International Journal of Control, Energy and Electrical Engineering (CEEE)
Vol.7 pp.53-58

Design of Fuzzy Controller rule base using Bat Algorithm

Nesrine TALBI
Department of electronic, LAJ Research Laboratory
Faculty of sciences and Technology, Jijel University
Jijel, ALGERIA
t_nesrine2003@yahoo.fr

Abstract— The work in this paper revolves fundamentally around the main axes of fuzzy control of the type Takagi-Sugeno (T-S) zero order for dynamic, complex nonlinear systems. In this paper, we present method for designing Fuzzy controller rule base using a new swarm intelligence algorithm, which is based on the Bat algorithm. The Bat algorithm is one of the most recent swarm intelligence based algorithms that simulates the intelligent hunting behavior of the bats found in nature. The main objective is to design the fuzzy rule base of fuzzy controller respecting the desired performance. To demonstrate the efficiency of the suggested approach, a control of a Magnetic Ball Suspension System is selected. Simulation results shows that the proposed approach could be employed as a simple and effective optimization method for achieving optimum determination of fuzzy rule base parameters.

Keywords- Fuzzy rule base, Bat algorithm, Fuzzy controller, Magnetic Ball Suspension System.

I. INTRODUCTION

Fuzzy logic is an important research topic on which focus many scientists. Its theoretical basis was formulated in 1965 by Professor Lotfi A. Zadeh, of the University of California, Berkeley [1]. He introduced the notion of a fuzzy subset to provide a means of representing and manipulating imperfectly described, vague or inaccurate knowledge.

At that time, the theory of fuzzy logic was not taken seriously except by some experts. As early as 1975, Mamdani and Assilian published the first results allowing this theory to be used in control systems [2]. By using a relatively simple controller structure, they have obtained better results when controlling certain processes than those provided by a standard PID controller.

Shortly after, in 1977, the Danish Ostergaard [3] applied fuzzy logic to the control of grinding tubes for the manufacture of cement. At that time, most studies of fuzzy logic control systems were conducted in Europe [4]. From around 1985, it was the Japanese [5] who began to make extensive use of fuzzy logic in industrial and consumer products to solve control and adjustment problems.

The fuzzy controllers make it possible to control complex systems or difficult to model them using a reasoning method of "if *condition* then *action*". Fuzzy controller (FC) depends mainly on the characteristics of three subsystems: fuzzification, fuzzy rules and defuzzification [6]. The fuzzification phase is perfectly specified when the membership functions of the linguistic terms describing the inputs are defined.

The Design of a fuzzy rule base is the process that led to the formalization in the form of rules and/or learned relations, from a set of examples between the inputs and outputs of a process. In many cases, the structure is determined empirically by choosing a priori the type of relational approximate reasoning, the number of fuzzy sets for each input variable, and taking all possible combinations to build the fuzzy rule base. So, it is important to mention the difficulty of ensuring consistency & interpretability of fuzzy rules, in particular for multivariable systems where the number of rules becomes very high [7].

The adjustment by successive trial of fuzzy controller parameters is quite long and tedious. Various techniques of optimization and learning fuzzy controllers have been developed. They are mostly based on a learning that makes it possible to iteratively define the best set of parameters for a given controller structure. For now, researchers are mainly focused on the following approaches:

- Optimization of membership functions,

- Optimization of fuzzy rules,

- Simultaneous optimization of membership functions and fuzzy rules.

several researches have been established to optimize the membership functions of a fuzzy controller using genetic algorithms [8]-[12], Thrift is the first to describe a method for optimizing fuzzy rules by genetic algorithm, using three bits to encode each rule [13], Lee and Takagi propose, in 1993, an optimization method that simultaneously takes into account the membership functions of fuzzification and the fuzzy rules [14]. several methods using the same concept were published [15][16]. Another technique of tuning fuzzy systems was the use of swarm intelligence [17]-[20].

The vast majority metaheuristic algorithms have been derived from the behavior of biological systems and/or physical systems in nature. For example, particle swarm

ISSN 2356-5608

optimization is inspired by social behavior of animals moving in swarm [21], while simulated annealing is based on the annealing process of metals [22]. New algorithms are also emerging recently, including harmony search [23], the firefly algorithm [24], and the Bat algorithm (BA) [25], [26]. The BA technique is introduced by Yang in 2010. It is based on the echolocation behavior of bats. The capability of echolocation of micro-bats is fascinating as these bats can find their prey and discriminate different types of insects even in complete darkness [27]. Bat algorithm was successfully applied to a number of very different problems [28]-[31]. BA is simple to implement and produces good results [31].

This paper presents a simple and effective method using Bat algorithm for designing a fuzzy controller of type Takagi-Sugeno zero order by optimizing the centers of membership functions and the fuzzy rule base. The remaining of this paper is organized as follows: The structure of fuzzy controller to be optimized is described in the next section. After, the standard BA algorithm is briefly presented in the section III. Section IV explained the method of designing the fuzzy controller by BA. The test of the effectiveness of the proposed method is made in section V. Finally, the paper is ended by a conclusion.

II. FUZZY CONTROLLER STRUCTURE

This section describes the fuzzy controller (FC) to be designed in this study. The FC is of type Takagi–Sugeno zero order. The i^{th} rule, which is denoted as R_i , is represented in the following form:

$$R_i: If\ e(k) is\ A_{i1}\ and\ \Delta e(k)\ is\ A_{in}$$
$$Then\ u(k)\ is\ o_i \tag{1}$$

where k is the time step, $e(k), \Delta e(k)$ are the input variables, $u(k)$ is the fuzzy controller output variable, A_{ij} is a fuzzy set, and o_i is a crisp value. A_{ij} is a fuzzy set which uses a triangular membership function defined by following equation:

$$\mu(x) = \max(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0) \tag{2}$$

where a, b and c represent respectively the locations of starting point, peak point and the ending point, for a triangle shaped membership function.

About the fuzzy rule base, the decision on the number of fuzzy rules is a very important issue because it plays a very important role in fuzzy control systems. Unfortunately, there is no systematic and effective procedure for selecting the number of the most appropriate rules, except for some proposals. A reasonable number of fuzzy rules, without losing too much information about the system to be controlled must be carefully obtained.

For flexibility of the implementation of fuzzy controller, the universe of discourse of inputs and output is limited to a range of [-1, 1], determined by the normalization of inputs and output [32]. To do this, the scale factors are used to have the desired dynamics.

This paper proposes a fuzzy rule base composed of only three enabled rules, extracted from analysis expressed as follows:

$\boldsymbol{R_1}: If\ e(k)\ is\ N(-1, a_1, a_2)\ and\ \Delta e(k)\ is\ N\ (-1, b_1, b_2)$

$then\ u(k) = o_1.$

$\boldsymbol{R_2}: If\ e(k)\ is\ Z(a_1, a_2, a_3)\ and\ \Delta e(k)\ is\ Z(b_1, b_2, b_3)$

$then\ u(k) = o_2.$

$\boldsymbol{R_3}: If\ e(k)\ is\ P(a_2, a_3, 1)\ and\ \Delta e(k)\ is\ P(b_2, b_3, 1)$

$then\ u(k) = o_3.$

where $e(k)$ is the difference between the desired output and the measured output of the controlled system. N, Z and P are fuzzy sets of input variables, and o_1, o_2, o_3 are real values of fuzzy controller output.

In the inference mechanism, the AND in the fuzzy rule is implemented by the algebraic product in the theory of fuzzy logic (according to Larsen) .Thus, given a set of input data $\vec{x} = (e, \Delta e)$, the degree of activation $\gamma_i(\vec{x})$ of Rule i is calculated by:

$$\gamma_i(\vec{x}) = \mu_{A_i}(e(k)) \cdot \mu_{B_i}(\Delta e(k)) \tag{3}$$

If there are n_r rules in fuzzy controller, the resulting output of the set of rules is given by the average of weighted individual outputs as follows:

$$u = \frac{\sum_{i=1}^{n_r} \gamma_i(\vec{x}). o_i}{\sum_{i=1}^{n_r} \gamma_i(\vec{x})} \tag{4}$$

where o_i is the value of the conclusion of the i^{th} rule.

In this paper, the optimization of the fuzzy controller includes the determination of all parameters of each fuzzy rule.

III. BAT ALGORITHM

Bat algorithm is a metaheuristic optimization algorithm developed by Xin-She Yang in 2010 [25]. The algorithm is based on echolocation of micro-bats with varying pulse rates of emission and loudness. Bats use sonar echoes to detect and avoid obstacles: they determine the size of an object, how far away they are, how fast they are travelling and even their texture, all in split in a second. After hitting and reflecting, bats transform their own pulse to useful information to gauge how far away the prey is. Bats are using wavelengths, that vary from range [0.7, 17] mm or inbound frequencies [20,500] kHz. By implementation, pulse frequency and rate has to be defined. Pulse rate can be simply determined from range 0 to 1, where 0 means there is no emission and by 1, bats are emitting maximum [33]- [35].

In order to transform these behaviors of bats to algorithm, Yang used three generalized rules [25]:

1) All bats use echolocation to sense distance, and they also guess the difference between food/prey and background barriers in some magical way.
2) Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission r from [0, 1], depending on the proximity of their target.

ISSN 2356-5608

3) Although the loudness can vary in many ways, we assume that the loudness varies from a positive large value A_0 to a minimum constant value A_{min}.

The initial position x_i, velocity v_i and frequency f_i are initialized for each bat. For each time step t , the movement of the virtual bats is given by updating their velocity and position using the following equations:

$$f_i = f_{min} + (f_{max} - f_{min}).\rho \quad (5)$$

$$v_i^j(t) = v_i^j(t-1) + \left[x_{gbest}^j - x_i^j(t-1)\right].f_i \quad (6)$$

$$x_i^j(t) = x_i^j(t-1) + v_i^j(t) \quad (7)$$

where $\rho \in [0,1]$ indicates randomly generated number, and x_{gbest}^j represents current global best solutions.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk:

$$x_i^{new} = x_i^{old} + \sigma.A_{mean}^{old} \quad (8)$$

Where, $\sigma \in [0,1]$ is a random number and represents direction and intensity of random-walk. A_{mean}^{old} is the average loudness of all the bats.

Based on these approximations and idealization, the basic steps of the Bat Algorithm (BA) can be summarized as the following pseudo code [26]:

```
Objective function f(x), x = (x_1, ..., x_d)^T
Initialize the bat population x_i and v_i for i = 1,n
Define pulse frequency Q_i ∈ [Q_min, Q_max]
Initialize pulse rates r_i and the loudness A_i
while (t < T_max) // number of iterations
   Generate new solutions by adjusting frequency, and
   updating velocities and locations/solutions
   if (rand(0,1) > r_i )
         Select a solution among the best solutions
         Generate a local solution around the best solution
   end if
   Generate a new solution by flying randomly
   If (rand(0,1) < A_i and f(x_i) < f(x))
              Accept the new solutions
              Increase r_i and reduce A_i
   end if
    Rank the bats and find the current best
end while
Post process results and visualization
```

IV. Designing Fuzzy Rule Base of Fuzzy Controller using Bat Algorithm

In this section, we propose to use Bat algorithm for designing a reduced fuzzy rule base of a fuzzy controller of type Takagi-Sugeno zero order, in order to obtain better performances of the system to control.

A. Optimization Procedure

The diagram of the closed loop control is shown schematically in Fig. 1. We can summarize the optimization procedure of the fuzzy controller using Bat algorithm through the following steps:

1. Generation of an initial population of solutions characterizing the controller settings.
2. For all solutions:
   - Evaluate the objective function.
   - Classify obtained solutions according to their fitness.
   - Construction of a new population by updating process of frequencies, velocities and solutions.

The step 2 is repeated until a maximum number of iterations is performed. After the process of evolution, the final iteration of the algorithm consists of the well-adjusted solution who provide best solution.

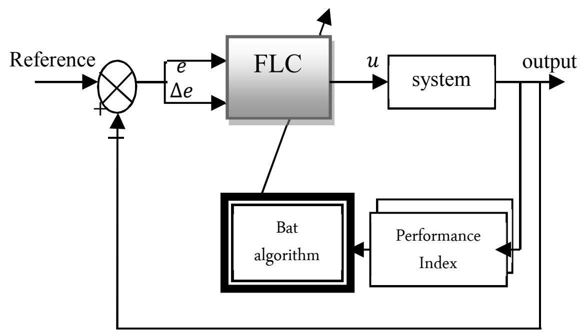

Fig.1. control structure and optimization

The inclusion of design constraints in the optimization process helps to preserve the semantics of fuzzy rules. For that, the constraints on the limits of the parameter vector to be identified, and limits on the control variables are imposed.

B. Parameters's Vector Representation of Fuzzy Controller

The parameter vector (solution) x of the fuzzy controller has nine parameters. These parameters represent the starting point locations, the pic point, and end point, for a triangular membership function belonging to the inputs of a fuzzy controller and fuzzy singleton for its release.

So $$x = [\, a_1\; a_2\; a_3\; b_1\; b_2\; b_3\; o_1\; o_2\; o_3 \,] \quad (9)$$

While respecting the following constraint:

$$\begin{cases} a_1 < a_2 < a_3 \\ b_1 < b_2 < b_3 \\ o_1 < o_2 < o_3 \end{cases} \tag{10}$$

C. Generation of Initial Population of solutions

Initial population is randomly generated from real-valued vectors with dimension d and number of bats n, by taking into account lower and upper boundaries. In this study, $d = 9$.

$$x_i^j = x_{min}^j + rand(0,1) * (x_{max}^j - x_{min}^j) \tag{11}$$

where $i = 1, n; j = 1, d$ and x_{min}^j and x_{max}^j are lower and upper boundaries for dimension j respectively.

V. APPLICATION

The test of effectiveness of the Bat algorithm is illustrated to designing fuzzy controllers of type Takagi-Sugeno zero order for control Magnetic Ball Suspension System.

The Bat algorithm parameters used in this section are given in Table I.

TABLE I
BA Control parameters values

Parameter	Designation	value
n	Population size	20
ng	Number of generation	100
f_{min}	Minimum frequency	0
f_{max}	Maximum frequency	1
A	Loudness	0.5
r	Pulse rate	0.5

A. Magnetic Ball Suspension System

The model of the magnetic ball suspension system shown in Fig. 6 is given by [36] :

$$M * \frac{d^2y(t)}{dt^2} = M.g - \frac{i^2(t)}{y(t)}$$
$$v(t) = R * i(t) + L * \frac{di(t)}{dt} \tag{12}$$

where $y(t)$ is the ball position, $M = 0.1$ kg is the ball mass, $g = 9.8$ m/s^2 is the gravitational acceleration, $R = 50\ \Omega$ is the winding resistance, $L = 0.5$ Henrys is the winding inductance, $v(t)$ is the input voltage, and $i(t)$ is the winding current. The position of the ball is detected by a position sensor (e.g., an infrared, microwave, or photo resistive sensor) and is assumed to be fully detectable over the entire range between the magnetic coil and the ground level. The ball will stay between the coil and the ground level [37]:

$$\begin{cases} \dfrac{dx_1(t)}{dt} = x_2(t) \\ \dfrac{dx_2(t)}{dt} = g - \dfrac{x_3^2(t)}{Mx_1(t)} \\ \dfrac{dx_3(t)}{dt} = -\dfrac{R}{L}\, x_3(t) + \dfrac{1}{L}\, v(t) \end{cases} \tag{13}$$

where $[x_1(t),\ x_2(t),\ x_3(t)] = [y(t),\ \frac{dy(t)}{dt},\ i(t)]$. Notice that the nonlinearities are induced by the $x_3^2(t)$ and $\frac{1}{x_1(t)}$ terms in the $\frac{dx_2(t)}{dt}$ equation. By linearizing the plant model in Equation (13), assuming that the ball is initially located at $x_1(t) = y(0)$, we can find a linear system by calculating the Jacobian matrix at $y(0)$. The linear state-space form of the magnetic ball suspension system is given as [37]:

$$\begin{cases} \dfrac{dx_1}{dt} = x_2(t) \\ \dfrac{dx_2(t)}{dt} = \dfrac{g}{y(0)} x_1(t) - 2\sqrt{\dfrac{g}{My(0)}}\ x_3(t) \\ \dfrac{dx_3(t)}{dt} = -\dfrac{R}{L}\, x_3(t) + \dfrac{1}{L}\, v(t) \end{cases} \tag{14}$$

The objective of this section is to control the position of magnetic levitation magnetic ball. The controller used is of fuzzy controller where its output is the command $u(t)$.

We consider a fuzzy controller of the type Takagi-Sugeno zero order, its inputs are the error $e(t)$ and its variation $\Delta e(t)$and an output $u(t)$ (the command applied to the system).

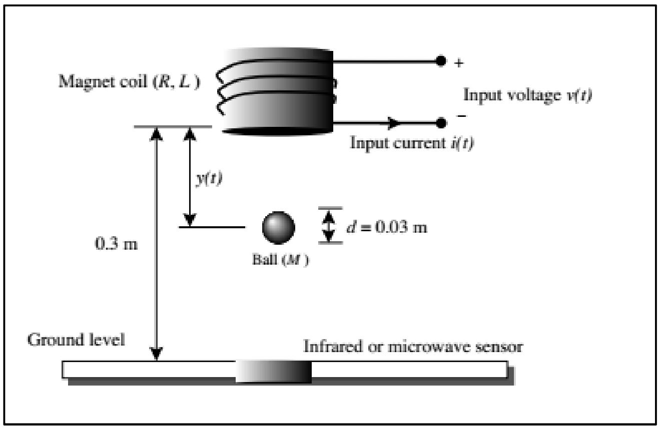

Fig.2. Magnetic ball suspension system [37]

The cost function is the mean square error calculated by the following equation:

$$MSE = \frac{1}{nT}\sum_{k=1}^{n} e^2(k) \qquad (15)$$

Where: n is the total number of samples and T the sampling time, $e(t)$ is the difference between the value of the desired output and the value of the measured output process under control.

After 100 generations, we obtain the best parameter vector which gives best performance of the system with only three rules extracted for analysis. Figure 7 shows the evolution of the cost function. At the end of program execution, the evaluation function (mean square error) is equal to 2.66×10^{-6} and the fuzzy rules obtained are as follows:

$Rule\ 1: if\ e\ is\ N\ (-1, -0.99, 0.69)\ and\ \Delta e\ is\ N\ (-1, -0.72, 0.16)\ then\ u = -1.04$

$Rule\ 2: if\ e\ is\ Z\ (-0.99, 0.69, 0.99)\ and\ \Delta e\ is\ Z\ (-0.72, 0.16, 0.79)\ then\ u = -0.08$

$Rule\ 3: if\ e\ is\ P(0.69, 0.99, 1.00)\ and\ \Delta e\ is\ P\ (0.16, 0.79, 1.00)\ then\ u = 1.02$

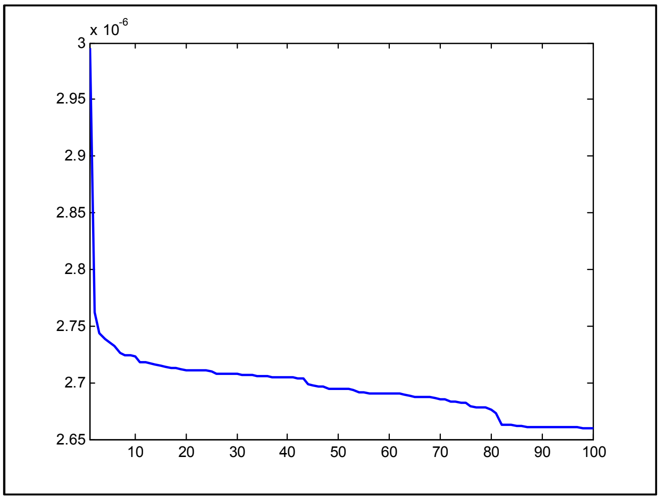

Fig.3. Evolution of fitness function

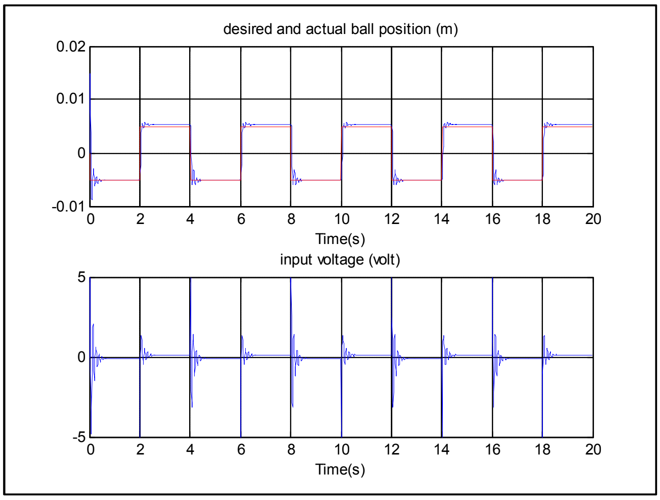

Fig.4 Responses of the system via nominal parameters

The results shows that the optimized fuzzy controller of electromagnetic voltage can stabilize the disturbance that would cause the metal sphere (ball) to fall or attach it to the electromagnet..

VI. Conclusion

In this paper, BAT algorithm is used to design a fuzzy controller for Takagi-Sugeno zero-order, The difficulty of obtaining the rule base and membership functions is indeed very disadvantageous when using fuzzy techniques. the BAT algorithm optimizes simultaneously:

- The membership functions of the input and output variables of the controller,
- The conclusions of fuzzy rules which are the base of rules itself, since the fuzzy controller is type of T-S zero order. The fuzzy rule base is composed of three rules extracted from analysis.

To demonstrate the effectiveness of the presented approach, a control of a magnetic ball levitation system is selected. Simulation results show that the proposed approach presented here is a powerful tool for the control of nonlinear systems. We can say that the BAT algorithm present a very powerful tool for the design of intelligent controllers.

References

[1] Zadeh, L. A. Soft computing and fuzzy logic. *IEEE Software*, 11(6), pp 48-56, 1994.

[2] Mamdani, E.H., Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Studies*, 7(1) : 1–13, 1975.

[3] Ostergaad, J.J. Fuzzy logic control of a heat exchange process. *in Fuzzy Automata and Decision Processes, M.M. Gupta, G.N. Saridis, and B.R. Gaines,Eds.*, pages 285–320, 1977.

[4] Willaeys, D., Malvache, N. Use of fuzzy model for process control. *IEEE International Conference on Cybernetics and Society*, 1978.

[5] Tamakawa, T. High speed fuzzy controller hardware system. *Proc. 2nd Fuzzy System Symp.*, pages 122–130, 1986.

[6] Jantzen, J. Foundations of Fuzzy Control. *John Wiley & Sons*, 2007

[7] M.S. Ju, D.L. Yang, "Design of adaptive fuzzy control based on natural control laws", Fuzzy Sets and Systems, Vol. 81, 1996, pp. 191-204.

[8] Karr, C.L. Design of an adaptive fuzzy linguistic controller using a genetic algorithm, *Proc. of the Fourth Int. Conf. Genetic Algorithms*, pages 450–457, 1991.

[9] Karr, C.L., Gentry, E.J. Fuzzy control of pH using genetic algorithms. *IEEE Trans. On Fuzzy Systems*, 1(1) : 46–53, 1993.

[10] Herrera, F., Lozano, M., Verdegay, J.L. Turning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning*, 12 : 299–315, 1995

[11] Shimojima, K., Fukuda, T., Hasegawa, Y. Self-turning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. *Fuzzy Sets and Systems*, 71 : 295–309, 1995.

[12] Liska, J., Melsheimer, S.S. Complete design of fuzzy logic systems using genetic algorithms. *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pages 1377–1382, 1994.

ISSN 2356-5608

[13] Thrift, P. Fuzzy logic synthesis with genetic algorithms. *Proc. of the Fourth Int. Conf.Genetic Algorithms*, pages 509–513, 1991

[14] Lee, M.A., Takagi, H. Dynamic control of genetic algorithms using fuzzy logic techniques. *Proceedings of the 5th Int'l Conf. on Genetic Algorithms*, pages 76–83, 1993.

[15] C.C. Chen and C.C. Wong, "Self-generating rule-mapping fuzzy controller design using a genetic algorithm", *In IEE Proceedings on Control Theory and Applications,* vol. 49, 2002, pp. 143–148.

[16] K. Belarbi, F. Titeli, W. Bourebia et K. Benmahammed, "Design of mamdani fuzzy logic controllers with rule base minimisation using genetic algorithm", *Engineering applications of artificial intelligence*, vol. 18, 2005, pp. 875–880.

[17] N. Talbi and K. Belarbi, "Optimization of Fuzzy Controller using Hybrid Tabu Search and Particle Swarm Optimization", *in Proc. IEEE, 11th International Conference on Hybrid Intelligent Systems (HIS)*, Malaysia, 2011, pp. 561-565.

[18] V. Mukherjee., .P. Ghoshal, "Intelligent particle swarm optimized fuzzy PID controller for AVR system". Electric Power Systems Research, Vol. 77, 2007, pp. 1689-1698.

[19] N. Talbi and K. Belarbi, "Designing Fuzzy Controllers for a Class of MIMO Systems using Hybrid Particle Swarm Optimization and Tabu Search", *International Journal of Hybrid Intelligent Systems, IOS Press,* vol. 10, n°1, 2013, pp. 1–9.

[20] C. F. Juang, P. H. Chang, "Designing fuzzy-rule-based systems using continuous Ant colony optimization", IEEE trans., *Fuzzy Syst.*, vol. 18, no. 1, 2010, pp. 138-149.

[21] J. Kennedy et R.C. Eberhart, " Particle swarm optimization", *IEEE International Conference on Neural Networks,* Piscataway*,* 1995, pp. 1942-1948.

[22] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing", Science, vol. 220,1983, pp. 671-680.

[23] Z.W. Geem, J. Kim, G.V Loganathan, "A new heuristic optimization algorithm: Harmony search", Simulation, vol. 76,2001, pp. 60-68.

[24] X.S. Yang, "Nature-inspired Metaheuristic Algorithms", Luniver Press, 2008.

[25] X.S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin,Springer,vol.284, 2010, pp. 65-74.

[26] X.S.Yang and A. H. Gandomi, "Bat Algorithm: A Novel Approach for Global Engineering Optimization", Engineering Computations, Vol. 29, Issue 5, 2012, pp. 464--483.

[27] K. Khan, A. Nikov,A. Sahai, "A fuzzy bat clustering method for ergonomic screening of office workplaces", *Advances in Intelligent and Soft Computin,* Springer , Vol. 101, 2011.; pp. 59–66.

[28] G.Q. Huang,W-J Zhao, Q.Q Lu, "Bat algorithm with global convergence for solving large-scale optimization problem", *Application Research of Computers*, 30(5), 2013, pp. 1323–1328.

[29] J. Zhang, G. Wang. "Image matching using a bat algorithm with mutation", *Applied Mechanics and Materials*, 203(1), 2012, pp.88–93.

[30] A. Alihodzic, M. Tuba, " Improved Hybridized Bat Algorithm for Global Numerical Optimization", *in Proc. IEEE, 16th International Conference on Computer Modelling and Simulation, 2014, pp.57-62.*

[31] K. Premkumar , B.V. Manikandan, « Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor », Engineering Science and Technology, an International Journal, vol. 19 , 2016, pp . 818–840

[32] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. Part 1", *IEEE Transactions on systems Man and Cybernetics*, Vol. 20, 1990, pp. 404-418.

[33] A.H. Gandomi, X.S. Yang, A.H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks", *Neural Computing & Applications*, pages 1–17, 2012.

[34] P.W. Tsai, J.S. Pan, B.Y. Liao, M.J. Tsai, and V. Istanda. "Bat algorithm inspired algorithm for solving numerical optimization problems", *Applied Mechanics and Materials*, 148:134–137,2012.

[35] X.S. Yang, "Review of meta-heuristics and generalised evolutionary walk algorithm", *International Journal of Bio-Inspired Computation*, 3(2):77–84, 2011.

[36] J.T. Spooner, R. Ordonez, and K. M. Passino, "Direct adaptive fuzzy control for a class of discrete time systems", In *Proceedings of the American Control Conf.*, pages 1814–1818, Albuquerque, NM, June 4-6, 1997.

[37] K.M. Passino, S. Yurkovich, "Fuzzy Control", Wesley Longman, 1998.

ISSN 2356-5608