

Genetic Design of an Interval Type-2 Fuzzy Controller for Velocity Regulation in a DC Motor

Regular Paper

Yazmin Maldonado¹ and Oscar Castillo^{1,*}¹ Tijuana Institute Technology, Division of Graduate Studies, Tijuana, Mexico

* Corresponding author E-mail: ocastillo@tectijuana.mx

Received 24 May 2012; Accepted 30 Jun 2012

DOI: 10.5772/51188

© 2012 Maldonado and Castillo; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract This paper proposes the design of a Type-2 Fuzzy Logic Controller (T2-FLC) using Genetic Algorithms (GAs). The T2-FLC was tested with different levels of uncertainty to regulate velocity in a Direct Current (DC) motor. The T2-FLC was synthesized in Very High Description Language (VHDL) code for a Field-programmable Gate Array (FPGA), using the Xilinx System Generator (XSG) of Xilinx ISE and Matlab-Simulink. Comparisons were made between the Type-1 Fuzzy Logic Controller and the T2-FLC in VHDL code and a Proportional Integral Differential (PID) Controller so as to regulate the velocity of a DC motor and evaluate the difference in performance of the three types of controllers, using the t-student test statistic.

Keywords Type-2 Fuzzy Control, Optimization, Genetic Algorithms

1. Introduction

The use of classical control techniques such as PID control in some cases provides a good alternative for controlling certain systems. However, the higher the accuracy required

in the system control, the more difficult the adjustment becomes as they are quite sensitive to noise signals and sometimes introduce variations when there are delays in the system. This kind of subjectivity has implications for modelling these systems through fuzzy logic.

There is great interest in the research and development of fuzzy systems - especially those based on type-2 fuzzy logic - due to their advantages in the management of uncertainty, which is why there are high expectations regarding their use in control systems as a possible way to compensate for errors due to instrumentation systems, among others. However, these systems require large computing resources which can be a difficulty for personal computers and so the investigation of different alternatives for their implementation is a topic of current research, such as fuzzy control implemented in programmable logic devices.

There are some works related to the optimization of fuzzy inference systems for a particular problem. In [2], a review of the methods used in the design of interval type-2 fuzzy controllers is presented. The fundamental focus of the work is based on the basic reasons for optimizing

type-2 fuzzy controllers for different areas of application. They consider the application of genetic algorithms, particle swarm optimization and ant colony optimization as three different paradigms that help in the design of optimal type-2 fuzzy controllers. In [3], they present a hybrid architecture which combines type-1 or type-2 fuzzy logic systems and genetic algorithms for the optimization of the membership function's parameters in order to solve the output regulation problem of a servomechanism with nonlinear backlash. The simulation results show the effectiveness of the optimized closed-loop system.

This paper is organized as follows: in Section 2, a general description of type-1 and type-2 fuzzy inference systems, FPGA and GAs is provided. The designs of T1-FLC, T2-FLC and PID controllers for regulating the velocity of a DC motor are given in Section 3. The testing and results of the comparisons of T1-FLC and T2-FLC compared with PID for regulating the velocity of a DC motor are provided in Section 4. Finally, Section 5 presents the conclusions drawn in this work.

2. Preliminaries

Fuzzy logic controllers tolerate imprecise information and can be used to model nonlinear functions of arbitrary complexity. They have been used successfully in many application areas, such as [14],[20]. A fuzzy inference system (T1-FIS) consists of three stages: fuzzification, inference and defuzzification [7],[23]. In Figure 1, the fuzzy system information processing is shown.

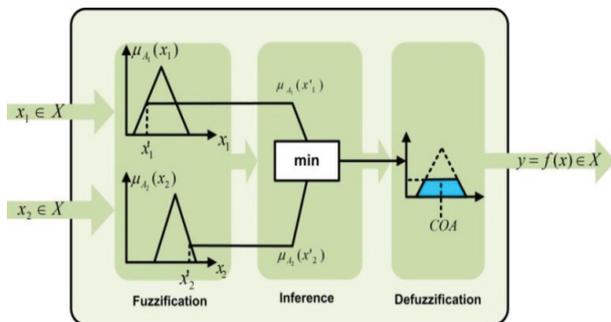


Figure 1. Type-1 Fuzzy System

The T1-FIS uses exact membership functions while type-2 fuzzy systems are described by membership functions with uncertainty. The Type-2 fuzzy logic systems are used increasingly due to their ability in relation to uncertain information [1]. Uncertainty is an attribute of information, which may be incomplete, inaccurate, vague, weak or contradictory, etc. [22].

Type-2 fuzzy systems consist of four stages: fuzzification, inference, type Reduction and defuzzification. We describe these stages below. The fuzzification maps a numeric value $x=(x_1, \dots, x_p)^T \in X_1 \times X_2 \times \dots \times X_p=X$ onto a type-2 fuzzy set A_x in X . A_x is a

singleton fuzzy set if $\mu_{A_x}=1/1$ for $x=x'$ and $\mu_{A_x}=1/0$ for all others $x \neq x'$ [10]. The inference stage consists of two blocks - the rules and the inference engine. It works in the same way as for type-1 fuzzy systems except that the antecedent fuzzy sets and the consequent are represented by type-2 fuzzy sets. The process consists of combining the rules and mapping the input to the output (type-2 fuzzy sets), using the Join and Meet operations. A type reducer is used to convert all of the type-2 fuzzy sets to type-1 fuzzy intervals on the output. There are several methods for calculating the reduced set, such as joint centre, centre of sums and height, among others [10]. The defuzzification stage consists of obtaining a numeric value for the output. Using the centre of sets type reducer, the defuzzification is an average value since the range is given by $[y_l, y_r]$ [10].

In Figure 2, the type-2 fuzzy system's information processing is shown.

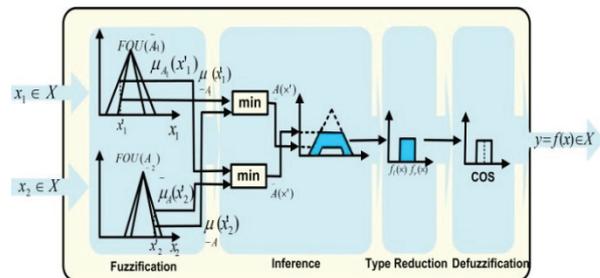


Figure 2. Type-2 Fuzzy System

A type-2 fuzzy system can be implemented on a general purpose computer or else by the specific use of a microelectronics realization, such as a FPGA.

A FPGA is a semiconductor device that contains such interior components as gates, multiplexers, etc. These are interconnected with each other according to a given design. These devices use the VHDL programming language, which is an acronym that represents the combination of VHSIC (Very High Speed Integrated Circuit) and HDL (Hardware Description Language).

FPGAs can be used to implement specific architectures in order to accelerate a particular algorithm. Applications that require a great number of simple operations are suitable for implementation on a FPGA. A processing element can be designed to perform this operation and several instances of it can be used to perform parallel processing [18],[21].

These systems are based on rules incorporating linguistic variables, linguistic terms and fuzzy rules. The acquisition of these rules is not an easy task for the expert and it is of vital importance in the operation of the controller. There are methods to optimize these elements, such as genetic algorithms (GAs) [4].

A GA is a stochastic optimization algorithm inspired by the natural theory of evolution. Following from the principle proposed by Holland [5], GAs have been used successfully to manage a wide variety of problems, such as control and search, etc. [16].

The notion of a GA is inspired by the mechanism of natural selection where stronger individuals are more likely to be the winners in a competitive environment. Here, the GA uses a direct analogy of such natural evolution. Through the genetic evolution method, an optimal solution can be found and represented by the final winner of the genetic game. Figure 3 shows the GA cycle.

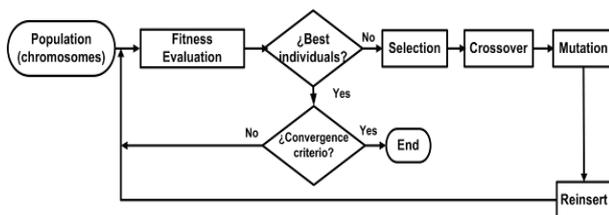


Figure 3. GA cycle

Of course, a GA does not provide the best way to solve every problem, however, GAs have proven to offer a good strategy because of their optimal results. GAs have applications in a wide variety of fields to develop solutions to complex problems = including the optimization of fuzzy systems - offering them learning and adaptation; they are commonly called ‘genetic fuzzy systems’ or ‘fuzzy system hybrids’.

This paper proposes a method for the genetic optimization of the triangular and trapezoidal membership functions (MF) of a type-2 fuzzy logic system (T2-FLC) for hardware applications such as FPGAs. This method involves taking only certain points of the membership functions in order to add greater efficiency to the algorithm. The GA has been tested in a T2-FLC and T1-FLC so as to regulate the speed of a direct current motor using the Matlab platform and the Xilinx System Generator (XSG), which is a toolbox of Matlab. Comparisons were made between the T1-FLC versus the T2-FLC in VHDL code and the T2-FLC versus a PID Controller with Ziegler-Nichols tuning in order to regulate the velocity of a DC motor and evaluate the difference in performance of the three types of controllers using the statistical t-student test.

3. Design of the T1-FLC, T2-FLC and PID controllers for the speed regulation of the dc motor

The T1-FLC is coded in VHDL and the T1-FLC for the fuzzification stage is able to instantly calculate the degree of membership. Using a method to calculate the slopes [9],[11], the inference works with the max-min composition [12],[15] and the defuzzification with the

heights method. The T1-FLC has two inputs and one output and each input and output contains three membership functions, two trapezoidal and one triangular. Figure 4 shows the triangular and trapezoidal membership functions (T1-MFs) that are used.

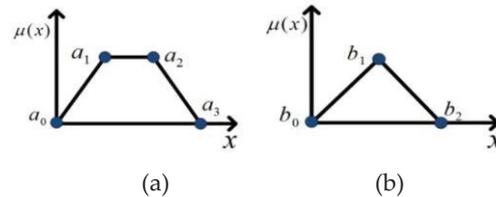


Figure 4. Parameters of the Membership Functions a) T1-MF Trapezoidal, b) T1-MF Triangular

For the optimization of the T1-FLC using GAs, one must define the chromosome that represents the information of the individual, which in this case is related to the universe of discourse and the linguistic terms. Figure 5 shows the chromosome of the GA.

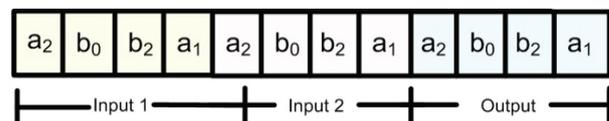


Figure 5. GA chromosome for the T1-FIS

The chromosome contains the information related to the T1-MF’s parameters for the T1-FIS design; the design proposes two trapezoidal T1-MFs to the sides and one triangular T1-MF in the centre. The T1-MFs parameters for each and the output are moved according to Table 1.

	Input 1	Input 2	Output
T1-MFs Parameters	First T2-MF: $a_0 = a_1 = 0$ $0 < a_2 < 128$ $a_3 = 128$	First T2-MF: $a_0 = a_1 = 0$ $0 < a_2 < 128$ $a_3 = 128$	First T2-MF: $a_0 = a_1 = 0$ $0 < a_2 < 128$ $a_3 = 128$
	Second T2-MF: $0 < b_0 < 128$ $b_1 = 128$ $128 < b_2 < 255$	Second T2-MF: $0 < b_0 < 128$ $b_1 = 128$ $128 < b_2 < 255$	Second T2-MF: $0 < b_0 < 128$ $b_1 = 128$ $128 < b_2 < 255$
	Third T2-MF: $a_0 = 128$ $128 < a_1 < 255$ $a_2U = a_3 = 255$	Third T2-MF: $a_0 = 128$ $128 < a_1 < 255$ $a_2U = a_3 = 255$	Third T2-MF: $a_0 = 128$ $128 < a_1 < 255$ $a_2U = a_3 = 255$

Table 1. Boundary t1-mf parameters of the chromosome

According to Figure 5, the chromosome of the T1-FIS is divided by each input and output; to each there corresponds 4 parameters (a_2 , b_0 , b_2 and a_1). The GA design only moves twelve of the thirty-three parameters to optimize; the remaining twenty-one parameters are fixed and, therefore, are not considered for the chromosome design, the main idea being that the runtime decreases for the GA.

We implemented the T2-FLC using the average method [17] in the FPGA. The T2-FLC is coded in VHDL and the

T2-FLC for the fuzzification stage is able to instantly calculate the degree of membership using a method to calculate the slopes [13]. The inference works with the max-min composition [19] and the defuzzification with the heights method.

Figure 6 shows the block diagram of the average T2-FIS for regulating the velocity of a DC motor. The system's inputs are the error (x_1) and change of error (x_2). The system has only one output (y).

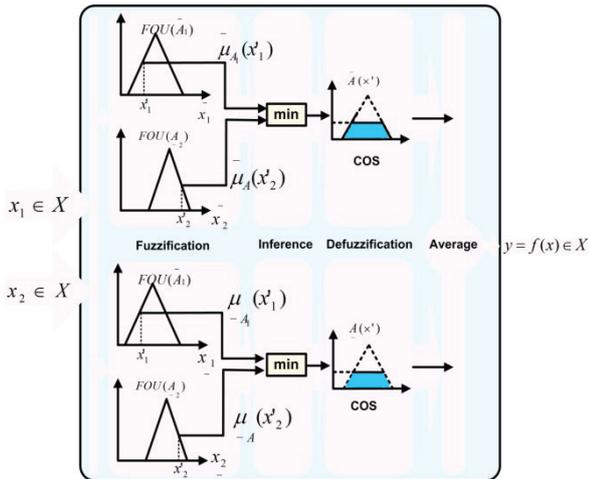


Figure 6. Block diagram of T2-FIS

The T2-FIS has two inputs and one output. Each input and each output contains three membership functions, two trapezoidal and one triangular. Figure 7 shows the type-2 triangular and trapezoidal membership functions (T2-MFs) that are used.

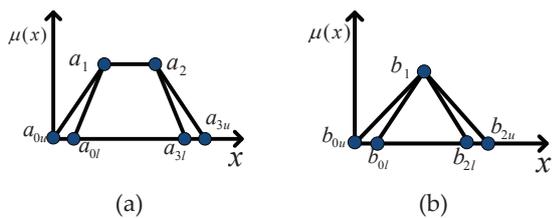


Figure 7. Parameters of the type-2 Membership Functions. a) MF Trapezoidal, b) MF Triangular

For the optimization of the T2-FLC using GAs, one must define the chromosome that represents the information of the individual, which in this case is related to the universe of discourse and the linguistic terms. Figure 8 shows the chromosome used for the GA.

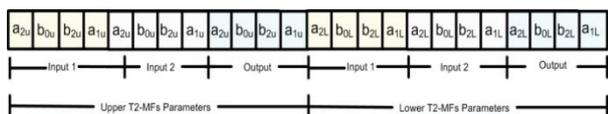


Figure 8. GA Chromosome for the T2-FIS

In Table 2, we show the boundary parameters of the chromosome.

	Input 1	Input 2	Output
Upper T2-MF Parameters	First T2-MF: $a_{0U} = a_{1U} = 0$ $0 < a_{2U} < 128$ $a_{3U} = 128$	First T2-MF: $a_{0U} = a_{1U} = 0$ $0 < a_{2U} < 128$ $a_{3U} = 128$	First T2-MF: $a_{0U} = a_{1U} = 0$ $0 < a_{2U} < 128$ $a_{3U} = 128$
	Second T2-MF: $0 < b_{0U} < 128$ $b_{1U} = 128$ $128 < b_{2U} < 255$	Second T2-MF: $0 < b_{0U} < 128$ $b_{1U} = 128$ $128 < b_{2U} < 255$	Second T2-MF: $0 < b_{0U} < 128$ $b_{1U} = 128$ $128 < b_{2U} < 255$
	Third T2-MF: $a_{0U} = 128$ $128 < a_{1U} < 255$ $a_{2U} = a_{3U} = 255$	Third T2-MF: $a_{0U} = 128$ $128 < a_{1U} < 255$ $a_{2U} = a_{3U} = 0$	Third T2-MF: $a_{0U} = 128$ $128 < a_{1U} < 255$ $a_{2U} = a_{3U} = 0$
Lower T2-MF Parameters	First T2-MF: $b_{0L} = a_{1L} = 0$ $a_{2L} < a_{2L} < 128$ $a_{3L} = 128$	First T2-MF: $a_{0L} = a_{1L} = 0$ $a_{2L} < a_{2L} < 128$ $a_{3L} = 128$	First T2-MF: $a_{0L} = a_{1L} = 0$ $a_{2L} < a_{2L} < 128$ $a_{3L} = 128$
	Second T2-MF: $b_{0L} < b_{0L} < 128$ $b_{1L} = 128$ $b_{2L} > b_{2L} < 255$	Second T2-MF: $b_{0L} < b_{0L} < 128$ $b_{1L} = 128$ $b_{2L} > b_{2L} < 255$	Second T2-MF: $b_{0L} < b_{0L} < 128$ $b_{1L} = 128$ $b_{2L} > b_{2L} < 255$
	Third T2-MF: $a_{0L} = 128$ $a_{1L} < a_{1L} < 255$ $a_{2L} = a_{3L} = 255$	Third T2-MF: $a_{0L} = 128$ $a_{1L} < a_{1L} < 255$ $a_{2L} = a_{3L} = 255$	Third T2-MF: $a_{0L} = 128$ $a_{1L} < a_{1L} < 255$ $a_{2L} = a_{3L} = 255$

Table 2. Boundary t2-mf parameters of the chromosome

For example, the a_{2U} parameter should be within the range of 0-128. If we consider that is 100, then the a_{2L} parameter must be greater than 100 and less than 128; these conditions are necessary to achieve the upper and lower of the T2-MF.

The GA design only moves twenty-four of the sixty-six parameters to optimize, of which eight are parameters for each input and output ($a_{2U}, a_{2L}, b_{0U}, b_{0L}, b_{2U}, b_{2L}, a_{1U}$ and a_{1L}) while the remaining forty-two parameters are fixed and, therefore, not considered for the chromosome's design.

Figure 9 shows the input of the T2-FIS with fixed and variable parameters. Each input and output has a size of 8 bits.

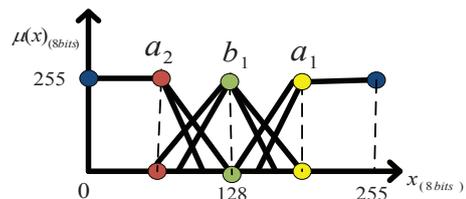


Figure 9. Points of Type-2 Membership Functions' Inputs and Output

The GA is a multi-objective GA [8] for the T1-FLC and T2-FLC, which means that in order to determine the best individual three evaluations are performed:

- a) Minimum overshoot

$$o_1 = \max(y(t)) - r \quad (1)$$

- b) Minimum undershoot

$$o_2 = |\min(y(t)) - r| \quad (2)$$

- c) Minimum output steady state error (sse)

$$sse = \sum_{t=201}^{1000} y(t) - r \quad (3)$$

where $y(t)$ is the output of the system and r is reference.

The T1-FLC and T2-FLC linguistic terms were optimized with the GA but the fuzzy rules are not changed. The process of the GA is described as follows: generate the initial population, fitness evaluation (o_1 , o_2 , sse), selection, crossover, mutation, reinsert and, finally, simulation using the XSG platform in Matlab-Simulink.

For the PID controller's design, we used the Ziegler-Nichols tuning method [24]. This method has the following steps:

1. Using only proportional control starting with a small gain value, increase the gain until the loop begins to oscillate.
2. Record the controller critical gain $K_p=K_c$ and oscillation period of the controller output P_c .
3. Adjust the controller parameters according to Table 3.

	K_p	K_i	K_d
P	$0.5K_c$		
PI	$0.45K_c$	$0.833P_c$	
PID	$0.6K_c$	$0.5P_c$	$0.125P_c$

Table 3. Controller parameters with ziegler-nichols

4. Tests and results for the comparisons with T2-FLC and T1-FLC in FPGA and PID controllers

The T1-FLC, T2-FLC and PID controllers were analysed; each was given a level of uncertainty and a comparison was made among them. The results were evaluated using t-student statistics.

To test the T2-FLC and T1-FLC, the speed control was simulated using a mathematical model of the plant in Matlab-Simulink. Figure 10 shows the T1-FLC and T2-FLC for regulating the velocity of a DC motor, the fuzzy incremental control model is used [6].

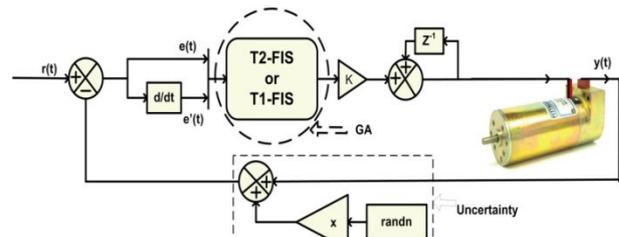


Figure 10. Model of T2-FLC and T1-FLC

The T1-FLC and T2-FLC have the inputs, error ($e(t)$) and change of error ($e'(t)$), and the output is the control signal ($y(t)$), the control objective AT2-FLC is:

$$\lim_{t \rightarrow \infty} |y(t) - r(t)| = 0 \quad (4)$$

where t is the sampling time.

The inputs are calculated as follows:

$$e(t) = r(t) - y(t) \quad (5)$$

$$e'(t) = e(t) - e(t-1) \quad (6)$$

The reference signal $r(t)$, is given by:

$$r(t) = \begin{cases} 15 & t > 0 \\ 0 & t \leq 0 \end{cases} \quad (7)$$

The uncertainty block represents an external perturbation, the goal is to disrupt the AT2-FLC and then the AT2-FLC retrieves its desired path. The uncertainty is represented by:

$$\hat{y}(t) = y(t) + x * randn \quad (8)$$

where x is the uncertainty level factor [0,1].

Each input and output of the T2-FIS and T1-FIS has three linguistic terms. For the linguistic variable error and change of error, the terms are {NB, Z, PB}, in this case NB is Negative Big, Z is Zero and PB is Positive Big. For the linguistic variable control signal, the terms are {BD, H, BI}, in this case BD is Big Decrement, H is Hold and BI is Big Increment.

The PID controller model is shown in Figure 11, the uncertainty block is used to test the PID controllers with different levels of noise.

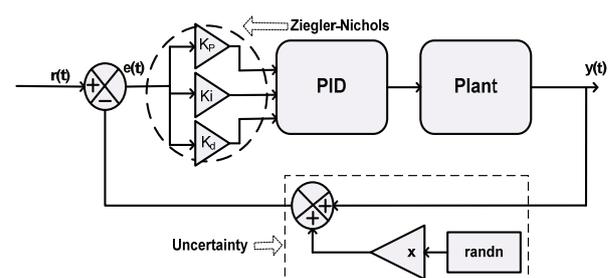


Figure 11. Model of PID

The input is calculated as (5), where the reference signal is (7).

In Table 4, we show the comparison of the T1-FLC, T2-FLC and PID for different levels of uncertainty. We note that the T2-FLC is better at different levels of uncertainty (noise), while the noise free T1-FLC exhibits similar behaviour to the T2-FLC and in this case PID is better.

No.	Controller	Uncertainty level factor (x)	Average Error
1	T2-FLC	0	0.6298
	T1-FLC	0	0.8249
	PID	0	0.6197
2	T2-FLC	0.001	0.6343
	T1-FLC	0.001	0.8283
	PID	0.001	0.8394
3	T2-FLC	0.005	0.6403
	T1-FLC	0.005	0.8279
	PID	0.005	0.8528
4	T2-FLC	0.008	0.6505
	T1-FLC	0.008	0.8215
	PID	0.008	0.8528
5	T2-FLC	0.05	0.6244
	T1-FLC	0.05	0.7975
	PID	0.05	0.8528
6	T2-FLC	0.08	0.6244
	T1-FLC	0.08	0.7946
	PID	0.08	0.8660
7	T2-FLC	0.1	0.6116
	T1-FLC	0.1	0.7532
	PID	0.1	0.8792
8	T2-FLC	0.5	0.6823
	T1-FLC	0.5	0.8301
	PID	0.5	0.8791
9	T2-FLC	0.8	0.7173
	T1-FLC	0.8	0.8932
	PID	0.8	0.8905
10	T2-FLC	1	0.7402
	T1-FLC	1	0.9387
	PID	1	0.8972

Table 4. T1-FLC, T2-FLC versus PID results for regulate the velocity of a dc motor

We analyse statistically the performance of the three controllers using the t-student test. Table 5 shows the statistical results of the three controllers.

Controller comparison	t-student
T1-FLC Vs T2-FLC	8.22
T2-FLC Vs PID	6.93

Table 5. T1-FLC, T2-FLC versus PID results for regulate the velocity of a dc motor

As shown in Table 5, the T2-FLC has better performance compared with the T1-FLC and PID, with a degree of confidence of more than 95 percent.

Figure 12 shows the velocities obtained with the three controllers with no level of uncertainty ($x=0$).

Figure 13 shows the velocities obtained with the three controllers with a particular level of uncertainty ($x=1$).

As shown in Figure 13, it is very difficult to determine which controller has better performance, for that reason we decided to test t-student statistics, as shown in Table 5, which illustrate that the T2-FLC is better compared to T1-FLC and PID controllers.

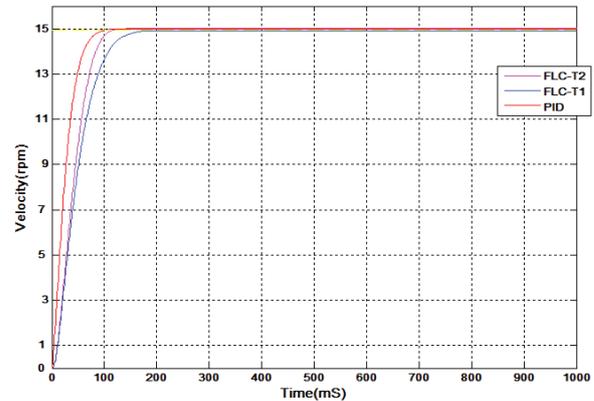


Figure 12. Behaviour of T2-FLC comparison with T1-FLC and PID controllers for velocity motor with $x=0$

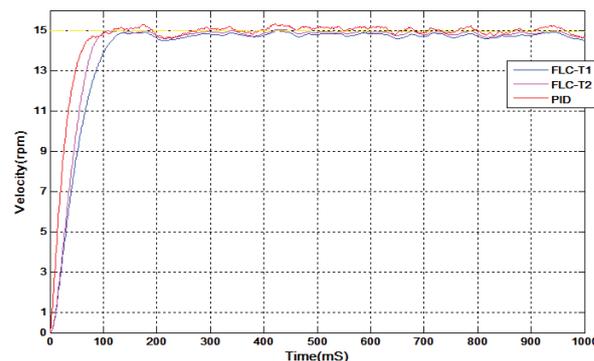


Figure 13. Behaviour of T2-FLC comparison with T1-FLC and PID controllers for velocity motor with uncertainty level ($x=1$)

5. Conclusions

We propose a tuning PID controller and the genetic optimization of T1-FLC and T2-FLC for regulating the velocity of a DC motor, where three triangular and trapezoidal membership functions for the two inputs and one output are used in the optimization. We propose the GA optimization for the T1-FLC and T2-FLC, the goal being to achieve an optimized T1-FLC and T2-FLC with a small runtime, and for this reason the fuzzy rules are not changed and we propose an optimization for T1-MFs and T2-MFs, where only some of these parameters are modified. The objective function of GA considers three characteristics: overshoot, undershoot and steady state error, so that makes it a multi-objective GA.

The T1-FLC and T2-FLC are codified in VHDL code for FPGA implementation. For the T1-FLC and T2-FLC

implementation, the fuzzification, inference and defuzzification stages are performed in parallel.

Each T1-FIS is simulated in a T1-FLC for regulating the velocity of the DC motor. The best T1-FLC was obtained in 70 generations with 85% crossover (crossover single point) and 10% mutation, with an error of convergence of 7.34×10^{-4} , in a runtime of 19.35 minutes with a 15 rpm of velocity.

The best T2-FLC was obtained in 30 generations with 85% crossover (crossover single point) and 10% mutation, with an error of convergence of 9.55×10^{-4} , in a runtime of 30.0088 minutes with a 15rpm of velocity.

The PID controller tuning was obtained with Ziegler-Nichols method, and the values of the constants obtained are $k_p=0.5$, $k_i=0.2$ and $k_d=0.025$.

Comparisons were made between the T1-FLC versus T2-FLC in VHDL code and T2-FLC versus PID controller, to regulate the velocity of a DC motor, to evaluate the difference in performance of three types of controllers, using the t-student statistical test, giving better results for the T2-FLC when there is uncertainty present.

Matlab-Simulink and XSG were used to perform the simulations in all cases.

6. Acknowledgments

We would like to express our gratitude to CONACYT and the Tijuana Institute of Technology for the facilities and resources granted for the development of the research reported in this paper.

7. References

- [1] M. Biglarbegan, W.W. Melek, J.M. Mendel, Design of Novel Interval Type-2 Fuzzy Controllers for Modular and Reconfigurable Robots: Theory and Experiments, *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 4, pp. 1371-1384, 2011.
- [2] O. Castillo, P. Melin: A Review on the Design and Optimization of Interval Type-2 Fuzzy Controllers, *Applied Soft Computing*, Vol. 12, No. 4, pp. 1267-1278, 2012.
- [3] N. R. Cazarez-Castro, L. T. Aguilar, O. Castillo, Fuzzy logic Control with Genetic Membership Function Parameters Optimization for the Output Regulation of a Servomechanism with Nonlinear Backlash, *Expert Systems with Applications*, Vol. 37, No.6, pp. 4368-4378, 2010.
- [4] D.E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Boston, MA: Addison-Wesley, 1989.
- [5] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA:MIT Press, 1992.
- [6] J. Jantzen, *Tuning of fuzzy PID Controllers*, Denmark.Tech. Report no 98-H 871 (fpid), Denmark University, pp. 1-22, 1998.
- [7] G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, USA 1995.
- [8] K.F. Man, K.S. Tang and S. Kwong, *Genetic algorithms*, Springer, Hong Kong, 2000.
- [9] Y. Maldonado, O. Montiel, R. Sepulveda and O. Castillo, Design and simulation of the fuzzification Stage through the Xilinx System the fuzzification Stage through the Xilinx System Generator, *Soft Computing for Hybrid Intelligent Systems, Studies in Computational Intelligence*, Springer, Heidelberg, Germany, pp. 297-305, 2008.
- [10] J.M. Mendel, R.I.B. John, Type-2 Fuzzy Sets Made Simple, *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 117-127, 2002.
- [11] O. Montiel, Y. Maldonado, R. Sepulveda and O. Castillo, Simple Tuned Fuzzy controller embedded into an FPGA, *IEEE NAFIPs Conference Proceedings*, pp. 1-6, 2008.
- [12] O. Montiel, J.A. Olivas, R. Sepulveda, O. Castillo, Development of an Embedded Simple Tuned Fuzzy Controller, *IEEE World Congress on computational Intelligence WCCI-2008*, pp. 555-561,2008.
- [13] O. Montiel, R. Sepulveda, Y. Maldonado and O. Castillo, Design and simulation of The Type-2 Fuzzification Stage: Using Active Membership Functions, *Evolutionary Design of Intelligent systems in Modeling, Simulation and Control*, Springer, pp. 273-293, 2009.
- [14] O. Montiel, R. Sepulveda, P. Melin, O. Castillo, M.A. Porta, I.M. Meza, Performance of a Simple Tuned Fuzzy Controller and PID Controller on a DC Motor, *FOCI 2007* , pp.531-537, 2007.
- [15] J.A. Olivas, R. Sepulveda, O. Montiel and O. Castillo, Methodology to Test and Validate a VHDL Inference Engine through the Xilinx System Generator, *Soft Computing for Hybrid Intelligent Systems, Studies in Computational Intelligence*, Springer, Heidelberg, Germany, pp. 325-331, 2008.
- [16] L.H. Randy, E.H. Sue, *Practical Genetic Algorithms*, Wiley, New York, USA, 2004.
- [17] R. Sepulveda, O. Castillo, P. Melin, O. Montiel, an Efficient computational Method to Implement Type-2 fuzzy Logic in Control Applications Analysis and Design of Intelligent Systems Using Soft Computing Techniques, *Advances in Soft Computing*, Springer-Verlag, Vol. 41, pp. 45-52, 2007.
- [18] R. Sepulveda, O. Montiel, O. Castillo, P. Melin, Embedding a high speed interval type-2 fuzzy controller for a real plant into an FPGA, *Applied Soft Computing*, Vol. 12, No. 3, pp. 988-998, 2012.

- [19] R. Sepulveda, O. Montiel, J.A. Olivas and O. Castillo, Methodology to Test and Validate a VHDL Inference Engine of a Type-2 FIS, through the Xilinx System Generator, *Evolutionary Design of Intelligent systems in Modeling, Simulation and Control*, Springer, pp. 295-308, 2009.
- [20] L. Tsoukalas, R.E. Ohring, *Fuzzy and Neural Approaches in Engineering*, Wiley-Interscience, New York, USA, 1997.
- [21] F.J.M. Velo, L. Baturone, S. Solano, A. Barriga, Rapid Design of Fuzzy Systems with XFUZZY, *IEEE International Conference on Fuzzy Systems FUZZ-IEEE-2003*, pp. 342-347, 2003.
- [22] L.A. Zadeh, *Fuzzy Sets, Information and Control* 8, pp. 338-353, 1965.
- [23] K. Zdenko, B. Stjepan, *Fuzzy Controller Design Theory and Applications*, Taylor and Francis, Boca Raton, FL, USA, 2006.
- [24] J. G. Ziegler, N. B. Nichols, Optimum Settings for Automatic Controllers, *Transactions of the ASME*, Vol. 64, pp. 759-768, 1942.