Genetic Optimization of Modular Type-1 Fuzzy Controllers for Complex Control Problems

Leticia Cervantes and Oscar Castillo

Tijuana Institute of Technology,
Tijuana, Mexico
ocastillo@tectijuana.mx

Abstract. In this paper a method to design modular type-1 fuzzy controllers using genetic optimization is presented. The method is tested with a problem that requires five individual controllers. Simulation results with a genetic algorithm for optimizing the membership functions of the five individual controllers are presented. Simulation results show that the proposed modular control approach offers advantages over existing control methods.

1 Introduction

This paper focuses on the field of fuzzy logic and control area, these areas can work together to solve various control problems. The problem of water level control for a three tank system is illustrated. This control is carried out by controlling five valves whose outputs are the inputs to the three tanks. The main idea in this paper is to apply a genetic algorithm to optimize the membership functions of the five controllers. Each controller has to open and close one of the valves. To control each of the valves we have five type-1 fuzzy systems and each fuzzy system has to control one valve of the three tanks. After that, the simulation is carried out using type-1 fuzzy systems, and then genetic algorithms are used to optimize the five controllers. Finally results are presented and compared.

The rest of the paper is organized as follows: In section 2 some basic concepts to understand the work are presented, Section 3 shows a case study, problem description and results are presented and finally in Section 4 conclusion is shown.

2 Background and Basic Concepts

In this section some basic concepts needed for this work are presented.

2.1 Genetic Algorithm

Genetic algorithms (GAs) were proposed by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of

P. Melin, O. Castillo (Eds.): *Soft Computing Appli. in Optimization,* STUDFUZZ 294, pp. 125–154.
DOI: 10.1007/978-3-642-35323-9_6

Michigan in the 1960s and the 1970s [2][3]. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems [15][19]. Holland's 1975 book Adaptation in Natural and Artificial Systems presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA [4][5]. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the “fitness” [17]. Holland's GA is a method for moving from one population of "chromosomes" (e.g., strings of ones and zeros, or "bits") to a new population by using a kind of "natural selection" together with the genetics inspired operators of crossover, mutation, and inversion [18]. Each chromosome consists of "genes" (e.g., bits), each gene being an instance of a particular "allele" (e.g., 0 or 1) [14][10]. The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average the fitter chromosomes produce more offspring than the less fit ones [28]. Crossover exchanges subparts of two chromosomes, roughly mimicking biological recombination between two single chromosome ("haploid") organisms; mutation randomly changes the allele values of some locations in the chromosome; and inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed. (Here, as in most of the GA literature, "crossover" and "recombination" will mean the same thing.) [7][16]. Some of the advantages of a GA include: Optimizes with continuous or discrete variables, doesn’t require derivative information, simultaneously searches from a wide sampling of the cost surface, deals with a large number of variables [13][29].

A typical algorithm might consist of the following:

1. Start with a randomly generated population of n l–bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness $f(x)$ of each chromosome x in the population.
3. Repeat the following steps until n offspring have been created:
   - Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.
   - With probability P_c (the "crossover probability" or "crossover rate"), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents. (Note that here the crossover rate is defined to be the probability that two parents will cross over in a single point. There are also "multipoint crossover" versions of the GA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)
   - Mutate the two offspring at each locus with probability P_m (the mutation probability or mutation rate), and place the resulting chromosomes in the

new population. If n is odd, one new population member can be discarded at random.

- Replace the current population with the new population.

Go to step 2 [30][31].

2.2 Fuzzy Systems

The idea of fuzzy systems appeared very early in the literature of fuzzy sets; it was originated by Zadeh (1965). The concept of a fuzzy system is intimately related to that of a fuzzy set. En order to make our discussion self-contained, it will be helpful to begin with a brief summary of some of the basic definitions pertaining to such sets. Research on fuzzy systems seems to have developed in two main directions. The first is rather formal and considers fuzzy systems as a generalization of nondeterministic systems. These have been studied within the same conceptual framework as classical systems. This approach has given birth to a body of abstract results in such fields as minimal realization theory and formal automata theory, sometimes expressed in the setting of category theory. The system is considered over a given period during which inputs, outputs, and relations may change [28][13].

A system will be called fuzzy as soon as inputs or outputs are modeled as fuzzy sets or their interactions are represented by fuzzy relations. Usually, a system is also described in terms of state variables. In a fuzzy system a state can be a fuzzy set. However, the notion of a fuzzy state is quite ambiguous and needs to be clarified. Note that generally a fuzzy system is an approximate representation of a complex process that is not itself necessarily fuzzy [20][21]. According to Zadeh, the human ability to perceive complex phenomena stems from the use of names of fuzzy sets to summarize information [22]. The notion of probabilistic system corresponds to a different point of view: all the available information at any time is modeled by probability distributions, built from repeated experiments. A fuzzy system can be described either as a set of fuzzy logical rules or as a set of fuzzy equations [23][24]. Fuzzy logical rules must be understood as propositions associated with possibility distributions. For instance, “if last input is small, then if last output is large, then current output is medium”, where “small” is a fuzzy set on the universe of inputs, and “medium” and “large” are fuzzy sets on the universe of outputs [25][26] . Let u_t , y_t, and s_t denote respectively the input, output, and state of a system S at time t. U, Y, S are respectively the set of possible inputs, outputs, and states [27][32]. Such a system is said to be deterministic if it is characterized by state equations of the form:

$$s_{t+1} = \delta(u_t, s_t), \qquad y_t = \sigma(s_t), \qquad t \in \mathbb{N}. \tag{1}$$

s_0 is called the initial state; δ and Ω are functions from U X S and from S to S and Y, respectively. S is said to be nondeterministic if S_{t+1} and / or Y_t, are not uniquely determined by U_t and S_t [33][1]. Let S_{t+1} and Y_t be the sets of possible values of S_{t+1} and Y_t, respectively, given U_t, and S_t. S_{t+1} and Y_t, may be understood as binary possibility distributions over S and Y, respectively. In some cases a fuzzy system is used to control complex problem to obtain better results [8][9][6].

2.3 Fuzzy Control Systems

Control systems theory, or what is called modern control systems theory today, can be traced back to the age of World War II, or even earlier, when the design, analysis, and synthesis of servomechanisms were essential in the manufacturing of electromechanical systems. The development of control systems theory has since gone through an evolutionary process, starting from some basic, simplistic, frequency-domain analysis for single-input single output (SISO) linear control systems, and generalized to a mathematically sophisticated modern theory of multi-input multi-output (MIMO) linear or nonlinear systems described by differential and/or difference equations.

It is believed that the advances of space technology in the 1950s completely changed the spirit and orientation of the classical control systems theory: the challenges posed by the high accuracy and extreme complexity of the space systems, such as space vehicles and structures, stimulated and promoted the existing control theory very strongly, developing it to such a high mathematical level that can use many new concepts like state-space and optimal controls. The theory is still rapidly growing today; it employs many advanced mathematics such as differential geometry, operation theory, and functional analysis, and connects to many theoretical and applied sciences like artificial intelligence, computer science, and various types of engineering. This modern control systems theory, referred to as *conventional* or *classical* control systems theory, has been extensively developed. The theory is now relatively complete for linear control systems, and has taken the lead in modern technology and industrial applications where control and automation are fundamental. Basically, the aim of fuzzy control systems theory is to extend the existing successful conventional control systems techniques and methods as much as possible, and to develop many new and special-purposed ones, for a much larger class of complex, complicated, and ill-modeled systems – fuzzy systems. This theory is developed for solving real-world problems [11].

Fuzzy controllers have been well accepted in control engineering practice. The major advantages in all these fuzzy-based control schemes are that the developed controllers can be employed to deal with increasingly complex systems to implement the controller without any precise knowledge of the structure of entire dynamic model. As a knowledge-based approach, the fuzzy controller usually depends on linguistics-based reasoning in design. However, even though a system is well defined mathematically, the fuzzy controller is still preferred by control engineers since it is relatively more understandable whereas expert knowledge can be incorporated conveniently. Recently, the fuzzy controller of nonlinear systems was studied by many authors and has also been extensively adopted in adaptive control of robot manipulators. It has been proven that adaptive fuzzy control is a powerful technique and being increasingly applied in the discipline of systems control, especially when the controlled system has uncertainties and highly nonlinearities [12].

3 Case Study

In this Section the problem description is presented and results are shown.

3.1 Problem Description

In this work the case study considers the problem of water level control for a 3 tanks system where the 3 tanks include valves that are opened or closed, these valves must be well controlled to give the desired level of water in each of the three tanks. The end tanks have a valve that fills and in the middle of the 3 tanks there are two valves that control the water level between tanks 1 and 2, and tanks 2 and 3. The water tank 3 has a valve to output more water flow, the case study model is made in Simulink and has three inputs (tank 1, tank2 and tank3), and these inputs correspond to the existing water levels in tank 1, tank2 and tank3. The outputs of the model made in Simulink has five valves, which provide water (v1 and v2) valves that are interconnected tanks (v13 and v32) and finally the output valve is responsible for the drainage of the three tanks (v20). The problem is shown in Figure 1.

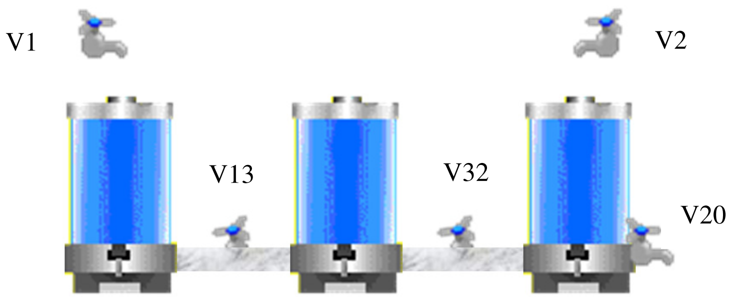

Fig. 1 Water control of 3 tanks

3.2 Type-1 Fuzzy System

For this case study it was necessary to use fuzzy systems to realize the simulation, each fuzzy system has one or two inputs depend on the valve. The Valves that are between 2 tanks are using 2 inputs (tank1 and tank2 or tank2 and tank3). The outputs are the valves, in total 5 fuzzy systems were used in this problem. The fuzzy systems are shown in Figures 2 to 6.

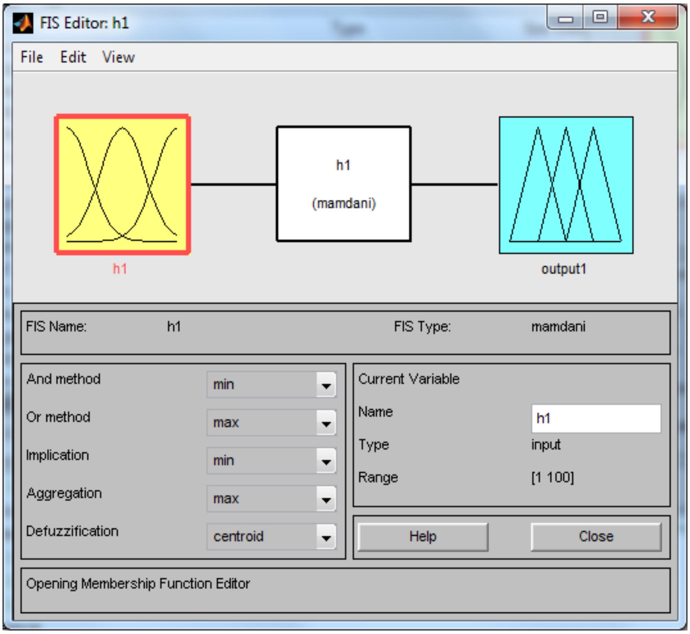

Fig. 2 Fuzzy system to control valve 1

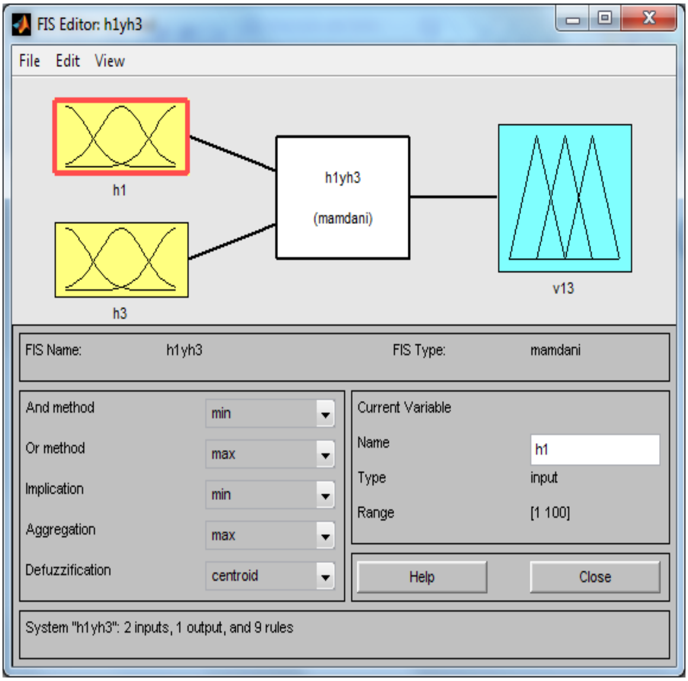

Fig. 3 Fuzzy system to control valve 13

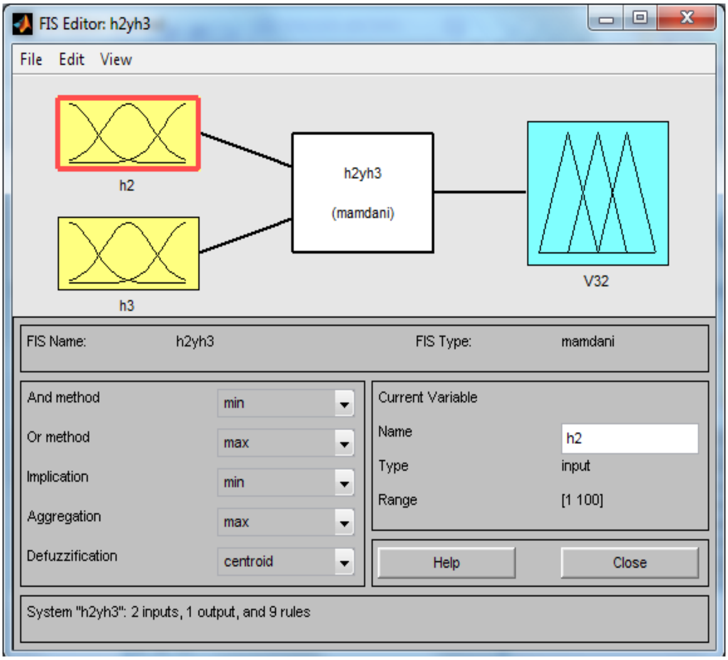

Fig. 4 Fuzzy system to control valve 32

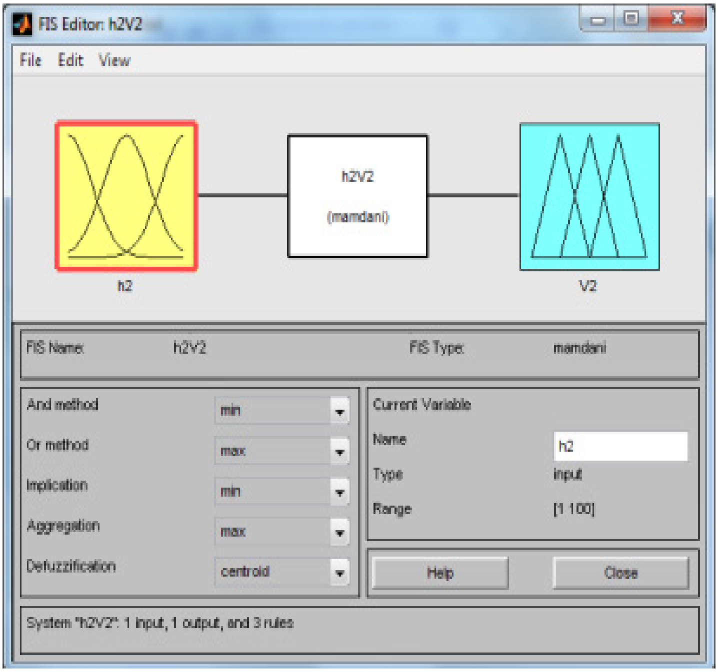

Fig. 5 Fuzzy system to control valve 2

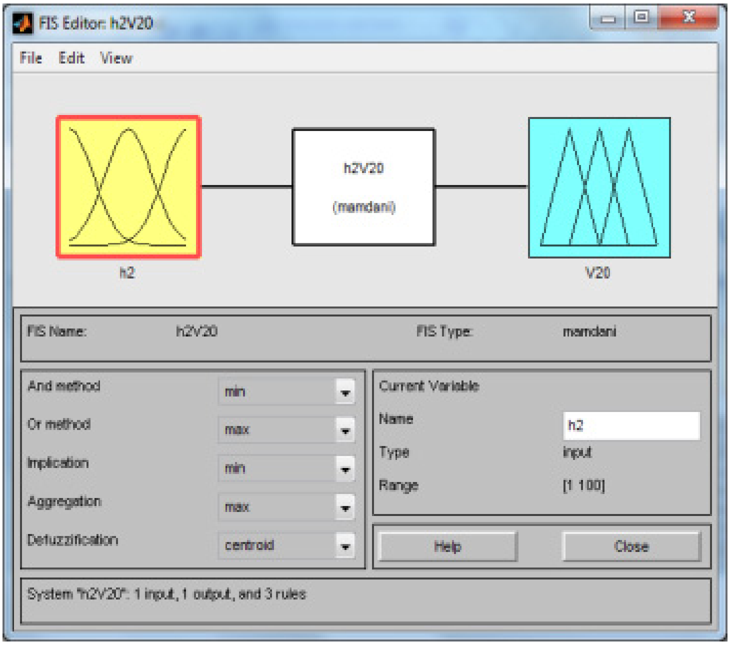

Fig. 6 Fuzzy system to control valve 20

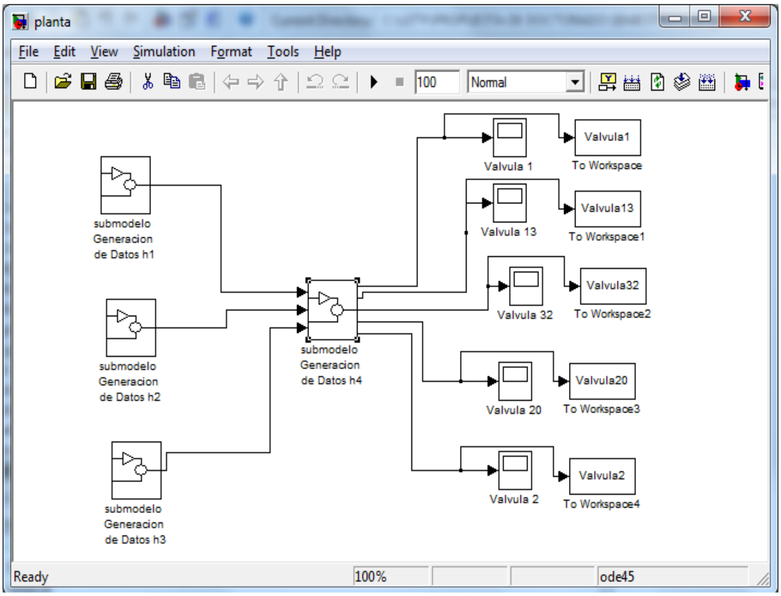

Fig. 7 Simulation plant

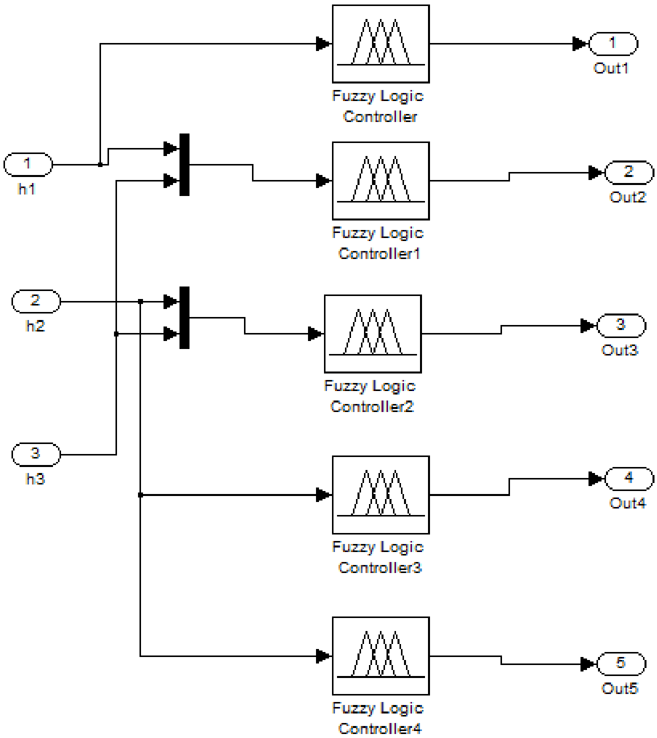

Fig. 8 Simulation plant showing inputs and outputs

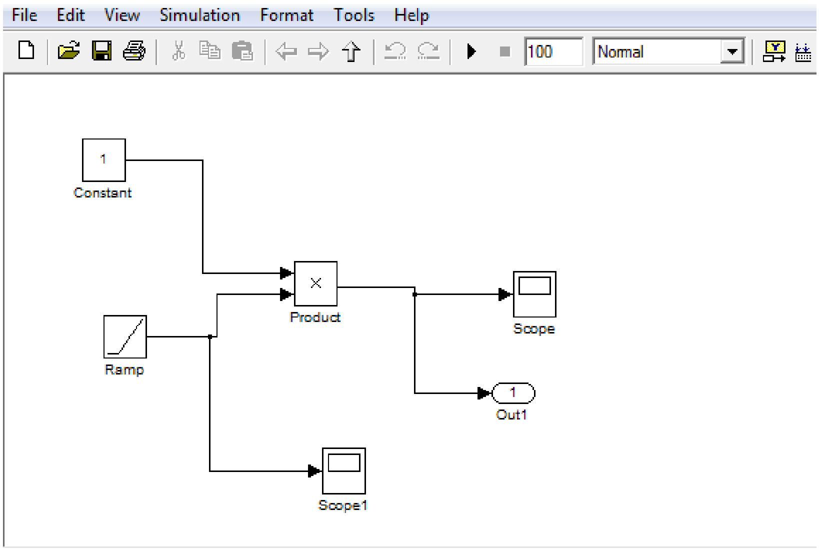

Fig. 9 Tank water simulation plant

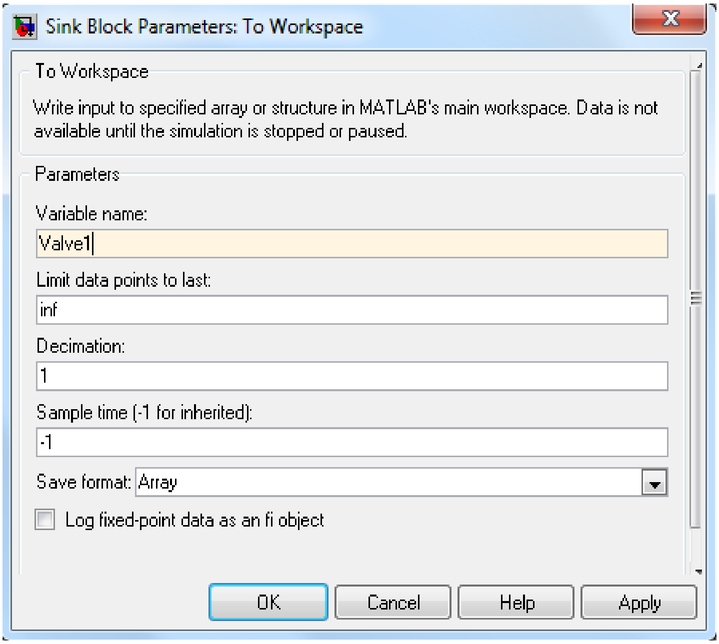

Fig. 10 Data block of the simulation plant

Having created the previous fuzzy systems, the simulation was performed using the Matlab language. The simulation plant is shown in Figures 7 to 10.

The simulation was carried out using the fuzzy systems shown before, the membership functions used in this case were triangular, Gaussian and trapezoidal, and the fuzzy systems with the different types of membership functions used in this case of study are shown in Figures 11 to 16.

All the valves in the inputs and outputs have 3 membership functions, all the membership functions in each input or output have the same position initially and this is because a genetic algorithm is applied to optimize each membership function.

When the genetic algorithm is used the membership functions start to move within the specified range. Later in section 3.3 the fuzzy system with genetic algorithm is presented where it shows new positions in all de membership functions. Figures 11 to 16 show the membership functions in the inputs and outputs of all fuzzy systems. The fuzzy systems that have one input are presented in Figures 11 to 13, and the fuzzy systems that have 2 inputs are presented in Figures 14 to 16.

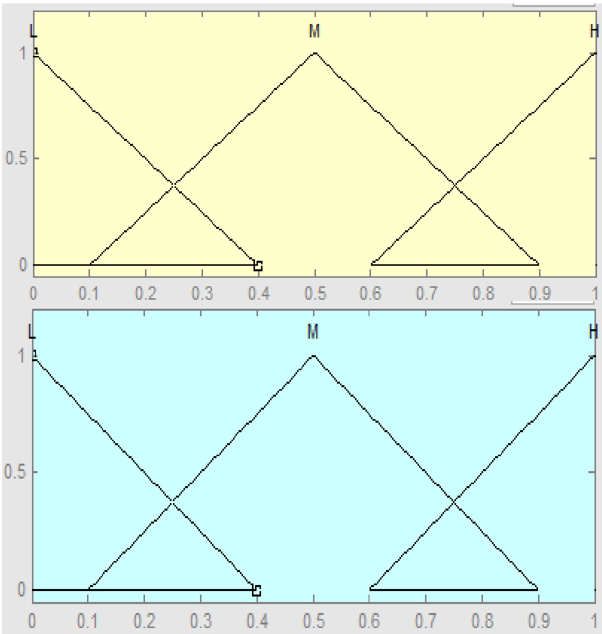

Fig. 11 Triangular membership functions use in valve 1, valve 2 and valve 20

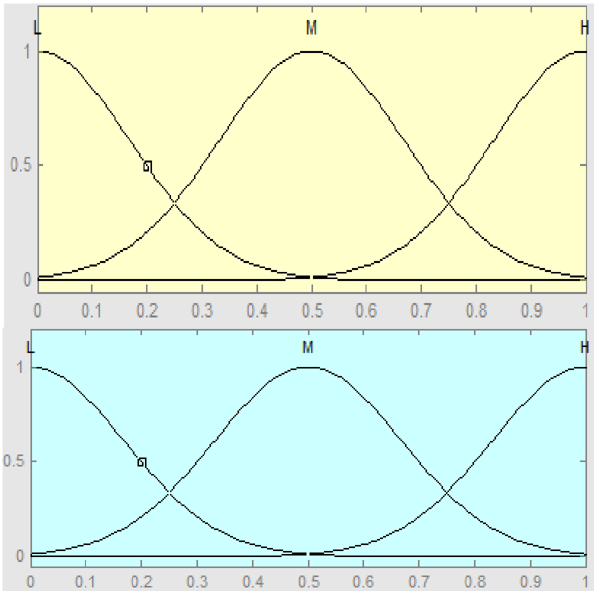

Fig. 12 Gaussian membership functions use in valve 1,valve 2 and valve 20

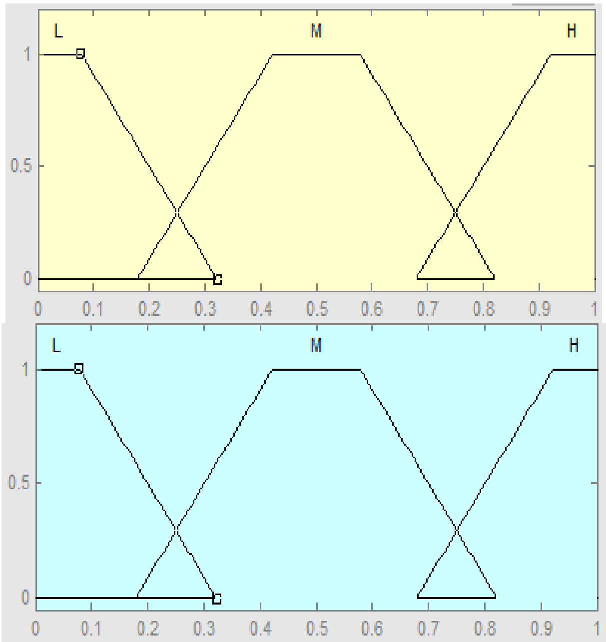

Fig. 13 Trapezoidal membership functions use in valve 1,valve 2 and valve 20

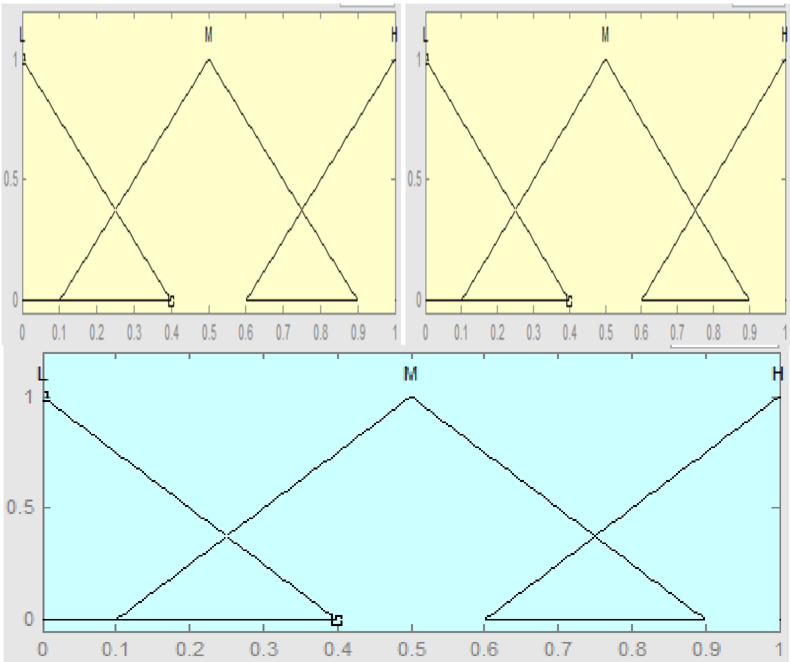

Fig. 14 Triangular membership functions use in valve 13 and valve 32

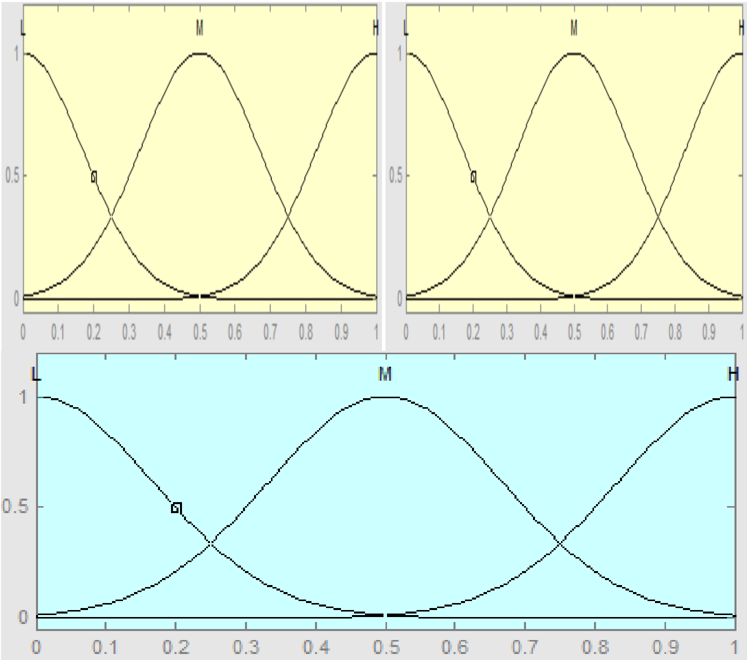

Fig. 15 Gaussian membership functions use in valve 13 and valve 32

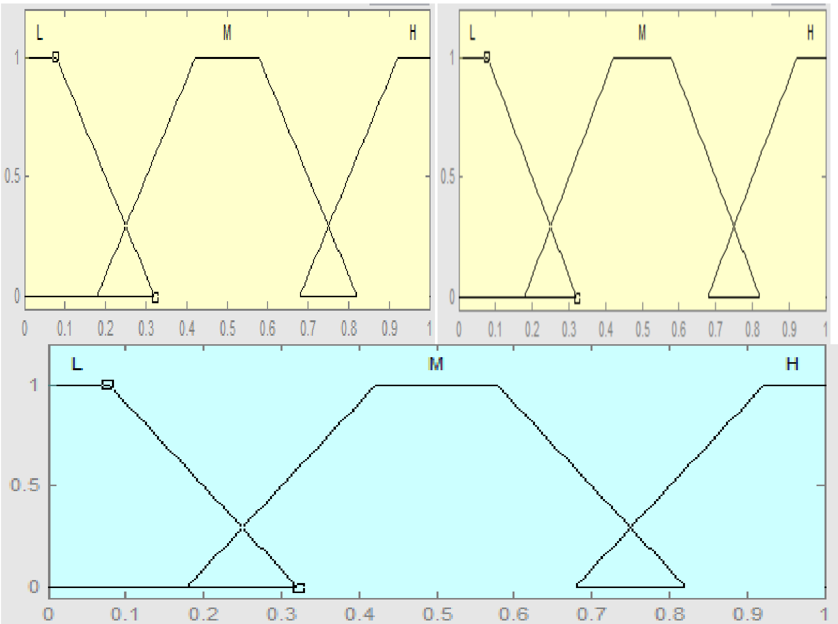

Fig. 16 Trapezoidal membership functions use in valve 13 and valve 32

Table 1 Results for the simulation plant using triangular membership functions

Using Triangular Membership Function	Error
valve 1	0.9246
valve 13	0.9278
valve 2	0.9278
valve 20	0.9279
valve 32	0.8341

Table 2 Results for the simulation plant using Gaussian membership functions

Using Gaussian Membership Function	Error
valve 1	0.898
valve 13	0.8994
valve 2	0.8994
valve 20	0.8995
valve 32	0.8463

Table 3 Results for the simulation plant using trapezoidal membership functions

Using Trapezoidal Membership Function	Error
valve 1	0.9522
valve 13	0.9551
valve 2	0.9504
valve 20	0.9551
valve 32	0.8233

The results with type-1 fuzzy systems are presented in Tables 1 to 3.

In the previous Tables the error of each valve was presented using different types of membership functions. The error when triangular membership functions were used is different as in trapezoidal and Gaussian. Is important to use different types of membership functions because the error can be vary and depends how complex is the problem to control sometimes the case of the study can be better using Gaussian membership function for the soft behavior or in other case can be better using another type, but is important to consider other types. In Figures 17 to 21 graphics are shown of each valve with the 3 types of membership functions used in this case.

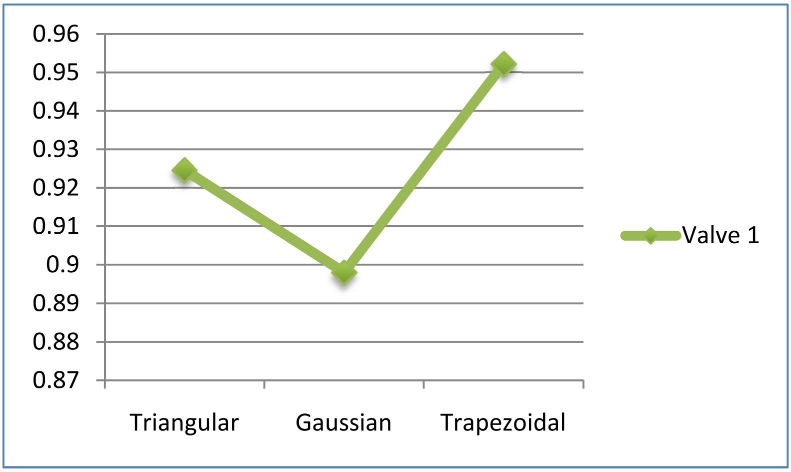

Fig. 17 Error of Valve 1 with 3 types of membership functions

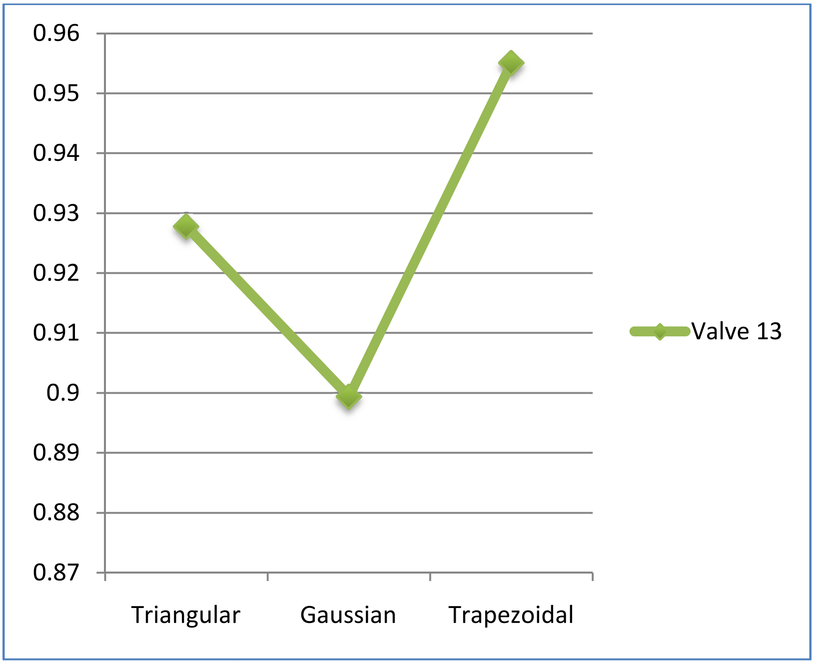

Fig. 18 Error of Valve 13 with 3 types of membership functions

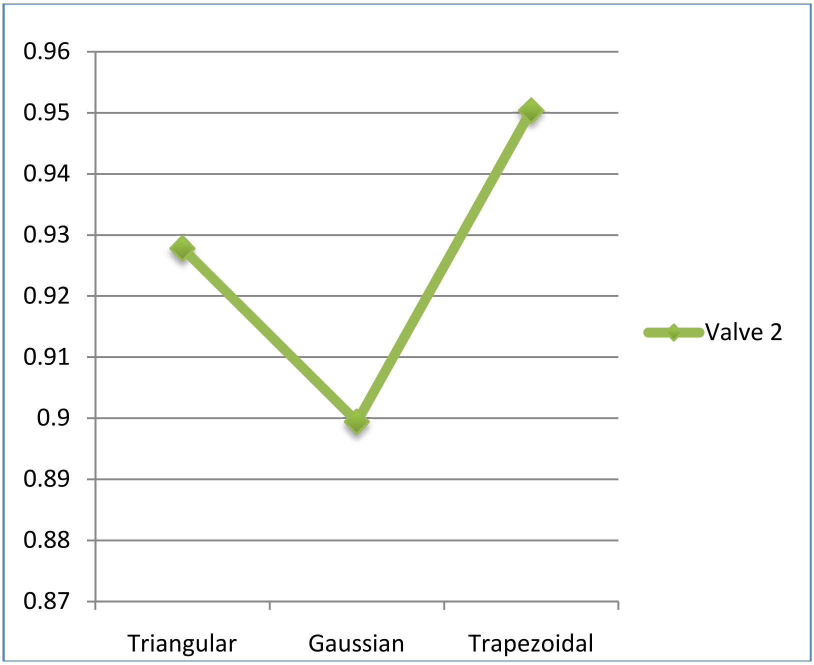

Fig. 19 Error of Valve 2 with 3 types of membership functions

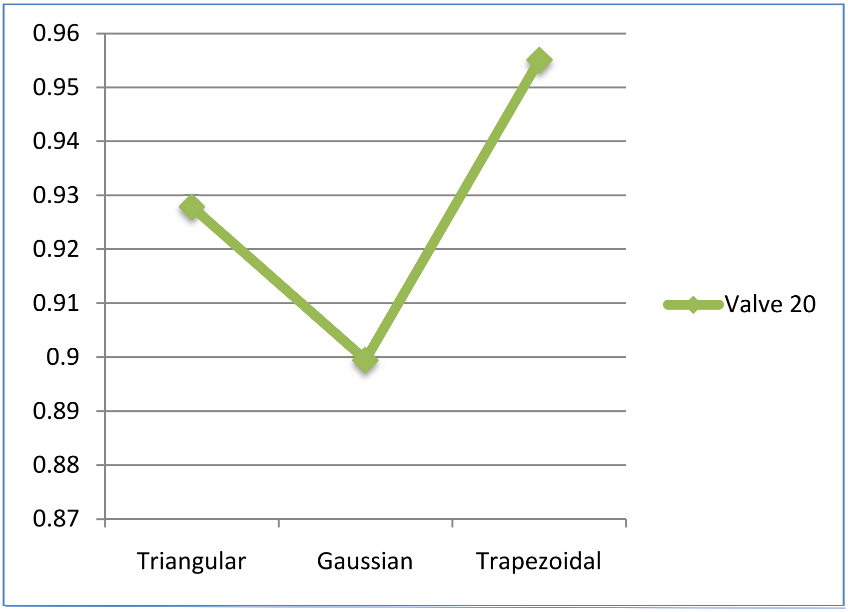

Fig. 20 Error of Valve 20 with 3 types of membership functions

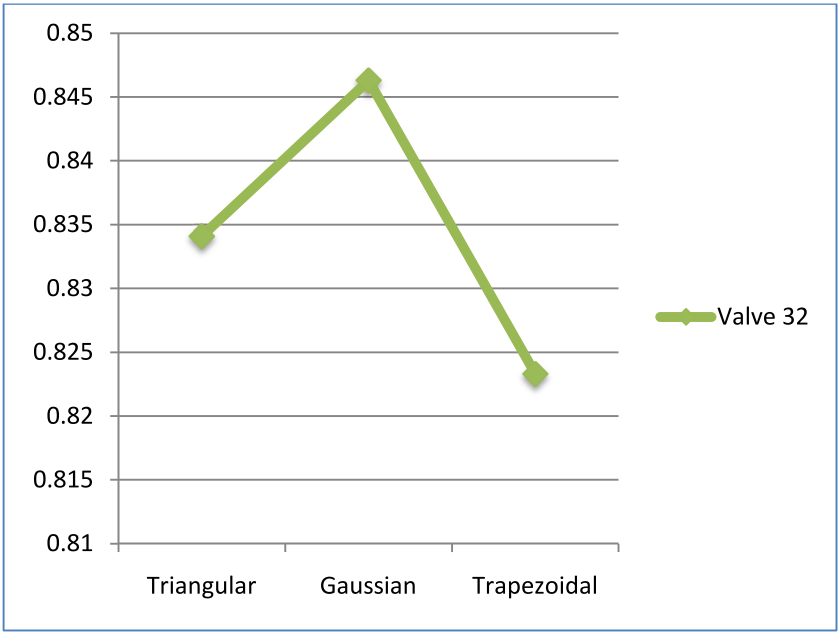

Fig. 21 Error of Valve 20 with 3 types of membership functions

1. If (h1 is BAJO) then (output1 is 1) (1)
2. If (h1 is NORMAL) then (output1 is 0) (1)
3. If (h1 is ALTO) then (output1 is m1) (1)

1. If (h1 is bajo) and (h3 is bajo) then (v13 is 1) (1)
2. If (h1 is bajo) and (h3 is normal) then (v13 is 1) (1)
3. If (h1 is bajo) and (h3 is alto) then (v13 is 1) (1)
4. If (h1 is normal) and (h3 is bajo) then (v13 is 0) (1)
5. If (h1 is normal) and (h3 is normal) then (v13 is 0) (1)
6. If (h1 is normal) and (h3 is alto) then (v13 is 0) (1)
7. If (h1 is alto) and (h3 is bajo) then (v13 is m1) (1)
8. If (h1 is alto) and (h3 is normal) then (v13 is m1) (1)
9. If (h1 is alto) and (h3 is alto) then (v13 is m1) (1)

1. If (h2 is bajo) then (V2 is 1) (1)
2. If (h2 is normal) then (V2 is 0) (1)
3. If (h2 is alto) then (V2 is m1) (1)

1. If (h2 is bajo) then (V20 is m1) (1)
2. If (h2 is normal) then (V20 is 0) (1)
3. If (h2 is alto) then (V20 is 1) (1)

1. If (h2 is bajo) and (h3 is bajo) then (V32 is 1) (1)
2. If (h2 is bajo) and (h3 is normal) then (V32 is 1) (1)
3. If (h2 is bajo) and (h3 is alto) then (V32 is 1) (1)
4. If (h2 is normal) and (h3 is bajo) then (V32 is 1) (1)
5. If (h2 is normal) and (h3 is normal) then (V32 is 0) (1)
6. If (h2 is normal) and (h3 is alto) then (V32 is 1) (1)
7. If (h2 is alto) and (h3 is bajo) then (V32 is 1) (1)
8. If (h2 is alto) and (h3 is normal) then (V32 is 1) (1)
9. If (h2 is alto) and (h3 is alto) then (V32 is 1) (1)

Fig. 22 Rules of the 5 type-1 fuzzy systems

The rules used to control in the case of the three tanks are shown in Figure 22.

The set of rules shown above rules are for the five fuzzy systems used to control the open and closed valves from the three tanks.

The first three rules are the controller number 1, the 9 following rules are controller 2, the third set of rules are the controller 3, the fourth set of 3 rules are the controller 4 and the last 9 rules are controller number 5.

The difference in the number of rules of each controller is because depending on the number of inputs, outputs and membership functions of fuzzy system will have a number of rules to be had. For example to control valve number one has only one input which is the tank 1, one output and has 3 membership functions therefore the number of rules are 3. The valves between 2 tanks need 2 inputs (tank1 and tank2 or tank 2 and tank3), these valves have one output and three membership functions therefore need 9 rules for fuzzy systems.

3.3 Genetic Algorithm

After obtaining the previous mentioned results, genetic algorithm optimization was performed. The genetic algorithm is used to optimize the membership functions of each fuzzy system (inputs and outputs).

In the genetic algorithm the membership functions of the 5 controllers were optimized.

In the algorithm the error of each controller is taken and finally the results of each controller were added, and the final result is divided between the number of controllers. The fitness function is shown in next equation:

$$f(y) = \frac{\left(\text{Error C1} = \frac{\sum_{i=1}^{n} \left| y^{i}_{REF_1} - y^{i}_{FS_1} \right|}{n} + \text{Error C2} = \frac{\sum_{i=1}^{n} \left| y^{i}_{REF_2} - y^{i}_{FS_2} \right|}{n} + \ldots \text{Error CN} = \frac{\sum_{i=1}^{n} \left| y^{i}_{REF_N} - y^{i}_{FS_N} \right|}{n} \right)}{N} \quad (2)$$

Where Y_{REF} is the reference, Y_{FS} is the output of the controller and **n** is the number of point used in comparison. Error C1 is the error of control 1 to N, and N in the number of the controllers.

The parameters used in the GA are shown in Figure 23.

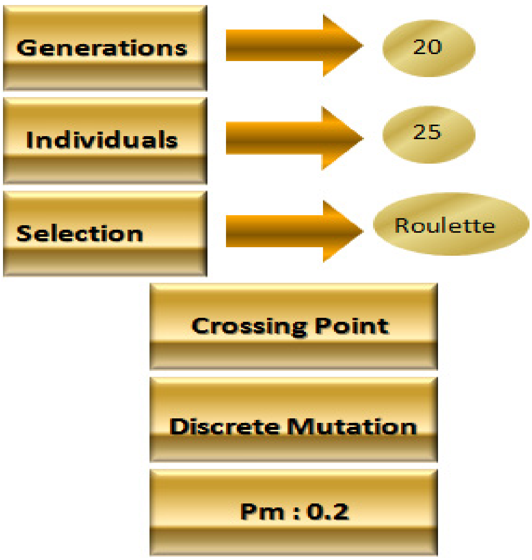

Fig. 23 Parameters of the genetic algorithm

After the use of the genetic algorithm the results obtained in the simulation are shown in Table 4.

Table 4 Results for the simulation plant using triangular membership functions and genetic algorithm

Error using triangular membership functions and genetic algorithm				
Valve 13	Valve 1	Valve 20	Valve 2	Valve 32
0.109	0.1146	0.0939	0.2077	0.218
0.131	0.1228	0.1329	0.1861	0
0.119	0.1275	0.111	0.239	0
0.115	0.1116	0.1092	0.2216	0
0.109	0.0908	0.1191	0.214	0
0.109	0.1132	0.0954	0.1922	0
0.117	0.1225	0.1003	0.1853	0
0.107	0.1102	0.1146	0.1938	0
0.105	0.0993	0.0851	0.2428	0
0.125	0.1196	0.113	0.1433	0
0.123	0.1191	0.1394	0.246	0
0.115	0.1114	0.091	0.1539	0
0.117	0.1231	0.101	0.1818	0
0.107	0.1444	0.0661	0.1366	0
0.117	0.1225	0.1003	0.1853	0

The above table shows a lower error in comparison with only using a type-1 fuzzy system. In the last table a genetic algorithm was used with triangular membership functions, the error is different in each valve even though the parameters are the same in all the tests. Some Graphics are shown in Figures 24 to 29 to present the behavior of each valve. In the last graphic the behavior of all valves is shown to observe all the behaviors.

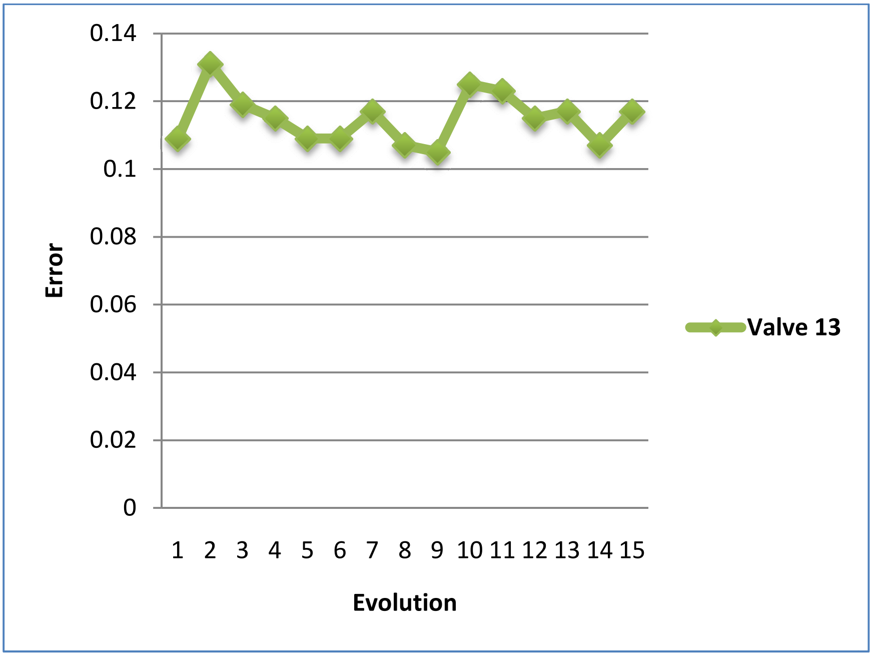

Fig. 24 Error of Valve 13 using a genetic algorithm

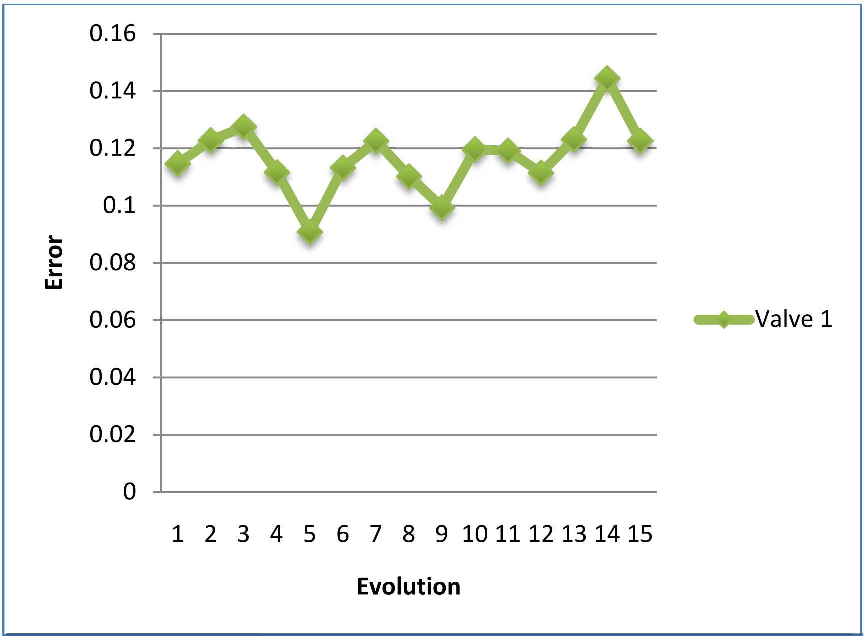

Fig. 25 Error of Valve 1 using a genetic algorithm

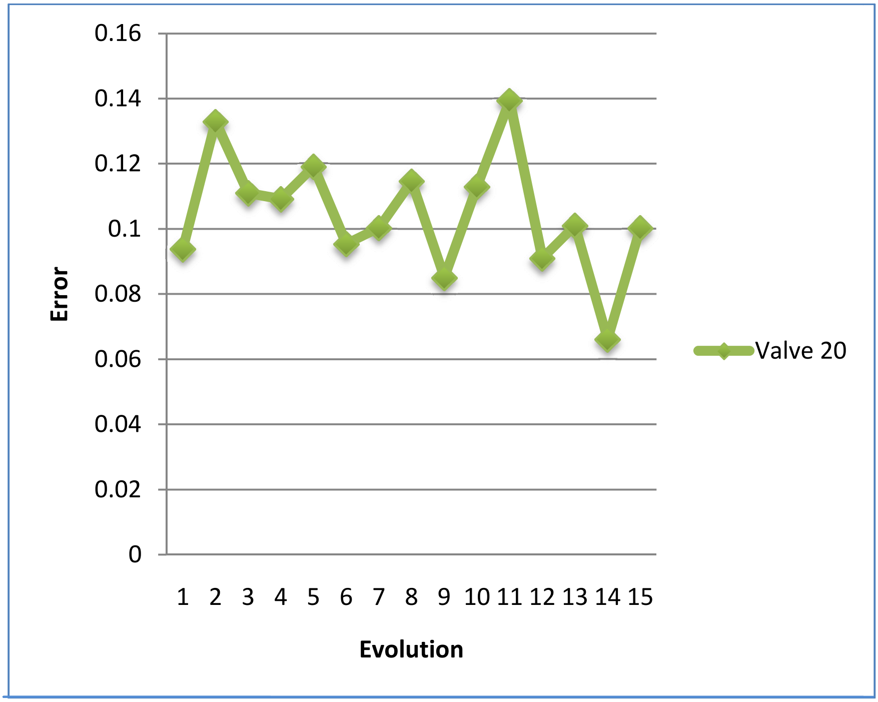

Fig. 26 Error of Valve 20 using a genetic algorithm

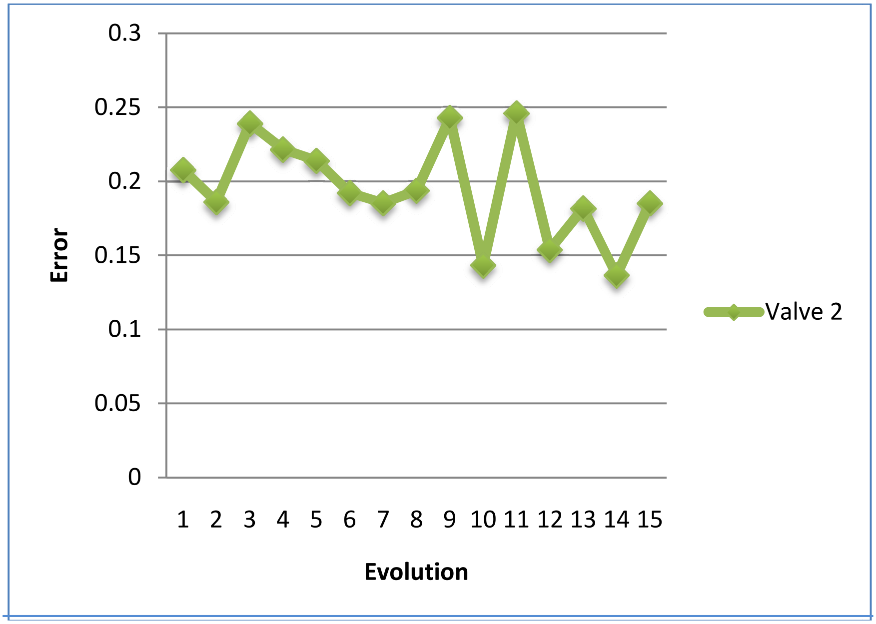

Fig. 27 Error of Valve 2 using a genetic algorithm

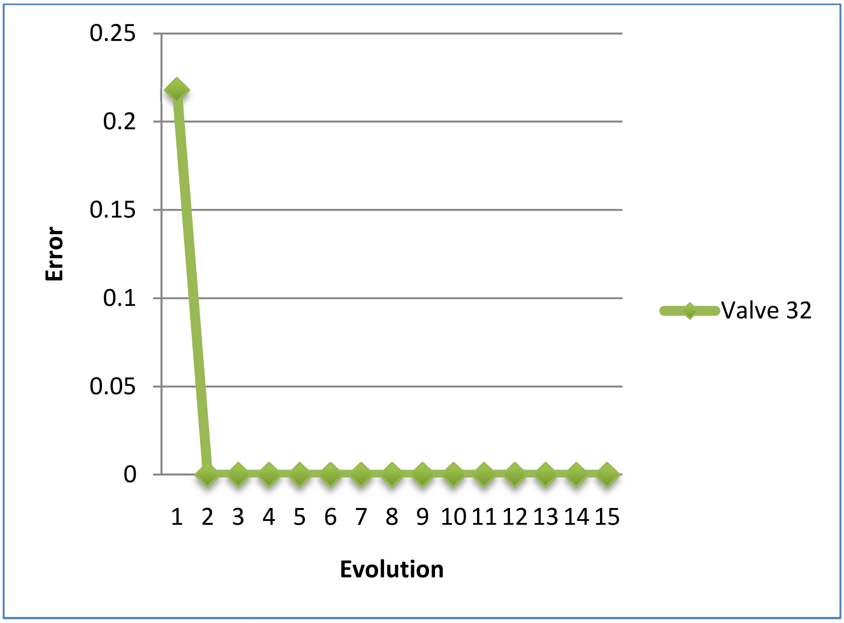

Fig. 28 Error of Valve 32 using a genetic algorithm

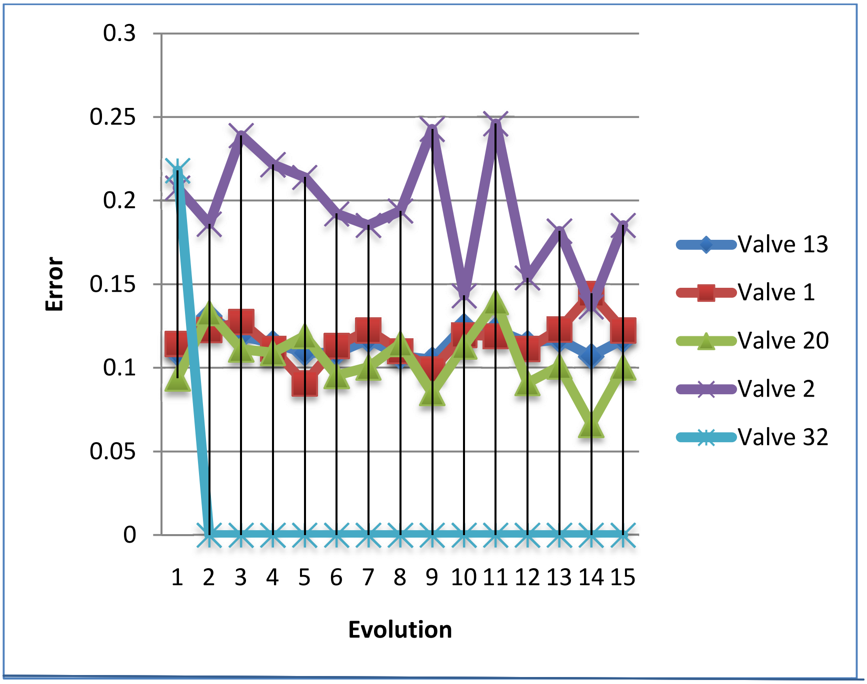

Fig. 29 Behavior of each valve using a genetic algorithm

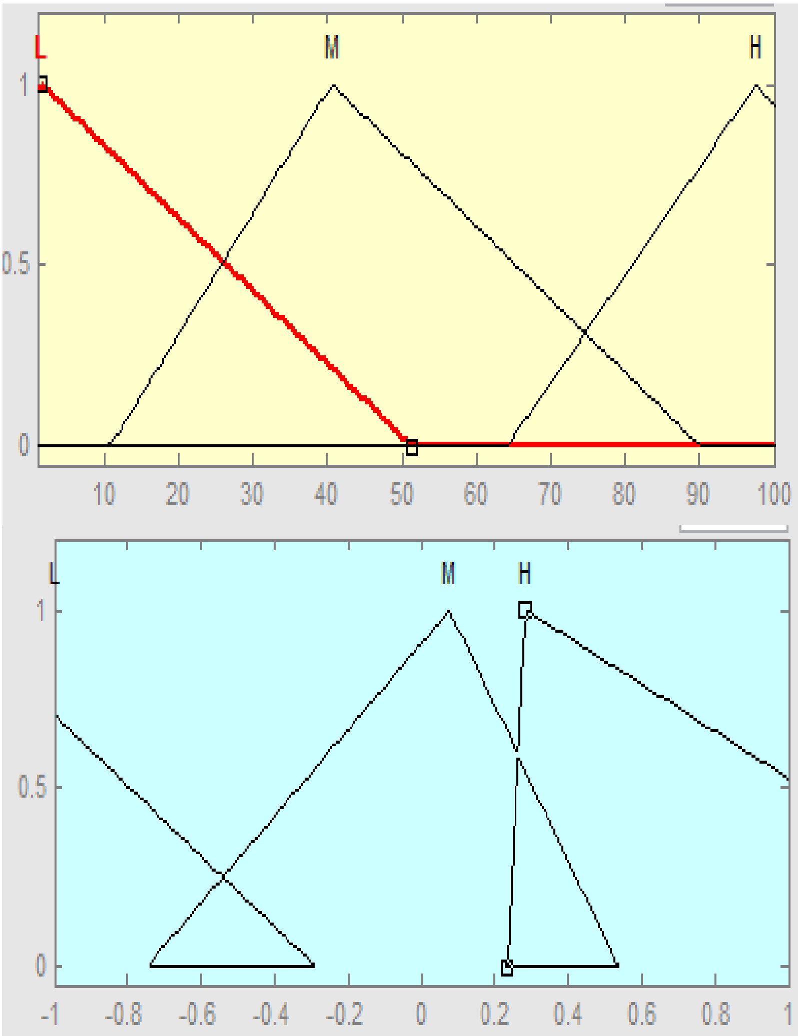

Fig. 30 Best Fuzzy system of valve 1 using genetic algorithm

Applying the genetic algorithm to a type-1 fuzzy system of each valve it was obtain the best fuzzy system of each valve as shown in Figures 30 to 34.

Last figure represents the best fuzzy system of valve 1 and its membership function of the input and the output. Yellow box is the input of the fuzzy system and the blue box is the output of the fuzzy system. In next fuzzy systems all the inputs of each are the yellow boxes and the outputs are the blue boxes.

All the fuzzy systems have 3 membership functions in the inputs and the outputs of each valve. When the genetic algorithm was implemented, more than 1 fuzzy systems were obtained, but in this case the best of the 15 evolutions is presented.

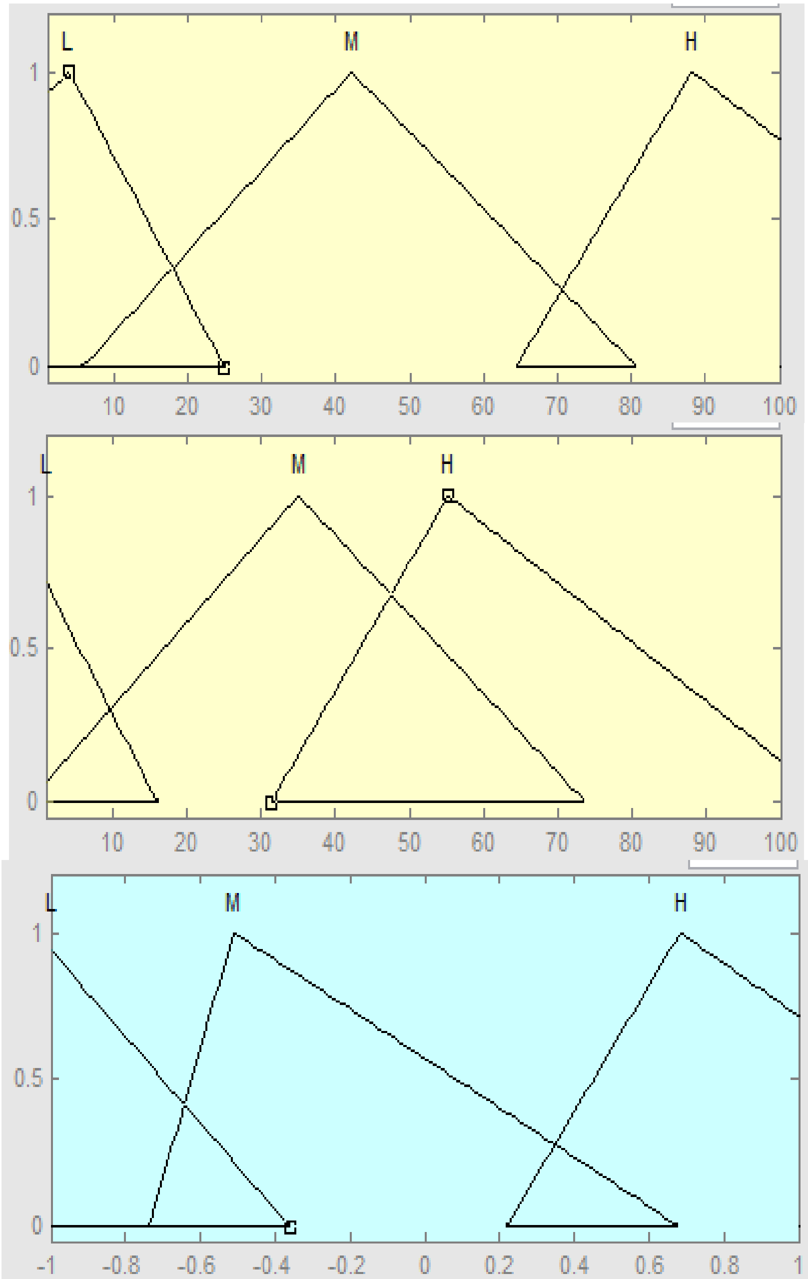

Fig. 31 Best Fuzzy system of valve 13 using genetic algorithm

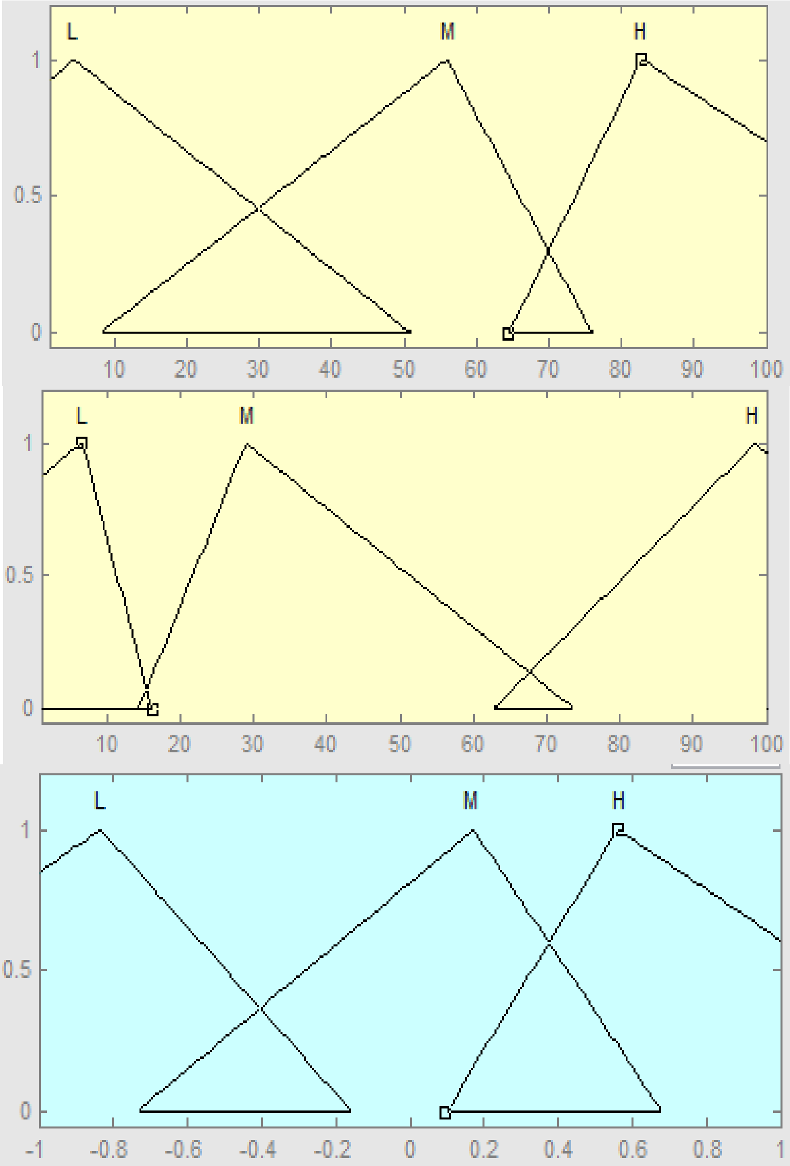

Fig. 32 Best Fuzzy system of valve 32 using genetic algorithm

Recall that this fuzzy system has two inputs because the valve 13 that is controlled is fed by two tanks (Tank 1 and Tank2).

This case is the same as that of the last fuzzy system, it needs two inputs to control the valve 32 because this valve is fed by two tanks (Tank 2 and Tank3). Valve 32 and valve 13 are the only ones needs two inputs, the reason is because as was explain those valves are between two tanks.

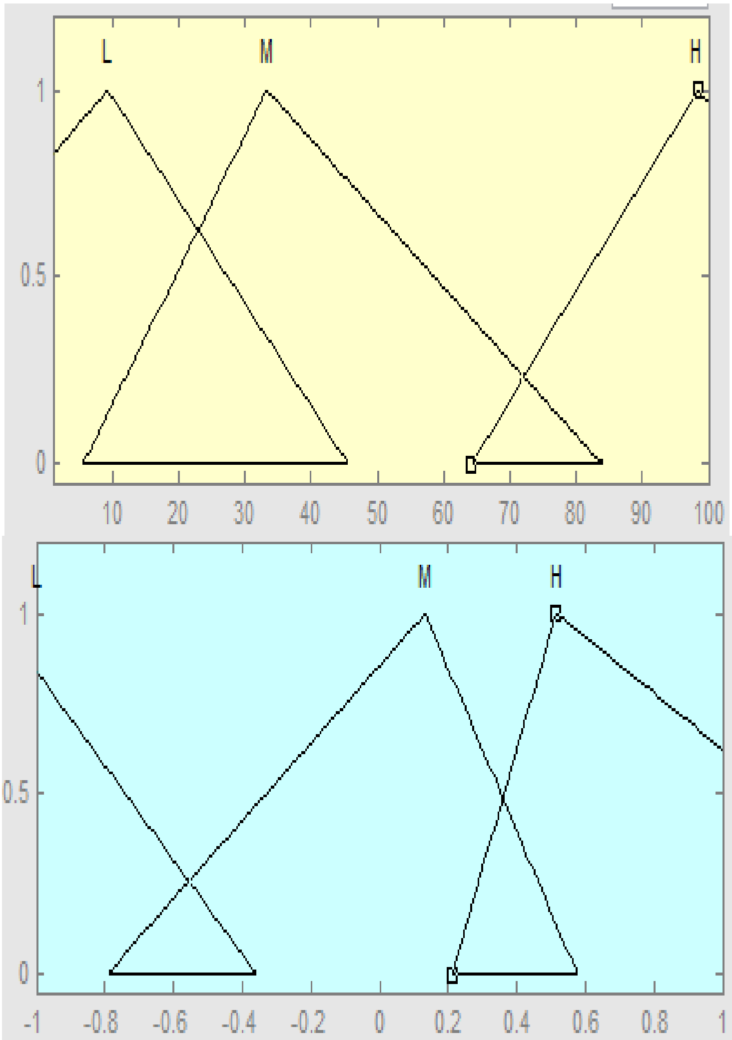

Fig. 33 Best Fuzzy system of valve 2 using genetic algorithm

Fuzzy systems have become a tool that can be useful to try and model the complex and nonlinear systems. And these fuzzy systems in this case study helps improve control valves. Membership functions can be varied to get more results. These fuzzy systems use three membership functions to establish the level of open or closed for the valves, the level of each membership function in the valves are open completely, half open and close.

The granulation of fuzzy systems may be increased and instead of using three membership functions it can be used 5 or another option, which could consider the valve as open medium, open, closed, half closed, fully closed. This depends on how you want to study the problem.

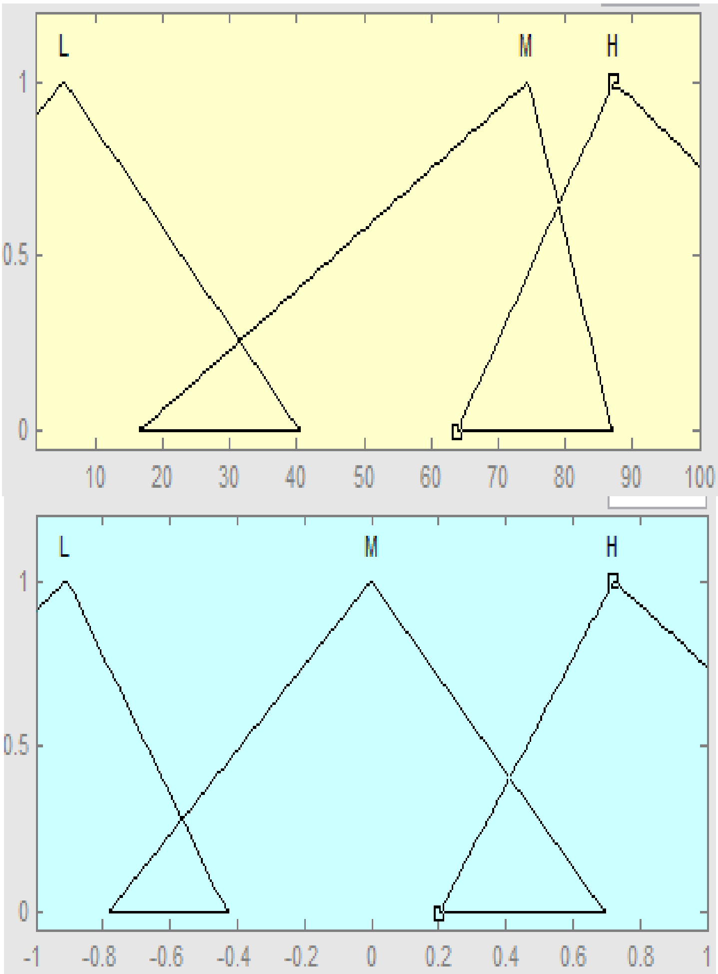

Fig. 34 Best Fuzzy system of valve 20 using genetic algorithm

4 Conclusions

A benchmark problem was used to test the proposed approach and based on the obtained results we can say that to achieve control of the present problem, a genetic algorithm is a good alternative to obtain a good fuzzy controller.

When a complex control problem is at hand, we start working on the case study, and once results are obtained with type-1 fuzzy systems it is a good choice to use a genetic algorithm for optimizing membership functions of the inputs and outputs of the controllers and to obtain better control, as was the case in this control problem. In the moment when genetic algorithm was used, results were better than with an initial type-1 fuzzy system, this is possible because in the moment that genetic algorithm is applied, it moves the parameters of the membership functions and the system has more options to control de valves and the genetic algorithm is evaluated to obtain the best fuzzy system to control the open and close valves and this is why better results are obtained by optimizing the membership functions.

References

1. Castillo, O.: Type-2 Fuzzy Logic in Intelligent Control Applications. STUDFUZZ, vol. 272. Springer, Heidelberg (2012)
2. Castillo, O., Martinez-Marroquin, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf. Sci. 192, 19–38 (2012)
3. Castillo, O., Melin, P.: A review on the design and optimization of interval type-2 fuzzy controllers. Appl. Soft Comput. 12(4), 1267–1278 (2012)
4. Castillo, O., Melin, P.: New fuzzy-fractal-genetic method for automated Mathematical Modelling and Simulation of Robotic Dynamic Systems. In: IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1182–1187 (1998)
5. Castillo, O., Melin, P., Montiel, O., Sepúlveda, R.: Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms. Soft Comput. 15(6), 1145–1160 (2011)
6. Castillo, O., Kacprzyk, J., Pedrycz, W.: Soft Computing for Intelligent Control and Mobile Robotics. Springer (2011)
7. Cázarez, N., Aguilar, L., Castillo, O.: Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash. Expert System Appl. 37(6), 4368–4378 (2010)
8. Cervantes, L., Castillo, O.: Design of a fuzzy system for the longitudinal control of an F-14 airplane. In: Castillo, O., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Intelligent Control and Mobile Robotics. SCI, vol. 318, pp. 213–224. Springer, Heidelberg (2010)
9. Cervantes, L., Castillo, O., Melin, P.: Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part II. LNCS, vol. 7095, pp. 1–12. Springer, Heidelberg (2011)
10. Coley, A.: An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific (1999)
11. Chen, G., Pham, T.: Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems (2001)
12. Dadios, E.: Fuzzy Logic-Controls, Concepts, Theories and Applications (2012)
13. Dubois, D., Prade, H.: Fuzzy sets and Systems: Theory and Applications (1980)
14. Gibbens, P., Boyle, D.: Introductory Flight Mechanics and Performance. University of Sydney, Australia (1999)
15. Haupt, R., Haupt, S.: Practical Genetic Algorithm. Wiley Interscience (2004)
16. Hidalgo, D., Melin, P., Castillo, O.: An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms. Expert Syst. Appl. 39(4), 4590–4598 (2012)

17. Martinez-Soto, R., Castillo, O., Aguilar, L.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inf. Sci. 179(13), 2158–2174 (2009)
18. Melin, P., Castillo, O.: A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks. In: IEEE International Conference on Fuzzy Systems, vol. 1, pp. 420–425 (2003)
19. Mitchell, M.: An Introduction to Genetic Algorithms. Massachusetts Institute of Technology (1999)
20. Rachman, E., Jaam, J., Hasnah, A.: Non-linear simulation of controller for longitudinal control augmentation system of F-16 using numerical approach. Information Sciences Journal 164(1-4), 47–60 (2004)
21. Reiner, J., Balas, G., Garrard, W.: Flight control design using robust dynamic inversion and time- scale separation. Automatic Journal 32(11), 1493–1504 (1996)
22. Sanchez, E., Becerra, H., Velez, C.: Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. Journal of Information Sciences 177(10), 1999–2022 (2007)
23. Sefer, K., Omer, C., Okyay, K.: Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. Expert Systems with Applications Journal 37(2), 1229–1234 (2010)
24. Song, Y., Wang, H.: Design of Flight Control System for a Small Unmanned Tilt Rotor Aircraft. Chinese Journal of Aeronautics 22(3), 250–256 (2009)
25. Walker, D.J.: Multivariable control of the longitudinal and lateral dynamics of a fly by wire helicopter. Control Engineering Practice 11(7), 781–795 (2003)
26. Wu, D.: A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems (July 22, 2010)
27. Wu, D., Mendel, J.: On the Continuity of Type-1 and Interval Type-2 Fuzzy Logic Systems. IEEE T. Fuzzy Systems 19(1), 179–192 (2011)
28. Zadeh, L.: Fuzzy Sets and Fuzzy Information Granulation Theory. Beijing Normal University Press, Baijing (2000)
29. Zadeh, L.: Fuzzy Sets, Information and Control, vol. 8(3), pp. 338–353 (1965)
30. Zadeh, L.: Shadows of Fuzzy Sets. Probl. Peredachi Inf. 2(1), 37–44 (1966)
31. Zadeh, L.: Fuzzy Logic. Neural Networks and Soft Computing Commun. ACM 37(3), 77–84 (1994)
32. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computting 2, 23–25 (1998)
33. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. SMC-3, 28–44 (1973)