DESIGN AN INTERVAL TYPE-2 FUZZY LOGIC CONTROLLER USING MODIFIED BIOGEOGRAPHY-BASED OPTIMIZATION

M.M.Sayed*, M.S.Saad, H.M. Emara, and E.E. Abou El-Zahab
Electric Power and Machines Dept., Faculty of Engineering Cairo University, Giza, Egypt.
*M.M.Sayed@ieee.org, Tel. (+201093456422), and Fax: (+20237125742)

Abstract: *In this paper we apply modified biogeography-based Optimization to design an interval type-2 fuzzy logic controller to improve the performance of the plant control system. Biogeography-based optimization is a novel evolutionary algorithm that is based on the mathematical models of biogeography. Biogeography is the study of the geographical distribution of biological organisms. In the Biogeography-based optimization model, problem solutions are represented as islands, and the sharing of features between solutions is represented as immigration and emigration between the islands. A modified version of the Biogeography-based optimization is applied to design interval type-2 fuzzy logic controller to get the optimal parameters of the membership functions of the controller. We test the optimal interval type-2 fuzzy logic controller obtained by modified biogeography-based Optimization using benchmark plants and the performance is compared with a Particle swarm optimization-based controller. Also this paper deals with the design of intelligent systems using interval type-2 fuzzy logic for minimizing the effects of uncertainty produced by instrumentation elements, environmental noise, etc. We found that the optimized membership functions for the inputs of a type-2 system to improve the performance of the system for high uncertainty (noise) levels.*

Key words: *Biogeography-Based Optimization, Particle Swarm Optimization, Process Control, and Type 2 Fuzzy Logic*.

1. Introduction

Optimization algorithms are search methods, where the goal is to find a solution to an optimization problem, such that a given quantity is optimized, possibly subject to a set of constraints [1, 2]. Some optimization methods are based on populations of solutions. Unlike the classic methods of improvement for trajectory tracking, in this case each iteration of the algorithm has a set of solutions. These methods are based on generating, selecting, combining and replacing a set of solutions. Since they maintain and they manipulate a set, instead of a unique solution throughout the entire search process, they used more computer time than other met heuristic methods. This fact can be aggravated because the "convergence" of the population requires a great number of iterations. For this reason a concerted effort has been dedicated to obtaining methods that are more aggressive and manage to obtain solutions of quality in a nearer horizon.

This paper is concerned with bio-inspired optimization methods like modified biogeography-based Optimization to design optimized interval type 2 fuzzy logic controller (IT2FLC) for a selection of some benchmark plants. This method is used to find the parameters of the membership functions achieving the optimal IT2FLC for plant control.

This paper is organized as follows: Section 2 reviews Biogeography-based optimization (BBO). Section 3, presents the modified Biogeography-based optimization (MBBO). In Section 4 is proposed an overview about type 2 fuzzy logic controller and, section 5 introduce the controller design where a MBBO is used to select the parameters. Robustness properties of the closed-loop system are achieved with a type-2 fuzzy logic control system using a Takagi-Sugeno model where the error and the change of error, are considered the linguistic variables. Section 6, presents the benchmark plants, performance criteria and provides a simulation study of the plant using the controller described in Sections 4 and 5. Finally, the conclusions are stated in Section 7.

2. Biogeography-Based Optimization

As its name implies, BBO is based on the science of biogeography. Biogeography is the study of the distribution of animals and plants over time and space. Its aim is to elucidate the reason of the changing distribution of all species in different environments over time. As early as the 19th century, biogeography was first studied by Alfred Wallace [3] and Charles Darwin [4]. After that, more researchers began to pay attention to this area. The environment of BBO corresponds to an archipelago, where every possible solution to the optimization problem is an island. Each solution H has a number of features called a suitability index variable (SIV). The number of SIV in each solution H corresponds to the problem dimension. The goodness of each solution is called its habitat suitability index (HSI), where a high HSI of an island means good performance on the optimization problem, and a low HSI means bad performance on the optimization problem.

Improving the population is the way to solve problems in heuristic algorithms. The method to generate the next generation in BBO is by emigrating solution features to other islands, and receiving solution features by immigration from other islands. The algorithm assumes high species' count in island having high HSI (i.e., for island corresponding to good solutions). The high species' count encourages species to leave the island sharing their good SIV with other island. Hence, islands with good HSI have high emigration rate and low immigration rate. Bad solutions (islands with low HSI) have small species count, low emigration rates and high immigration rates. Mutation is performed for the whole population in a manner similar to mutation in GAs. The basic procedure of BBO is as follows:

1. Define the island modification probability, mutation probability, and elitism parameter. Island modification probability is similar to crossover probability in GAs. Mutation probability and elitism parameter are the same as in GAs.
2. Initialize the population (n islands).
3. Calculate the immigration rate (rate of species arrival to an island) and emigration rate (rates of species departing from an island) for each island.
4. Probabilistically choose the immigration islands based on the immigration rates. Use roulette wheel selection based on the emigration rates to select the emigrating islands.
5. Migrate randomly selected SIVs based on the selected islands in the previous step.
6. Probabilistically perform mutation based on the mutation probability for each island.
7. Calculate the fitness of each individual island.
8. If the termination criterion is not met, go to step 3; otherwise, terminate.

In 2008, BBO performance was tested and compared with seven other evaluation algorithms (EA) optimization techniques (including PSO, genetic algorithm (GA) and five other different methods). Comparison of the performance of BBO with other standard methods using fourteen benchmarks optimization problems shows significant advantages of the BBO algorithm [5]. The author also provides an example of the use of BBO in engineering application by applying it to the design of aircraft engine [5]. After 2008, oppositional Biogeography-Based Optimization for Combinatorial Problems is developed by Mehmet Ergezer and Dan Simon [6]. Biogeography-Based Optimization with Blended Migration for Constrained Optimization Problems is introduced by Haiping Ma and Dan Simon [7]. A Hybrid Differential Evolution with Biogeography-Based Optimization (DE/BBO) for Global Numerical Optimization is proposed by Wenyin Gong and colleagues [8]. Equilibrium species counts and migration model tradeoffs for Biogeography-Based Optimization is developed by Haiping Ma and colleagues [9]. The BBO is applied to electric power system applications like the solution of the power flow problem as introduced by Rick Rarick and colleagues [10].

3. Modified Biogeography-Based Optimization

In BBO, there are two main operators: migration and mutation. The basic BBO provides a discrete solution space ranging from a user specified minimum to a specified maximum and with a specified granularity. The authors proposed a new migration operator called modified migration, in order to improve the convergence performance and provide a continuous solution space for the optimization problems [11].

3.1 Mutation Operator

Mutation is a probabilistic operator that randomly modifies a solution's SIV based on its a priori probability of existence. Namely, a randomly generated SIV replaces a selected SIV in the solution HI according to a mutation probability. Although mutation is not the most important factor in BBO, the improvement of solutions is obtained by perturbing the solution after the migration operation. For classic BBO, the mutation probability is inversely proportional to the solution probability [5], and is defined by

$$m_I = m_{max}\left\{1 - \frac{P_I}{P_{max}}\right\} \quad (1)$$

Where m_{max} is the user-defined maximum mutation probability, $P_{max} = ARG_{MAX}(P_I)$, I=1,……., n (n is population size), and P_I is the solution probability. For more details see [5].This mutation scheme tends to increase diversity among the population.

3.2 Modified Migration Operator

In biogeography, migration is the movement of species between different habitats. In BBO, migration is a probabilistic operator that adjusts each solution H_I (uniquely defined for each island) by sharing features between solutions. In the original BBO work [5], the probability that the solution H_I is selected as the immigrating habitat is proportional to its immigration rate λ_I, and the probability that the solution H_J is selected as the emigrating habitat is proportional to the emigration rate μ_J. Migration can be expressed as:

$$H_I(SIV) \leftarrow H_J(SIV) \quad (2)$$

In biogeography, an SIV is a suitability index variable which characterizes the habitability of a habitat [5]; that is, the HSI is determined by many SIVs. In BBO, an SIV is a solution feature, equivalent to a gene in a GA. In other words, an SIV is a search variable and the set of all possible SIVs is the search space from which an optimal solution will be determined. Eq. (2) shows that a solution feature of solution HI is replaced by a feature from solution H_J. In BBO, each HI resides in an island having its own immigration rate λ_I and emigration rate μ_J. A good solution has relatively high μ and low λ, while the converse is true for a poor solution. The immigration rate and the emigration rate are functions of the fitness of the solution. They can be calculated as:

$$\lambda_I = I\left\{(1 - \frac{K_{(I)}}{n})\right\}, \mu_I = E\left\{\frac{K_{(I)}}{n}\right\}, \quad (3)$$

Where I is the maximum possible immigration rate; E is the maximum possible emigration rate; $k_{(I)}$ is the fitness rank of the Ith individual (I is worst and n is best); and n is the number of candidate solutions in the population. I and E are often set equal to 1, or slightly less than 1. Note that the migration function (3) is a linear curve, but in general it might be a more complicated curve.

Haiping Ma and Dan Simon [7] propose the blended migration operator, which is a generalization of the standard BBO migration operator. The blended migration operator is motivated by blended crossover in GA. In blended crossover, instead of copying a parent's gene to a child chromosome, the off spring are obtained by combining parents' genes, so that equation (2) will be replaced by:

$$H_I(SIV) \leftarrow \alpha H_I(SIV) + (1 - \alpha)H_J(SIV) \quad (4)$$

Where $\alpha \in [0, 1]$

In [7], α is a constant selected by the user. The best results shown in the paper corresponds to α = 0.5. We proposed a new migration operator called modified migration operator defined as [11]:

$$H_I(SIV) \leftarrow \alpha(H_I)H_I(SIV) + \left(1 - \alpha(H_I)\right)H_J(SIV) \quad (5)$$

Where

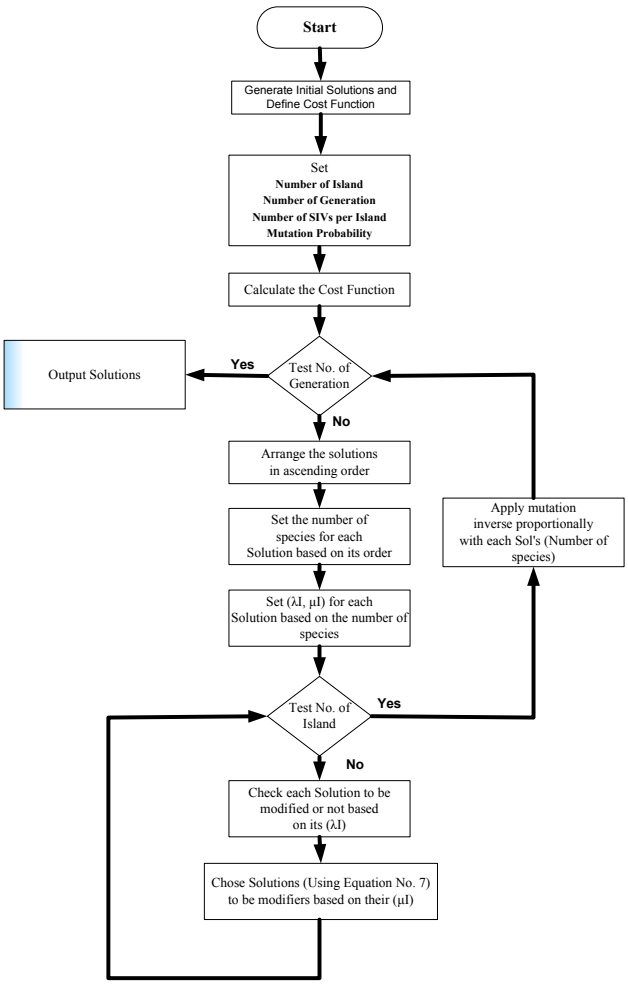
$$\alpha(H_I) = \frac{K_{(I)}}{K_{(I)} + K_{(J)}} \quad (6)$$

$K_{(I)}$: is the fitness rank of the solution.

It should be noted that both equation (4) and (5) can provide a new member in the solution space, while (2) just provide re-construction of the new solution vectors using a selection from a constant pool. When using (2), the mutation operator is the only way to generate new SIV not members of the set of SIVs of the initial population. Hence, an algorithm using (2) may need high initial population count or high mutation rate. Finally after substituting equation (6) in (5) to get the modified migration operator form, defined as:

$$H_I(SIV) \leftarrow \frac{\{K_{(I)}H_I(SIV) + K_{(J)}H_J(SIV)\}}{\{K_{(I)} + K_{(J)}\}} \quad (7)$$

Modified migration is an attractive BBO modification from a couple of different viewpoints. On one hand, good solutions will be less likely to be degraded due to migration. On the other hand, poor solutions can still accept a lot of new features from good solutions. The BBO algorithm, generalized for modified migration, is summarized in the next flow chart as:



4. Interval Type 2 Fuzzy Logic Systems

Similar to type 1 fuzzy sets, type 2 fuzzy sets provide some linguistic information about the variable of interest. Moreover, type 2 fuzzy sets provide information about the uncertainty associated with the linguistic information. For example, if several chemical process operators are asked to

define the range in which they consider the pressure of a reactor to be "Normal", the result obtained from one operator may be represented by the triangular type-1 membership function (MF) depicted in Fig. 1-a.By overlapping the set of MFs obtained from different operators, a blurred triangle is obtained as in Fig. 1-b. To accommodate with the uncertainty of the fuzzy sets, at a specific value of x, say ẋ we can express the MF of ẋ as an interval that takes on values in the interval [a(ẋ), b(ẋ)] where the vertical line intersects the blur. In type 2 Fuzzy sets, we associate a fuzzy set (secondary fuzzy set) that describe the possibility we associate with each value of the interval [a (ẋ), b (ẋ)].The most commonly used form of fuzzy type 2 sets is the interval type 2 Fuzzy sets. In this case, the membership function of the secondary fuzzy set is equal to one for all points inside [a (ẋ), b (ẋ)], and is zero otherwise. To represent interval type 2 fuzzy sets, we only need to identify the interval [a (ẋ), b (ẋ)] for all possible ẋ (since the secondary membership function is known). This is performed by defining the two T1 membership functions that bound the blurred area. These two T1 MFs are usually denoted as UMF (upper membership function) and LMF (Lower MF) as shown in Fig. 2 [12].

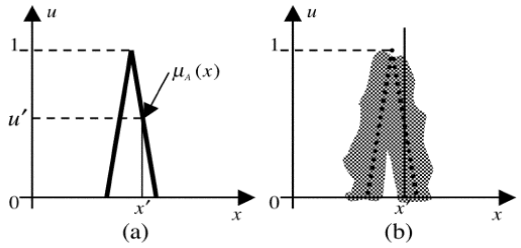


Fig. 1: (a) Type1 MF (b) Blurred type1 MF

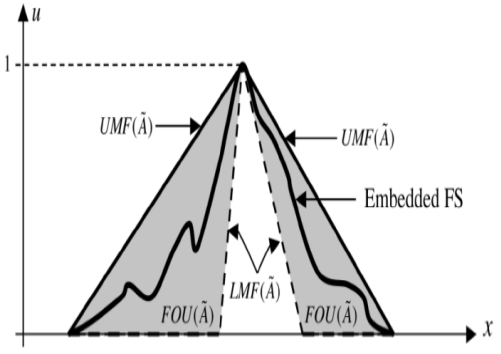


Fig. 2: Upper and Lower Membership Functions

Interval type 2 fuzzy logic systems (IT2FLS) are quite similar to type 1 fuzzy logic systems. The main difference is that the antecedent and/or consequent sets in a type-2 fuzzy logic system are type-2, so that each rule output set is a type-2. There are five principal parts in a type-2 fuzzy logic system: fuzzifier, Rule Base, Inference Engine, Type-Reducer and defuzzifier as shown in Fig. 3 [13].The type-reducer performs a type-reduction operation which is an extended version of type-1 defuzzification. Type reduction yield a type-1 set from the type-2 rule output sets. It was proposed by Karnik and Mendel [14], and [15]. The resulting type-1 set is called type-reduced set.

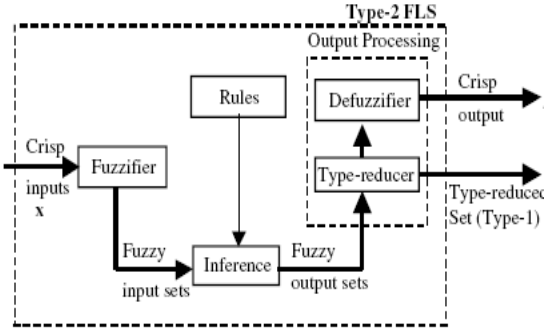


Fig. 3: Type 2 fuzzy logic system block

The type-reduced set can then be defuzzifiedusing conventional type fuzzy system defuzzificattion rules.Type-2 fuzzy rule base consists of a collection of IF-THEN rules [16] in the following form:

$$R^i: \text{IF } x_1 \text{ is } \tilde{F}_1^i \ldots\ldots \text{and } x_n \text{ is } \tilde{F}_n^i \text{ then } y \text{ is } \tilde{G}^i$$

Where $\tilde{F}_j^i$ is the interval type 2 antecedent sets (j=1,2…n), yϵY the output, $\tilde{G}^i$ is the interval type 2 consequent sets and i=1,2…M, where M is the total number of rules of the rule base.In IT2FSC with meet under minimum or product t-norm the firing interval of the i^{th}rule $W^i = [\underline{w}^i, \overline{w}^i]$ can be calculated using equation (8)

$$\begin{aligned} \underline{w}^i &= \underline{\mu}_{\tilde{F}_1^i}(x_1) X \ldots\ldots X \underline{\mu}_{\tilde{F}_n^i}(x_n) \\ \overline{w}^i &= \overline{\mu}_{\tilde{F}_1^i}(x_1) X \ldots\ldots X \overline{\mu}_{\tilde{F}_n^i}(x_n) \end{aligned} \quad (8)$$

Where $\underline{\mu}_{\tilde{F}_k^i}(x_k)$ is the firing of the LMF of $\tilde{F}_k^i$, $\overline{\mu}_{\tilde{F}_k^i}(x_k)$ is the firing of the UMF of $\tilde{F}_k^i$, and X is the t-norm operator.There are several methods of type reduction introduced in [17], and [18]. In this paper we will use the center of sets type reduction [19]. In this case, the type reduced set$Y_{tr} = [y_l, y_r]$is determined by two end points$y_l$and y_r which can be calculated from the following equations (9) and (10).

$$y_l = \frac{\sum_{i=1}^{M} w_l^i y_l^i}{\sum_{i=1}^{M} w_l^i} \quad (9)$$

$$y_r = \frac{\sum_{i=1}^{M} w_r^i y_r^i}{\sum_{i=1}^{M} w_r^i} \quad (10)$$

Where y_l^i and y_r^i are the two end points of centroid interval of interval type 2 consequent as described in [14],$w_l^i \epsilon [\underline{w}^i, \overline{w}^i]$ denotes the firing strength membership grade contributing to the left-most point y_l ,and $w_r^i \epsilon [\underline{w}^i, \overline{w}^i]$ denotes the firing strength

membership grade contributing to the right-most point y_r.w_l^iandw_r^i can be computed using KM algorithms as in [18]. The defuzzified output of y will be the average of y_l and y_r that is present is equation (11)

$$y = \frac{y_l + y_r}{2} \qquad (11)$$

5. IT2FLC Design

In this section we design interval type 2 fuzzy logic controller (IT2FLC) by using MBBO as shown in Fig. 4. For the IT2FLC a Takagi-Sugeno type of fuzzy system is used with two inputs a) error, and b) error change, with Seven membership functions for each input, "High Negative, medium Negative, Negative, Zero, Positive, medium Positive, High Positive" (Gaussian), and one span output, defined with constant values and 49 fuzzy rules (IF-THEN) [12,20-25]. Now, the number of SIVs per island will be reduced from 57 SIV to 3 SIV (real values that represent the span of the two inputs: error "S1"and error change "S2"and one span for output constant value "S0") as indicated in the equations to follow. the standard deviation of the two inputs are

$$\sigma_{1_U} = \frac{S_1}{2 * (n-1) * \sqrt{(2 * Ln(2))}} \qquad (12)$$

$$\sigma_{2_U} = \frac{S_2}{2 * (n-1) * \sqrt{(2 * Ln(2))}} \qquad (13)$$

$$\sigma_{1_L} = \gamma * \sigma_{1_U} \quad (14)$$

$$\sigma_{2_L} = \gamma * \sigma_{2_U} \qquad (15)$$

The mean of the two inputs are

$$\mu_{1_U}(i) = \frac{-S_1}{2} + \frac{(i-1) * S1}{n-1} \qquad (16)$$

$$\mu_{2_U}(i) = \frac{-S_2}{2} + \frac{(i-1) * S2}{n-1} \qquad (17)$$

$$\mu_{1_L}(i) = \mu_{1_U}(i) \qquad (18)$$

$$\mu_{2_L}(i) = \mu_{2_U}(i) \qquad (19)$$

$$Mo(i) = \frac{-S0}{2} + \frac{(i-1) * S0}{n-1} \qquad (20)$$

Where:

S1: Error Span

S2: Error change span

S0: Output span

σ_{1_U}: Standard deviation for upper membership function "Error"

σ_{2_U}: Standard deviation for upper membership function "Error Change"

σ_{1_L}: Standard deviation for Lower membership function "Error"

σ_{2_L}: Standard deviation for Lower membership function "Error Change"

$\mu_{1_U}(i)$: Mean for upper membership function "Error"

$\mu_{2_U}(i)$: Mean for upper membership function "Error Change"

$\mu_{1_L}(i)$: Mean for Lower membership function "Error"

$\mu_{2_L}(i)$: Mean for Lower membership function "Error Change"

$Mo(i)$: Output for membership function

γ: is constant from 0 to 1, set by user. i: From 1 to n ,

Where; n: is the number of the membership functions for each input. All parameters above are shown in Fig. 6, 7 and (8). Once we obtain the IT2FLC design, we set the parameters of MBBO method as presented in Table 1.

6. Benchmark plants, performance criteria and Simulation Results

In this section we present the benchmark Plants, performance criteria and provide a simulation study of the plant using the controller described in Sections 4 and 5.

6.1. Benchmark Linear Plants

To test the optimized IT2FLC obtained by the modified biogeography-based Optimization and Particle swarm optimization methods; we used different linear systems. We first consider two benchmark plants called Plant 1 and Plant 2 with different levels of complexity [26]. Plant 1 is given by the following second order transfer function:

$$G(s) = \frac{W_n^2}{S^2 + 2\varepsilon W_n S + W_n^2}, \varepsilon = 0.5, W_n = 2 \qquad (21)$$

Where, W_n is the natural frequency and ε is the coefficient of damping. Plant 2 is given by the following transfer function:

$$G(s) = \frac{1}{S^2 + 4} \qquad (22)$$

6.2 Benchmark Non-Linear Plant

For the non-linear test system, we choose the simple inverted pendulum system [27]. The dynamic equation is,

$$\ddot{\theta} = -\frac{g}{l}\sin(\theta + \delta) - \frac{k}{m}\dot{\theta} + u \qquad (23)$$

Where, θ is the position angle; m is the mass of the bob, l is the length of the rod, g is the gravity, k is the friction coefficient. Assume the rod is rigid and has zero mass. The bob of the pendulum moves in a circle of radius l. u repersents the torque applied to the pendulum bob.

The state space equations are shown below

$$\dot{X_1} = f_1(x_1, x_2) = x_2, \quad \dot{X_2} = f_2(x_1, x_2) = -a\sin(x_1) - b\,x_2 + u \qquad (24)$$

Where

$$a = \frac{g}{l} = a_0 + \Delta a; \quad b = \frac{k}{m} = 0.5; \; -1 \le \Delta a \le 1, \; a_0 = 1$$

6.3 Performance Criteria

For evaluating and comparing the transient closed-loop response of the plant control system, we can use the Integral of Square Error (ISE) as shown in Fig. 5 and following equation.

$$ISE = \int_0^\infty [e(t)]^2 dt \qquad (25)$$

6.4 Simulation Results

In this section we evaluate, through computer simulations performed in MATLAB® and SIMULINK®, the designed IT2FLC for the benchmark plants (Plant 1, Plant 2 and the non-linear plant) as shown in Fig. 5. We include the results for interval type-2 controller designed with MBBO and PSO. All results for the case employing a PSO-based controller presented by Castillo and colleagues [28-29].

Plant 1 using MBBO: Table 2 presents the main results of IT2FLC obtained with MBBO with the best result show on the fourth row. Fig. 6 shows the membership functions (MFs) of input 1 and input 2 of the optimized FLC obtained by the MBBO.

Plant 2 using MBBO: Table 3 presents the main results of IT2FLC obtained with MBBO with the best result show on the fourth row. Fig. 7 shows the membership functions (MFs) of input 1 and input 2 of the optimized FLC obtained by the MBBO.

Non-linear Plant using MBBO: Table 4 presents the main results of IT2FLC obtained with MBBO with the best result show on the fourth row. Fig. 8 shows the membership functions (MFs) of input 1 and input 2 of the optimized FLC obtained by the MBBO.

Plant 1 using PSO: Table 5 presents the main results of IT2FLC obtained by PSO with the best tresult show on the first row [28-29].

Plant 2 using PSO: Table 6 presents the main results of IT2FLC obtained by PSO with the best tresult show on the first row [28-29].

Fig. 9 shows the step response results of IT2FLC for plant 1 obtained by MBBO and PSO.

Fig. 10 shows the step response results of IT2FLC for plant 2 obtained by MBBO and PSO.

Fig. 11 shows the closed loop states (angular position(θ) angular velocity ($\dot{\theta}$))response results of IT2FLC for non-linear plant obtained by MBBO.

The variations of the cost function with number of generations for plants 1, 2 and non-linear plant using MBBO are shown in Fig 12, 13 and 14 respectively.

We were also interested in improving the controller by adding uncertainty (noise) to the plant. We decided to use Random noise number block function in the MATLAB SIMULINK by add to plant output to simulate uncertainty in the control process as shown in Fig. 15. To check the system performance and stability due to uncertainty in the plant output, Plant 1 with uncertainty using MBBO: Table 7 presents the main results of IT2FLC obtained with MBBO showing the ISE for each case. Fig. 16 shows the step response results of IT2FLC for plant 1 with uncertainty obtained by MBBO. Plant 2 with uncertainty using MBBO: Table 6 presents the main results of IT2FLC obtained with MBBO showing the ISE for each case. Fig. 17 shows the step response results of IT2FLC for plant 2 with uncertainty obtained by MBBO.

6.5 Examining the results

We can note the following: Table 2, 3, 5, and 6 indicates that the maximum number of cost function evaluations for MBBO is 2000 while that of PSO is 140000. Hence it is clear that employing the MBBO to design IT2FLC produces better results with lower computation cost. Fig. 9 and Fig. 10 present the unit step response for the test plants using the two optimization algorithms, PSO and MBBO for optimizing the IT2FLC parameters. In all cases MBBO results in lower overshoot and steady state error compared to PSO [28-29]. It is important to note that when testing with higher levels of noise for in all cases, the step response result show the system is still tracking the set point and stable.

Table I: MBBO Parameters

Parameter	Value
Number of Generation	*20*
Number of SIVs per island	*3*
Island modification probability	*1*
Mutation probability	*0.005*
Elitism parameter	*1*

Table 2: Results of the IT2FLC for plant 1 obtained by MBBO

No	N0.of Island	Number of Generation	No. of SIVs per Island	Mutation	MBBO Time	Error
1	25	20	3	0.005	00:47:34	0.034726
2	50	20	3	0.005	01:50:10	0.037662
3	75	20	3	0.005	03:03:56	0.034311
4	**100**	**20**	**3**	**0.005**	**04:10:24**	***0.033171***

Table 3: Results of the IT2FLC for plant 2 obtained by MBBO

No	No. of Island	Number of Generation	No. of SIVs per Island	Mutation	MBBO Time	Error
1	25	20	3	0.005	00:39:34	0.035247
2	50	20	3	0.005	01:38:42	0.035232
3	75	20	3	0.005	02:56:32	0.035233
4	**100**	**20**	**3**	**0.005**	**03:48:59**	**0.035226**

Table 4: Results of the IT2FLC for non-linear plant obtained by MBBO

No	No. of Island	Number of Generation	No. of SIVs per Island	Mutation	MBBO Time	Error
1	25	20	3	0.005	01:13:12	0.00182
2	50	20	3	0.005	03:02:57	0.00112
3	75	20	3	0.005	04:06:21	0.00082
4	**100**	**20**	**3**	**0.005**	**05:32:02**	**0.00061**

Table 5: Results of the IT2FLC for plant 1 obtained by PSO [28-29]

No	SW	It	C1	C2	Inertia	PSO Time	Error
1	***200***	***70***	***0.5149***	***.3317***	***0.6808***	***07:05:20***	***0.08028***
2	200	70	0.8149	.9059	0.1706	12:44:19	0.08794
3	200	70	0.8129	.8159	0.1906	10:52:09	0.08894
4	200	70	0.7646	.9229	0.1096	09:50:45	0.12521
5	200	70	0.8168	.9359	0.4806	11:05:43	0.12630

Table 6: Results of the IT2FLC for plant 2 obtained by PSO [28-29]

No.	SW	It	C1	C2	Inertia	PSO Time	Error
1	**200**	**70**	**0.317**	**.541**	**0.7012**	**08:51:28**	**0.08338**
2	200	70	0.769	.208	0.8755	08:09:51	0.09031
3	200	70	0.388	.639	0.7304	08:50:16	0.09609
4	200	70	0.951	.801	0.6141	09:15:29	0.10198
5	200	70	0.787	.632	0.5724	10:00:28	0.10201

Table 7: Results of the IT2FLC for plant 1, and 2 with uncertainty level factor obtained by MBBO

Plant Number	Case Number	Uncertainty level factor	ISE
1	1	0.001	**0.033371**
1	2	0.01	**0.039091**
1	3	0.1	**0.065439**
1	4	1	**0.109008**
2	1	0.001	**0.035726**
2	2	0.01	**0.040018**
2	3	0.1	**0.070652**
2	4	1	**0.110639**

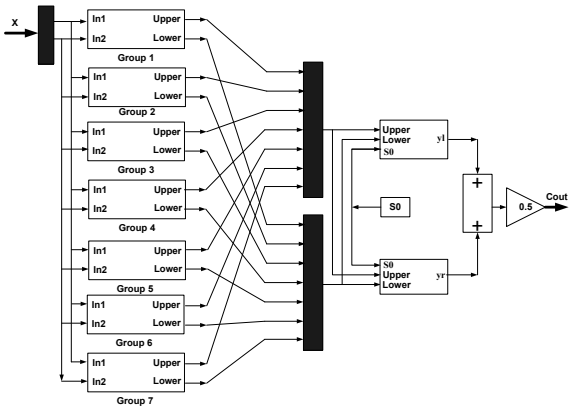


Fig. 4: Block diagram of a IT2FLC

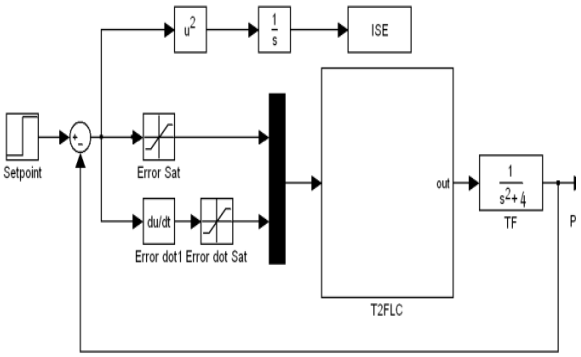


Fig. 5: Block diagram of a test plant with a IT2FLC controller

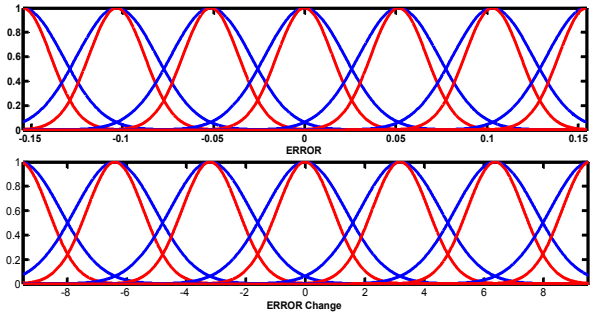


Fig. 6: Error and Error Change membership function of the optimized IT2FLC for plant 1.

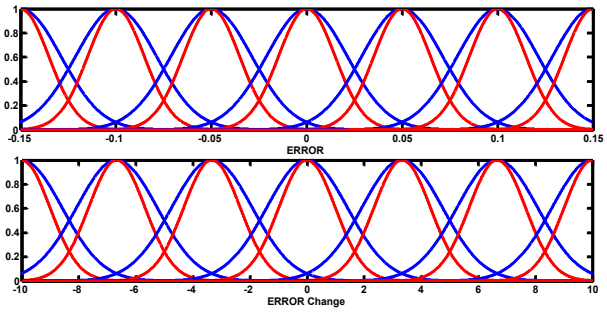


Fig. 7: Error and Error Change membership function of the optimized IT2FLC for plant 2

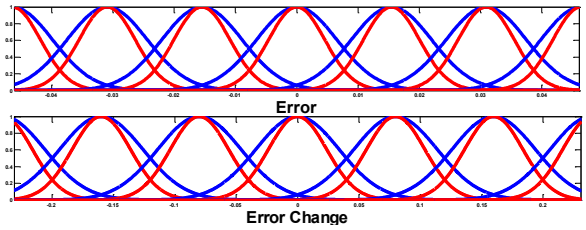


Fig. 8: Error and Error Change membership function of the optimized IT2FLC for Non-Linear plant

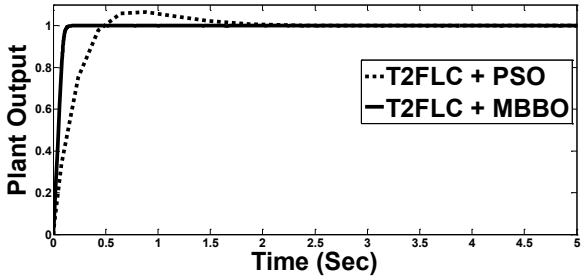


Fig.9. step response results for Plant 1

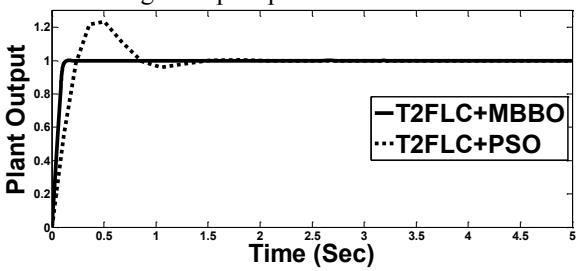


Fig.10. step response results for Plant 2

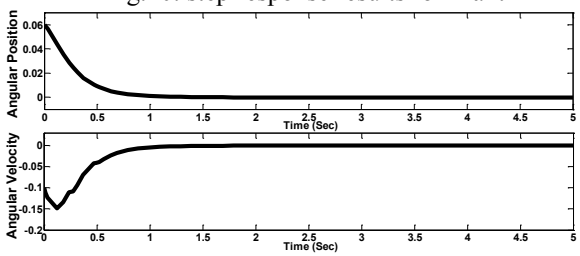


Fig. 11 shows the closed loop states response results

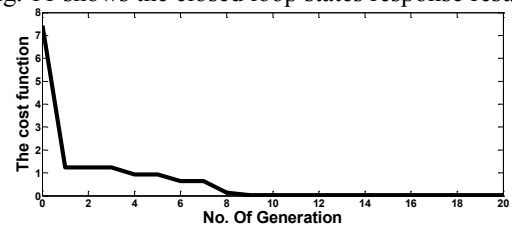


Fig.12: The cost function with number of generations for plant 1 (case no. 4) using MBBO

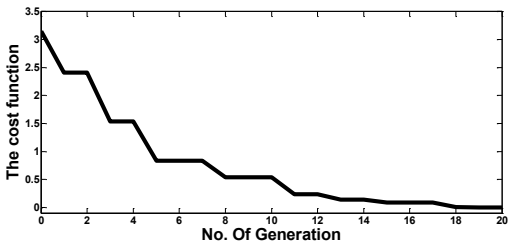


Fig.13: The cost function with number of generations for plant 2 (case no. 4) using MBBO

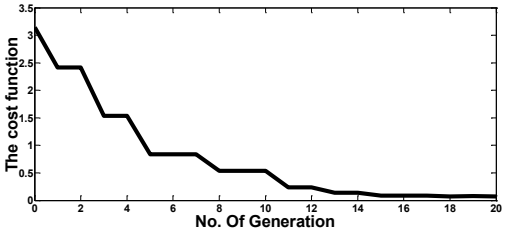


Fig.14: The cost function with number of generations for Non-Linear Plant using MBBO

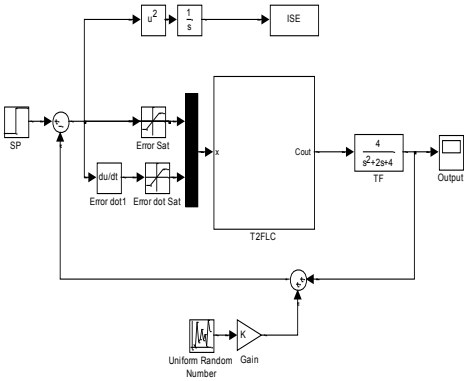


Fig. 15: Block diagram of a test plant with a IT2FLC controller plus the noise source

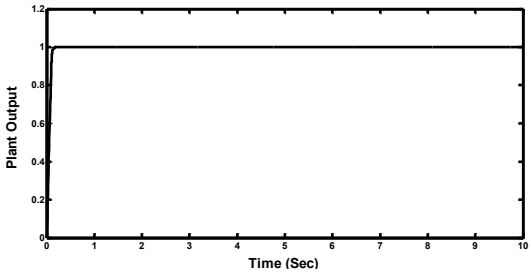


Fig. 16-Case 1: Step response result for Plant 1 with uncertainty level factor = 0.001

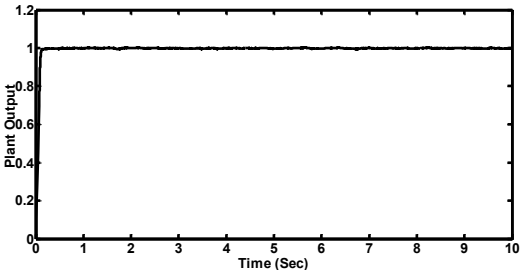


Fig. 16-Case 2: Step response result for Plant 1 with uncertainty level factor = 0.01

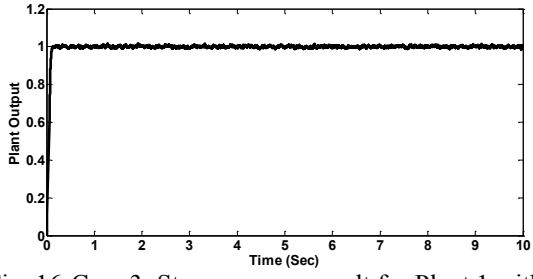


Fig. 16-Case 3: Step response result for Plant 1 with uncertainty level factor = 0.1

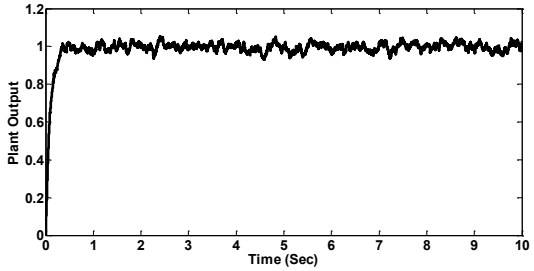


Fig. 16-Case 4: Step response result for Plant 1 with uncertainty level factor = 1

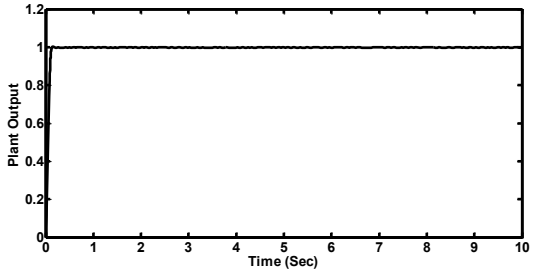


Fig. 17-Case 1: Step response result for Plant 2 with uncertainty level factor = 0.001

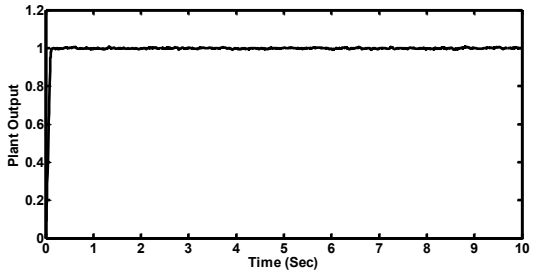


Fig. 17-Case 2: Step response result for Plant 2 with uncertainty level factor = 0.01

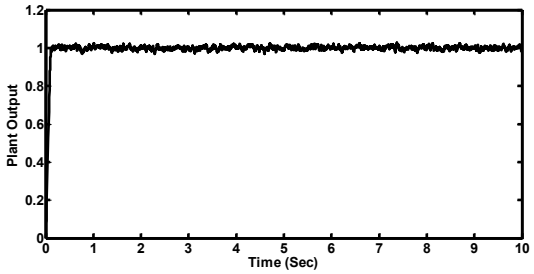


Fig. 17-Case 3: Step response result for Plant 2 with uncertainty level factor = 0.1

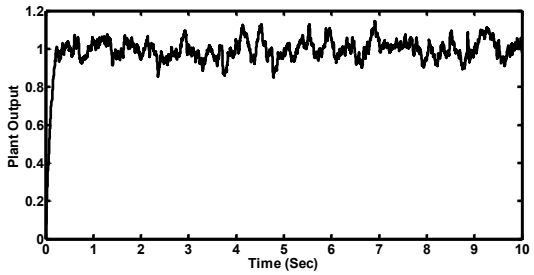


Fig. 17-Case 4: Step response result for Plant 2 with uncertainty level factor = 1

7. CONCLUSION

We described in this paper the application of bio-inspired methods to design optimized IT2FLC using MBBO and PSO. To test the optimized IT2FLC, we use different systems. In particular, we presented results of MBBO and PSO applied to two linear systems, using two different levels of complexity and uncertainty. Also we presented result of MBBO applied to a non-linear system. The results show that the IT2FLC obtained by MBBO and PSO gets stable in less than 10 seconds. On the other hand, the IT2FLC obtained by MBBO is better than the IT2FLC obtained by PSO, because the MBBO is less time consuming in the process and achieves lower overshoot in all plants: the plots of the results shows this difference. We have achieved satisfactory results with MBBO; the next proposed step by the authors is to solve the problem in a perturbed environment and considering multiple objective optimization to obtain more improved results.

References

1. Astudillo, L., Melin, P., Castillo, O. *A new optimization method based on a paradigm inspired by nature*. In Soft Computing for Recognition Based on Biometrics. SCI, vol. 312, pp. 277–283. Springer, Heidelberg (2010).
2. Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L. *Ten years of genetic fuzzy systems: current framework and new trends.* Fuzzy Sets and Systems 141, 5–31 (2004)
3. A. Wallace. *The Geographical Distribution of Animals (Two Volumes).* Adamant Media Corporation, 2005.
4. C. Darwin. *The Origin of Species, Gramercy*, 1995.
5. Dan Simon. *Biogeography-based optimization*. IEEE Transactions on Evolutionary Computation (12) pp. 702-713, December 2008.
6. M. Ergezer and Dan Simon. *Oppositional Biogeography-Based Optimization for Combinatorial Problems*. 2011 IEEE Congress of Evolutionary Computation, CEC 2011, p 1496-1503, 2011.
7. Haiping Ma and Dan Simon. *Biogeography-Based Optimization with Blended Migration for Constrained Optimization Problems*. Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10, p 417-418, 2010.
8. Wenyin Gong, ZhihuaCai, and Charles X. Ling. *DE/BBO: A Hybrid Differential Evolution with Biogeography-Based Optimization for Global Numerical Optimization.* Soft Computing, v 15, n 4, p 645-665, April 2011
9. H. Ma, Suhong Ni, and M. Sun. *Equilibrium Species Counts and Migration Model Tradeoffs for Biogeography-Based Optimization*. Joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conference Shanghai, P.R. China, December 16-18, 2009.
10. R. Rarick, D. Simon, F. E. Villaseca, and B. Vyakaranam. *Biogeography-Based Optimization and the Solution of the Power Flow Problem*. In Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, p 1003-1008, 2009.
11. M. M. Sayed, M. S. Saad, H. M. Emara, and E. E. Abou El-Zahab. *A Novel Method for PID Tuning Using a Modified Biogeography-Based Optimization Algorithm*. Proceeding of the 24th Chinese Control and Decision Conference (CCDC), 2012, Page(s): 1642 – 1647.
12. Oscar Castillo. *Type-2 Fuzzy Logic in Intelligent Control Applications*. Springer, Berlin, 2012
13. J. M. Mendel. *Advances in type-2 fuzzy sets and systems.* Inf. Sci.,vol. 177, no. 1, pp. 84–110, Jan. 2007.
14. Jerry M. Mendel. *Type-2 fuzzy logic systems: Type-reduction*. In IEEE Syst.,Man, Cybern. Conf., San Diego, CA, Oct. 1998.
15. N.N. Karnik, J.M. Mendel and Q. Liang. *Type-2 fuzzy logic systems.* IEEE Trans. Fuzzy Syst., vol. 7, pp.643–658, Dec. 1999.
16. J, Mendel. *Fuzzy logic systems for engineering: a tutorial.* IEEE Processing's, vol. 83, pp. 345–377, Nov. 1995
17. J, Mendel. *Computing derivatives in interval type-2 fuzzy logic systems.* IEEE Transactions on Fuzzy Systems, vol. 12, pp. 84–98,Feb. 2004.
18. D. Wu and J. Mendel. *Enhanced Karnik-Mendel Algorithms.* IEEE Transactions on Fuzzy Systems, vol. 17, pp. 923–934, Aug. 2009.
19. Jerry M. Mendel, Robert I. John, and Feilong Liu. *Interval Type-2 Fuzzy Logic Systems Made Simple.* IEEE Transactions on Fuzzy Systems, vol. 14, No. 6, Page 808-821, Dec. 2006.
20. Z. Chi, H. Yan, T. Pham. *Fuzzy Algorithms: With Applications to Image Processing and Pattern recognition*. World Scientific, Singapore, 1996.
21. D. Driankov, H. Hellendoorn, M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, Berlin, 1993.
22. T. Fukao, H. Nakagawa, N. Adachi. *Adaptive Tracking Control of a NonHolonomic Mobile Robot*. IEEE Trans. On Robotics and Automation, Vol. 16, No. 5, pp. 609-615, October 2000.
23. T. H. Lee, F. H. F. Leung, P. K. S. Tam. *Position control for wheeled mobile robot using a fuzzy controller*. IEEE Proceedings, pp 525-528, 1999.
24. R. Martinez, O. Castillo, L.T. Aguilar and A. Rodriguez. *Evolutionary Optimization of type-2 Fuzzy Systems Applied to Linear Plants*. Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control, Springer, pp. 17- 31, October, 2009.
25. R. Martinez, O. Castillo, L. T. Aguilar. *Intelligent control for a perturbed autonomous wheeled mobile robot using type-2 fuzzy logic and genetic algorithms*. Journal of Automation, Mobile Robotics & Intelligent Systems, Vol. 2, 2008.
26. Martinez, R., Castillo, O., Aguilar, L.T., Rodriguez, A. *Optimization of type-2 fuzzy logic controllers using PSO applied to linear plants*. In Soft Computing for Intelligent Control and Mobile Robotics. Studies in Computational Intelligence, vol. 318, pp. 181–193. Springer, Heidelberg (2010).
27. Bader M. Badreddine and Feng Lin. *Adaptive PID controller for stable/unstable linear and Non-linear systems*. Proceesing of 2001 IEEE International Conference on Control Applications. September, 2001.
28. Oscar Castillo. *Type-2 Fuzzy Logic in Intelligent Control Applications*. Chapter 12, Springer, Berlin, 2012.
29. Martinez, R., Castillo, O., Aguilar, L.T., Rodriguez, A. *Type-2 Fuzzy Logic Controllers Optimization using Genetic Algorithms and Particle Swarm Optimization*. Proceeding of the 2010 IEEE International Conference on Granular Computing, Page(s): 724 – 727.