



FPGA BASED P/PI/PD/PID-LIKE INTERVAL TYPE -2 FLC DESIGN FOR CONTROL SYSTEMS

Mohammed Y. Hassan¹ and Saif F. Abulhail²

¹ Assistant professor, Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq. Email: myhazawy@yahoo.com

² MSc student, Department of Control and Systems Engineering Department, University of Technology, Baghdad, Iraq. Email: saifabulhail80@gmail.com

<http://dx.doi.org/10.30572/2018/kje/090312>

ABSTRACT

Type-2iFuzzyilogicicontrol contains footprint of uncertainty (FOUi) that is able to handle the numerical uncertainties, nonlinearities, and linguistic associated with the inputs and outputs.

This paper presents the structure of P/PI/PD/PID-like Interval Type-2 Fuzzy Logic Controller (IT2-FLC) that is designed and simulated using MATLAB SIMULINK. This controller is designed by combining the advantages of fuzzy inference and different structures of controller. The type of the controller is selected using two bits control inputs. The inputs to the controller are: input-output gains, set-point, and the actual output. Karnick-Mendel (KM) algorithm is used to implement the IT2-FLC. IT2FLC has two triangular shaped membership functions. The range of the universe of discourse (-1.5 to 1.5) is selected for each input and output. The defuzzification technique is selected as Centroid method. The sampling time is (0.01), and the choice of rules is by using trial and error. The programs are written in MATLAB SIMULINK and converted into VHDL and Verilog languages. The program generated must achieve small FPGA size and as a result less execution time must achieve vs. acceptable accuracy.

Linear and nonlinear models were used to test the Controller. Trial and error method was used to tune the gain factors to obtain the best response with minimum overshoot, minimum settling time, minimum Root Mean Squared Error (RMSE), and minimum structure design.

KEYWORDS: P/PI/PD/PID-like IT2FLC; Footprint of uncertainty; Karnik-Mendel; FPGA.

1. INTRODUCTION

Interval Type-2 fuzzy logic controller has a significant interest because it is deal with the uncertainties and non-linear ties in three dimensions, it has the ability for handling the problem of uncertainty parameters (Biglarbegian et al., 2010). Hassan and Sharif, (2007), presented the PID-like fuzzy logic controller (PIDFLC). On FPGA chip. This design is aimed to combine the type-1 fuzzy logic algorithm with the aid of conventional PID control. Maldonado et al., (2011) have used Genetic algorithm in order to reach the optimization in the design of membership functions (triangular and trapezoidal) of a fuzzy logic that was implemented in FPGA chip. Mani and Barjeev, (2011) used Fuzzy Logic Controller (FLC) and have developed as one of the most promising areas for Industrial Applications. The simulation results of the Fuzzy Logic Controller that is implemented on FPGA chip have been compared with the simulation results of the PID Controller and Fuzzy Logic Controller using MATLAB SIMULINK. Maldonado et al., (2013) have used Particle Swarm Optimization (PSO) method and they have applied for react to optimum design of IT-2 FLC and they have used VHDL code for a Xilinx Spartan FPGA. The results were compared with the IT2 FLC designed using genetic algorithms as well. Maldonado et al., (2014) have presented the IT2 FLC to control and regulate the speed of a DC motor. They were used an FPGA chip which provides implementation in real time. Mehmet. et al., (2014) presented a Boundary Function (BF) based type reduction and defuzzification method for IntervalType-2 Fuzzy logic with PID controller (IT-2 PID). Jun. et al., (2015) presented a new controller to control the feed water under the automatic power regulating system for an advanced boiling water reactor. This controller was designed and implemented on FPGA device.

In this work, the design of P/PI/PD/PID-like Interval Type-2 Fuzzy Logic Controller (IT2-FLC) is explained. The general layout of the controller chip is shown in Fig. 1. The type of the controller is selected using two bits control inputs. This P/PI/PD/PID-like IT2FLC design in this paper is combining the advantages of type-2 fuzzy algorithm and different structures of controller. For controlling efficiently to the real systems and to achieved small FPGA size and as a result less execution time must be achieved vs. acceptable accuracy. Simulations of the linear and nonlinear models are explained.

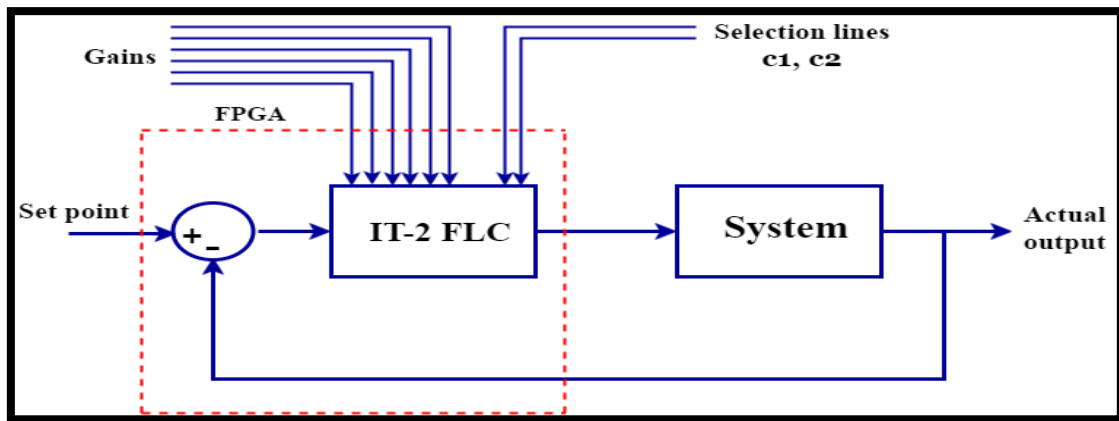


Fig. 1. Layout of the proposed controller in a unity feedback control system.

2. INTERVAL TYPE-2 FLC

Type-2iFuzzyisetsimembership function are fuzzy and contain the footprint of uncertainty FOU as shown in Fig. 2.that is handle and model the uncertainties. nonlinearities and linguistic related with the inputs and output of the fuzzy logic control by modeling them and reducing their effect. FOU is the area between lower membership and upper membership. Allowing each input to have two membership grade values related with it; an upper ($UMF(x_i)$) and lower ($LMF(x_i)$) There are two types of T2FLC: Mamdani type and the Takagi Sugeno Kang (TSK) type. The difference between the two types. The Sugeno output membership function is linear or constant and the Mamdani output is shape the membership.

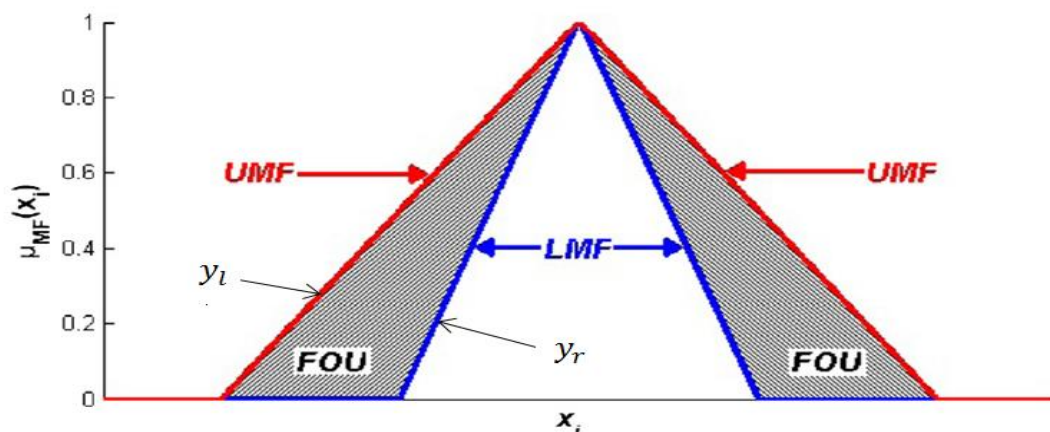


Fig. 2. IT2 membership function structure adapted from (Karnik et al., 1999).

Furthermore the Mamdani type needs type reducer method while the Takagi Sugeno type doesn't need type reduction operation. The structure of the Interval Type-2 Fuzzy Logic Control (IT-2FLC) is shown in Fig. 3; (Mendel et al., 2006; and Karnik et al., 1999).

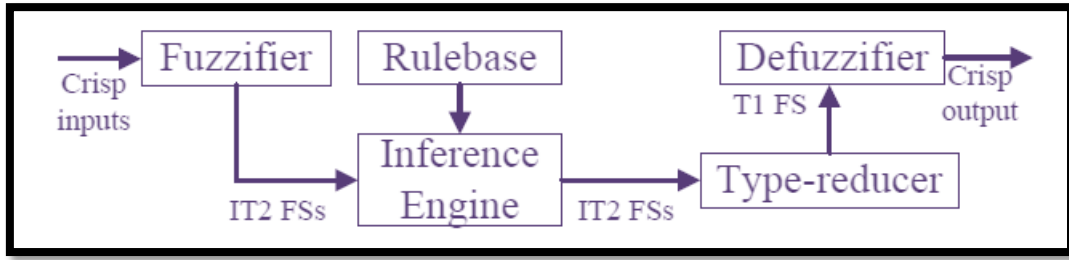


Fig. 3. The schematic diagram of an IT2 FLC.

3. COMPUTATIONS OF IT2-FLC

An interval type two fuzzy logic control containing at least one interval fuzzy set. Without loss of generality, consider if an IT2 FLC consisting of N rules. The rule base has the following form:

$\tilde{R}^n : y = Y^n$ If x_1 is $\tilde{X}_1^n$ and ... and x_I is $\tilde{X}_I^n$, where $\tilde{X}_i^n$ ($i = 1, \dots, I$) are IT2 fuzzy sets, and $Y^n = [\underline{y}^n, \bar{y}^n]$ is an interval that can be understood as the centroid of a consequent IT2 fuzzy set, or the simplest Takagi-Sugeno-Kang (TSK) model. Each rule consequent is represented by a crisp number in many applications (Wu and Tan, 2006).

For an input vector $X' = (x'_1, x'_2, \dots, x'_I)$, typical computations in an interval type two fuzzy logic control include the following steps (Wu and Tan, 2006):

1) Compute the membership interval of x'_i on each $\tilde{X}_i^n$, $[\mu_{\underline{\tilde{X}_i^n}}(x'_i), \mu_{\overline{\tilde{X}_i^n}}(x'_i)]$, $i = 1, 2, \dots, I$, $n = 1, 2, \dots, N$.

2) Calculate the firing interval of the n^{th} rule, F^n :

$$F^n = [\mu_{\underline{\tilde{X}_1^n}}(x'_1) \times \dots \times \mu_{\underline{\tilde{X}_I^n}}(x'_I), \mu_{\overline{\tilde{X}_1^n}}(x'_1) \times \dots \times \mu_{\overline{\tilde{X}_I^n}}(x'_I)] \equiv [\underline{f}^n, \bar{f}^n], n = 1, 2, \dots, N.$$

3) The third step is that Perform type-reduction. The most commonly used one is the center of sets type reducer:

$$Y_{cos} = \frac{\sum_{n=1}^N Y^n}{\sum_{n=1}^N F^n} = [y_l, y_r] \quad 1$$

$$y_l = \min_{k \in [1, N-1]} \frac{\sum_{n=1}^k \underline{y}^n \bar{f}^n + \sum_{n=k+1}^N \underline{y}^n \underline{f}^n}{\sum_{n=1}^k \bar{f}^n + \sum_{n=k+1}^N \underline{f}^n} \quad 2$$

$$y_r = \max_{k \in [1, N-1]} \frac{\sum_{n=1}^k \bar{y}^n \underline{f}^n + \sum_{n=k+1}^N \bar{y}^n \bar{f}^n}{\sum_{n=1}^k \underline{f}^n + \sum_{n=k+1}^N \bar{f}^n} \quad 3$$

In equation (2) and equation (3) K is a potential switchpoint. In an complete search method all k in $[1, N - 1]$ need to be calculated until the correct switchpoint is recognized. Fortunately, y_l and y_r , as shown in Fig. 2 can also be calculated more efficiently by the KM algorithm introduced in the next subsection.

4) Compute the defuzzified output as:

$$y = \frac{y_l + y_r}{2} \quad 4$$

3.1. Karnik-mendel algorithm (KM)

The Karnik-Mendel algorithm (KM) for computing y_l in (2) and y_r in (3) is consist of the following steps:

$$f_l(k) = \frac{\sum_{n=1}^k \underline{y}^n \bar{f}^n + \sum_{n=k+1}^N \underline{y}^n \underline{f}^n}{\sum_{n=1}^k \bar{f}^n + \sum_{n=k+1}^N \underline{f}^n} \quad 5$$

$$f_r(k) = \frac{\sum_{n=1}^k \bar{y}^n \underline{f}^n + \sum_{n=k+1}^N \bar{y}^n \bar{f}^n}{\sum_{n=1}^k \underline{f}^n + \sum_{n=k+1}^N \bar{f}^n} \quad 6$$

Where: k is an integer in $[1, N - 1]$ and $(\underline{y}^n)$ and $(\bar{y}^n)$ have been sorted in ascending order. respectively. Furthermore, y_l in (2) and y_r in (3) can be re expressed as (Wu and Tan, 2006):

$$y_l = \frac{\sum_{n=1}^L \underline{y}^n \bar{f}^n + \sum_{n=L+1}^N \underline{y}^n \underline{f}^n}{\sum_{n=1}^L \bar{f}^n + \sum_{n=L+1}^N \underline{f}^n} \quad 7$$

$$y_r = \frac{\sum_{n=1}^R \bar{y}^n \underline{f}^n + \sum_{n=R+1}^N \bar{y}^n \bar{f}^n}{\sum_{n=1}^R \underline{f}^n + \sum_{n=R+1}^N \bar{f}^n} \quad 8$$

where L and R are switch points satisfying

$$\underline{y}^L \leq y_l < \underline{y}^{L+1} \quad 9$$

$$\bar{y}^R < y_r \leq \bar{y}^{R+1} \quad 10$$

The KM Algorithm for Computing y_l and y_r are specified in Table 1. The main idea is to find the switch points for y_l and y_r . $\underline{y}^n$ increases along the horizontal axis from the left to the right. for computing y_l switch from the upper firing level to the lower firing level. and switch from the lower firing level to the upper firing level for Computing y_r as shown Fig 4.

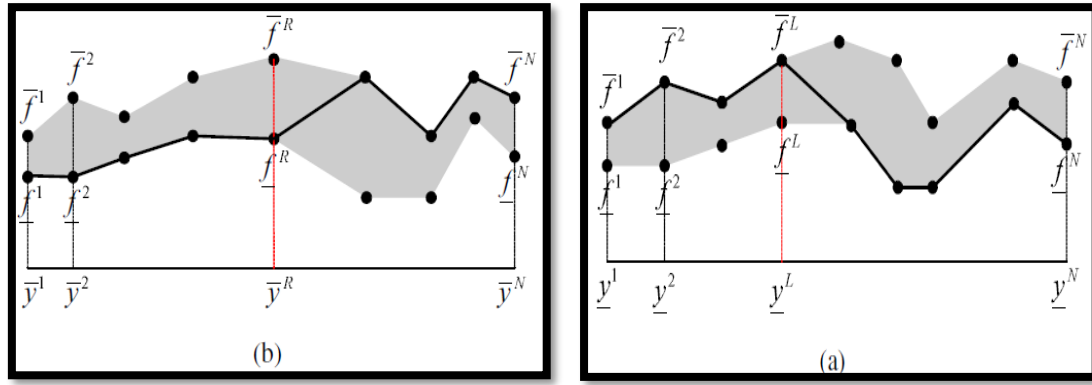


Fig. 4. Illustration of the switch points in computing y_l and y_r . (a) Computing y_l : switch from the upper bounds to the lower bounds; (b) Computing y_r : switch from the lower bounds to the upper bounds.

Table 1. The KM Algorithm. Note that $\{y^n\}_{n=1, \dots, N}$ and $\{\bar{y}^n\}_{n=1, \dots, N}$ have been sorted in ascending order, respectively.

Step	For computing y_l	For computing y_r
1.	Initialize $f^n = \frac{f^n + \bar{f}^n}{2}$ and compute $y = \frac{\sum_{n=1}^N y^n f^n}{\sum_{n=1}^N f^n}$	Initialize $f^n = \frac{f^n + \bar{f}^n}{2}$ and compute $y = \frac{\sum_{n=1}^N \bar{y}^n f^n}{\sum_{n=1}^N f^n}$
2.	Find $l \in [1, N-1]$ $\underline{y}^l \leq y_l < \underline{y}^{l+1}$	Find $r \in [1, N-1]$ $\bar{y}^r < y_r \leq \bar{y}^{r+1}$
3.	Set $f^n = \begin{cases} \bar{f}^n, & n \leq l \\ f^n, & n > l \end{cases}$ and compute $y' = \frac{\sum_{n=1}^N y^n f^n}{\sum_{n=1}^N f^n}$	Set $f^n = \begin{cases} f^n, & n \leq l \\ \bar{f}^n, & n > l \end{cases}$ and compute $y' = \frac{\sum_{n=1}^N \bar{y}^n f^n}{\sum_{n=1}^N f^n}$
4.	If $y' = y$, stop and set $y_l = y$ and $L = l$; Otherwise, set $y = y'$ and go to step 2.	If $y' = y$, stop and set $y_r = y$ and $R = r$; Otherwise, set $y = y'$ and go to step 2.

4. STRUCTURE OF THE PROPOSED CONTROLLER

It is a PID-like IT2FLC. This controller is designed in discrete form with minimum structure form of the conventional PID controller. In order to design a PID-like IT2FLC, it is required to design a fuzzy inference system with three inputs that represent the proportional, derivative, and integral components. A fuzzy controller with three inputs may not be preferred because it is very difficult to design and needs too much work. For example, If eight fuzzy sets are used for each input. Then a $(8 \times 8 \times 8 = 512)$ rules will be required for the controller. Instead, PID-

likeiIT2FLC can be designed as a parallel structure of a PD-like IT2FLC and a PI-like IT2FLC and the output of the PID-like IT2FLC is formed by algebraically adding the outputs of the two fuzzy control blocks. However, a PD-like IT2FLC may be employed to serve as PI-like IT2FLC in incremental form. Equation (11) shows a PD-like IT2FLC obtained in position form, while (12) shows a PI-like IT2FLC in incremental form (Biglarbegian et al., 2010; and Hassan and Sharif, 2007):

$$u_{(k)} = K_p e_{(k)} + K_d \dot{e}_{(k)} \quad 11$$

$$\Delta u_{(k)} = K_p \dot{e}_{(k)} + K_i e_{(k)} \quad 12$$

This method will reduce the number of rules required to $8*8+8*8=128$ rules only, as shown in Fig. 5.

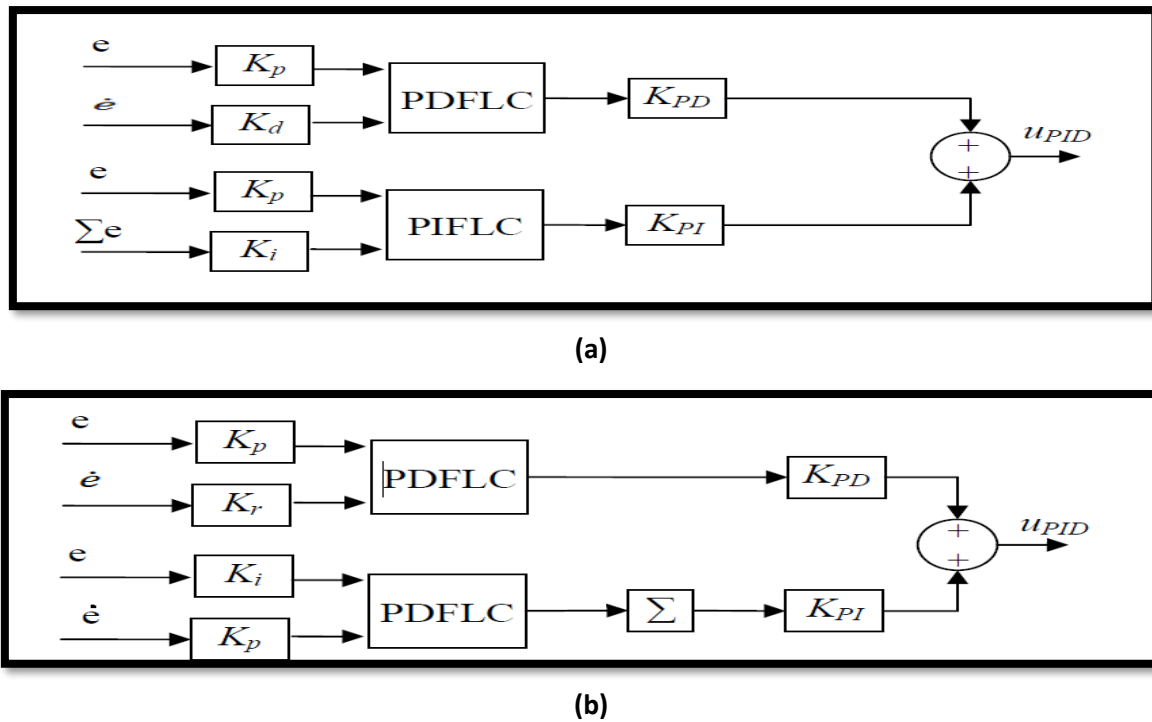


Fig. 5. (a) Three input PID-like IT2FLC (b) Two input PID-like IT2FLC.

Where: $e_{(k)}$ is a sampled error signal. $\dot{e}_{(k)}$ is the rate of change of sampled error signal. and index (k) represents the present sampling instant. Now by comparing (11) and (12). one sees that the PD Controller in position form becomes the PI controller in incremental form if $e_{(k)}$ and $\dot{e}_{(k)}$ exchange positions. kd is replaced by ki and $u(k)$ is replaced by $\Delta u_{(k)}$ the outputs of the PD-like IT2FLC and PI-like IT2FLC are summed together to form the PID-like IT2FLC output. Since each PD-like IT2FLC has its own gains and rules. The final design could act as a P-like IT2FLC, a PD-like IT2FLC, a PI-like IT2FLC, and a PID-like IT2FLC depending on the two input selection lines c1 and c2. As shown in Table 2 (Wu and Tan, 2006).

4.1. Membership functions

The parameters of the P/PI/PD/PID-like IT2FLC use two triangular shaped membership functions as shown in Fig. 6. The universe of discourse $(-1, 1)$ are selected for each inputs and output. The defuzzification technique is selected to be Centroid method.

Table 2. Selection lines setting.

Control signals		Controller type
C1	C2	
0	0	P-Like IT2FLC
0	1	PD-Like IT2FLC
1	0	PI-Like IT2FLC
1	1	PID-LikeIT2FLC

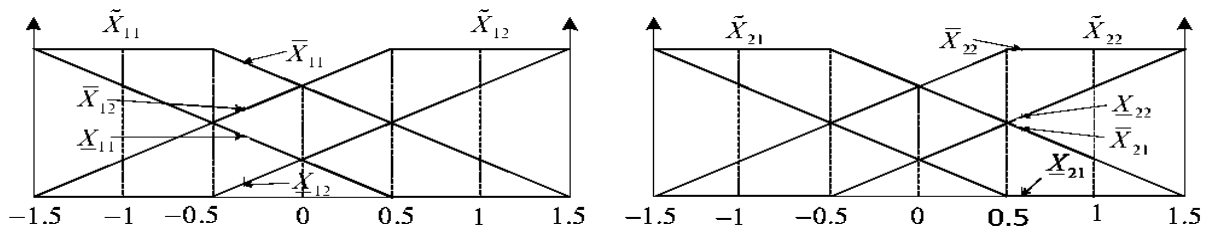


Fig. 6. (a) Input MFs of e . (b) Input MFs of $\dot{e}$.

4.2. Rule base

The rule base and rule consequent for computing the output is listed in Table 3. The choice of rules is by using trial and error.

First rule : Y is Y^1 when e is X_{11} and $\dot{e}$ is X_{21} .

Second rule: Y is Y^2 when e is X_{11} and $\dot{e}$ is X_{22}

Third rule : Y is Y^3 when e is X_{12} and $\dot{e}$ is X_{21}

Fourth rule: Y is Y^4 when e is X_{12} and $\dot{e}$ is X_{22}

Table 3. Rule base and consequent of the P/PI/PD/PID-like IT2FLC.

X2 X1	$\widetilde{X}_{12}$	$\widetilde{X}_{12}$
	$\widetilde{X}_{11}$	$\widetilde{X}_{12}$
$\widetilde{X}_{11}$	$Y^1 = [\underline{y}^1 \cdot \overline{y}^1] = [-1, -.9]$	$Y^2 = [\underline{y}^2 \cdot \overline{y}^2] = [-.6, -.4]$
$\widetilde{X}_{12}$	$Y^3 = [\underline{y}^3 \cdot \overline{y}^3] = [.4, .6]$	$Y^4 = [\underline{y}^4 \cdot \overline{y}^4] = [.9, 1]$

4.3. Scaling factors

In this paper the P/PI/PD/PID -like IT2FLC scaling factors are tuned manually and a set of gains are obtained. The SIMULINK Top level view of the P/PI/PD/PID-like IT2FL structure simulated in MATLAB SIMULINK environment, as shown in Fig. 7.

5. SIMULATIONS RESULTS OF THE PROPOSED CONTROLLER

The SIMULINK structure of P/PI/PD/PID-like IT2FLC is shown in Fig. 8. the type of the controller is selected using two bits control input (c1 and c2). In order to reach minimum overshoot, minimum undershoot, minimum settling time, minimum steady state error and minimum structure design trial and error method was used to tune the gain factors to obtain the best response. The programs are written in MATLAB and converted to VHDL and Verilog formats.

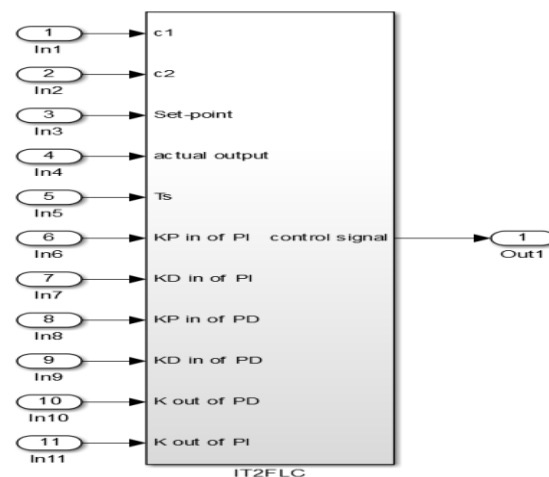


Fig. 7. SIMULINK Top level view of the P/PI/PD/PID-like IT2FLC.

This proposed controller requires FPGA size with (67) multipliers. (894) adders / subtractors. (2) registers. and (678) multiplexers, as shown in Fig. 9. In order to examine this proposed controller. Linear and nonlinear models are used.

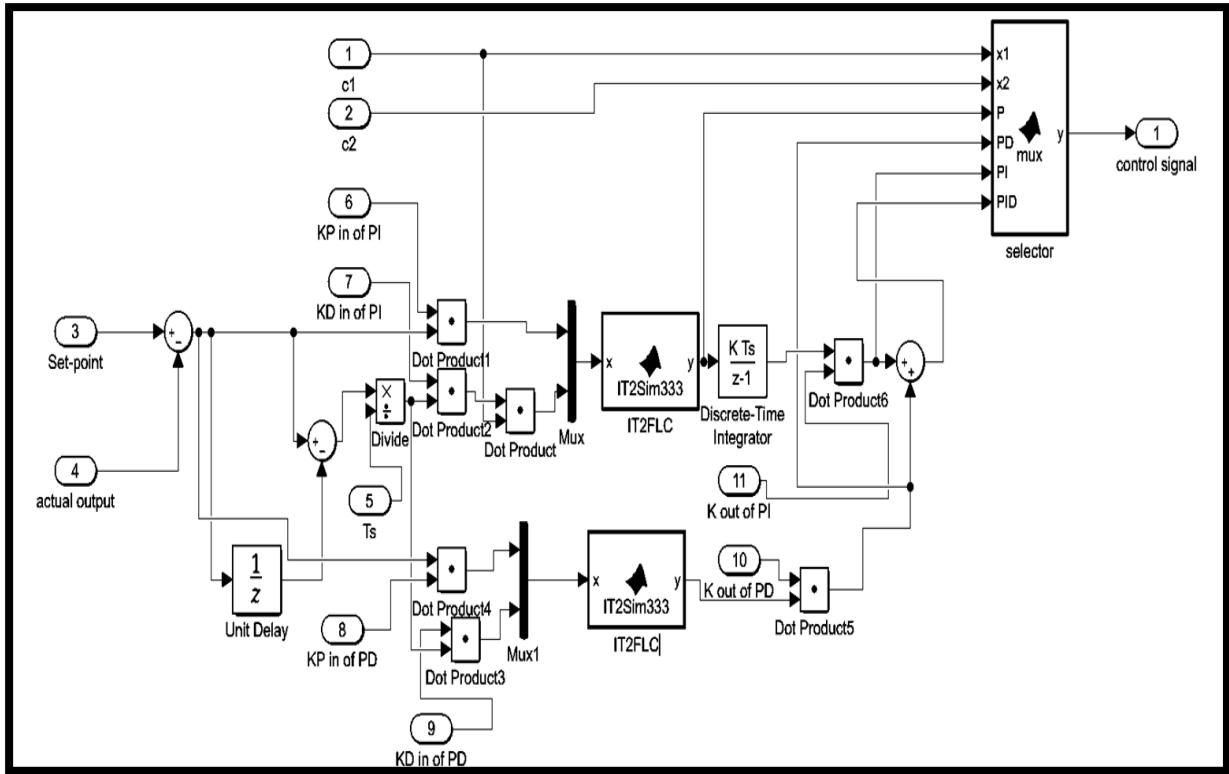


Fig. 8. SIMULINK structure of P/PI/PD/PID-like IT2FLC.

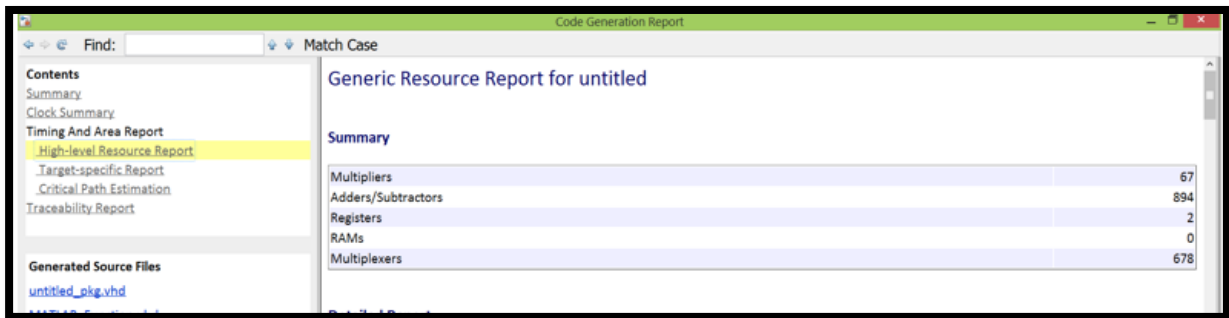


Fig. 9. MATLAB report generated by the MATLAB SIMULINK.

5.1. Linear Model

The choice of linear model is selected as an example and represented by the following discrete equation:

$$y(z) = \frac{0.00995z}{z-0.99} \quad 13$$

The step response of this model using P/PI/PD-like IT2FLC is shown in Fig. 10. The step response of this model using PID-like IT2FLC is shown in Fig. 11. The PID-like IT2FLC has the best response with the performance index root mean squared error (MSE) is equal to 0.5093.

The rise time is equal to 0.85. The overshoot is equal to zero and the settling time is equal to 0.9.

5.2. Nonlinear servo motor model

The DC motor is a particular sort of motors, which is classified as one of the principal machines to generate mechanical power from electrical power. mathematical modeling of servo actuator system is represented by a second order dynamic system with friction (Xie, 2007).

$$J\ddot{x} = ui - iF - T_{Li} \quad 14$$

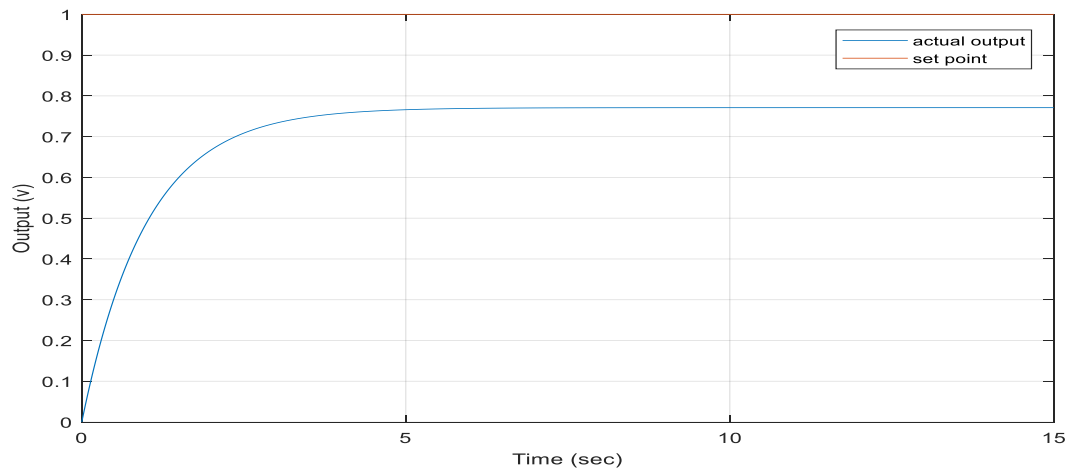
Where the moment of inertia is represented by J ; the acceleration is represented by $\ddot{x}$. The control input torque is represented by u . F is the friction torque and T_L is the load torque. The friction torque is represented by static friction phenomena, which include: coulomb friction. Stiction friction. and the viscous friction. i.e. (Xie, 2007).

$$F = \left\{ F_s \exp\left(-\left(\frac{\dot{x}}{\dot{x}_s}\right)^2\right) + F_c \left(1 - \exp\left(-\left(\frac{\dot{x}}{\dot{x}_s}\right)^2\right)\right) + \sigma |\dot{x}| \right\} * \text{sgn}(\dot{x}) \quad 15$$

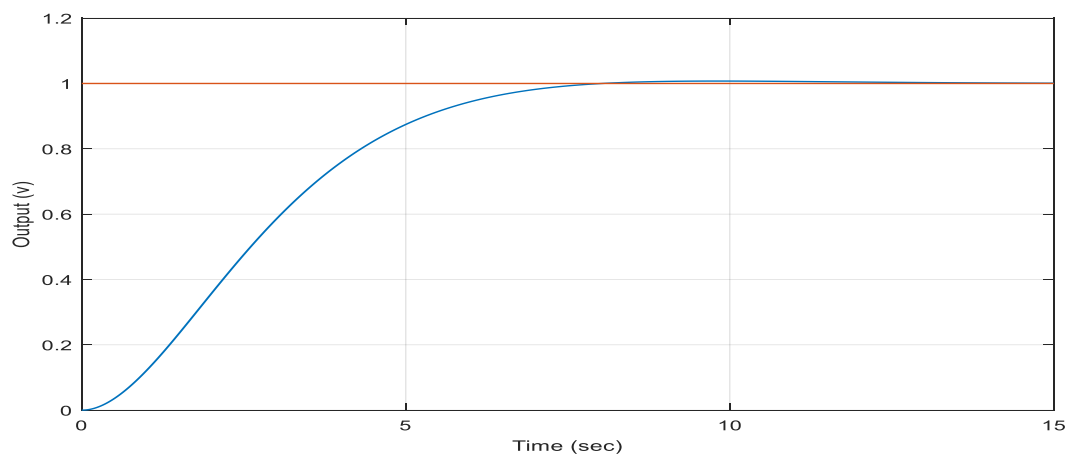
Where: the coulomb friction is represented by F_c . the stiction friction is represented by F_s . the stiction velocity is represented by $\dot{x}_s$. and σ is the viscous friction coefficient.

The step response of this model using PID-like IT2FLC is shown in Fig. 12.

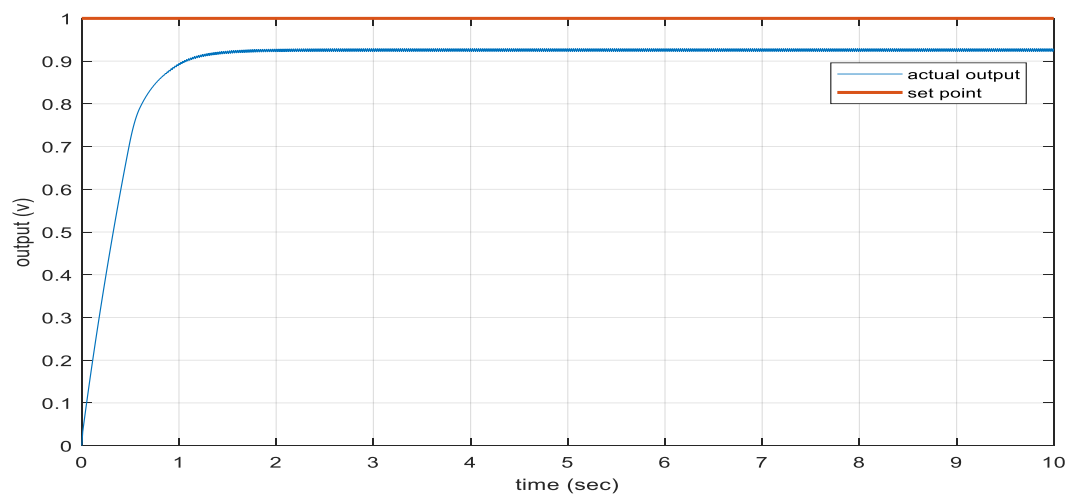
- a) The simulation without applying disturbance to the system with the performance index Root Mean. Squared error (RMS) is equal to 0.5352. The rise time is equal to 1.5 and the overshoot is equal zero.
- b) The simulation with applying uncertainty parameters to the system equal to 0.2 with the performance index RMS is equal to 0.5502. The rise time is equal to 2 and the overshoot is equal to zero.



(a)



(b)



(c)

Fig. 10. Simulation of the linear model.

a) P-like IT2FLC (b) PI-like IT2FLC (c) PD-like IT2FLC

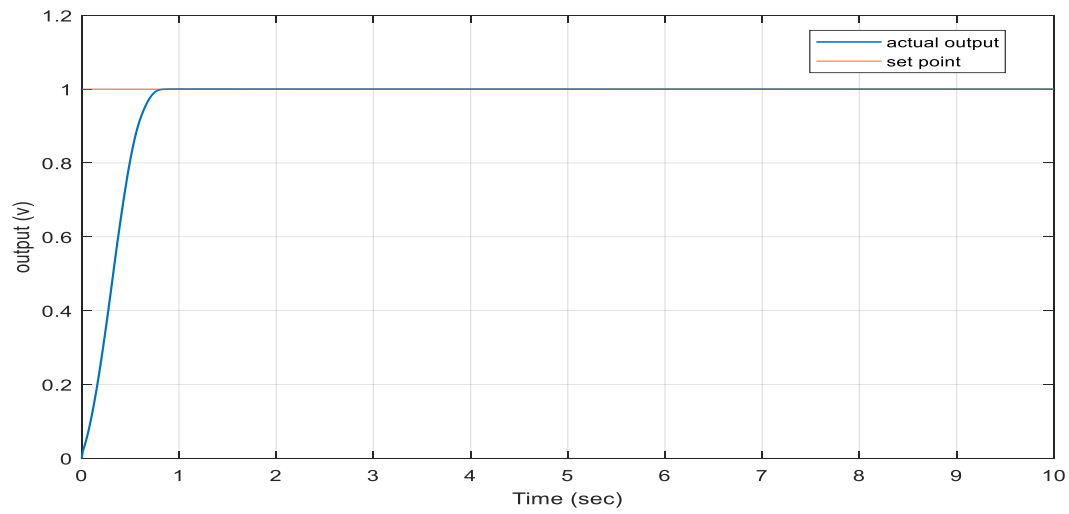


Fig. 11. Simulation of PID-like IT2FLC using linear model.

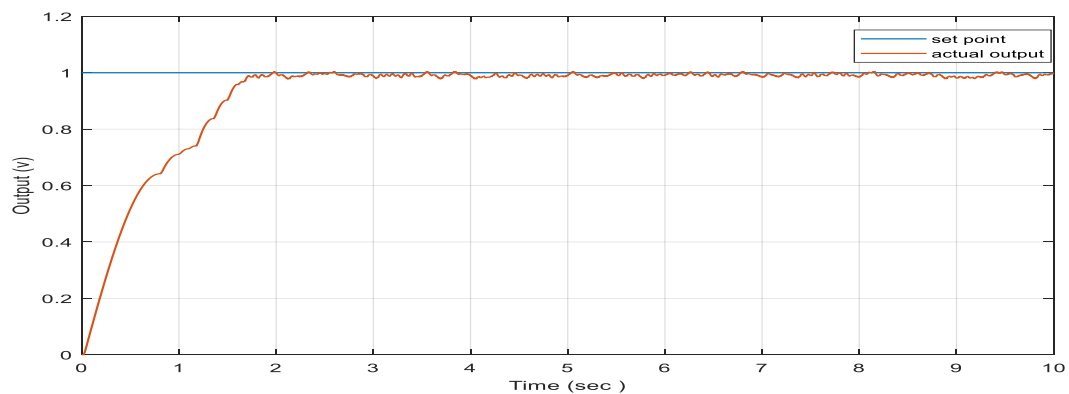
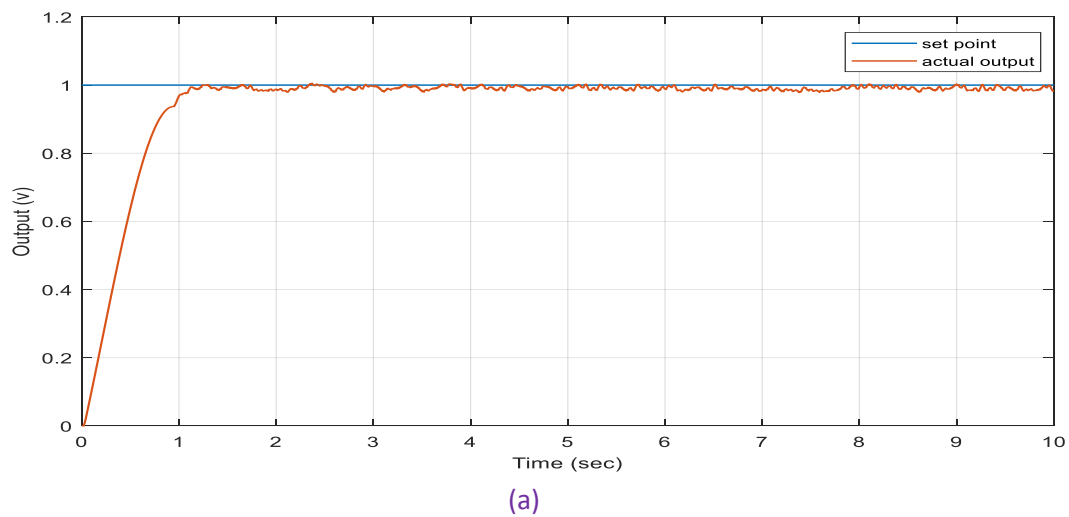


Fig. 12. Simulation of the nonlinear servomotor model.

(a) PID-like IT2FLC without uncertainty. (b) PID-like IT2FLC with uncertainty equal to

6. CONCLUSION

In this paper The P/PI/PD/PID-like IT2FLC with six gain factors on FPGA chip have been designed. This proposed controller was designed to control linear and nonlinear models that including the nonlinearity and uncertainty parameters and to reduce FPGA size and as a result less execution time must achieve vs. acceptable accuracy. The type of the controller is selected using two bits control input. The simulation results show that the step response of nonlinear model controlled by the proposed controller after changing the uncertainty parameters with 20% was close to response of the same model before this changing.

7. REFERENCES

- Biglarbegian, M., Melek, W. W. and Mendel J. M. "On the stability of interval type-2TSK fuzzy logic control systems". IEEE Transactions Systems Man and Cybernetics. vol. 40. No. 3. pp. 798 – 818. 2010.
- Hassan, M. Y. and Sharif W. F. "Design of FPGA based PID-like Fuzzy Controller for Industrial Applications". International Journal of Computer Science. 2007.
- Maldonado, Y., Castillo, O. and Melin P. "Optimal design of type-2 fuzzy controllers with a multiple objective genetic algorithm for FPGA implementation". Fuzzy Information Processing Society (NAFIPS), 2011.
- Mani, A. S. and Barjeev, T. "Design and Implementation of Fuzzy Controller on FPGA" I. J. Intelligent Systems and Application. pp. 35-42. September 2012.
- Maldonado, Y., Castillo, O. and Melin, P. "Particle swarm optimization of intervaltype-2 fuzzy systems for FPGA applications". Applied Soft Computing. vol. 13. No. 1. pp. 496–508. (2013).
- Maldonado, Y., Castillo, O. and Melin, P. "A multi-objective optimization of Type-2 fuzzy control speed in FPGAs." Applied Soft Computing. vol. 24. pp.1164–1174.November 2014.
- Mehmet, D. F., Tufan K., Ahmet, S. and Engin, Y. "Boundary Function based Karnik- Mendel Type Reduction Method for Interval Type-2 Fuzzy PID Controllers" IEEE International Conference on Fuzzy Systems .pp.6-11. July 2014. Beijing. China.
- Jun, L. J., Hsuan, H. H. and Hwai C. P. "Evaluation of an FPGA-based fuzzy logic control of feed-water for ABWR under automatic power regulating." Progress in Nuclear Energy. vol. 79. pp. 22–31. 2015.

Mendel, J. M., John, R. I. and Liu F. "Interval Type-2 Fuzzy Logic Systems Made Simple". IEEE Transactions on Fuzzy Systems. Vol. 14. No. 6. December 2006.

Karnik, N. N., Mendel J. M. and Liang Q. "Type-2 fuzzy logic systems." IEEE Trans. on Fuzzy Systems. vol. 7. pp. 643–658. 1999.

Wu, D. and Tan, W. W. "Genetic learning and performance evaluation of type-2 fuzzy logic controllers." Engineering Applications of Artificial Intelligence. vol. 19. No. 8. pp. 829–841. 2006.

Wu, D. and Tan, W. W. "A simplified type-2 fuzzy controller for real-time control." ISA Transactions. vol. 15. No. 4. pp. 503–516. 2006.

Xie, W. "Sliding mode observer based adaptive control for servo actuator with friction." IEEE Transaction on Industrial Electronics. vol. 54. No. 3. pp. 1517-1527. June 2007.