DOI: 10.2478/ijasitels-2020-0002

# Imperialist competitive algorithm for determining the parameters of a Sugeno fuzzy controller

*Stelian Ciurea*[1]
"Lucian Blaga" University of Sibiu, Faculty of Engineering, Department of Computer and Electrical Engineering, E. Cioran Str, No. 4, Sibiu-550025, ROMANIA;
Research Center in Informatics and Information Technology.
E-mail: stelian.ciurea@ulbsibiu.ro

**Abstract**
We used an imperialist competitive algorithm to determine the parameters of a fuzzy controller of type Sugeno that would ensure a good unit step response of a second-order single-input and single-output automatic system.

**Keywords**: **S**ingle-input and single-output second-order linear system, Fuzzy Controller og type Sugeno, Imperialist Competitive Algorithm

## 1. Introduction

First described in 2007 by Gargari et al [1], the imperialist-competitive algorithm (hereinafter referred to as ICA) is an alternative to other heuristic methods of finding solutions as close as possible to the optimal ones in complete hard-NP problems. In previous papers we have studied how this type of algorithm can solve discrete problems such as the traveler salesman problem [15], or continuous problems such as determining the minimum for real functions with several arguments [16]. In this article we present how a competitive imperialist algorithm was used to determine the parameter values for a fuzzy controller of type Sugeno that was used to regulate a SISO second order linear system. The representation of the controller was done by 4 integer values and 199 real values. For this reason, ICA-specific operations have been implemented modeled on those used in the articles we referred to earlier ([15], [16]).

## 2. The mathematical model used for the second-order linear system

We have implemented an ICA (imperialist competitive algorithm) in order to determine the parameters of a controller so as to enable the best possible unit step response of a second-order automatic system. The function at the output of such a system is given by the following differential equation [2]:

$$a_2 \frac{d^2 y}{dt^2} + a_1 \frac{d\ y}{dt} + a_0 y(t) = b_2 \frac{d^2 u}{dt^2} + b_1 \frac{d\ u}{dt} + b_0 u(t) \tag{1}$$

where u(t) is the input variable and y(t) the output variable. In case of downtime τ in the transmission of the data, the corresponding differential equation is the following:

$$a_2 \frac{d^2 y}{dt^2} + a_1 \frac{d\ y}{dt} + a_0 y(t) = b_2 \frac{d^2 y}{dt^2} + b_1 \frac{d\ y}{dt} + b_0 u(t-\tau) \tag{2}$$

Many papers such as [3], [4], [8], [10], [12] use this type of equation in describing the behavior of DC motors. We have considered the standard structure of an automated system consisting of a second-order system and a fuzzy controller, illustrated in Fig. 1 ([2],[8]).

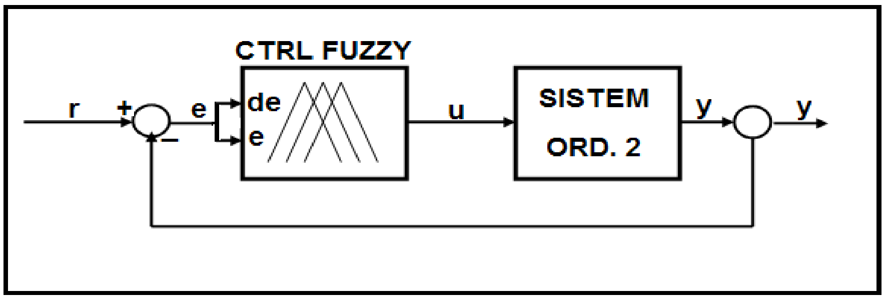


Figure 1. Automated regulating system

Our aim is for output value y to follow the evolution of reference value r as accurately as possible.

To simulate the operation of the system, we have considered its discrete model. The method, described in [2], consists in considering relation (1) at time intervals equal to dt and integrating the terms from that formula twice:

$$\int_{t-dt}^{t}\int_{t-dt}^{t} (a_2 \frac{d^2 y}{dt^2} + a_1 \frac{d\ y}{dt} + a_0 y(t)) = \int_{t-dt}^{t}\int_{t-dt}^{t} (b_2 \frac{d^2 u}{dt^2} + b_1 \frac{d\ u}{dt} + b_0 u(t)) \tag{3}$$

By using the trapezoidal rule for the calculation of integrals, we have obtained the following relation:

$$y_k = \frac{(b_2 + b_1\Delta + b_0\Delta^2)u_k + 2(b_0\Delta^2 - b_2)u_{k-1} + (b_2 - b_1\Delta + b_0\Delta^2)u_{k-2}}{a_2 + a_1\Delta + a_0\Delta^2} - \frac{2(a_0\Delta^2 - a_2)y_{k-1} - (a_2 - a_1\Delta + a_0\Delta^2)y_{k-2}}{a_2 + a_1\Delta + a_0\Delta^2} \quad (4)$$

where $y_k$, $y_{k-1}$ and $y_{k-2}$ are output values y at times $t_k$, $t_{k-1}$ and $t_{k-2}$, $t_k$ - $t_{k-1}$ = $t_{k}$-1 - $t_{k-2}$ = dt; analogous $u_k$, $u_{k-1}$ and $u_{k-2}$ are values u, and $\Delta$ =(dt/2)$^2$.

We have aimed at the operation of this system within the range of [0; tmax] seconds, considering that:

• at time $t_0$=0, a step signal of an amplitude equal to the unit is applied;

• prior to time $t_0$, the system was idle;

• dt=0.005 seconds.

## 3. Designed fuzzy controller

We have implemented a fuzzy controller of type Sugeno of type PD (for details see [13] and [14]) with two inputs and one output. The two entries have been the following:

- "error", marked *e*, is the percentage difference between the reference value and the system output. We have considered the universe of discourse for this value to be interval [-100; 100]. Its value at time $t_k$ is given by the formula:

$$e_k = \frac{r_k - y_k}{r_k} \times 100 \quad (5)$$

- "error variation", marked *de*, having universe of discourse [-5; 5] and calculated at time tk by means of the following formula:

$$de_k = \frac{e_k - e_{k-1}}{\Delta t} \quad (6)$$

For the output of the controller noted *u*, the interval [-200; 200] was chosen.

We set seven fuzzy sets for the two entries. The associated linguistic terms were in the case of both inputs NB (big negative), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), PB (positive big). The membership functions we employed are trapezoidal or built on two Gauss curves (Fig. 2 and Fig. 3).

We set seven fuzzy sets for the two entries. The associated linguistic terms were in the case of both inputs NB (big negative), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), PB (positive big). The membership functions we employed are trapezoidal or built on two Gauss curves (Fig. 2 and Fig. 3).

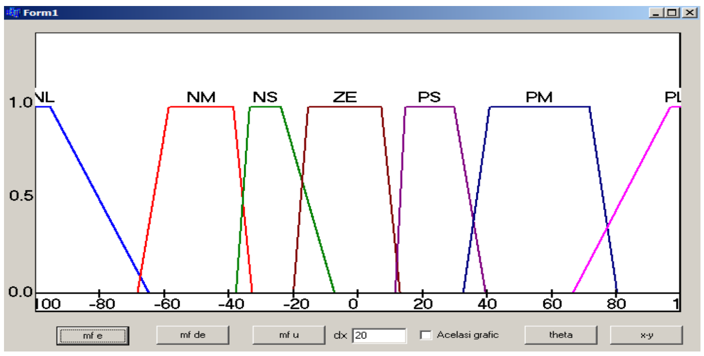


Figure 2. Example of trapezoidal members 1

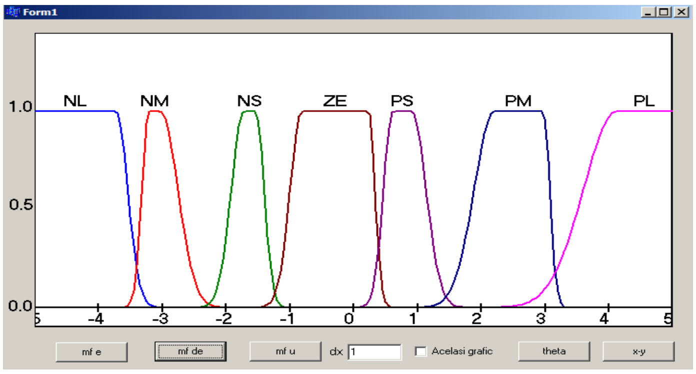


Figure 3. Example of membership functions shaped as a combination of two Gauss curves for variable *de*

For output variable u, the membership functions are linear functions in input variables. The rules used for this type of fuzzy controller are of the type below:

IF *e* is NB and *de* is NB, THEN $u_1 = a_1e + b_1de + c_1$

IF *e* is NB and *de* is NM, THEN $u_2 = a_2e + b_2de + c_2$ (7)

With 7 fuzzy sets for *e* and 7 fuzzy sets for *de*, we will have 49 controller rules (one rule for each pair of fuzzy sets *e* - *de*). Since the three coefficients in the above expressions ($a_i$, $b_i$ si $c_i$, i=1,2,…,49) are different for each pair, we need 3 × 49 = 147 parameters for the controller rules.

The defuzzification method used for this type of control is the “weighted average”; thus, denoting by $MFe_i$ one of the fuzzy functions that determine input *e*, by $MFde_i$ one of the fuzzy functions that determine input *de*, and by $e_v$ and $de_v$ the values of *e* and *de* at a given time, we calculate $w_i$ = the weight of the appropriate inference rule:

$w_i = \text{AndMethod}(MFe_i(e_v), MFde_i(de_v))$

We calculate the value provided by the controller using the following formula:

$$u = \frac{\sum_{i=1}^{49} w_i \theta u_i}{\sum_{i=1}^{49} w_i} \qquad (8)$$

It follows that, for a complete definition of the fuzzy controller, we need parameters defining the type and position of the fuzzy functions on the discourse universe axis for the two input values (*e* and *de*), the implementation of the fuzzy AND operation, and the values of the 147 parameters, $a_i$,$b_i$ and $c_i$ (i=1,2,…,49).

## 4. The Designed ICA

Imperialist competitive algorithms (ICA) are part of the category of heuristic search algorithms used to determine solutions as close as possible to optimal solutions for hard NP-complete problems. In these problems, the solutions are usually made up of a series of integer and/or real values. The basic idea is to generate a predetermined number of possible solutions (called in ICA „countries”), evaluate them, preserve the best (called in the ICA „metropoli”) and explore their neighborhoods by attracting other solutions (called in the ICA „colonies”) to the metropolis through the assimilation operation. To avoid stagnation of the solution set, the operation called revolution is used, which is applied to some of the colonies. By this some colony type solutions are thrown out of the vicinity of the metropolis type solutions.

The ICA algorithm we used in our application is similar to the one presented by Gargari et al in [1]:

```
1. Generate a random set of countries;
2. Evaluate these solutions and the best ones become metropolises;
3. Generate the initial empires by distributing the colonies;
4. Repeat
        5. Assimilation
        6. If a colony has better results than the imperialist country then
            a. Interchange the colony with the imperialist country
        7. Competition between empires:
          a. Assess empires
          b. Distribute the weakest colony of the weakest empire by another empire
          c. If the weakest empire has no colonies left then
                i. Remove this empire
  Until one of the stopping conditions is reached.
```

The stopping conditions are:
• only one empire remained;
• a predetermined number of iterations of the repeat… until loop were performed.

We will present the particularities of the algorithm specific to the problem it has to solve.

### 4.1 Representation of controllers

The controllers (or “countries” from the point of view of ICA) have been represented in the form of structures having the following makeup presented in detail in [11]:

- 1 integer field for coding fuzzy sets for the three linguistic variables corresponding to the two inputs and to the output: trapezoidal or a combination of two exponential functions;

- 1 integer field representing the number of fuzzy sets in each linguistic variable (seven in our case);
- 1 boolean field stating that the membership functions of fuzzy sets have a symmetry with respect to the central value of the universe of discourse;
- 52 real fields for coding the form and locations on the axis of the universe of discourse of the fuzzy sets within each linguistic variable (e and de), with the following implications for any of these variables:
  - 6 values representing the ratio between the larger base of the trapezium and the average value of the universe of discourse – a value within the range [0.25; 2.0];
  - 6 values representing the percentage of the larger base of the trapezium overlapping the larger base of the trapezium placed on its left - a value within the range [0.05 ; 0.4];
  - 7 values representing the ratio smaller base/larger base – a value within the range [0.01; 0.65];
  - 7 values representing the position of the smaller base for CE – a value within the range [0.05 ; 0.95] out of the available interval calculated depending on the large basis.
- 1 integer fields that specify the mathematical formula for the logical operations AND (0=min, 1=product)
- 147 real fields for the coefficients that appear in the expressions [7] ($a_i$, $b_i$ si $c_i$, i=1,2,…,49).

This mode of representation is useful because thespecific operations of ICA provide results with values in the same ranges, so correct. But for performing the operations necessary for fuzzy control or for graphical representations, some simple formulas (and related calculations) are needed to determine the classic shape of fuzzy sets.

## 4.2 Generating the initial set of countries

This has been done randomly: for the 147 specific coefficients the range [-100; 100]. These limits determine for the command u, the theoretical interval [-300; 300]

## 4.3 Evaluation of colonies

To evaluate the colonies in order to implement some of the specific ICA operations (empire determination, colony distribution, competition between empires), we have simulated the behaviour of the system when applying a step signal at time 0. The controller performance was calculated with the formula:

$$performance = \left(\sum_{t=0}^{2.5} |r_k - y_k|\right) + 10\, max(yfin_{max}) \qquad (9)$$

where $y_{max}$ is the maximum that the output value takes in the studied range, and $y_{fin}$ is the final value of the output of the system.

This evaluation function was thought of as a penalty: the lower its value, the more efficient the controller. Thus, the value of the first term of this expression is lower the shorter the response time and the lower the steady-state error. The second term is lower the less the system exceeds the steady state value.

## 4.4 The assimilation operation

In the application, we have used different formulae depending on the two types of parameters that make up a colony:

• for integer parameters, we have taken into account that the value of any of these parameters does not condition the value of another parameter. This led to the following formula: the colony parameter receives the value of the corresponding parameter in the metropolis with a given probability, hereinafter referred to as assimilation probability and marked $prob_a$;

• for real parameters, we have used the following formula:

$$paramcol_i = paramcol_{i-1} + p(1 \pm d)(parammet_{i-1} - paramcol_{i-1}) \tag{10}$$

where $paramcol_i$ is the value of the colony parameter after iteration i, $parammet_i$ is the value of the parameter corresponding to the metropolis of the respective colony after iteration i, and d is a subunit random value, $d \in [-d_a/2 \; ; \; d_a/2]$; $d_a$ is the assimilation deviation.

## 4.5 The revolution operation

We have used different formulae depending on the type of parameter whose value is to be changed.

• for each integer parameter, the following formula was applied: with a probability equal to value $d_r$ (revolution deviation)

$$paramcol = nrval - 1 - paramcol \tag{11}$$

where paramcol is the integer parameter and nrval is the number of values available for that parameter.

• for real parameters, we have used the following formula:

$$paramcol_i = paramcol_{i-1} - p(1 \pm d)(parammet_{i-1} - paramcol_{i-1}) \tag{12}$$

$d \in [-d_r/2 \; ; \; d_r/2]$, in this case $d_r$ being the revolution deviation.

# 5 Findings

The following values were chosen for the second-order system parameters:

$a_0=20$; $a_1=10$; $a_2=1$; $b_0=1$; $b_1= b_2=0$.

These values of the coefficients determine, for the system analyzed in open loop, a transfer function with two real poles having the values -7.2361 and -2.7639.

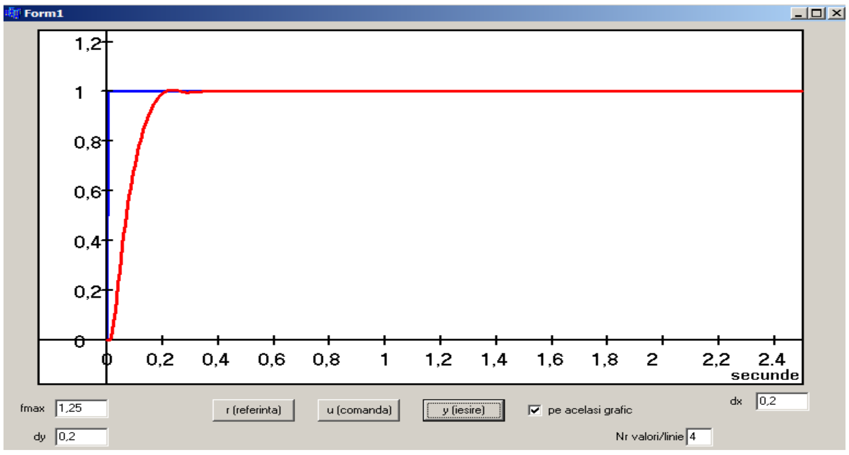
The observation interval was [0;2.5] seconds. Tests with ten initial sets of countries have been performed. The ICA parameters have been chosen as follows:

- number of countries in a set: 108;
- initial number of empires in a set: 8;
- maximum number of iterations in the algorithm: 300;
- share of a colony in the performance of the empire (w): 0.001, fixed;
- approaching step (p): 0.1 and 0.9;
- assimilation and revolution deviations ($d_a$ , $d_r$): 0.01 and 0.9;
- the probability with which the revolution operation is applied to a country: 10% and 30%.

The entire application has been implemented in the C language.

The results obtained for each of the ten sets of countries (marked with #0, #1, ..., #9) are presented in tables 1. We have noted the following:

- the average performance of the best controllers was better for higher values of the parameters p, $d_a$ and $prob_{rev}$;
- the yield of the algorithm representing the ratio between the average performance of the best controllers in each set after the first iteration and the average performance after the last iteration had the minimum value of 3.21 and a maximum of 4.19;
- In the case of the combination of ICA parameters for which the best average performance was obtained, in all ten tests the performance had values less than or approximately equal to 20 which implies a very good behavior of these controllers;
- the best value of the performance function obtained for a controller is 15.48. The simulation of the behaviour of the system with the help of this controller is presented in Fig 4. The system has an override equal to 0.7% and a response time of 0.19s. the system did not present oscillation around the reference value in steady state.



fmax 1,25
dy 0,2
dx 0,2
Nr valori/linie 4

Fig 4 The behaviour of an automated regulating system with the help of the best controller provided by ICA

Tabel 1 The performance of the best controllers obtained with the help of ICA

| 108 countries | | Step | | | | Step | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 0.1 | | | | 0.9 | | | |
| | | Assimilation deviation $d_a$ | | | | Assimilation deviation $d_a$ | | | |
| | | 0.01 | | 0.01 | | 0.9 | | 0.9 | |
| | | $prob_{rev}$ | | $prob_{rev}$ | | $prob_{rev}$ | | $prob_{rev}$ | |
| Set | Initial perform. | 0.1 | 0.3 | 0.1 | 0.3 | 0.1 | 0.3 | 0.1 | 0.3 |
| #0 | 31.28 | 18.56 | 16.58 | 17.24 | 17.62 | 19.87 | 16.29 | 18.55 | 17.03 |
| #1 | 29.08 | 16.48 | 16.57 | 16.78 | 16.69 | 18.75 | 16.59 | 15.92 | 16.77 |
| #2 | 77.24 | 16.8 | 17.71 | 16.34 | 18.86 | 20.87 | 16.85 | 16.97 | 15.78 |
| #3 | 81.16 | 20.13 | 18.51 | 19.43 | 18.23 | 18.78 | 17.48 | 19.99 | 15.66 |
| #4 | 46.2 | 17.26 | 17.87 | 16.44 | 18.82 | 15.75 | 16.30 | 15.78 | 15.62 |
| #5 | 71.53 | 17.82 | 16.86 | 17.13 | 17.48 | 23.58 | 20.30 | 16.39 | 17.17 |
| #6 | 147.2 | 17.74 | 17.00 | 19.14 | 17.85 | 25.92 | 17.98 | 19.08 | 16.06 |
| #7 | 84.40 | 17.27 | 17.36 | 18.4 | 18.08 | 18.32 | 17.47 | 16.64 | 16.16 |
| #8 | 78.04 | 23.27 | 18.76 | 22.85 | 17.27 | 31.95 | 15.81 | 20.29 | 16.37 |
| #9 | 30.56 | 17.29 | 16.71 | 16.6 | 18.06 | 16.82 | 16.91 | 16.60 | **15.48** |
| Average performance | 67.669 | 18.26 | 17.39 | 18.03 | 17.89 | 21.06 | 17.19 | 17.62 | 16.12 |
| Average yield | | 3,70 | 3,89 | 3.75 | 3.78 | 3.21 | 3.93 | 3.84 | 4.19 |

The ICA determined the minimum method for implementing the fuzzy AND operation. The fuzzy sets for the two inputs are illustrated in Fig. 5 and Fig. 6:

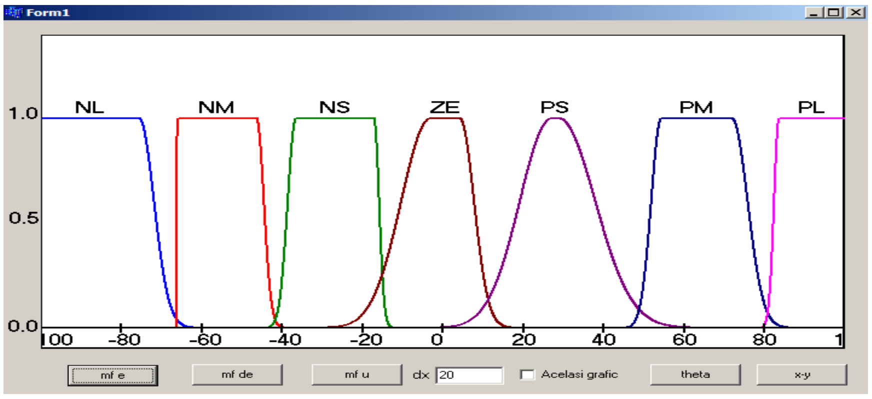


Fig. 5 Membership functions for input *e*

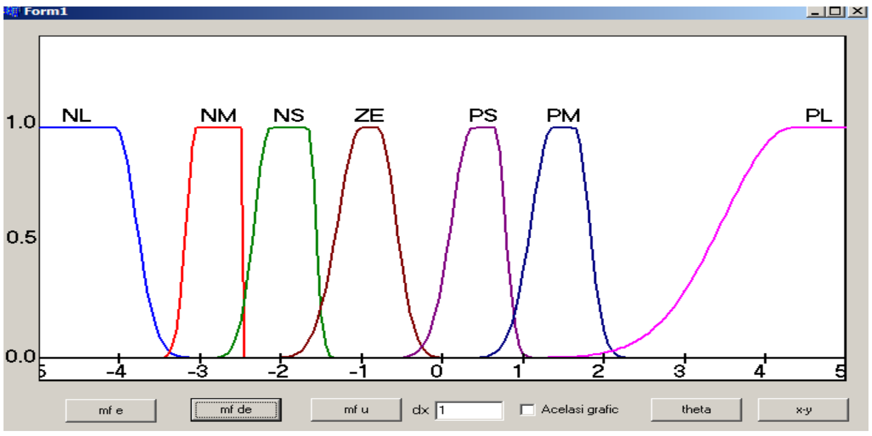


Fig. 6 Membership functions for input *de*

The values of the 147 coefficients are given in table 2.
Figure 7 shows the performance variation over the 300 iterations in the test that provided the best controller.

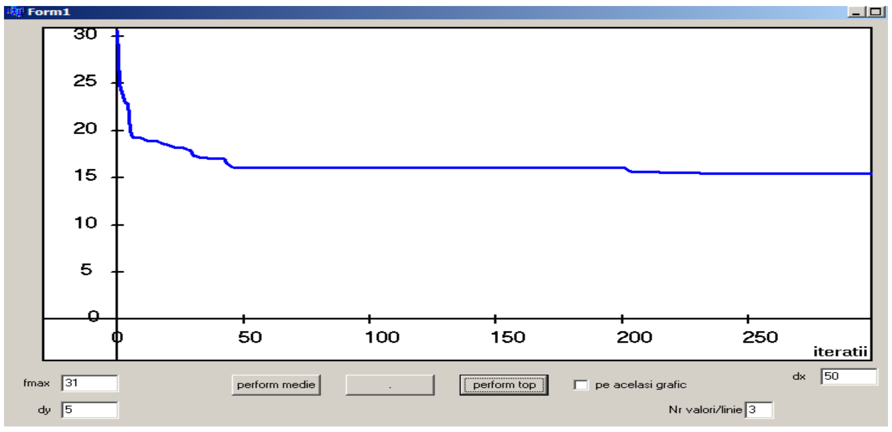


Fig.7 Performance of the best Sugeno controller vs iterations

Table 2 Inference coefficient values for Sugeno controller

| i | $a_{i1}$ | $a_{i2}$ | $b_i$ | | i | $a_{i1}$ | $a_{i2}$ | $b_i$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 20.85 | 70.12 | -53.63 | | 26 | 35.30 | 29.39 | 20.00 |
| 2 | -5.61 | -29.85 | -9.81 | | 27 | -109.8 | 60.65 | 9.23 |
| 3 | -26.18 | -4.885 | -14.49 | | 28 | 4.048 | -0.262 | -42.66 |
| 4 | -55.35 | -0.978 | -40.58 | | 29 | -0.742 | 118.5 | -41.80 |
| 5 | 15.61 | 11.27 | 16.07 | | 30 | -26.10 | 74.35 | -12.64 |
| 6 | 3.314 | 29.55 | -18.94 | | 31 | 4.725 | 12.03 | -13.10 |
| 7 | -16.81 | -28.01 | 12.88 | | 32 | -69.34 | -6.803 | -0.238 |
| 8 | -58.71 | -0.971 | 30.26 | | 33 | -24.85 | -10.99 | -86.02 |
| 9 | 0.59 | -18.71 | -43.46 | | 34 | -26.61 | 10.85 | -2.355 |
| 10 | -15.47 | -16.77 | -15.59 | | 35 | 33.53 | 3.745 | 17.70 |
| 11 | 3.144 | -49.80 | 2.58 | | 36 | -34.51 | -26.10 | 28.34 |
| 12 | 9.687 | 91.86 | -44.88 | | 37 | 120.4 | 54.30 | -1.045 |
| 13 | -20.31 | -5.146 | 19.99 | | 38 | 40.42 | -16.14 | 3.061 |
| 14 | 8.541 | -98.76 | 22.47 | | 39 | -7.942 | 27.90 | 0.809 |
| 15 | 2.390 | -20.40 | 2.371 | | 40 | 74.58 | -4.596 | -39.50 |
| 16 | -16.19 | 10.45 | -14.65 | | 41 | -16.11 | -1.064 | -5.001 |
| 17 | -11.74 | -3.282 | 15.52 | | 42 | -62.15 | -2.779 | -13.51 |
| 18 | 8.887 | 9.813 | -68.76 | | 43 | 11.75 | 97.83 | 36.00 |
| 19 | -66.07 | 32.52 | -6.918 | | 44 | -0.983 | -70.08 | 68.86 |
| 20 | 50.68 | 106.0 | 7.714 | | 45 | 6.143 | 32.92 | -20.92 |
| 21 | -51.87 | -22.60 | -7.216 | | 46 | 5.611 | 6.246 | -52.41 |
| 22 | -7.590 | -11.84 | -79.67 | | 47 | 5.206 | 7.311 | 12.74 |
| 23 | -16.58 | 4.308 | -3.190 | | 48 | -14.20 | 16.07 | 55.55 |
| 24 | 6.983 | 18.66 | -91.15 | | 49 | -13.61 | 10.52 | 1.807 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 25 | 31.61 | 3.862 | -65.30 | | | | | |

## 6. Conclusions

The imperialist competitive algorithm has been able to determine parameters for a fuzzy controller to ensure the efficient operation of the second-order automated regulating system, even in difficult conditions (non-negative controls of the controller). Better results have been obtained in the tests in which the values of the parameters p, $d_a$ and $prob_{rev}$ was hight. Applying a step signal to an automated regulating system, the best controller ensured the operation of the system with an override equal to 0.7%, a response time of 0.19s. The system did not present oscillation around the reference value in steady state.
The average running times on a computer with an Intel Core i3 microprocessor at 2.93 GHz was equal with 37 seconds (for the sets of 108 countries and a maximum of 300 iterations). Sugeno-type fuzzy controllers are much faster than Mamdami-type fuzzy controllers (for the same values of ICA parameters, on the same type of computer Mamdami controllers had average running times of 117 minutes and 21 seconds).

## References

[1] Esmaeil Atashpaz Gargari, Caro Lucas, *Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition*, IEEE Congress on Evolutionary Computation, CEC 2007.
[2] Babutia I., Dragomir T.L., Mureşan I., Proştean O.: *Conducrea automată a proceselor*, Ed. Facla, Timişoara, 1985
[3] Bhartiya, S., Sehra, S., Bhuria V.: Dc Motor Speed Control using PID- Fuzzy Logic Based Controller, http://ijair.jctjournals.com/jan2013/t50.pdf IJAIR, ISSN: 2278-7844, 2013
[4] Bindu R., Mini Namboothiripad K.: *"Tuning of PID Controller for DC Servo Motor using Genetic Algorithm"*, International Journal of Emerging Technology & Advanced Engineering, Volume 2, Issue 3, March 2012, ISSN 2250-2459
[5] Ashkan MohammadZadeh Jasour, Esmaeil Atashpaz Gargari, Caro Lucas, *Vehicle Fuzzy Controller Design Using Imperialist Competitive Algorithm*, Second joint congress on fuzzy and intelligent systems, Tehran, Iran, 2008
[6] Danilo Pelusi, *Optimization of a Fuzzy Logic Controller using Genetic Algorithms*, 2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) 2011
[7] Z. Michalewicz, "*Heuristic Methods for Evolutionary Computation Techniques*", Journal of Heuristics, Vol.1, No.2, 1995, pp.177-206
[8] Dumitrache I., Călin S., Boţan C., Niţu C.: *Automatizări şi echipamente electronice*, Ed. Didactică şi Pedagogică, Bucureşti, 1982
[9] Stelian Ciurea, *"Determining the Parameters of a Sugeno Fuzzy Controller Using a Parallel Genetic Algorithm*", Proceedings of the 9th International Conference on Control Systems and Computer Science, University Politehnica of Bucharest, Romania, 2013", ISBN 798-0-7685-4980-4, pg 36-43.

[10] Kuma D., D., Dhakar, B., Yadav, R.: "*Tuning a PID controller using Evolutionary Algorithms for an Non-linear Inverted Pendulum on the Cart System*", International Conference on Advanced Developments in Engineering and Technology (ICADET-14), INDIA, International Journal of Emerging Technology and Advanced Engineering, website: www.ijetae.com (ISSN 2250-2459 (Online), Volume 4, Special Issue 1, February 2014.

[11] Stelian Ciurea, "*A Better Genetic Representation of a Fuzzy Controller Enabling the Determination of Its Parameters with the Help of a Genetic Algorithm*", Proceedings of MIDS 2013, ISSN 2067-3965, pg. 49-58, Sibiu, Romania

[12] Yazdani A. M., Ahmadi, A., Buyamin, S., Fua'ad Rahmat, M., Davoudifar F., Abd Rahim, H.: "*Imperialist competitive algorithm-based fuzzy pid control ethodology for speed tracking enhancement of stepper motor*", International Journal on Smart Sensing and Intelligent Systems, vol. 5, no. 3, September 2012

[13] Terano, T., Asai, K., Sugeno, M.: *Fuzzy Systems Theory and Its Applications*, Academic Press, 1992.

[14] "*Fuzzy Logic Toolbox™ User's Guide R2011b*", The MathWorks, Inc., 2011

[15] Stelian Ciurea, *An Imperialist Competitive Algorithm Optimized to Solve the Travelling Salesman Problem ,* Proceedings of MIDS 2017

[16] Stelian Ciurea, *Imperialist Competitive Algorithm with Variable Parameters to Determine the Global Minimum of Functions with Several Arguments,* Proceedings of MIDS 2015