Proceedings of the Tenth IASTED International Conference
CONTROL AND APPLICATIONS (CA 2008)
May 26-28, 2008 Quebec City, Quebec, Canada

USING GENETIC ALGORITHMS TO DESIGN A FUZZY LOGIC CONTROLLER FOR A PH REACTOR: AN OBJECT APPROACH

Orlando Reyes*, Gustavo Sanchez*, Miguel Strefezza**
*Departamento de Tecnologia Industrial, **Departamento de Procesos y Sistemas
Universidad Simón Bolívar, Sartenejas
Venezuela
* orlandoreyes@usb.ve, * gsanchez@usb.ve, **strefeza@usb.ve

ABSTRACT
This paper presents a Fuzzy Logic Controller (FLC) designed through Genetic Algorithms (GA). The genetic representation follows an "object approach", with attributes and methods for each controller element: membership functions, conclusions tables and fitness values. Special mutation and crossover operators are also discussed. Results show that the proposed method is able to find good controllers for the target plant: a pH reactor operating near to the neutral point.

KEY WORDS
Fuzzy Control, Genetic Algorithms, pH reactor, neutralization.

1. Introduction

Due to the economic importance of pH controllers, a great effort has been put on improving its performance. The objective is to drive the pH in the system to a setpoint (pH=7) in as short a time as possible, and to keep it there.

In other hand, FLC and GA have been successfully combined to solve difficult control problems, for highly non-linear plants [1], [2], [3], [4]. For example, in [5] a FLC for a laboratory pH system was tuned using GA. The coding scheme was multiparameter, mapped and fixed-point: 32 parameters defining the linguistic terms were coded in 224-bit strings. The adaptive GA-FLC was capable of accomplishing the control objective in approximately 75 seconds, despite "spikes" caused by the external addition of acid or base. In [7] a simple method for designing FLCs with symmetrically distributed fuzzy sets was proposed. The design parameters included the parameters of the membership functions and the rule base. The method was based on a network implementation of the FLC with real and binary weights with constraints, and was also tuned using GAs. In [8], an enhanced GA which constraints the optimization of FLCs to produce well-formed fuzzy sets and rules was presented. The authors devised several new genetic operators and used a parallel GA with three populations. They used a novel method for creating migrants between the populations to increase the chance of optimization. In [9] a nonlinear gain scheduling control strategy, based on a neuro-fuzzy network, was presented. The process operation was partitioned into several fuzzy operating regions, and within each region, a local linear model was used to model the process. A nonlinear controller was developed by combining several local linear controllers that were tuned on the basis of the local model parameters. In [6] a GA-based optimal fuzzy controller was proposed. The design procedure was accomplished by establishing an index function as the consequent part of the fuzzy control rule. The inputs of the controller were utilized by the index function for computing the output linguistic value.

It is very difficult to adjust the setpoint in pH=7, reader could verify it comparing several studies in this matter, i.e. [1], [2]. Is common see control far neutral point, because in medium zone (Fig 1) ph reactor is highly non linear and control design is a hard task to achieve.

In the present paper, a GA is used in order to design a FLC for a pH reactor. The genetic representation follows an "object approach", with attributes and methods for each controller element: membership functions, conclusions tables and fitness values. Special mutation and crossover operators are also discussed. Results show that the proposed method is able to find good controllers for the target plant. First, in section 2 a basic background is presented. In section 3, the mathematical model of the pH reactor is given. In section 4, the FLC is explained. In section 5, numerical results are given. Finally, some conclusions are drawn.

2. Basic background

Next, a briefing about Fuzzy Logic (FL) and GA is presented.

2.1 Fuzzy Logic

The Fuzzy Logic concept was proposed in a seminal paper written in 1965 by Lofti A. Zadeh [10]. One of the first FLCs was developed in [11], attempting control the speed of a steam engine.

An important issue in FLC design is searching for good parameters for both membership functions and conclusion tables. Heuristic techniques are useful to perform this task. A FLC consists of a rule set that, in a linguistic manner, tells how the system must work. The output of the FLC will be the control action. Linguistic rules are constructed like statements, with cause and consequences, as follows:

IF cause_1 ***AND*** cause_2 ***THEN***
consequence_1 ***AND*** consequence_2

In this work the defuzzification process is carried out following the Takagi-Sugeno (T-S) method, with five membership functions for each input, and twenty five rules. The FLC output is calculated using the weighted averaging defuzzification method (see eq. 1).

$$CF_i = \frac{\sum \alpha_j * C_{i,j}}{\sum \alpha_j} \tag{1}$$

where:

$C_{i,j}$: conclusion i , rule j; α_j: activation degree of rule j;
CF_i: defuzzificated (crisp) value.

An extended FLC review, presented in [12], gives the reader a clue of their broad application.

2.2 Genetic Algorithms

A Genetic Algorithm [13], [14] is an iterative stochastic optimization process based in how the nature selects the best individual to survive within a given environment. They are now accepted by both the optimization and control communities to solve problems for which classical methods (i.e mathematical programming) can not be used or are not efficient enough. A GA starts with a scattered random population in a bounded space. An adaptation (fitness) value is assigned to every individual, which will be used to give a selection probability for crossover, survivor or mutation operations. The choice of the best individuals for crossover will give good "chromosomes" to children. The process is iterated with the hope to obtain better individuals when the algorithm stops.

3. The pH reactor model.

The equations for the pH dynamic were developed in [15]. The main issue is to keep the process around the neutral point, where the system is very sensitive and highly non linear. The interested reader can easily verify this fact by the construction of the neutralization or titration curve (TC). An experimental method to obtain the TC is based on holding the base concentration constant, slowly adding the acid and then plotting the pH versus the acid concentration. Three operating zones are commonly considered: low, medium, high (see Fig 1). The pH is usually controlled by the mixture of two solutions with different concentrations, one basic and other acid. In this work, we validated our SIMULINK® model by comparing the resulting TC with the one presented in [9].

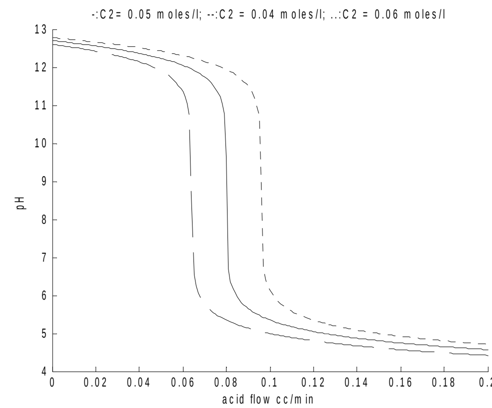

Figure 1. Titration curve, zones low, medium, high, pH approximately 0~6, 6~11.5, 11.5~14, respectively

The neutralization process takes place within a Continuous Stirred Tank Reactor (CSTR). There are two flows to the CSTR. One is acetic acid of concentration C_1 at flow rate F_1, and the other is sodium hydroxide of concentration C_2 at flow rate F_2. The mathematical equations of the CSTR can be stated as:

$$V\frac{d\xi}{dt} = F_1 C_1 - (F_1 + F_2)\xi \tag{2}$$

$$V\frac{d\varsigma}{dt} = F_2 C_2 - (F_1 + F_2)\varsigma \tag{3}$$

$$\begin{aligned}&[H^+]^3 + (K_a + \varsigma)[H^+]^2 + \\ &\{K_a(\varsigma - \xi) - K_w\}[H^+] - K_w K_a = 0\end{aligned} \tag{4}$$

$$pH = \log_{10}[H^+] \tag{5}$$

$$\xi = [HAC] + [AC^-] \tag{6}$$

$$\varsigma = [NA^+] \tag{7}$$

Table 1 shows the parameters and model variables.

4. The Fuzzy Logic Controller (FLC)

In this work, we propose to implement the FLC following an object approach. Data structures and storage schemes are important issues to improve numerical software.

Table 1
Description and values for parameters and variables

Name	Description	Value
V	Volume of tank	1 L
F_1	Flow rate of acid	0.081 L/min
F_2	Flow rate of base	0.512 L/min
C_1	Concentration of acid in F1	0.32 mol/L
C_2	Concentration of acid in F2	0.05005 mol/L
K_a	Acid equilibrium constant	1.8×10^{-5}
K_w	Water equilibrium constant	1.0×10^{-14}
$[H^+]$	Hydrogen ion	-
[HAC]	Acetic acid	-
$[AC^-]$	Acetate ion	-
$[NA^+]$	Sodium ion	-

4.1 Object-Oriented Approach

There is not a widely-agreed definition of Object Oriented Programming (OOP). However, there are some essential concepts, identified in the strong majority of OOP definitions. These are: class or structure: defines the abstract characteristics of a thing (its attributes or properties like dimensions, shape, weight, etc...) and the things it can do (its behaviors or methods, response or reaction to an stimulus). The Object: a particular instance of a class. Method: an object's abilities. Message passing: The process by which an object sends data to another object (like in crossover). Inheritance: in some cases, a class will have "subclasses," more specialized versions of a class. Encapsulation: conceals the exact details of how a particular class works from objects that use its code or send messages to it. Abstraction: Simplifying complex reality by modeling classes appropriate to the problem, and working at the most appropriate level of inheritance for a given aspect of the problem. Polymorphism: is behavior that varies depending on the class in which the behavior is invoked, that is, two or more classes can react differently to the same message [16]. In our framework, every individual (FLC) is an object, with the attributes and methods presented in figure 2.

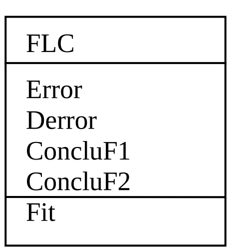

Figure 2. FIS Object

The FLC object contains the following attributes: current error (Error), rate of error change (Derror), conclusions values for control of the base valve (ConcluF1) and acid valve (ConcluF2) The FLC object contains also the method Fit (individual fitness value). FLC variables are initialized declaring type and variable name. Every attribute of FLC has also a range. For the attribute Error it could be expressed as [Emin, Emax].

4.2 Variables Initialization

The population is randomly initialized, using the following procedure:

i.- three points are generated inside [a b]

$$v(i) = a + (b-a) * \lambda(i) \\ i = 1,2,3;\ 0 \le \lambda \le 1 \tag{7}$$

ii.- sort the vector v(i)
iii.- Build an extended vector with [a v(i) b]

```
x(1) = a
For k = 2 to 4
     x(k) = v(k-1)
End
x(5) = b
```

Vector x contains the vertex for each triangular membership function (Fig 3).

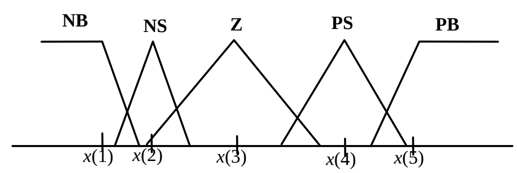

Figure 3. Membership functions vertex points. N:Negative, P: Positive, S: Small, B:Big, Z: Zero. i.e. NS means Negative Small

iv.- Vector x allows to find the rest of membership function parameters (Fig 4)

For trapezes

To left (NB)

```
a = x(1)
b = (x(1)+x(2))/2+(x(2)-
(x(1)+x(2))/2)*λ
ind(1).error = [a b]
```

To right (PB)

```
a = x(4)+((x(4)+x(5))/2-x(4))* λ
b = x(5)
ind(5).error = [a b]
```

For inner triangles

```
for k = 2:4
     a = x(k-1)+((x(k-1)+x(k))/2-x(k-
1))*λ
b = x(k)
c = (x(k)+x(k+1))/2+(x(k+1)-
(x(k)+x(k+1))/2)* λ
ind(k).error = [a b c]
end
```

v.- the flow 1 and 2 conclusions tables are created.

```
for k = 1:5
    ind(k).concluF1 = con(1) +
(con(2)-con(1))*rand(1,5);
    ind(k).concluF2 = con1(1) +
(con1(2)-con1(1))*rand(1,5);
end
```

In step v, “con” is a vector with two elements containing the domain for the values generated between the conclusion table for the flow 1 and 2. The instruction rand(1,5) generate one vector of five columns, randomly distributed between 0 and 1. The instructions in step v.- generate five vectors for every table.

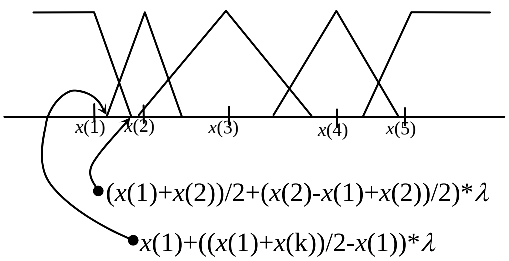

Figure 4. Calculation of membership functions base parameters.

Variables initialization is important because the algorithm starts with a defined number of creatures that make up the initial population (see figure 5).

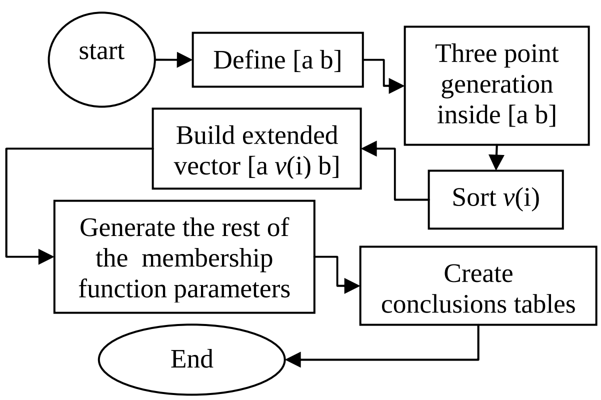

Figure 5. initialization variables flow diagram

4.3 Rules conclusions and defuzzification

Rules are created with all possible combinations between Error and Derror fuzzy values. **NB**: Negative Big, **NS**: Negative Small, **Z**: Zero, **PS**: Positive Small, **PB**: Positive Big.

IF Error is ***NB AND*** Derror is **PB** ***THEN*** Ci,1
: : : : : : : : : : :
IF Error is ***PB AND*** Derror is **NB** ***THEN*** Ci,25

Next table show how to construct the rules.

Table 2

Rule table, i = 1 affects valve of acid, i = 2 otherwise

Derror \ Error	NB	NS	Z	PS	PB
PB	$C_{i,1}$	$C_{i,2}$	$C_{i,3}$	$C_{i,4}$	$C_{i,5}$
PS	$C_{i,6}$	$C_{i,7}$	$C_{i,8}$	$C_{i,9}$	$C_{i,10}$
Z	$C_{i,11}$	$C_{i,12}$	$C_{i,13}$	$C_{i,14}$	$C_{i,15}$
NS	$C_{i,16}$	$C_{i,17}$	$C_{i,18}$	$C_{i,19}$	$C_{i,20}$
NB	$C_{i,21}$	$C_{i,22}$	$C_{i,23}$	$C_{i,24}$	$C_{i,25}$

Defuzfication is done calculating CF1: conclusion at flow 1 and CF2: conclusion at flow 2, with the equation 1. Where α_j is the max value between both membership degree Error and Derror at the FLC input.

5. Problem formulation

5.1 Fitness Function

A GA needs a fitness function to be defined in order to minimize it. The control problem can be stated as following a reference signal (rs) or set point (sp). Usually, the quadratic error between the output (y) and the sp is taken as the fitness function [6].

$$\min \int_0^{\tau} \left| sp(t) - y(t) \right|^2 dt \qquad (8)$$

5.2 Object Codification

The representation of a FLC as an object allows reducing the search space size. The code presented before shows how to call a particular attribute. For example, the value of the error corresponding to the individual number 5 is called as:

```
ind(5).error
```

The five parameters of the membership functions could be invoked using the following syntax:

```
VariableName(Individual).Atribute
```

Thus, crossover could be implemented changing membership functions between two parents as a whole.

6. Experiment Design

A population of 80 individuals (FLC) were generated. System simulations were carried out for every FLC, encapsulating its fitness value. A roulette wheel method selects the best parents to give good information to children. Crossover is achieved shifting the attributes between parents, as a whole. Next, the child or survivor is mutated: one or more attributes are changed randomly and/or one or more conclusion table parameters are also changed.

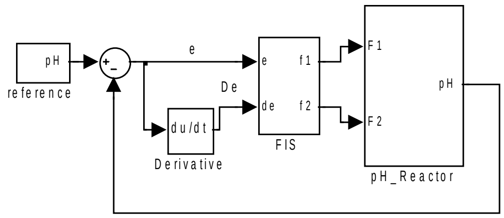

Figure 6. Closed loop system

Table 3
GA input

Parameter	value
Initial population	80
Number of generations	35
Crossover probability	0.8
mutation probability	0.05
Error domain	[-7 7]
Derror domain	[-7 7]
Conlusion 1 table domain	[0 10]
Conlusion 2 table domain	[0 10]

6.1 Numerical Results for the Best Solution

Membership functions parameters for Error

-7.0000	-4.5970		left. trapeze
-3.9690	-0.7567	0.7389	triangle
-0.5158	1.2012	2.5650	‘’
4.1622	4.9115	6.3640	‘’
5.9257	7.0000		right. trapeze

Membership functions parameters for Derror

-7.0000	-3.6538		left. trapeze
-6.7713	-6.4489	-0.6029	triangle
-1.6339	2.6860	3.5215	‘’
1.0894	5.2727	6.9430	‘’
3.4257	7.0000		right. trapeze

Consequence for flow 1

3.5666	3.2062	0.2088	1.7257	6.8197
5.3040	2.6541	0.8110	6.9210	5.4860
0.4779	3.9873	4.6005	1.2861	3.0425
3.4719	2.1831	0.4307	0.3410	4.5134
5.7997	1.0954	5.1420	2.2434	2.2721

Consequence for flow 2

2.0226	0.4519	4.8556	3.6185	1.4102
0.0984	4.5968	4.1440	4.1144	4.2632
6.6044	2.5693	4.7068	4.4349	0.9079
3.4714	0.2592	0.8381	4.9771	0.9853
0.6440	1.0529	1.4037	4.6385	1.5999

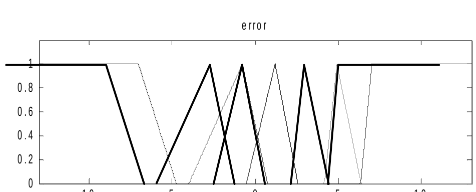

Figure 7a. Membership functions of the best error

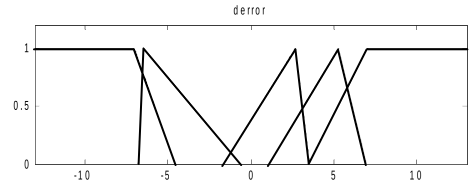

Figure 7b. Membership functions of the best derror

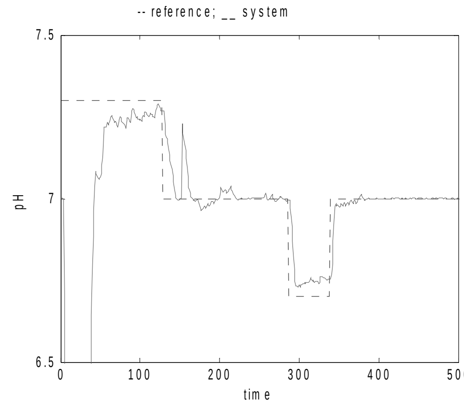

Figure 8. Response near to neutral point, changing the reference signal

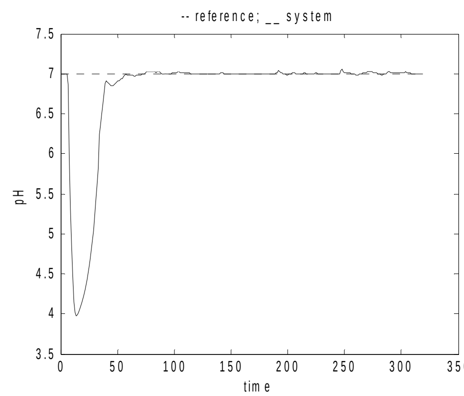

Figure 9. Response in the neutral point

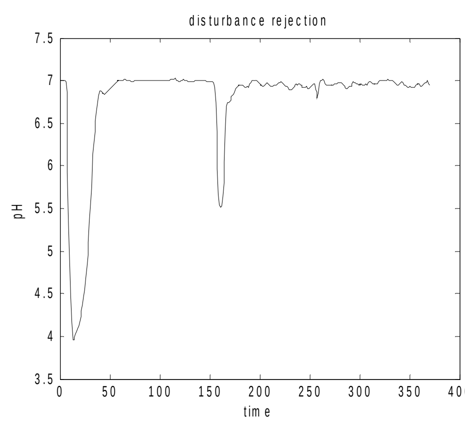

Figure 10. Disturbance trying down response

7. Conclusion

GA is an useful tool to find good FLC parameters when classic methods are unable. It is not easy to keep the system stable because it is highly nonlinear near neutral point. Results show how disturbances that try to up pH (Fig. 11) are less supported than otherwise (Fig 10). In fact pH under 6.5 in the zone called low is well controlled because the range to change acid flow is wider for pH variation (Fig 1). Over pH = 7, pH magnitude has a wider range with respect to the variation of acid flow. The obtained FLC shows good performance near to and in the neutral point (Fig. 8 and 9). For each GA run, we have observed the same trend for both the best and mean population error (Fig 12, 13), as could be expected for a good GA implementation.

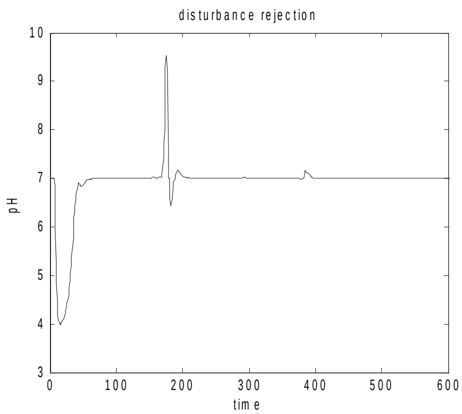

Figure 11. Disturbance tying up response

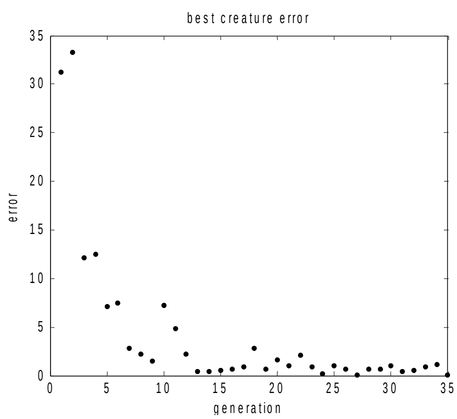

Figure 12. Best individual error

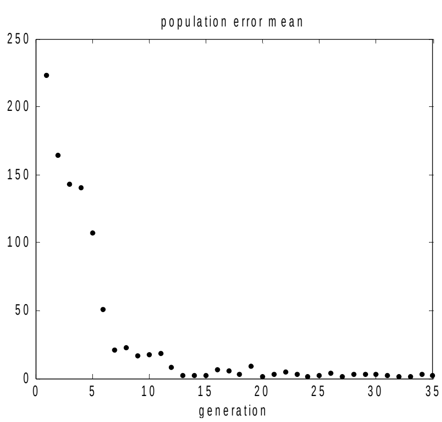

Figure 13. Actual population mean error

References

[1] Mahmoud Reza Pishvaie and Mohammad Shahrokhi. Control of pH processes using fuzzy modeling of titration curve. *Fuzzy Sets and Systems 157,* 2006, 2983 – 3006.

[2] M.J. Fuente, C. Robles, O. Casado, S. Syafiie and F. Tadeo. Fuzzy control of a neutralization process. *Engineering Applications of Artificial Intelligence 19*,2006 905-14.

[3] Yi-Sheng Zhou and Lin-Ying Lai. Optimal Design for Fuzzy Controllers by Genetic Algorithms. *IEEE Transactions on Industry Applic. Vol. 36(1).*2000

[4] Yu-Chiun Chiou and Lawrence W. Lan. Genetic fuzzy logic controller: an iterative evolution algorithm with new encoding method, *Fuzzy Sets and Systems, Volume 152, Issue 3, 16.*2005, 617-635.

[5] Karr, C. Genetic Algorithms for Modelling, Design and Process Control , *Proceedings of the CIKM, USA.*1993.

[6] Chih-Hsun Chou. Genetic Algorithm-Based Optimal Fuzzy Controller Design in the Linguistic Space, *IEEE Transactions on Fuzzy Systems, 14(3).* 2006

[7] Khaled Belarbi and Faouzi Titel. Genetic Algorithm for the Design of a Class of Fuzzy Controllers: An Alternative Approach, *IEEE Transactions on Fuzzy Systems, 8(4).*2000.

[8] F. Cheon & R. Lai. Constraining the optimization of a Fuzzy Logic Controller Using an Enhance Genetic Algorithm. *IEEE Transactions on Systems, Man and Cybernetics. 30(1)*. 2000. 31-46.

[9] Zhang, J. A Nonlinear Gain Scheduling Control Strategy Based on Neuro-Fuzzy Networks. *Ind. Eng. Chem. Res. 40.* 2001. 3164 –3170.

[10] L. A. Zadeh. Fuzzy Sets. *Information and Control 8*, 1965. 338 – 353.

[11] E. H. Mandani and S. Assilian. An Experiment In Linguistic Synthesis with a Fuzzy Logic Controller, *Int. J. Man Mach. Stud. Vol. 7.* 1975. 1-13.

[12] Gang Feng. A Survey on Analysis and design of Model-Based Fuzzy Control Systems. *IEEE Transactions on Fuzzy Systems,14(5).* 2006.

[13] Holland, J. Adaptation in Natural and Artificial Systems, *University of Michigan Press, Ann Arbor.* 1975.

[14] David E. Golberg, Genetic Algorithms in Search Optimization & Machine Learning. *Addison Wesley Longman Inc.* 1953.

[15] McAvoy, T. J.; Hsu, E.; Lowenthal, S. Dynamics of pH in a Controled Stirred Tank Reactor. *Ind. Eng. Chem. Process Des. Dev*. *11.*1975. 68 –70.

[16] Kostas Zotos. Object-oriented programming in Mathematics, *Applied Mathematics and Computation 188,* 2007. 1562-1566.