i \neq j} \left\| c_i - c_j \right\|^2} \tag{18}$$

Figure 5 shows the change of XB's index for this example. Thus, the optimal number of fuzzy rules is found to be 4, therefore, resulting in 16 parameters. The curve bellow gives the optimal number of rules.

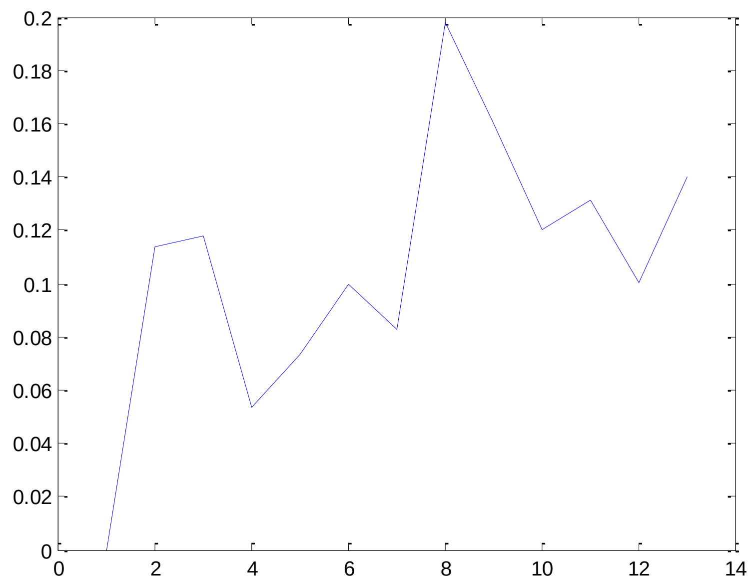


Figure 5. Xie Beni cluster validity index

The final average RMSE of the training data for the GD-LSE method, PSO-LSE method and GA-LSE method are approximated to 0.0042, 0.0011 and 0.0021, respectively.

Figure 6 shows the outputs of these methods for the input ($u(k) = \sin(\frac{2.\pi.k}{25})$). The RMSE values of the different training methods are given in Table 3.

Table 3. The performance comparison with different methods

	HELA [32]	GA [33]	SE [34]	ACO-ACO [35]	GA-GA [35]	PSO-PSO [35]	GD-LSE	PSO-LSE	GA-LSE
RMSE (avg)	0.0051	0.11	0.012	0.13	0.06	0.002	0.0042	0.0011	0.0021
RMSE (best)	0.004	0.10	0.009	0.032	0.031	0.0001	0.0042	0.0007	0.0012

Table 3 shows that PSO-LSE gives the lowest RMSE compared to GA-LSE and GD-LSE using in this work and the other methods in literature.

Figure 7 shows the learning curves of the three methods. In this figure, we find that the PSO-LSE method gives the lower RMSE compared to the others methods.

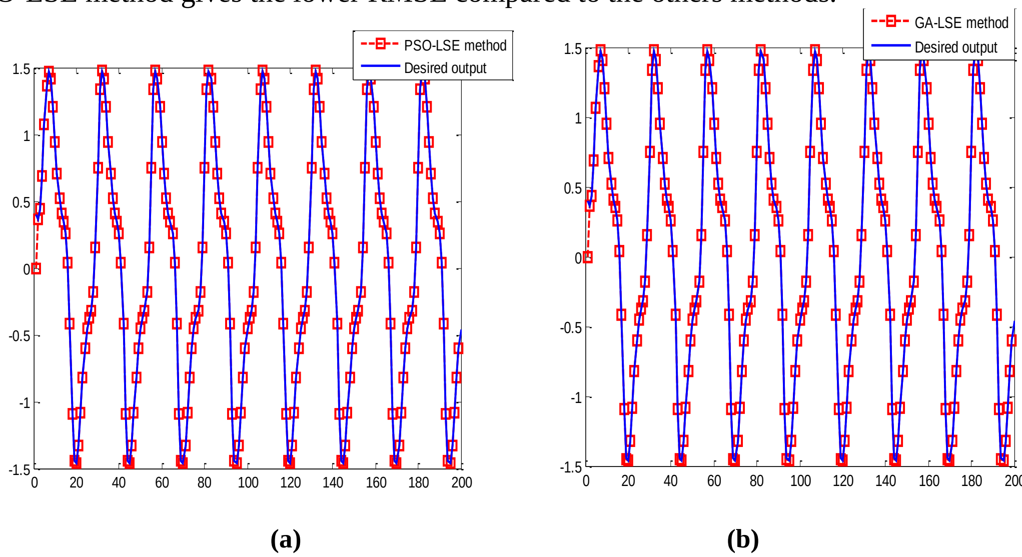


Figure 6. Results of desired output, (a) PSO-LSE output and (b) GA-LSE output

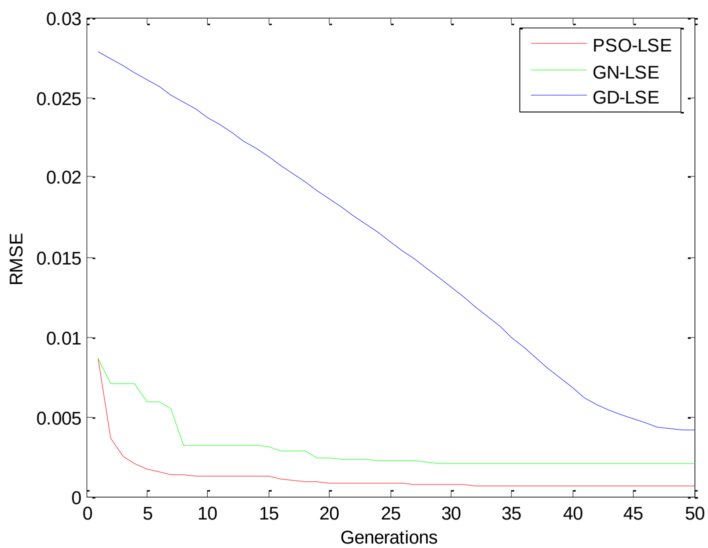


Figure 7. The RMSE curves of the different methods GD-LSE, PSO-LSE and GA –LSE

Example 2: Control of temperature of a water bath system

In this section, the proposed hybrid algorithm is applied for the control of the temperature of a water bath system, which can be described by the following difference equation:

$$y(k+1) = e^{-\alpha T_s} y(k) + \frac{\frac{\beta}{\alpha}(1-e^{-\alpha T_s})}{1+e^{0.5y(k)-40}} u(k) + (1-e^{-\alpha T_s}) y_0 \tag{19}$$

The system parameters used in this example are α=1.0015e-4, β=8.67973e-3 and y_0=25(°C), which were obtained from a real water bath plant. The sampling period is Ts=30s [30].

The general learning for control purpose is performed in three steps:

- In the learning phase, a fuzzy inverse model with the hybrid algorithm is obtained based on input-output data generated from the water bath system.
  The model is identified as follows:
  $$\hat{u} = f(y(k+1), y(k)) \tag{20}$$
  It is a two-input-single-output fuzzy model. There are 120 input-target data sets chosen as training data.
- In the application phase, the desired output (yref(k+1)) and the water bath system output (y(k)) are used as input to the fuzzy inverse model obtained in the first phase. This model represents a controller that generates the control action to the bath water system.

To establish the inverse model for control purpose, we use an on-line strategy. In fact, the fuzzy inverse model with the hybrid algorithm occurs at each time step to adjust the consequent and premises parameters of the fuzzy controller. The overall control structure is illustrated in Figure 8.

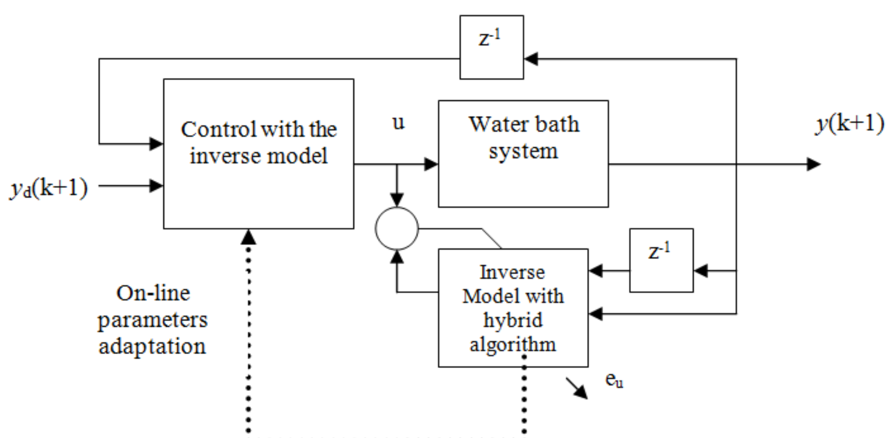


Figure 8. On-line strategy of control

To test the performance of PSO-LSE and GA-LSE methods, we use two tasks. The first task is to control the simulated system to follow three set points:

$$y_{ref} = \begin{cases} 35°C & for\ k \le 40 \\ 55°C & for\ 40 < k \le 80 \\ 75°C & for\ 80 < k \le 120 \end{cases} \tag{21}$$

The control performance of the different training algorithms is shown in Figure 9. Simulation results demonstrated that the inverse model controller with evolutionary method is able to achieve satisfactory response characteristics. Figure 10 shows that the errors obtained using the PSO-LSE method was smaller compared to the values found using the other two controllers.

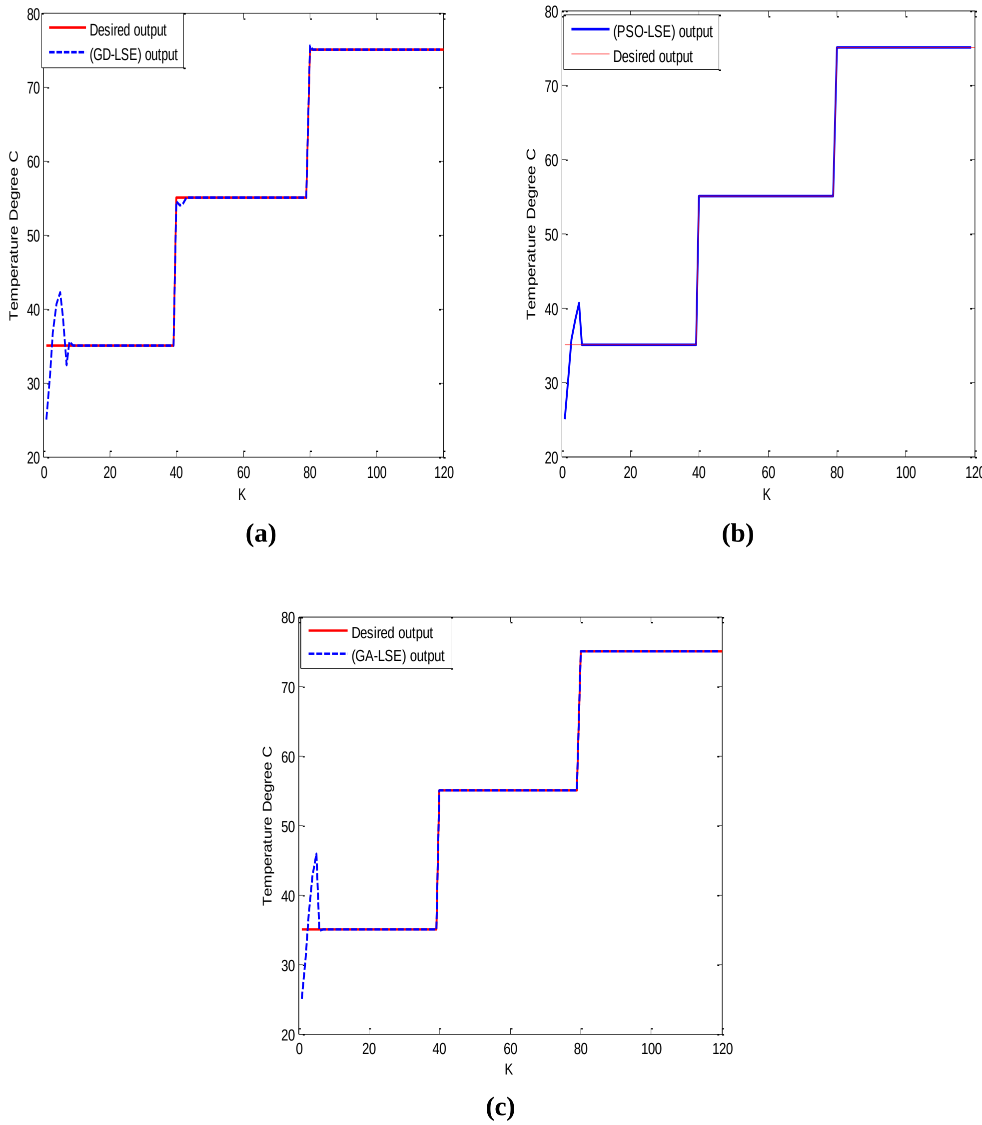


Figure 9. The control performance of the different methods (a) GD-LSE, (b) PSO-LSE and (c) GA-LSE

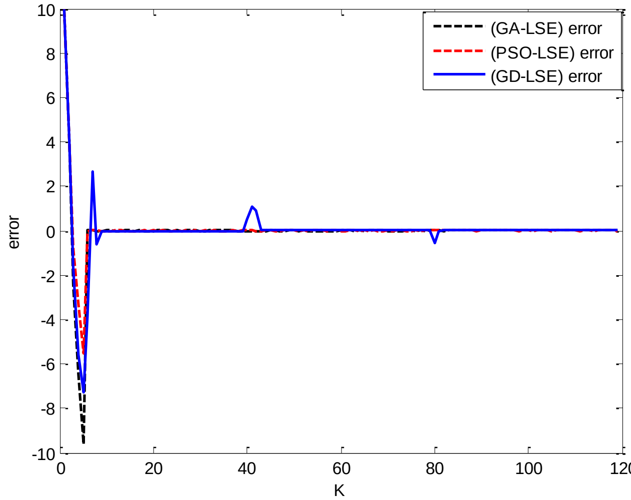
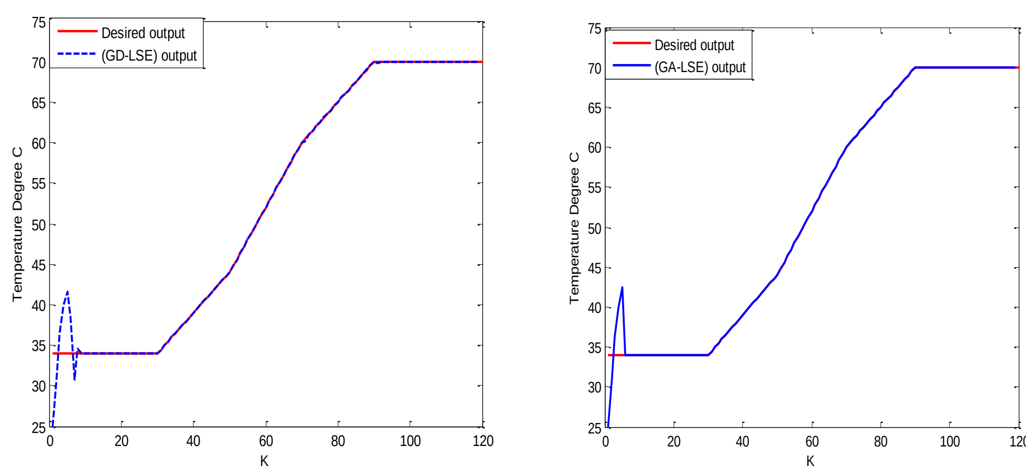


Figure 10. The error curves of the different methods

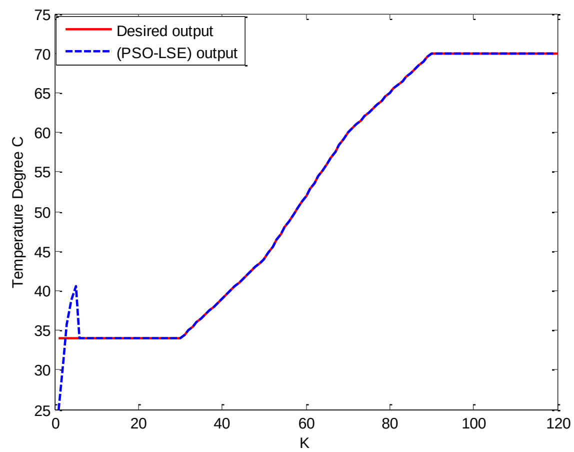
The second set of simulations is carried out for the purpose of studying the tracking capability of the hybrid methods with respect to ramp-references signals.

$$y_{ref}\begin{cases} 34°C & for\ k\le 30 \\ (34+0.5\cdot(k-30))°C & for\ 30<k\le 50 \\ (44+0.8\cdot(k-50))°C & for\ \ 50<k\le 70 \\ (60+0.5\cdot(k-70))°C & for\ \ 70<k\le 90 \\ 70°C & for\ \ 90<k\le 120 \end{cases} \tag{22}$$

The tracking performance of the different methods PSO-LSE, GA-LSE and GD-LSE is shown in Figure 11. The corresponding errors of these methods are shown in Figure 12.



(a) **(b)**



(c)

Figure 11. The tracking capability of the different methods (a) (GD-LSE), (b) (GA-LSE) and (c) (PSO-LSE)

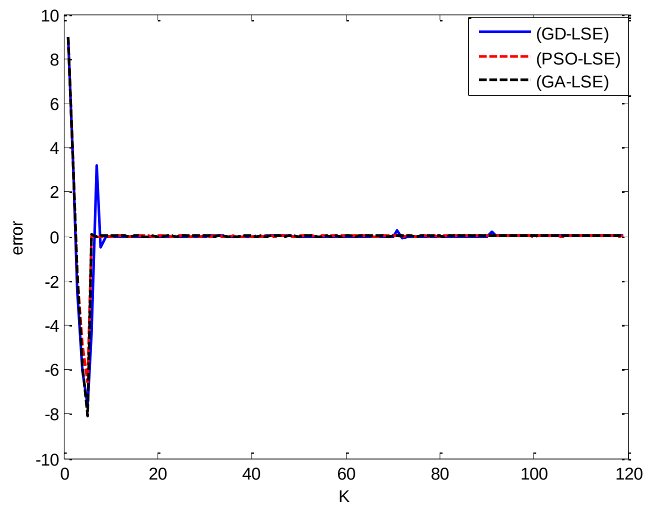


Figure 12. The error curves of the different methods

We also compare the performance of those methods with other existing in literature HELA [32], GA [33], SE [34] and PID. The sum of absolute error is used to test the regulation and tracking performance, it is defined by:

$$\text{sum of absolute error}=\sum_k \left|y_{ref}(k)-y(k)\right| \tag{23}$$

Table 4 details the values of sum of absolute error for the used methods. We can notice that compared with methods used in this work GD-LSE and GA-LSE or other methods HELA, PID, GA and SE in literature applied in the same example, PSO-LSE method reaches lowest sum of absolute error 25.92 for regulation performance and 25.22 for tracking performance.

Nevertheless, the PSO-PSO employing in [36] in the same example gives a lowest sum of absolute error compared to PSO –LSE used in this work. This is because PSO is an evolution algorithm compared to LSE which is a backpropagation one. But the PSO-PSO algorithm used in [36] is not considered as hybrid algorithm such as ANFIS or PSO-LSE.

Table 4. Performance comparison of different methods

Sum of absolute error	HELA [32]	PID	GA [33]	SE [34]	PSO-PSO [36]	GD-LSE	GA-LSE	PSO-LSE
Regulation performance	358.07	418.5	378.02	365.20	25.87	38.83	34.76	25.92
Tracking performance	50.71	100.6	100.22	87.18	22.55	37.5	27.63	25.22

6. Conclusion

In this paper, a hybrid learning algorithm combining meta-heuristic algorithms (GA, PSO) and least-squares method (LSE) was proposed to solve identification and control problems. The optimization technique was applied in the case of inverse control with on-line adaptative learning. Simulation results indicate that PSO algorithm applied to the antecedent part of the ANFIS gives best results compared to the GD and GA methods in terms of RMSE in identification and especially in on-line control of system where the PSO algorithm can be implemented easily to Real Time system.

References

[1] D. Nauck, "Neuro-fuzzy systems: review and prospects", European congress on Intelligent Techniques and Soft Computing EUFIT'97, Aachen, Germany, **(1997)**, pp. 1044-1053.

[2] Y. Hayashi and J. J. Buckley, "Approximations Between Fuzzy Expert Systems and Neural Networks", International Journal of Approximate Reasoning, vol. 10, **(1994)**, pp. 63-73.

[3] H. Li and C. L. P. Chen, "The Equivalence Between Fuzzy Logic Systems and Feed forward Neural Networks", IEEE Transactions on Neural Networks, vol. 11, no. 2, **(2000)**, pp. 356-365.

[4] Y. Zhou, S. Li and R. Jin, "A new fuzzy neural network with fast learning algorithm and guaranteed stability for manufacturing process control", Fuzzy Sets Syst, **(2002)**, pp. 201–216.

[5] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modelling and control", IEEE , Trans. Syst. Man Cybern., **(1985)**, pp. 116–132.

[6] J. Shing and R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", IEEE, Transaction on systems, Man, and Cybernetics, vol. 23, no. 3, **(1993)**.

[7] D. P. Mandic and J. A. Chambers, "Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures, and Stability", Wiley, New York, **(2001)**.

[8] D. E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning", Addison-Wesley, Reading, MA, **(1989)**.

[9] J. K. Koza, "Genetic Programming On the Programming of Computers by Means of Natural Selection", MIT Press, Cambridge, MA, **(1992)**.

[10] L. J. Fogel, "Evolutionary programming in perspective The top-down view", Computational Intelligence Imitating Life, IEEE Press, Piscataway, NJ, **(1994)**.

[11] I. Rechenberg, "Evolution strategy. Computational Intelligence Imitating Life", IEEE Press, Piscataway, NJ, **(1994)**.

[12] R. Eberchart and J. Kennedy, "A new optimizer using particle swarm theory", Proc. of 6th Int. Sym. on Micro Machine and Human Science, Nagoya, Japan, **(1995)**, pp. 39–43.

[13] C. L. Karr, "Design of an adaptative fuzzy logic controller using a genetic algorithm", Proceedings of the 4th International Conference on Genetic Algorithms, **(1991)**, pp.450-457.

[14] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms", IEEE, Transactions on Fuzzy Systems, vol. 3, **(1995)**, pp.129-139.

[15] W. Y. Wan and Y. H. Li, "Evolutionary learning of BMF fuzzy neural networks using a reduced form genetic algorithm", IEEE, Transactions on Systems, Man and Cybernetics, Part B, vol. 33, **(2003)**, pp. 966-976.

[16] C. F. Juang, J. Y. Lin and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design", IEEE, Transactions on Systems, Man and Cybernetics, **(2000)**, Part B, vol.30, pp. 290-302.

[17] O. Cordon, F. Gomide, F. Herrera, F. Hoffman and L. Magdalena, "Ten years of genetic fuzzy systems Current framework and new trends", Fuzzy Sets and Systems", vol.141, no. 1, **(2004)**, pp. 5–31.

[18] W. Wei, "Synthèse d'un controleur flou par Algorithme Génétique: Application au règlage dynamique des paramètres d'un système", Université de Lille, France, **(1998)**.

[19] G. E. Goldberg, "Algorithmes génétiques", Editions Addison-Wesley, Paris, **(1994)**.

[20] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference On Neural Networks, **(1995)**, pp. 1942-1948.

[21] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", IEEE, Trans Syst. Man Cybern, vol. 34, no. 2, **(2004)**, pp. 997–1006.

[22] R. Eberchart and J. Kennedy, "A new optimizer using particle swarm theory", Proc. of 6th Int. Sym. on Micro Machine and Human Science", Nagoya, Japan, **(1995)**, pp. 39–43.

[23] C. F. Juang and C. Y. Wang, "A self-generating fuzzy system with ant and particle swarm cooperative optimization", Expert Systems with Applications, **(2009)**, pp. 5362–5370.

[24] C. F. Juang, I. F. Chung and S. K. Chen, "Group-Based Evolutionary Swarm Intelligence for Recurrent Fuzzy Controller Design", Fuzzy Systems, 2006 IEEE International Conference On, **(2006)**, pp. 1710 – 1714.

[25] C. F. Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design", IEEE Trans Syst Man Cybern B Cybern, vol. 34, no. 2, **(2004)**, pp. 997-1006.

[26] G. Zhang, M. Dou and S. Wang, "Hybrid Genetic Algorithm with Particle Swarm Optimization Technique", International Conference on Computational Intelligence and Security, **(2009)**, 978-1-4244-5411-2.

[27] V. Oliveira and R. Schirru, "Applying particle swarm optimization algorithm for tuning a neuro-fuzzy inference system for sensor monitoring", Progress in Nuclear Energy, vol. 51, **(2009)**, pp. 177-183.

[28] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh and M. A. Khamesar, "Identification using Anfis with intelligent hybrid stable learning algorithm approaches and stability analysis of training method", Applied Soft Computing, **(2009)**, pp. 833-850.

[29] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Trans. Neural Netw, **(1990)**, pp. 4-27.

[30] J. Tanomaru and S. Omatu, "Process control by self-constructing trained neural controllers", IEEE, Trans. Ind. Electron, **(1992)**, pp. 511-521.

[31] G. Hanifi and S. Abdulkadir, "Comparison of clustering algorithms for analog modulation Classification", Expert Systems with Applications, **(2006)**, pp. 642–649.

[32] C. J. Lin and Y. J. Xu, "A hybrid evolutionary learning algorithm for TSK-type fuzzy model Design", Mathematical and Computer Modelling, vol. 43, **(2006)**, pp. 563–581.

[33] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm", Proc. Fourth Int. Conf. Genetic Algorithms, **(1991)**, pp. 450–457.

[34] C. F. Juang, J. Y. Lin and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design", IEEE Trans. Syst. Man Cybern. B, **(2006)**, pp. 290–302.

[35] S. Bouzaida, M. Turki, A. Sakly and F. M'sahli, "Meta-heuristic Methods for Tuning a Neuro-Fuzzy Inference System", The 12th International conference on Sciences and Techniques of Automatic control & computer engineering, STA 2011, Tunisia, **(2011)**.

[36] M. Turki, S. Bouzaida, A. Sakly and F. M'sahli, "Adaptive Control of Nonlinear System Using Neuro-Fuzzy Learning by PSO Algorithm", Proceedings of the 16th IEEE Mediterranean Electrotechnical Conference, MELECON 2012, Tunisia, **(2012**), pp. 519 – 523.

Authors

Mourad TURKI

He was born in Tunisia in 1982. He received the Electrical Engineering diploma in 2006 from the National Engineering school of Monastir. He is currently a phD student at National Engineering School of Monastir, and Technological Teacher in Higher Institute of technological studies of Mahdia. His research interests include Modeling, Intelligent Control of Nonlinear Systems based on Soft Computing techniques and Implementation in embedded system.

Sana BOUZAIDA

She was born in Tunisia in 1986. She received the Electrical Engineering diploma in 2010 from the National Engineering school of Monastir. She is currently a phD student at National Engineering School of Monastir, and Technological Teacher in Higher Institute of technological studies of Nabeul. Her research interests include Modeling, Adaptive Control of Nonlinear Systems based on Soft Computing techniques.

Anis SAKLY

He was born in Tunisia in 1970. He received the Electrical Engineering diploma in 1994 from National Engineering School of Monastir (ENIM), then the PhD degree in Electrical Engineering in 2005 from National Engineering School of Tunis (ENIT). He is currently Professor at ENIM. His research interests are in analysis and synthesis of intelligent control systems, particularly soft computing-based control approaches.

Faouzi M'SAHLI

He received his Mastery of Science and DEA from ENSET, Tunis, Tunisia in 1987 and 1989, respectively. In 1995 he obtained his Doctorate Degree in Electrical Engineering from ENIT, Tunisia. He is currently Professor of Electrical Engineering at National School of Engineers, Monastir, Tunisia. His research interests include Modeling, Identification, Predictive and Adaptive Control of Linear and Nonlinear Systems. He has published over 80 technical papers and co-author of a book “Identification et Commande Numérique des Procédés Industriels” Technip editions, Paris, France.

