AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1
ISSN: 1998-4456

Hardware Implementation of IT2FLC using FPGA for Control Applications

Assist. Prof. Dr. Mohammed Y. Hassan

University of Technology, Control and Systems Department, Baghdad, Iraq

E mail: **60003@uotechnology.edu.iq**

Saif Faris Abulhail

University of Technology, Control and Systems Department, Baghdad, Iraq

E mail: **saifabulhail80@gmail.com**

Lecturer Dr.Waleed Fawwaz Shareef

University of Technology, Control and Systems Department, Baghdad, Iraq

E mail: 60026@uotechnology.edu.iq

Received on 4 December 2017 Accepted on 22 January 2018 Published on 14 May 2018

DOI: 10.30772/qjes.v11i1.519

Abstract: Interval Type2 Fuzzy Logic Control (IT2FLC) has been applied to a number of industrial, medical, home and military applications. Hardware implementation of IT2FLC can be achieved in a number of ways. Programmable Gate Array (FPGA). One of these ways is the use of a Field

In this paper, the design and implementation of an IT2FLC using FPGA has been presented. The proposed controller is of Mamdani type. It works in different structures (P/PI/PD/PID like IT2FLC) depending on two control lines, different number of triangular shape memberships (2-7) depending on three control lines, six tunable gains and within a range of sampling time of (0.01-1024) seconds. Three type reduction algorithms are used and it is found that the Enhanced Iterative Algorithm with Stop Condition (EIASC) produced the minimum reduction in FPGA size. Thus less execution time. The reduction size is about 75% than Karnick Mendel (KM) and is about 3% than Enhanced KM (EKM). Linear and nonlinear models are used to test the designed Controller. Gains are tuned manually to reach minimum overshoot, settling time and steady state error.

Simulation and Implementation results showed that the proposed controller works in an efficient way under no-load, load and uncertainty in the nonlinear model parameters.

Keywords: Footprint of uncertainty, Karnick Mendel, Enhanced Karnick Mendel and FPGA

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

Nomenclatures

Symbols	Definition
$\tilde{A}$	Type-2 Fuzzy sets
$\overline{f^n}$	Upper Firing Level
$\tilde{R}^n$	Fuzzy rule number
F	Friction torque
F_c	Coulomb friction
g	Gravity
J	Moment of inertia
$\tilde{X}_i^n$	Inputs fuzzy sets
$e_{(k)}$	Sampled error
$\underline{f^n}$	Lower Firing Level
$\underline{y}^n$	Lower end point
$\overline{y}^n$	Upper end point
F^n	Firing interval
Y_{cos}	Centre of Sets Type Reduction
y_l	Left end point
y_r	Right end point
R	Right switch point
L	Left switch point
U	Control action
$\mu\,\underline{\tilde{X}_i^n}$	Lower Membership Function of Input $\tilde{X}_i^n$
$\mu\,\overline{\tilde{X}_i^n}$	Upper Membership Function of Input $\tilde{X}_i^n$
J_x	Type1 fuzzy set
$\overline{FOU(\tilde{A})}$	Upper Footprint of uncertainty

INTRODUCTION

Type2 FLC contain footprint of uncertainty (FOU) that is able to handle the numerical uncertainties, nonlinearities and linguistic associated with the inputs and outputs. **Hassan and Sharif [1],** in 2006 proposed a Proportional-Integral-Derivative Type1 FLC (PIDT1FLC). Fuzzy sets and programmable rule table were designed using Very High Description Language (VHDL) for implementation on FPGA device. The controller could serve as a Proportional-Derivative T1FLC (PDT1FLC), Proportional-Integral T1FLC (PIT1FLC), or Proportional-Integral-Derivative T1FLC (PIDT1FLC). Two versions of this controller were designed; the first one is 6-bits FPGA-Based Design, which uses 6 bits for I/O variables, while the second one is 8-bit FPGA Based Design, which uses 8 bits. The results of the simulation showed that the 8-bit FPGA-Based Design is superior to the 6-bit FPGA-Based Design. **Maldonado, et al. [2]**, in 2011 presented the design of the optimal T2FLC obtained using Genetic Algorithms (GA) for optimization. The optimization of Triangular and Trapezoidal Membership Functions (MFs) of a fuzzy system is done for hardware representations such as the FPGA. The GA uses only certain points of the MFs, while the fuzzy rules did not

1.

Vol. 11, No. 1
ISSN: 1998-4456

change. The GA was tested in a T2FLC to regulate the direct current motor speed, using the MATLAB/SIMULINK and VHDL code. Comparisons were made between the T1FLC versus T2FLC, to evaluate the difference in performance of both types of controllers. **Mani and Barjeev [3],** in 2012 developed the implementation of a T1FLC through the use of the VHDL code. FLC is designed for an armature control DC motor speed control. VHDL has been used to develop FLC on FPGA. A Sugeno type FLC structure has been used to obtain the controller output. The controller algorithm developed, synthesized, simulated and implemented on FPGA Spartan 3E xc3s500e-4fg320 board. **Panda, et al. [4],** in 2012 designed an IT2FLC for an automatic voltage regulator system. For controller design, memberships of system variables are represented using interval value fuzzy sets. The effectiveness of this controller has been investigated through simulation studies. The simulation showed performance of the controller and compared with a PID controller. **Jun, et al. [5]**, in 2014 presented a new feed-water controller under the automatic power regulating system for an advanced boiling water reactor. The new feed water controller is designed by using a rule-based hierarchical FLC and is implemented by using the FPGA technology. The results demonstrated that the FPGA-based hierarchical FLC is a practical approach for automatic power operations in advanced nuclear power plant applications**. Schrieber and Biglarbegian [6],** in 2015 developed hardware implementation and performance comparison of IT2FLC for real ime applications. IT2FLC with parallel processing using FPGA was designed and implemented. Design and implementation are using three different inference mechanisms of IT2FLC on hardware. **Li, et al. [7]**, in 2015 designed sampled data controller for IT2FLC with actuator fault. The IT2FLC and the IT2 state-feedback controller share different MFs.

It was found that there has not been a major focus on the design and implementation of P/PI/PD/PID like IT2FLC using FPGA chip. In this paper, an IT2FLC can work with different number of memberships and several type reduction algorithms with high resolution data. Several promising approaches have used other types in the design of controllers. The proposed IT2FLC is designed 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. Hardware setup and experimental results of the linear and nonlinear models are explained.

The rest of this paper is organized as follows: section 2 introduces to an IT2FLC and its computations. Section 3 describes the designed controller, including the performing of an IT2FLC with different structures and type reduction algorithms. Section 4 illustrates the implementation and hardware setup of an IT2FLC, including ALTERA DE2 board, National Instrument (NI) 6212 device. It describes the connections between these devices with PC using hardware in the loop approach. Section 5 shows the practical results of the designed IT2FLC that is connecting with linear and nonlinear models. Section 6 presented the conclusions from this work.

1. INTERVAL TYPE2 FLC

Type2 Fuzzy sets membership function are fuzzy and contain FOU that handles and models the uncertainties, nonlinearities and linguistic related with the inputs and output of the FLC. FOU is the area between lower membership and upper membership. It is allowing each input to have two membership grade values related with it; an Upper MF (UMF) and Lower MF (LMF). There are two types of T2FLC: Mamdani type and Takagi Sugeno Kang (TSK) type. The difference between these two types, the Sugeno output membership fucntion is linear or constant and the Mamdani output is shape the membership. Mamdani type needs type reducer method while the Takagi Sugeno type does not need type reduction operation. The structure of the IT2FLC is shown in Figure (1) [8, 9].

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1
ISSN: 1998-4456

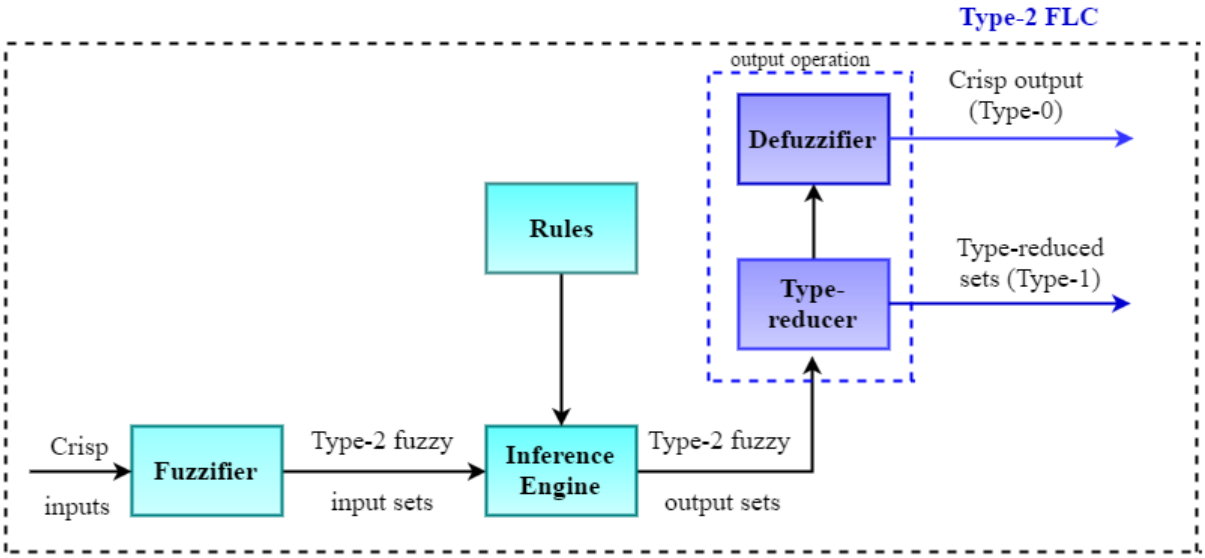


Figure 1: Structure of T2FLC.

1.1 COMPUTATIONS OF IT2FLC

An IT2FLC containing at least one interval fuzzy set without loss of generality. If an IT2 FLC consisting of N rules, the rule base has the following form:

$\tilde{R}^n : y = Y^n$ If $x1$ is $\tilde{X}_1^n$ and … and x_I is $\tilde{X}_I^n$, where $\tilde{X}_i^n$ $(i = 1 \ldots. I)$ are IT2 fuzzy sets. In addition, $Y^n = \left[\underline{y}^n . \overline{y}^n\right]$ is an interval that can be understood as the centroid of a consequent IT2 fuzzy set or the simplest TSK model. Each rule consequent is represented by a crisp number in many applications [10]. For an input vector $X' = (x'_1 . x'_2 . \ldots.. x'_I)$. Typical computations in an IT2FLC include the following steps [10]:

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

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

$F^n = \left[\mu \underline{\tilde{X}_1^n}(x'_1) \times \ldots \times \mu \underline{\tilde{X}_I^n}(x'_I), \mu \overline{\tilde{X}_1^n}(x'_1) \ldots \times \mu \overline{\tilde{X}_I^n}(x'_I)\right] \equiv \left[\underline{f^n} . \overline{f^n}\right] . n$ =1,2…N.

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

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

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

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

In equation (2) and equation (3) K is a potential switch point.

4) Compute the defuzzified output as:

$$y = (yl + yr)/2 \quad (4)$$

2. CONTROLLER DESIGN

The general block diagram of the proposed controller is shown in Figure (2). This proposed IT2FLC is designed by combining the advantages of fuzzy inference and different structures of controller. The type of the controller is selected using two bits selection lines, as illustrated in Table (1). Structure design of the proposed IT2FLC is shown in Figure (3). The inputs to the controller are: input-output tunable gains,

sampling time, set-point and the control signal. The output of the controller is the control action signal. The range of the universe of discourse is within $[-1\ to\ 1]$ for each input and output.

The proposed controller can work with (2-7) input-output triangular shaped MFs. The user can select the number of MFs using three control signals (C3, C4 and C5) that is listed in Table (2). The shapes of MFs design are shown in Figure (4). The defuzzification technique is selected as Centroid method. The sampling time is selected by user depending on the model that is connecting with the proposed controller. Rule-base is written in a lookup table. This Rule-base is designed by the user and located in the controller. Mamdani-type is used to perform fuzzy inference of the controller.

Three algorithms KM, EKM and EIASC are used to implement the type reduction method of the IT2FLC. EKM has three improvements over the KM algorithm. First, a better initialization is used to reduce the number of iterations. Then, the termination condition of the iterations is changed to remove one unnecessary iteration. Finally, a subtle computing technique is used to reduce the computational cost of each iteration. EIASC algorithm enumerate the switch point for y_l from 1 to N−1 until $fl_{(k)}$ stops decreasing, at which point y_l is obtained. Similarly, they enumerate the switch point for y_r from 1 to N−1 until $fr_{(k)}$stops increasing, at which point $y_r$is obtained. The comparisons among three type reduction algorithms are listed in Table (3). It can be concluded from Table (3) that using EKM algorithm gives small FPGA chip size than KM algorithm and as a result generates less execution time and using EIASC algorithm produces small FPGA chip size than EKM algorithm and as a result generates less execution time. It can be notice that the use of EIASC with the proposed controller give us less FPGA size. Thus, EIASC will be used in the implementation of the IT2FLC. The programs are written using MATLAB / SIMULINK and MATLAB functions, then converted into VHDL and Verilog formats using HDL coder. HDL coder technique has many restrictions and limitations in MATAB functions, such as (for loop, if statement, floating point and FIND function). Therefore; these limitations were solved by designing MATLAB functions that overcome the HDL coder restrictions. The program is generated to achieve minimum FPGA size and as a result, less execution time. Furthermore, In order to design a PID like IT2FLC, t 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 difficult to design. For example, if eight fuzzy sets are used for each input, then a (8*8*8=512) rules will be required for the controller. Instead, PID like T2FLC can be designed as a parallel structure of a PD like IT2FLC and a PI like IT2FLC. The output of the PID like IT2FLC is formed by algebraically summing the outputs of the two fuzzy control structure. This method will reduce the number of rules required to (8*8+8*8=128) rules only, as shown in Figure (5). A PD like IT2FLC may be employed to serve as PI like IT2FLC in incremental form. Equation (5) shows a PD controller equation obtained in a position form, while Equation (6) shows a PI controller equation in an incremental form in discrete time domain:

$$u_{D(k)} = KP\ e_{(k)} + KD\ \Delta e_{(k)} \tag{5}$$

$$\Delta uI_{(k)} = KP\ \Delta e_{(k)} + KD\ e_{(k)} \tag{6}$$

$$uI(k) = uI_{(k-1)} + \Delta\ uI_{(k)} \tag{7}$$

$$\Delta e(k) = \frac{e_{(k)} - e_{(k-1)}}{Ts} \tag{8}$$

where $uI_{(k-1)}$ is the previous control signal, $e_{(k)}$is a sampled error signal, $e_{(k-1)}$ is a previous sampled error signal $\Delta e_{(k)}$ is the rate of change of sampled error signal, Ts is the sampling time and index (k) represents the present sampling instant. Now by comparing equations (4) and (5) , it show that the PD like IT2FLC in a position form becomes a PI like IT2FLC in incremental form if $e_{(k)}$ and $\Delta e_{(k)}$ exchange positions, KD is replaced by KI and $uI_{(k)}$ is replaced by $\Delta uI_{(k)}$. The output of a PI like IT2FLC is obtained by summing the change of control signal $\Delta\ uI_{(k)}$ and the previous control signal$uI_{\circ}$. 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 controller design can act as a P like IT2FLC by setting KD of PD and PI like IT2FLC to zero, a PD like IT2FLC, a PI like IT2FLC, and a PID like IT2FLC depending on two bits control signal C1 and C2 as illustrated previously.

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1
ISSN: 1998-4456

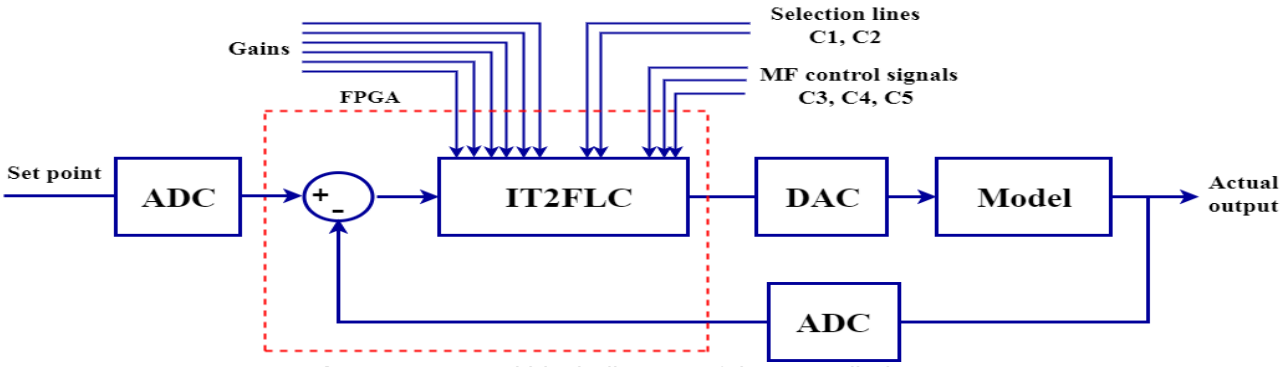


Figure 2: General block diagram of the controlled system

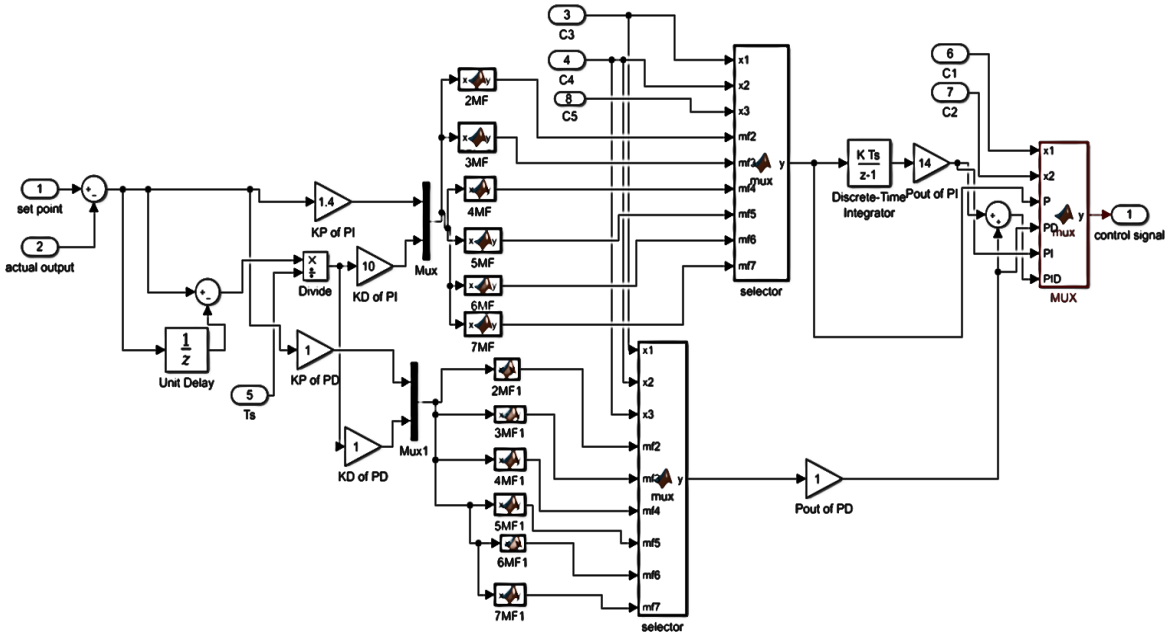
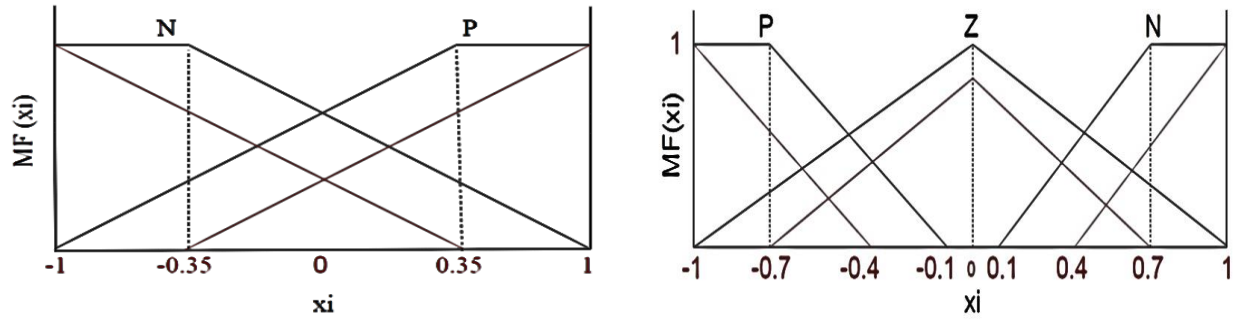


Figure 3: Structure of the proposed controller

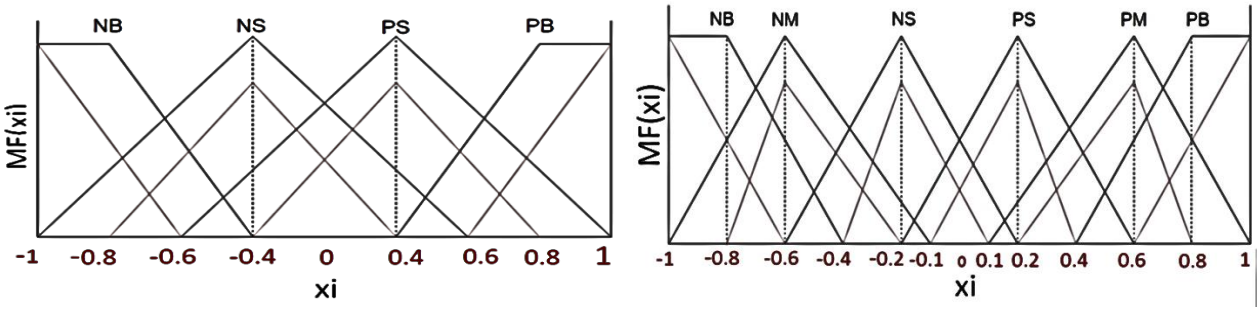


(a) Two MFs (b) Three MFs

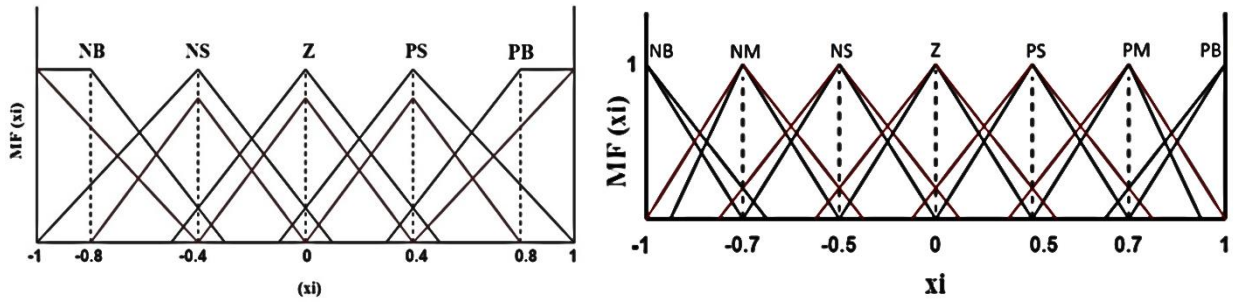
AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

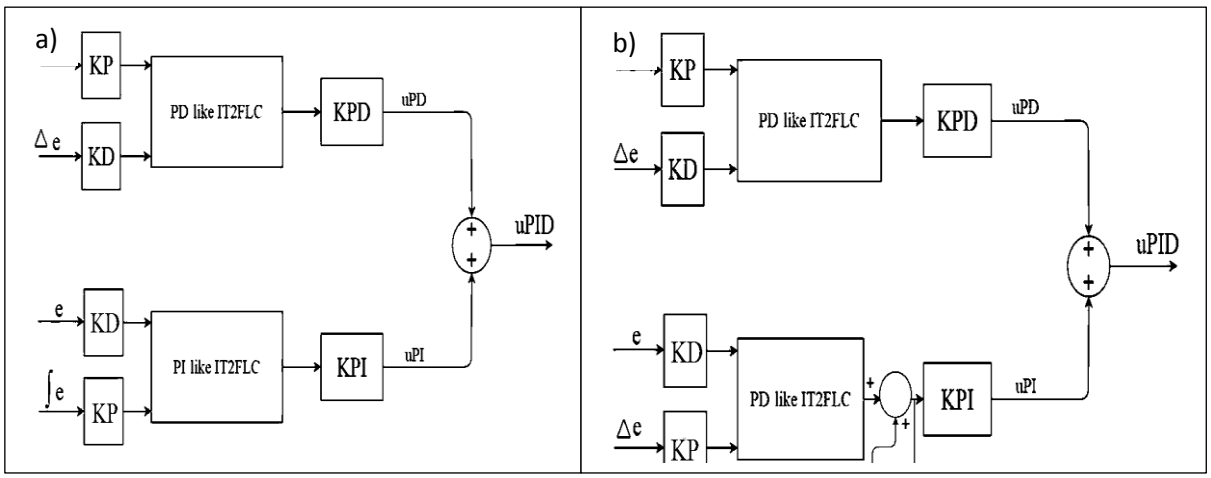


(C) Four MFs (d) Six MFs



(e) Five MFs (f)Seven MFs

Figure (4): MFs design



(a) Three inputs PID like IT2FLC (b) Two inputs PID like IT2FLC

Figure 5: Minimizing rule-base of PID like IT2FLC

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

3. IMPLEMENTATION Of THE PROPOSED IT2FLC

3.1 Altera Development FPGA Board

Altera DE2 board is shown in Figure (6), it provides two 40-pin expansion headers. Each header connects directly to 36 pins on the Cyclone LL FPGA, and also provides DC +5V (VCC5) and two GND pins [11].

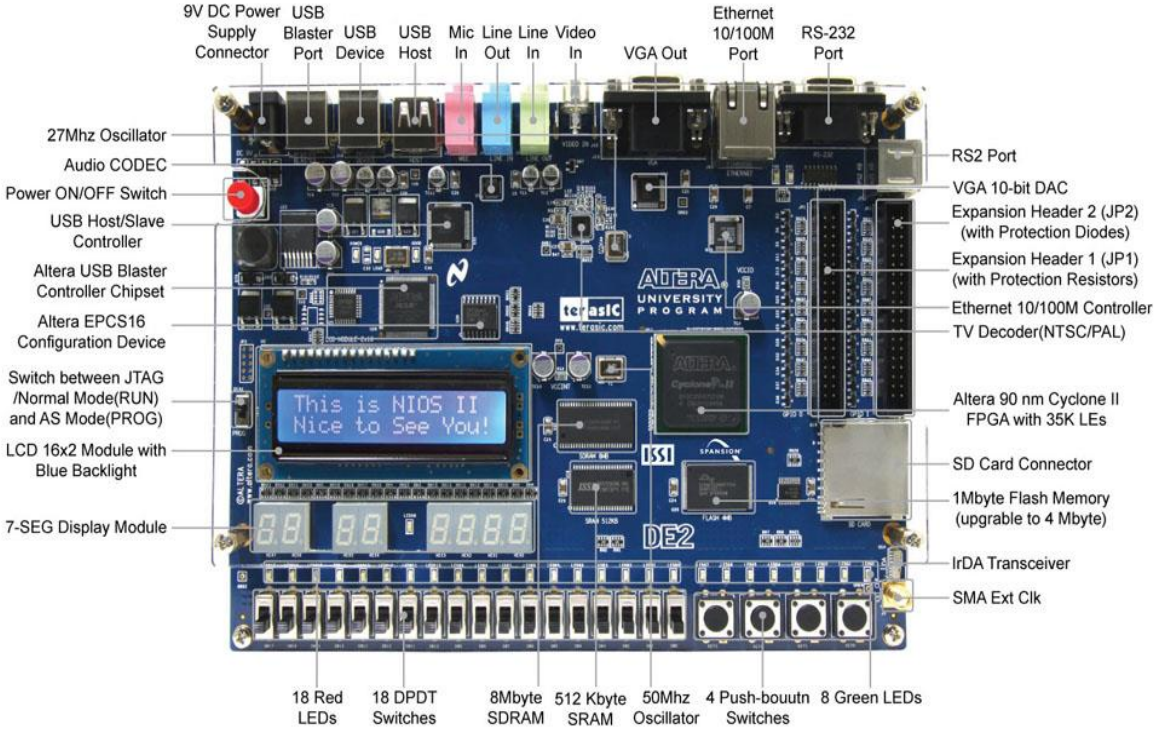


Figure 6: ALTERA FPGA board

3.2 NI USB 6212-DAQ Device

Data acquisition systems is shown in Figure (7), it incorporate signals, sensors, actuators, signal conditioning, data acquisition devices and application software. NI USB-6212 DAQ is a simple and low cost multifunction I/O device from National Instruments.

Figure 7: NI DAQ 6212

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

3.3 Hardware Connections

The advantage of NI USB 6212 DAQ in this design is to transfer data between the PC and Altera DE2 board. Verilog HDL language is used in Quartus II software program to implement the proposed IT2FLC. Verilog code that represents the proposed IT2FLC is synthesized and downloaded into the FPGA using Quartus II 13.0 web edition software (64-Bits). Quartus II summary report, shown in Figure (8), describes the requirement FPGA gates. The connection between the model and the proposed controller is using hardware in the loop approach as shown in Figure (9), where the model is simulated in MATLAB and the controller is working in FPGA ALTERA DE2 Board. NI USB 6212-DAQ acts as the interfacing between PC and the proposed controller. Linear and nonlinear models are simulated in the MATLAB/ SIMULINK environment. Data Acquisition Toolbox software is installed in (MATLAB 2017a) to provide a set of tools for analog input, alog output, and digital input/output. The control action generated from the proposed controller is sent to the PC in a parallel way through NI USB 6212-DAQ device. NI USB 6212-DAQ is sending it in serial mode to the PC using USB port. The feedback signal is returned to the controller through DAQ device using USB port. The required ALTERA DE2 pins are connected with NI USB 6212-DAQ pins using wires. Thirty-two wires are connected between NI USB 6212 DAQ and ALTERA DE2 board, 16 for inputs and 16 for outputs. The pins from (A0 – A15) are setting as output (control action) and (A15 -A31) are setting as input (actual output). Two switches (C1) and (C2) are using to select the type of the proposed controller as illustrated in Table (1). Three switches (C3, C4 and C5) are used to select the number of MFs, see Table (2). One push-Bottom is used to reset the controller. Finally, the photograph of the hardware setup is shown in Figure (10).

Flow Summary	
Flow Status	Successful - Sun Sep 17 17:27:01 2017
Quartus II 64-Bit Version	13.0.0 Build 156 04/24/2013 SJ Web Edition
Revision Name	pidIT2_1tunable
Top-level Entity Name	pidIT2_1tunable
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	15,715 / 33,216 (47 %)
Total combinational functions	15,715 / 33,216 (47 %)
Dedicated logic registers	13 / 33,216 (< 1 %)
Total registers	13
Total pins	78 / 475 (16 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	34 / 70 (49 %)
Total PLLs	0 / 4 (0 %)

Figure 8: Quartus II Summary Report

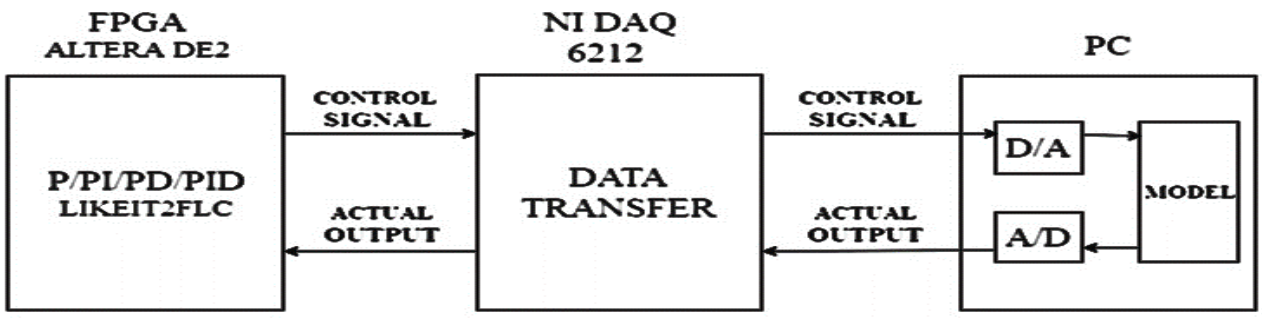


Figure 9: Hardware In The Loop Connections

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

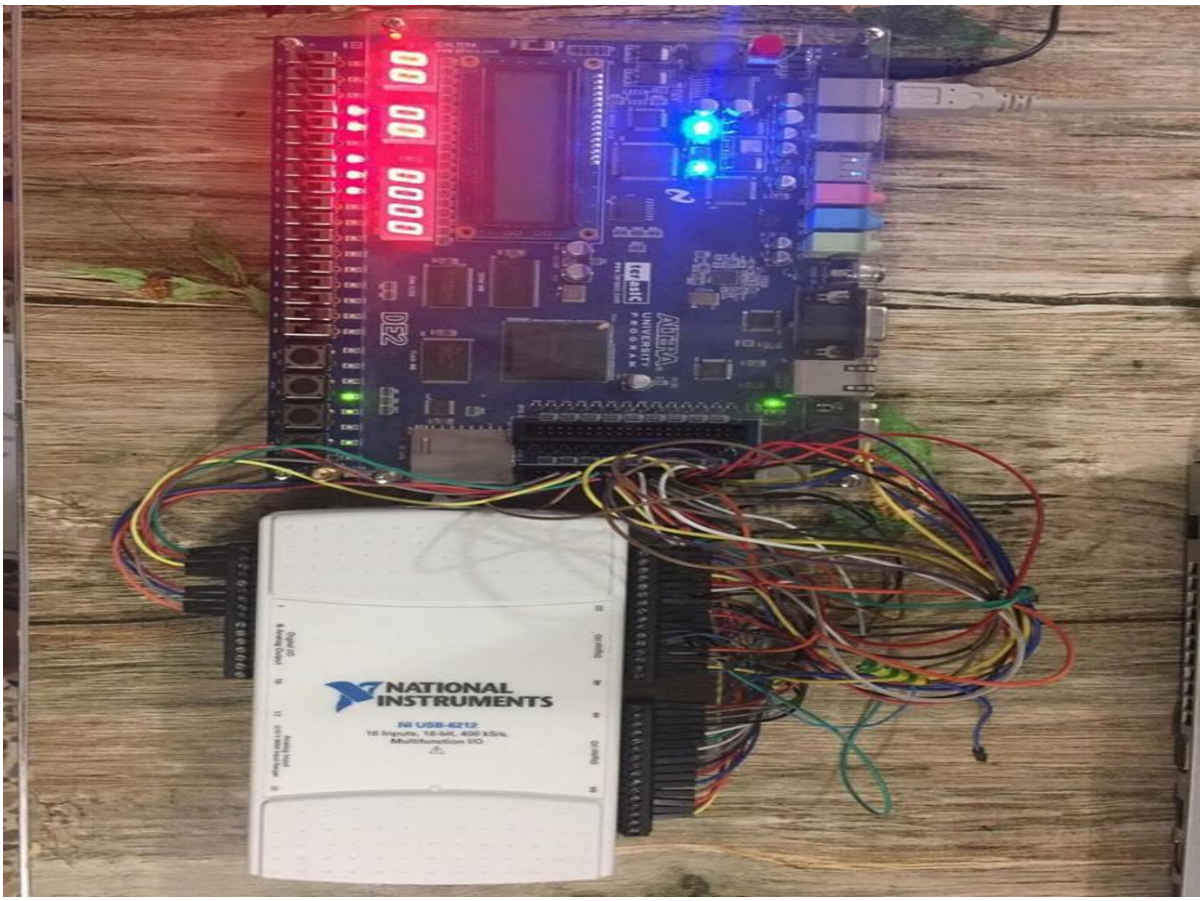


Figure 10: Connections between NI USB 6212-DAQ and ALTERA DE2 board

4. PRACTICAL RESULTS

In order to examine this proposed controller, linear and nonlinear models are used. In this design sampling time is set to 0.01 seconds.In order to reach minimum overshoot, minimum undershoot, minimum settling time, minimum steady state error and minimum structure design, trial and error method is used to tune the gains and rules as shown in Table (4) to obtain the best response. The desired position is chosen as a unit step value. Seven Triangular memberships are chosen for testing purposes.

4.1 Linear Model

A linear model is selected as an example and represented by the following transfer function:

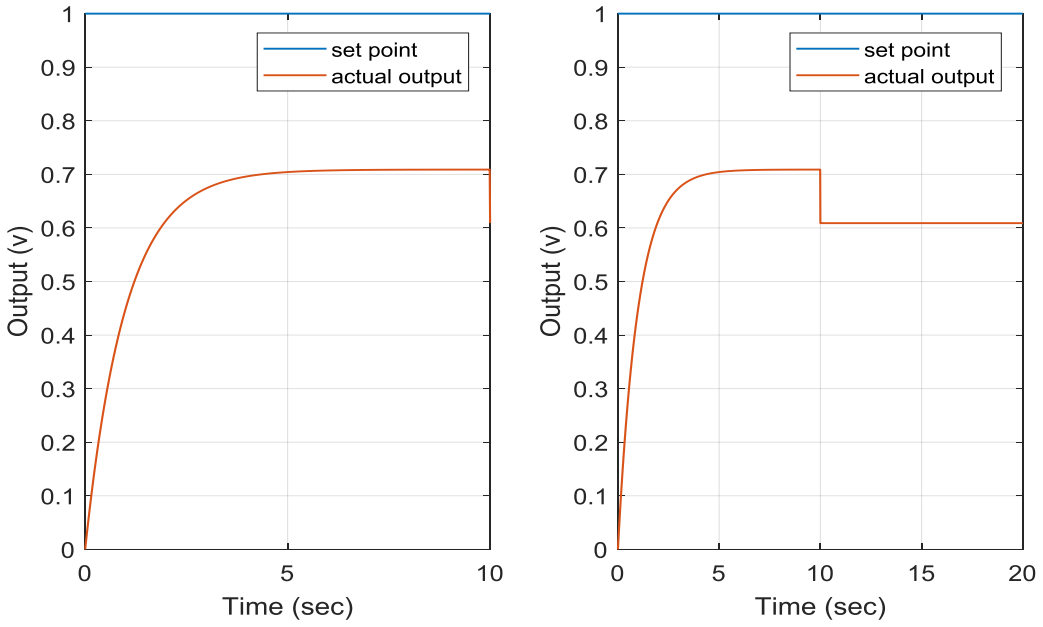
$$y_{(z)} = \frac{0.00995z}{z-0.99} \tag{9}$$

Gains are tuned manually and they are listed in Table (5). The unit step response of this model using P like IT2FLC as shown in Figure (11), PD like IT2FLC as shown in Figure (12), PI like IT2FLC as shown in Figure (13) and PID like IT2FLC is shown in Figure (14).

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

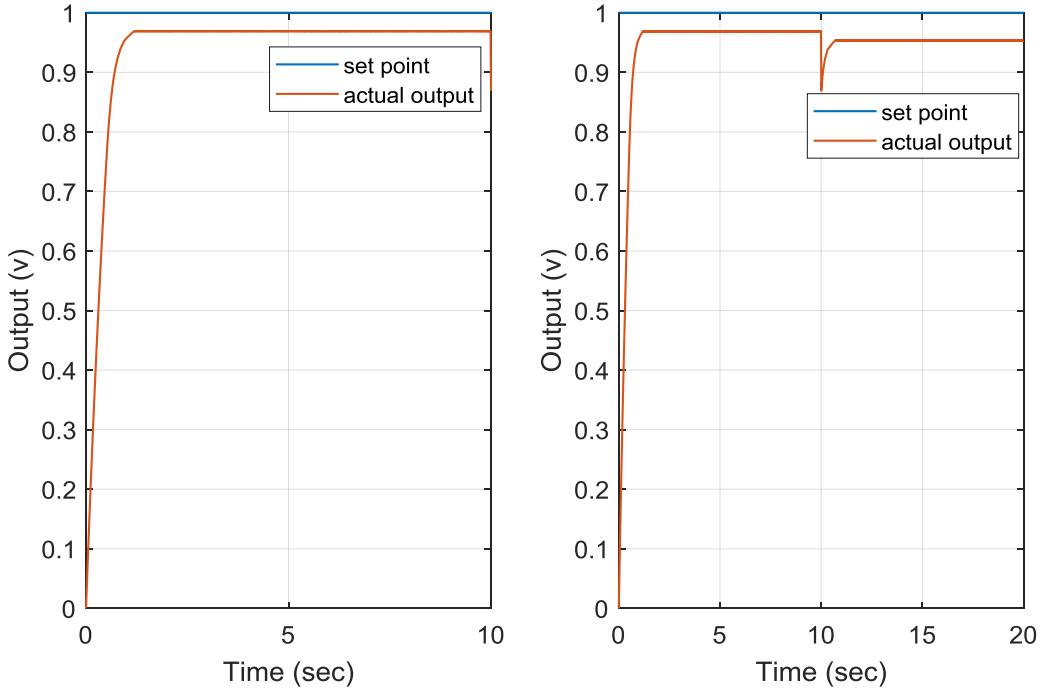
Vol. 11, No. 1

ISSN: 1998-4456



(a) with no load (b) with load =10% from step input

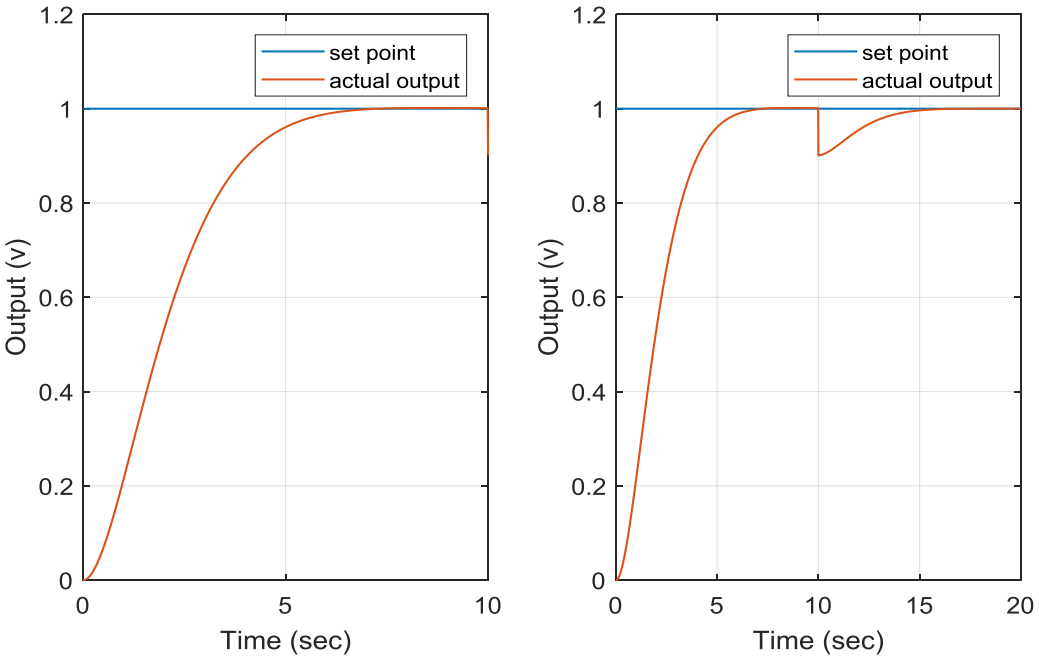
Figure 11: Output time response of the linear model controlled by the P like IT2FLC



(a) with no load (b) with load =10% from step input

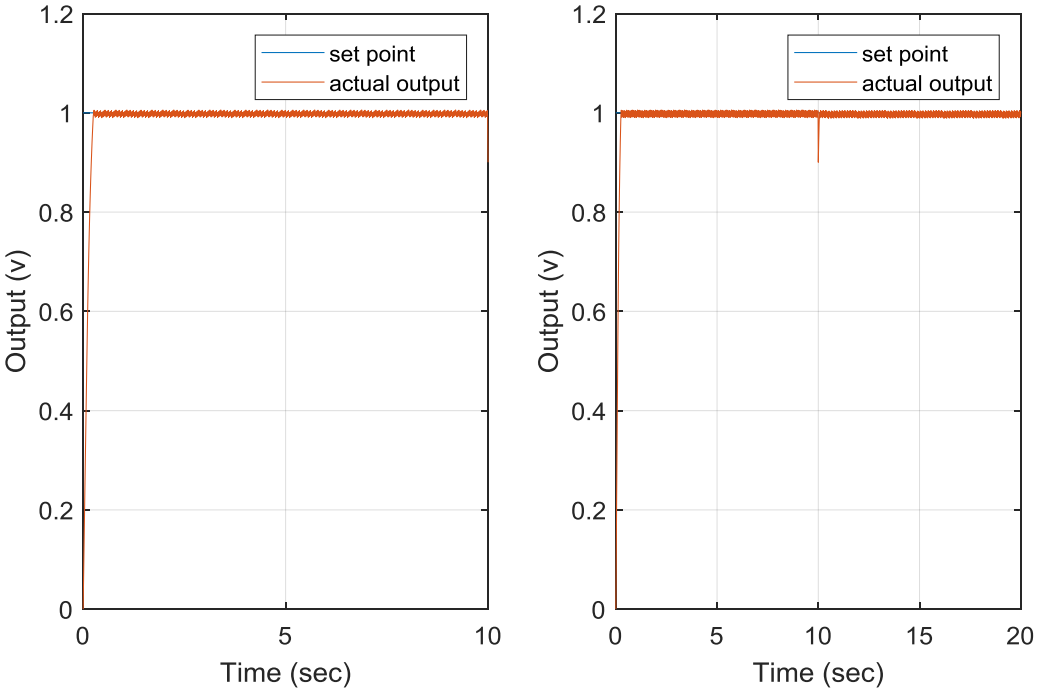
Figure 12 : Output time response of the linear model controlled by the PD like IT2FLC

Vol. 11, No. 1
ISSN: 1998-4456



(a) with no load (b) with load =10% from step input

Figure13: Output time response of the linear model controlled by the PI like IT2FLC



(a) with no load (b) with load =10% from step input

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456

Figure 14: Output time response of the linear model controlled by the PID like IT2FLC

4.2 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. The mathematical model of servo motor system is represented by a second order dynamic system with friction as follows [12]:

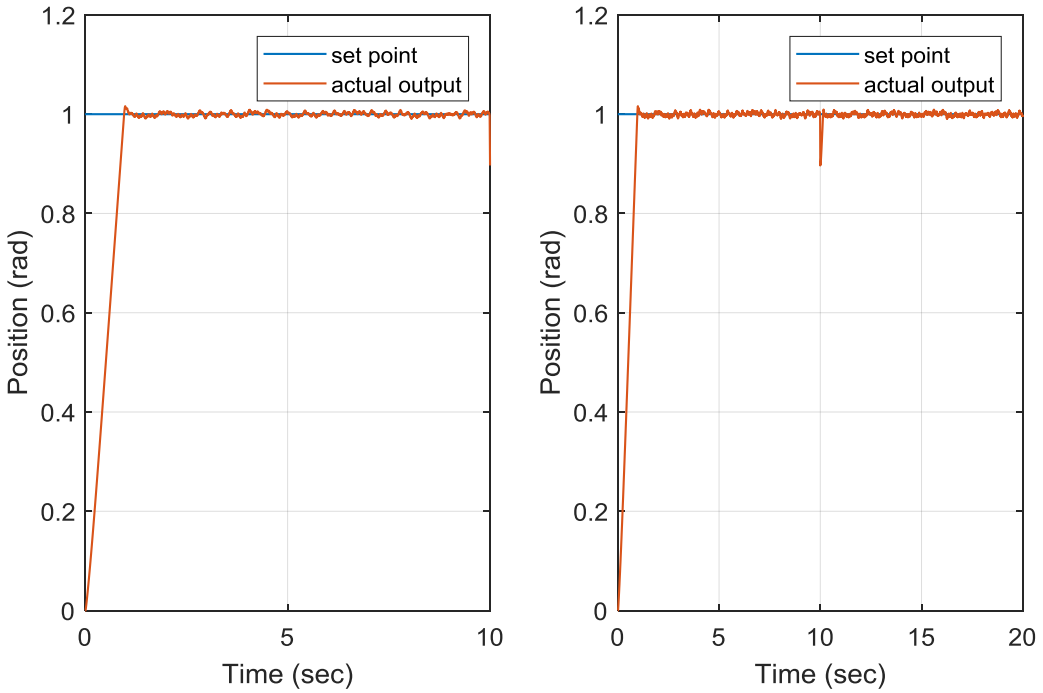
$$J\ddot{x} = u - F - T_L \tag{10}$$

where the moment of inertia is represented by J; the acceleration is represented by $\ddot{x}$. The control input torque is represented by . 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.. [14].

$$F = \left\{F_s \exp\left(-\left(\frac{\dot{x}}{\dot{x}_s}\right)^2\right) + F_c\left(1 - \exp\left(-\left(\frac{\dot{x}}{\dot{x}_s}\right)^2\right)\right) + \sigma|\dot{x}|\right\} * sgn(\dot{x}) \tag{11}$$

where the coulomb friction is represented by F_c. the stiction friction is represented by F_s. the stribeck velocity is represented by $\dot{x}_s$. and σ is the viscous friction coefficient [12].

Gains are tuned manually and they are listed in Table (6).The unit step response of this model using PID like IT2FLC without applying uncertainty to the system is shown in Figure (15). Moreover, the unit step response with applying uncertainty about 10% from unit step input to the friction and moment of inertia parameters is shown in Figure (16).



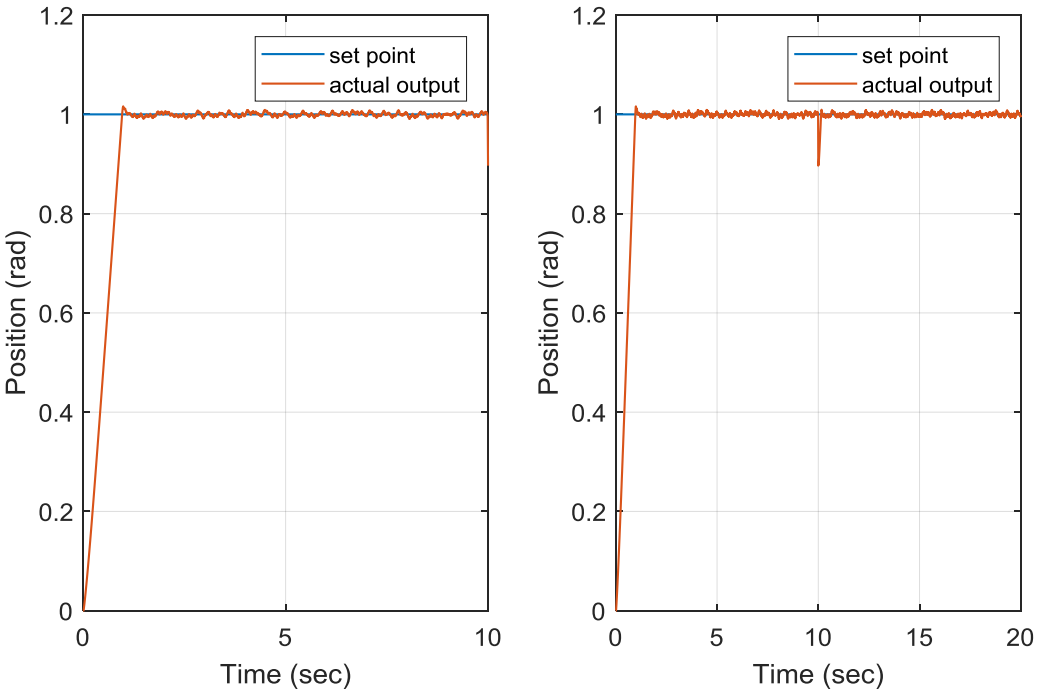
(a) with no load (b) with load =10% from step input

Figure 15: Output time response of the servo motor controlled by the PID like IT2FLC

AL-QADISIYAH JOURNAL FOR ENGINEERING SCIENCES

Vol. 11, No. 1

ISSN: 1998-4456



(a) with no load (b) with load =10% from step input

Figure 16: Output time response of the servo motor with uncertainty parameter of about 10 % controlled by the PID like IT2FLC

CONCLUSION

In this paper, the P/PI/PD/PID like IT2FLC has been implemented on FPGA chip. This proposed controller is designed to control linear and nonlinear models. The practical results showed that the unit step response of nonlinear model controlled by the proposed controller after changing the uncertainty parameters with 10% was close to response of the same model before this changing. It can be used with industrial, home, medical and military applications. The end user can select sampling time, the structure of the controller, number of memberships and tune the gains only.

REFERENCES

1. **M. Y. Hassan and W. F. Sharif**." Design of FPGA based PID like Fuzzy Controller for Industrial Applications", International Journal of Computer Science, 2007.
2. **Y. Maldonado. O. Castillo and P. Melin**. "Optimal design of type-2 fuzzy controllers with a multiple objective genetic algorithm for FPGA implementation", Fuzzy Information Processing Society (NAFIPS), 2011.
3. **A. S. Mani and T. Barjeev** "Design and Implementation of Fuzzy Controller on FPGA" International Journal Intelligent Systems and Application, pp. 35-42. September 2012.
4. **M. K. Panda, G. N. Pillai and V. Kumar**, "Design of an Interval Type2 Fuzzy Logic Controller for Automatic Voltage Regulator System", Electric Power Component and System Vol.40, 219–235, 2012.
5. **L. J. Jun . H. H. Hsuan. C. P. Hwai**. "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.
6. **M. D. Schrieber and M. Biglarbegian**, "Hardware implementation and performance comparison of interval Type-2 fuzzy logic controllers for real time applications", Application Soft Computing, Vol. 32, PP. 175–188, 2015.

7. **H. Li, X. Sun, P. Shi and H.K. Lam,** "Control design of interval Type2 fuzzy systems with actuator fault: Sampled data control aPProach", Information Science, Vol. 302, PP. 1–13, 2015.

8. **J. M. Mendel. R. I. John and F. Liu**. "Interval Type-2 Fuzzy Logic Systems Made Simple" IEEE Transactions on Fuzzy Systems. Vol. 14. No. 6. December 2006.
9. **N. N. Karnik. J. M. Mendel. and Q. Liang**. "Type-2 fuzzy logic systems." IEEE Trans. on Fuzzy Systems. vol. 7. pp. 643–658. 1999.
10. **D. Wu and W. W. Tan**. "Genetic learning and performance evaluation of type-2 fuzzy logic controllers." Engineering Applications of Artificial Intelligence. vol. 19, no. 8, pp. 829–841, 2006.
12. **Donald G. Bailey,** "Design for Embedded Image Processing on FPGAs", New Zealand, John Wiley & Sons , 2011.
13. **W. Xie.** "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.