International Journal of Hybrid Information Technology
Vol. 9, No.8 (2016), pp. 243-264
http://dx.doi.org/10.14257/ijhit.2016.9.8.22

Design FPGA-Based Fuzzification Algorithm for Model-free Control Techniques

Farzin Piltan, Maryam Rahmani,Omid Mahmoudi, Meysam Esmaeili, Mohammad Ali Tayebi, Mahsa Piltan and Hamid Cheraghi

Intelligent Systems and Robotics Lab, Iranian Institute of Advanced Science and Technology (IRAN SSP), Shiraz/Iran
Email: piltan_f@iranssp.org, WWW.IRANSSP.ORG/ENGLISH

Abstract

Many of fuzzy control applications require real-time operation; higher density programmable logic devices such as field programmable gate array (FPGA) can be used to integrate large amounts of logic in a single IC. This work, proposes a developed method to fuzzifier algorithm with optimal-tunable gains method-using FPGA. The maximum frequency in FPGA-based design is about 72.4 MHz and the delay time in this design is about 13.78 ns. It is observed that this algorithm is able to make as a fast response at 13.78 ns clock period with 72.4 MHZ of a maximum frequency and 2.1 ns for minimum input arrival time after clock. From investigation and synthesis summary, 24.3 ns for maximum input arrival time after clock with 13.9 MHZ frequencies, this design has 13.78 ns delays for each controller to 46 logic elements and the offset before CLOCK is 82.1 ns.

Keywords: *real-time operation, Field Programmable Gate Array (FPGA), improved partly sliding mode control, PUMA robot manipulator, VHDL, Xilinx, sampling time*

1. Introduction

According to classical nonlinear control theory such as sliding mode controller or computed torque controller, these types of controllers are worked based on manipulator dynamic model. Based on equivalent part in conventional nonlinear controllers, in complex and highly nonlinear systems these controllers have many problems for accurate response. Conventional nonlinear controllers need to have accurate knowledge of dynamic formulation of system and it is one of the main challenges. In recent years, artificial intelligence theory has been used in control of nonlinear systems. In most of research papers, neural network, fuzzy logic and neuro-fuzzy are used in nonlinear, time variant and uncertain systems (*e.g.,* robot manipulator) for system identification and control. Fuzzy logic controller (FLC) is one of the most important applications of fuzzy logic theory. This controller can be used to control of nonlinear, uncertain, and noisy systems. After the invention of fuzzy logic theory in 1965 by Zadeh, this theory was used in wide range applications such as control theory or system modeling. The nonlinear dynamic formulation problem in highly nonlinear system (*e.g.,* robot manipulator) can be solved by fuzzy logic theorem. Fuzzy logic theory is used to estimate the system dynamics. This type of controller is free of mathematical dynamic parameters of plant.

The foundation and introduction of fuzzy logic theory is discussed in this section [1-2]. Supposed that U is the universe of discourse for example in this research error, change of error and torque are the sample of universe of discourse and x is the element of U it means that if error is positive and small or if change of error to be defined as positive and small, all these part of the elements of error and change of error. A crisp set to be defined as a set such as error or change of error which consists of different elements (x) such as

ISSN: 1738-9968 IJHIT

negative and small in error will all or no membership in a defined set. In crisp set the membership degree is zero or one such as logical systems. A crisp set (C) in a universe of discourse U is defined by a following membership function ($\boldsymbol{\mu_C}$);

$$\boldsymbol{\mu_C : U \rightarrow \{0, 1\}} \quad \textbf{(1)}$$

A fuzzy set is a set such as error or change of error that different element has different membership grade between zero to one. A fuzzy set (F) in a universe of discourse U is defined by a following membership function ($\boldsymbol{\mu_F}$);

$$\boldsymbol{\mu_F : U \rightarrow [0, 1]} \quad \textbf{(2)}$$

The membership function $(\mu_F(x))$ of fuzzy set F has value between zero to one which each element of universe of discourse$\boldsymbol{U}$ mapped a values between zero to one and this value is called membership degree**.** Figure 1 shows the fuzzy membership function.

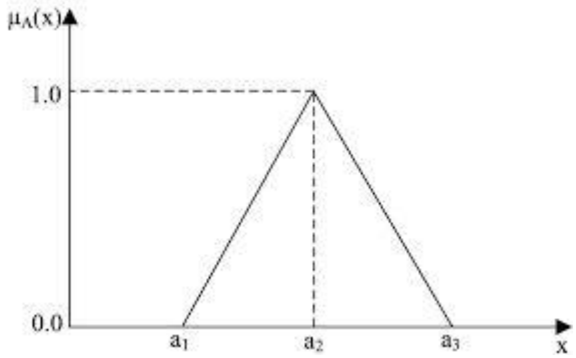

Figure 1. Fuzzy Membership Functions [1]

If the membership functions value $\mu_C(x)$ equal to zero or one it is a crisp set. Based on literature numerical membership function and functional membership function are two important technique to define fuzzy membership function. In numerical membership function the numerator of fraction shows the degrees of membership function and the denominator of fraction shows the universe of discourses point. In functional membership function standard functions in fuzzy sets is used to define membership function and membership degrees. According to the literature the famous functional membership function in practical applications are:

- triangular function
- trapezoidal form
- Gaussian form

According to the literature the formulation of Trapezoidal membership function is;

$$\boldsymbol{\mu_{F(x)}} = \begin{cases} \mathbf{0}, & (x < a) or\ (x > d) \\ \dfrac{\boldsymbol{x-a}}{\boldsymbol{b-a}}, & a \le x < b \\ \dfrac{\boldsymbol{d-x}}{\boldsymbol{d-c}}, & c \le x < d \\ \mathbf{1}, & b \le x \le c \end{cases} \quad \textbf{(3)}$$

Figure 2 shows the Trapezoidal membership function.

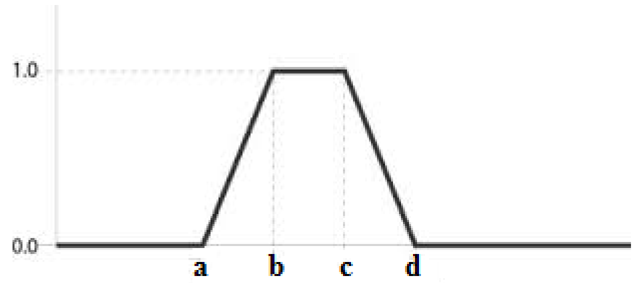

Figure 2. Trapezoidal Membership Functions

The formulation of Triangular membership function is;

$$\mu_{F(x)} = \begin{cases} \mathbf{0}, & x < a \\ \dfrac{\boldsymbol{x-a}}{\boldsymbol{m-a}}, & a \le x < m \\ \dfrac{\boldsymbol{b-x}}{\boldsymbol{b-m}}, & m \le x \le b \\ \mathbf{0}, & x > c \end{cases} \quad \textbf{(4)}$$

Figure 3 shows the Triangular membership function.

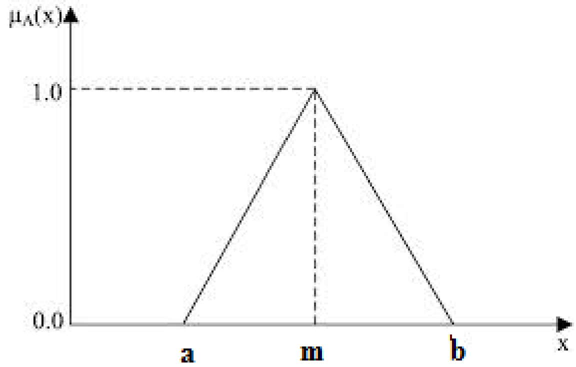

Figure 3. Triangular Membership Functions

Based on literature the Gaussian membership function is formulated by:

$$\mu_{F(x)} = e^{\frac{-(x-m)^2}{2K^2}} \quad \textbf{(5)}$$

Figure 4 shows the Gaussian membership function.

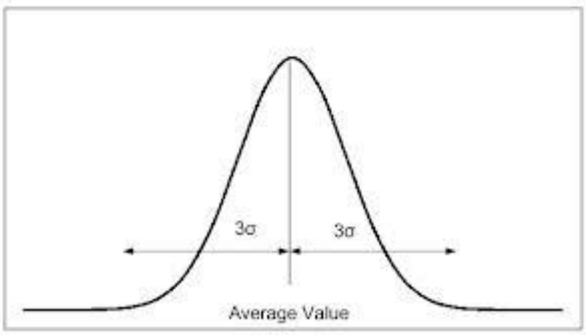

Figure 4. Gaussian Membership Functions

The variable in crisp set is numerical variable but in fuzzy logic theory the variables are linguistic variable. This item is used to describe the values of universe of discourse such as error by any words and sentences. Linguistic variable opens a window to application of fuzzy logic theory in many applications. In a fuzzy logic theory all numerical variables are replaced by words or sentences. For example various words of linguistic variables for any inputs or outputs are: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Figure 5 shows the membership function and linguistic variables in this graph the universe of discourse is error, the membership function is triangular and the linguistic variables are Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Big (PB).

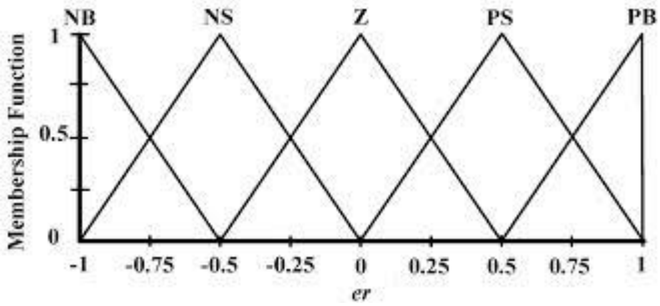

Figure 5. Membership Function and Linguistic Variables

According above discussion, fuzzy set used a membership function to support the membership value of its elements. Universe of discourse is defined by several linguistic variable and different membership functions. To define the behaviour of fuzzy logic set rule base is played important role. If-then rule is the main part to design rule base and rule table. If-then rule statements are used to formulate the condition statements in fuzzy logic theory. A fuzzy if-then rule is divided into two main groups:

- single fuzzy if-then rule
- Multiple fuzzy if-then rule

According to the literature if-then rule has two main parts:

- Antecedent part
- Consequent part

In single fuzzy if-then rule the antecedent part has one variable and it has only one part, following rule shows the single fuzzy if-then rule:

$$\boldsymbol{FR}: \boldsymbol{If\ e\ is\ NB\ Then\ U\ is\ PB} \quad \textbf{(6)}$$

In (6) e is input variable which in design controllers defined by fuzzy logic input, U is output variable, NB and PB are the Linguistic variables that can be defined by fuzzy set, the part of "$e\ is\ NB$" is called the antecedent part and the part of "$U\ is\ PB$" is called the Consequent or Conclusion part. In most of fuzzy controllers antecedent part has multiple parts; the following rule shows the multiple antecedent parts:

$$\boldsymbol{FR}: \boldsymbol{If\ e\ is\ NB\ and\ \dot{e}\ is\ ML\ then\ T\ is\ PB} \quad \textbf{(7)}$$

In (7) $e\ and\ \dot{e}$ are inputs, T is output, Negative Big (NB), Medium Left (ML) and Positive Big (PB) are the linguistic variables, the part of "$\boldsymbol{If\ e\ is\ NB\ and\ \dot{e}\ is\ ML}$" is antecedent part and it multiple, in this state the fuzzy operations (AND/OR) is used and the part of "$\boldsymbol{then\ T\ is\ PB}$" is the consequent part. In most of fuzzy logic controllers, fuzzy controller inputs are used in antecedent part and the output of controller defined by consequent part.

The common method to design a fuzzy controller is to realize it as a computer program. However, most of fuzzy control applications require real time operations with high speed constrains. Therefore, the common method can not be considered as a suitable solution for these type applications. Higher density programmable logic devices such as FPGA can be used to integrate large amounts of logic in a single IC. FPGA provide additional flexibility than ASIC, and they can be used with tighter time-to-market schedules. The term programmable highlights the customization of the IC by the user. Many researchs have discussed the design of hardware implementation of fuzzy logic controller. Number of these were specialized in control application [3-9], and were aim to get better control responses. These researches have concerned using new techniques in fuzzy control, in order to get higher processing speed versus low utilization of chip resource.

This paper is organized as follows; section 2, has served as an introduction to the system dynamics and kinematics and intro to fuzzy logic controller. Part 3, introduces and describes the methodology based on FPGA. Section 4 presents the simulation results and discussion of this algorithm applied to a robot manipulator and the final section is describing the conclusion.

2. Theory

The equation of an *n-DOF* robot manipulator governed by the following equation:

$$M(q)\ddot{q} + N(q,\dot{q}) = \tau \qquad (8)$$

Where τ is actuation torque, M (q) is a symmetric and positive define inertia matrix, $N(q,\dot{q})$ is the vector of nonlinearity term. This robot manipulator dynamic equation can also be written in a following form:

$$\tau = M(q)\ddot{q} + B(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 + G(q) \qquad (9)$$

Where B(q) is the matrix of coriolios torques, C(q) is the matrix of centrifugal torques, and G(q) is the vector of gravity force. The dynamic terms in equation (2) are only manipulator position. This is a decoupled system with simple second order linear differential dynamics. In other words, the component $\ddot{q}$ influences, with a double integrator relationship, only the joint variableq_i, independently of the motion of the other joints. Therefore, the angular acceleration is found as to be:

$$\ddot{q} = M^{-1}(q).\{\tau - N(q,\dot{q})\} \qquad (10)$$

This technique is very attractive from a control point of view.

The data collection to implement the single joint robot arm is the relationship between the rate of torque and end-effector position. Regarding to this information Figure 6 shows the system dynamic behavior.

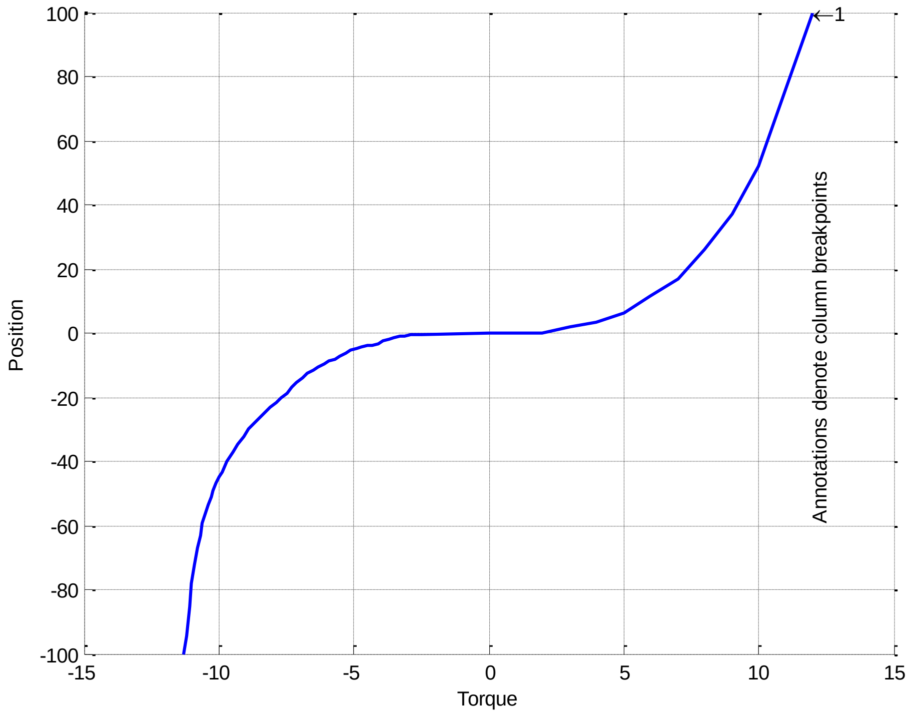

Figure 6. Single Joint Robot Arm System's Lookup Table

A robot manipulator consider as a nonlinear and MIMO system with outputs available measurement and possibility to input change. According to design sliding mode controller, this controller can be used as a controller when it has a satisfying result and system is easy to implement but if some information about robot manipulator operation or controller is available and it can be formulated as a set of rules and fuzzy controller is used to control of robot manipulator. However the application of fuzzy logic controller is really wide, all types of fuzzy logic controllers consists of the following parts;

- Choosing inputs
- Scaling inputs
- Input fuzzification (binary-to-fuzzy[B/F]conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B]conversion)
- Scaling output

Figure 7 shows the block diagram of fuzzy controller operation.

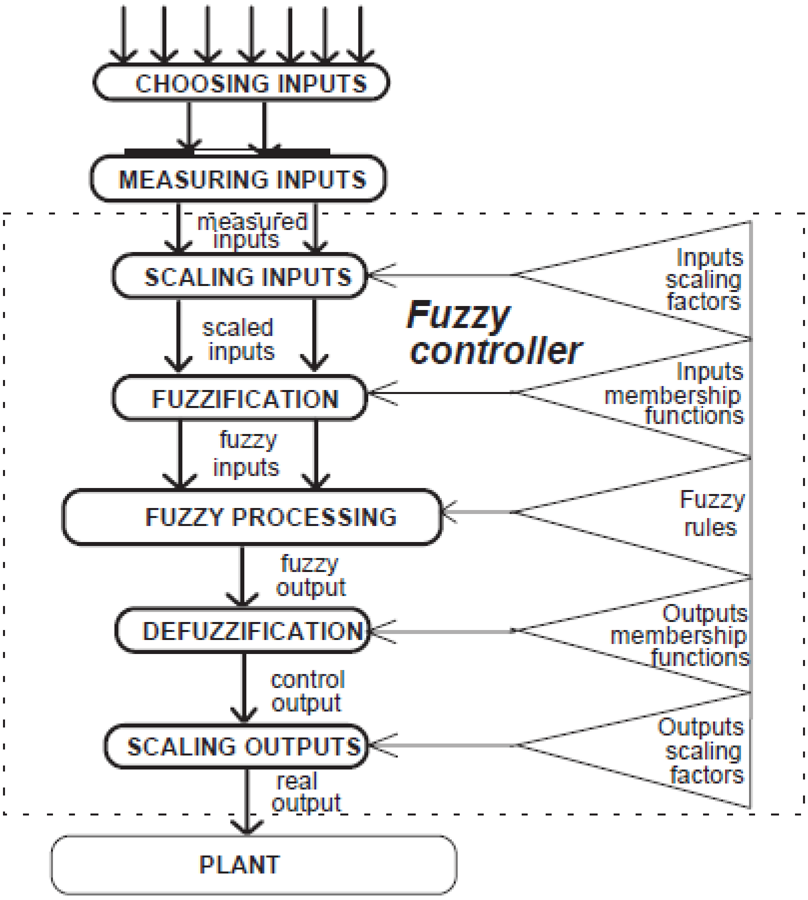

Figure 7. Block diagram of Fuzzy Controller Operation [1]

Define the inputs and control variables: based on controllers' design select the type of inputs is very important to design controller. In most of industrial controllers error and the functional of error are used as inputs to design controller. According to design the linear controller, PI, PD and PID are three types of linear controller. If PI like fuzzy controller is design, error and integral of error are introduced as fuzzy input. In PD like fuzzy controller error and change of error are used to define as controllers' inputs. According to design PID controller, PID like fuzzy controller has three inputs; error, change of error and integral of error. To design fuzzy controller, if one has made a choice of designing a type of PD like fuzzy controller, PI like fuzzy controller or PID like fuzzy controller, this already dealing the choice of process state and control output variables, as well as the content of the antecedent and consequent parts for each rule [2].

Scaling inputs/outputs: in fuzzy logic controller to define membership function, first of all one needs to consider the universe of discourse for all inputs and outputs linguistic variables. If universe of discourse indicates by small range of scaling input or output, the data can be off the scaling. Figure 8 shows the wrong small scaling input/output. Conversely, if universe of discourse indicates by large range of input scaling, the membership function area can be wide on the left or right side if scaling input or output. Figure 9 shows the wrong large scaling input/output. The role of a right choice of scaling factors is obviously shown by the fact that if your choice is bad, the actual operating area of the inputs/outputs will be transformed into a saturation or narrow situation. Input scaling factors have played important role to basic sensitivity of the controller with respect to the optimal choice of the operating areas of the input signals. When the scale output is scaled, the gain updating factor of the controller is scaled. This item affects the closed loop gain and caused to modify the stability and oscillation tendency. Because of its strong impact on stability and reduce the oscillation, this factor is important factor to design fuzzy controller. The right choice of input/output scaling factor shows in Figure 10.

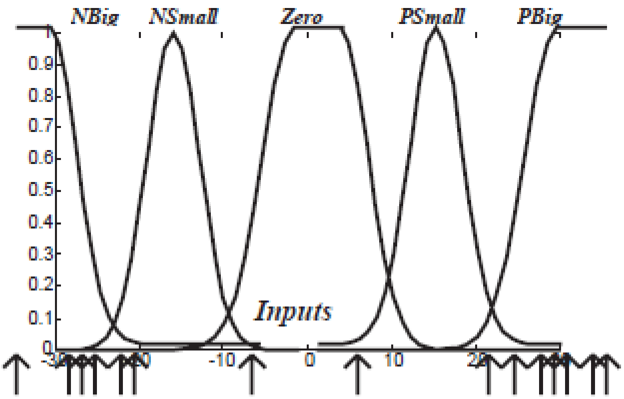

Figure 8. Choice of the Scaling Factor (Small Scaling Factor) [2]

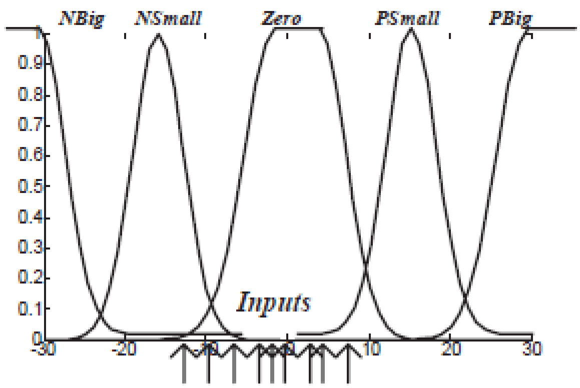

Figure 9. Choice of the Scaling Factor (Large Scaling Factor) [2]

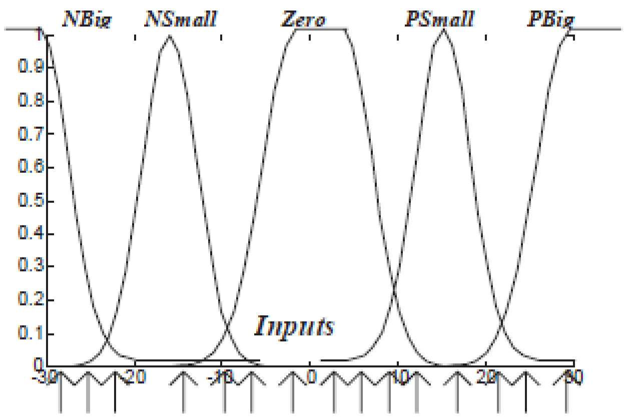

Figure 10. Right Choice of the Scaling factor [1]

Input fuzzification (binary-to-fuzzy [B/F] conversion): fuzzification is used to change the crisp set into fuzzy set. This part is divided into three main parts;

- Linguistic variables

- Scaling factor (normalization factor)
- Inputs membership function

A linguistic variable is a natural language based on the quantity of interest. These variables are words or sentence and this is the main difference between linguistic variable and numerical variable. Linguistic variables can be divided into three sub parts:

- Primary terms, which are the labels of location of the universe of discourse (e.g. $Negative\ or\ Positive\ or\ Zero$).
- Connective terms $AND, OR,$ and NOT.
- Limitation terms such as $Small, Medium\ and\ Big$.

A linguistic variable is defined by;

- Symbolic name of inputs/outputs variables such as $error, change\ of\ error$ and $Torque$.
- Set of linguistic values that can take on some variables such as, Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB).
- Scaling factor as actual physical domain over which the meaning of the linguistic value, for example and based on literature [7] and experience knowledge if error is between -1 and +1.
- Interpretation of linguistic value in terms of the quantities values.
- Select the membership function has a below challenges;
- Select the general parameters , such as the number of membership functions to support all the values of the linguistic variable on the universe of discourse
- The location of membership functions on the universe of discourse
- Width of the membership functions
- Continous parameters, such as the shape of a particular membership function

Figure 11 shows the fuzzification part for PD like fuzzy controller system, this PD like fuzzy controller has two inputs (error and change of error), any input is described with five linguistic values; Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Big (PB), they are quantized into four levels between -2 and +2, and triangular membership functions are used for error and change of error inputs.

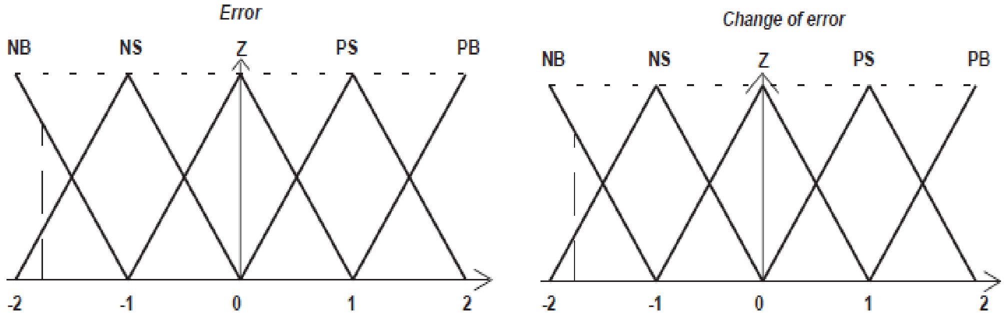

Figure 11. Fuzzification Part in PD like Fuzzy Controller [1]

Fuzzy rule Formulation: the role of the rules in fuzzy logic controller is extremely significant and the main approaches and source of fuzzy logic controller rules are;

- Expert experience and knowledge base
- Learning based on operators' control action
- Identification of fuzzy model system under control action
- The application of learning technique

According to above, the main approach comes from an expert knowledge of system because any fuzzy controller is expert system to solve the control problem. Based on linguistic variables, fuzzy rule base provides a natural theory for human thinking and knowledge base formulation. In practice to find the rule base two method are introduced;

- Design rule base based on redeveloping literature, manuals and research papers
- Design and develop of rule base based on the inquiry operator using questionnaire.

In practice fuzzy rule base divided into main three parts: antecedent part, consequent part and connective terms (AND, OR, NOT). The OR term of two fuzzy set e and $\dot{e}$ is a new fuzzy set which the new membership function is given by

$$S(a,b) = \mu_{e\ OR\ \dot{e}} = \max\{\mu_{\dot{e}(u)}, \mu_{e(u)}\}, \qquad \forall u \in U \tag{11}$$

The AND term of two fuzzy set e and $\dot{e}$ ($T - norm$) is a new fuzzy set which the new membership function is given by

$$T(a,b) = \mu_{e\ AND\ \dot{e}} = \min\{\mu_{e(u)}, \mu_{\dot{e}(u)}\} \tag{12}$$

In fuzzy set and the $NOT\ e$ operation can be replaced by $1 - e$ operation in fuzzy set.

Frequently the rules are formulated one by one base on experience knowledge and any other methods and rule table is design after complete all rule bases. Rule table is the dynamic behavior of fuzzy logic theory and it necessity to have three significant parts;

- Continuous
- Complete
- Consistent

Inference Engine (Fuzzy rule processing): The fuzzy inference engine recommends a fuzzy method to transfer the fuzzy rule base to fuzzy set. Fuzzy rule processing is divided into two main techniques;

- Mamdani method
- Sugeno method

Mamdani anticipated controlling the system by realizing various fuzzy rule bases. In Mamdani controllers' design, in order to improve the control quality, he increased the number of control inputs and used the change of variable error in his design. He designed his controller on the PDP-8 computer. It contained 24 fuzzy rule bases. According to his experiments, the quality of the fuzzy controller based on the 24 fuzzy rules was found to be better than the best result of the fixed conventional controller, as a result fuzzy controller opening a new epoch in a design controller. This type of fuzzy inference is easily understandable by human experts, simple to formulate rules and proposed earlier and commonly used.

Michio Sugeno is changed a part of the rules, in consequent part of fuzzy rule base. According to this method, the consequent part is a mathematical function of the input variables. This type of fuzzy inference is more efficient computationally, more suitable in mathematical analysis, and guarantee the output continuity surface. The following definition shows the Mamdani and Sugeno fuzzy rule base

$$\begin{aligned} &Mamdani \quad F.R^1: if\ e\ is\ NB \quad and \quad \dot{e}\ is\ NB \quad then \quad T\ is\ PB \\ &Sugeno \quad F.R^1: if e\ is\ NB \quad and \quad \dot{e}\ is\ NB \quad then\ 0.3 \times e + 0.6 \times \dot{e} is\ PB \end{aligned} \tag{13}$$

In these two types inference engine the antecedent part are the same but the main difference between these two methods are in consequent part. Fuzzy inference system has four main parts;

- Rule evaluation
- Activation degree
- Aggregation degree
- Defuzzification

Rule evaluation is used to illustrate the fuzzy operation (AND/OR) impact to the antecedent part of the fuzzy rules. The activation degree is the rule of antecedent part's membership degree into consequent part membership degree and the aggregation degree is used to aggregate the consequent part. This part is used to aggregate two neighbouring fuzzy rules and makes a new consequent part. It is note that activation degree is the impact of antecedent part to consequent part and aggregation degree is the impacts of the first modify consequent part in the second one. Several methodologies are used to calculate aggregation degree;

- Max-Min aggregation
- Sum-Min aggregation
- Max-bounded product
- Max-drastic product
- Max-bounded sum
- Max-algebraic sum
- Min-max aggregation

The formulation of Max-min aggregation is

$$\mu_U(x_k, y_k, U) = \mu_{\bigcup_{i=1}^{r} FR^i}(x_k, y_k, U) = \max\left\{\min_{i=1}^{r}\left[\mu_{R_{pq}}(x_k, y_k), \mu_{p_m}(U)\right]\right\} \quad (14)$$

The formulation of Sum-min aggregation is;

$$\mu_U(x_k, y_k, U) = \mu_{\bigcup_{i=1}^{r} FR^i}(x_k, y_k, U) = \sum \min_{i=1}^{r}\left[\mu_{R_{pq}}(x_k, y_k), \mu_{p_m}(U)\right] \quad (15)$$

where r is the number of fuzzy rules activated by x_k and y_k and also $\mu_{\bigcup_{i=1}^{r} FR^i}(x_k, y_k, U)$ is a fuzzy interpretation of i^{th} rule. Defuzzification it is used to convert fuzzy logic set to crisp or logic set. To calculate defuzzification some methods introduced by researchers which the name of two famous methods are: Centre of gravity method (COG) and Centre of area method (COA).

The formulation of COG method is;

$$COG(x_k, y_k) = \frac{\sum_i U_i \sum_{j=1}^{r} . \mu_u(x_k, y_k, U_i)}{\sum_i \sum_{j=1}^{r} . \mu_u(x_k, y_k, U_i)} \quad (16)$$

The formulation of COA method is

$$COA(x_k, y_k) = \frac{\sum_i U_i . \mu_u(x_k, y_k, U_i)}{\sum_i \mu_U . (x_k, y_k, U_i)} \quad (17)$$

According to above two formulations, $COG(x_k, y_k)$ and $COA(x_k, y_k)$ illustrates the value of defuzzification output, U_i is the element of the fuzzy set, $\mu_U(x_k, y_k, U_i)$ is the membership function, and r is the number of fuzzy rule bases.

3. Methodology: Design FPGA-Based Fuzzification Algorithm

In recent years, many researches discussed the implementation techniques of fuzzy logic; these techniques involve two main types, analog and digital Implementation. This section presents an overview on these types with their challenges. The requirements for the hardware implementation are listed below:

- High-speed performance.
- Low complexity, which means that algorithms for fuzzy processing, fuzzification and defuzzification have to be very simple and demand as small amount of memory as possible for their realization.
- High flexibility, which means the ability of the hardware to be used successfully in different applications and configurations.

These conditions contradict each other. So it is not easy to choose the right way, especially if one takes into account some other factors, such as manufacturing cost (very important for consumer product fuzzy controllers) or design cost (important in research and development).

The variables in fuzzy systems are analog by nature. Thus, analog implementation eliminates the need for analog-to-digital and digital-to-analog conversions. The fuzzy systems also require massive parallelism, making analog circuits particularly suited for their implementation. Furthermore, the physical characteristics of transistors can be utilized in realizing the nonlinear functions required, whether it is a fuzzy operation, or a membership function. Analog implementations, however, have typically very restricted possibilities for programmability. Analog implementation techniques include voltage mode and current mode realizations, in addition to mixed mode (current and voltage) realizations. One difficult aspect of analog circuit implementations is devising a reliable analog memory module.

Many hardware description languages (HDL) have become widely used for designing digital architecture from ever and ever highest design levels. Although fuzzy chips may have limited input/output capabilities, they have particularly useful applications in real-time control systems. Analog fuzzy values must be converted to binary digital signals. Analog-to-digital conversion can lead to quantization errors in both input signals and membership values. Thus, decline in the fuzzy processing may occur if an insufficient number of bits are used to represent the analog signals. On the other hand, using a large number of bits can slow down the process. This is the trade-off between precision and speed. Fuzzy dedicated circuits are characterized in the following points [3-9]:

- The number of inputs and outputs.
- The number and shapes of membership functions.
- Inference techniques including operators, consequences, and size of the premises.
- Defuzzification method.
- The number of fuzzy logic inferences per second, FLIPS.
- Physical size.
- Power consumption.
- Software available to support the design.

The cells on the FPGA can be more complex than the simple gates provided on the gate array. The FPGA designer can constructs more complex functions from these cells. The term *field programmable* highlights the customization of the IC by the user. Different FPGA manufacturers have developed different basic cells, seeking to provide the most useful functionality in the cells for generation of overall FPGA functions. Cells range from fine-grained cells consisting of basic gates through medium-grained cells providing more complex programmable functions to large-grained cells. Different FPGA manufacturers also provide different approaches to the programming step. FPGA programming technologies include one-time programming or multiple-time programming capabilities with the programming either nonvolatile (*i.e.,* programming is retained when power is turned off) or volatile (*i.e.,* programming is lost when power is turned off). Some advantages of using FPGA are: it can ensure an error-free design before the permanent ASIC implementation, allows the shortening of the time to correct any design problem and it allows the use of the prototype board for the hardware testing of other system components. Several different types of structures for FPGAs are available commercially, all of them have a basic structure that consists of many logic blocks or logic cells, accompanied by a large number of pre-laid lines for connecting these logic blocks.Special purpose structures are placed into the array depending on the FPGA type, these are configurable RAM blocks and clock distribution elements, , adders, multipliers *etc.* To achieve to these goals, Spartan 3E FPGA is selected. The information of this device is introduced as the following Table 1.

Table 1. Summary of XA Spartan-3E FPGA Attributes

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits(1)	Block RAM bits(1)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XA3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XA3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XA3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	190	77
XA3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XA3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

Notes:

1. By convention, one Kb is equivalent to 1,024 bits.

The following formulation shows the derivative algorithm:

$$d(e) = \frac{Din(t) - Din(t-1)}{\Delta t} = \big(Din(k+1) - Din(k)\big) \times sample\ time \qquad \textbf{(18)}$$

The following Figures (Figure 12 and Figure 13) show the outline and interior view of derivative design in HDL. Regarding to this Figure this system has 40 bits-inputs and 40 bits-output.

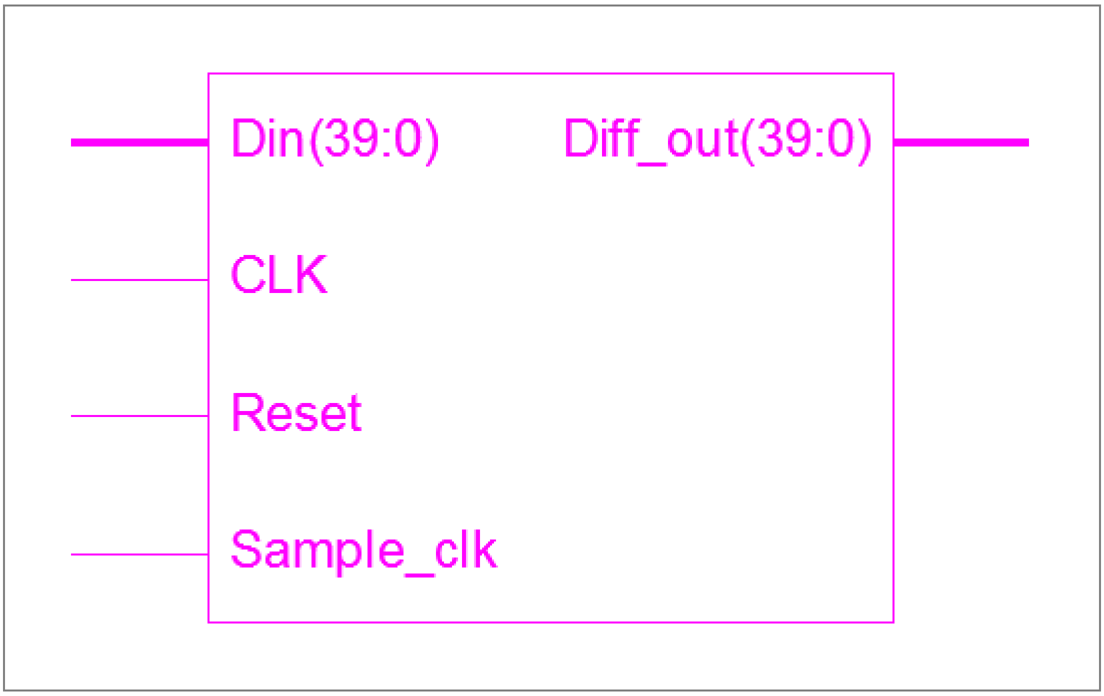

Figure 12. Outline Derivative Algorithm in HDL using Spartan 3E

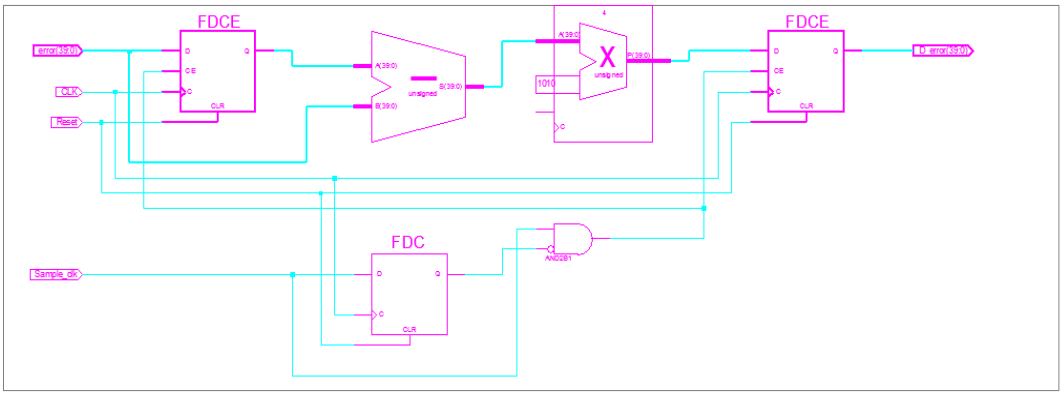

Figure 13. Interior View: Derivative Algorithm in HDL using Spartan 3E

The following Figure (Figure 14) shows the Derivative Program based on VHDL code.

```
entity Derivative_control is
   Port(error : in  STD_LOGIC_VECTOR (39 downto 0);
        D_error : out  STD_LOGIC_VECTOR (39 downto 0);
        Reset : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        Sample_clk : in  STD_LOGIC);
end Derivative_control;

architecture Behavioral of Derivative_control is
-----------------------------------------------------------------------------------------------
constant sample_rate : std_logic_vector(4 downto 0)  := "01010";
signal last_error : std_logic_vector(39 downto 0);
signal data_sample_error : std_logic_vector(39 downto 0);
signal diff_data : std_logic_vector(44 downto 0);
signal last_sample_clk : std_logic;
signal sample_clk_edge : std_logic;
-----------------------------------------------------------------------------------------------
begin
-----------------------------------------------------------------------------------------------
sample_clk_edge <= (not last_sample_clk) and Sample_clk;
process(CLK, Reset)
begin
   if(Reset = '1')then
      last_sample_clk <= '0';
   elsif(rising_edge(CLK))then
      last_sample_clk <= Sample_clk;
   end if;
end process;
-----------------------------------------------------------------------------------------------
process(CLK)
begin
   if(Reset = '1')then
      last_error <= (others => '0');
      D_error <= (others => '0');
   elsif(rising_edge(CLK) and sample_clk_edge = '1')then
```

Figure 14. VHDL Code: Derivative Algorithm in HDL Spartan 3E Device

The Formulation of PD control is as follows:

$$U_{PID} = K_p \times e + K_v\left(\frac{de}{dt}\right) = K_p \times e + K_v \dot{e} \qquad (19)$$

The following Figures (Figure 15 and Figure 16) show the outline and interior view of PD control design in HDL. Regarding to this Figure this system has 30 bits-inputs and 35 bits-output.

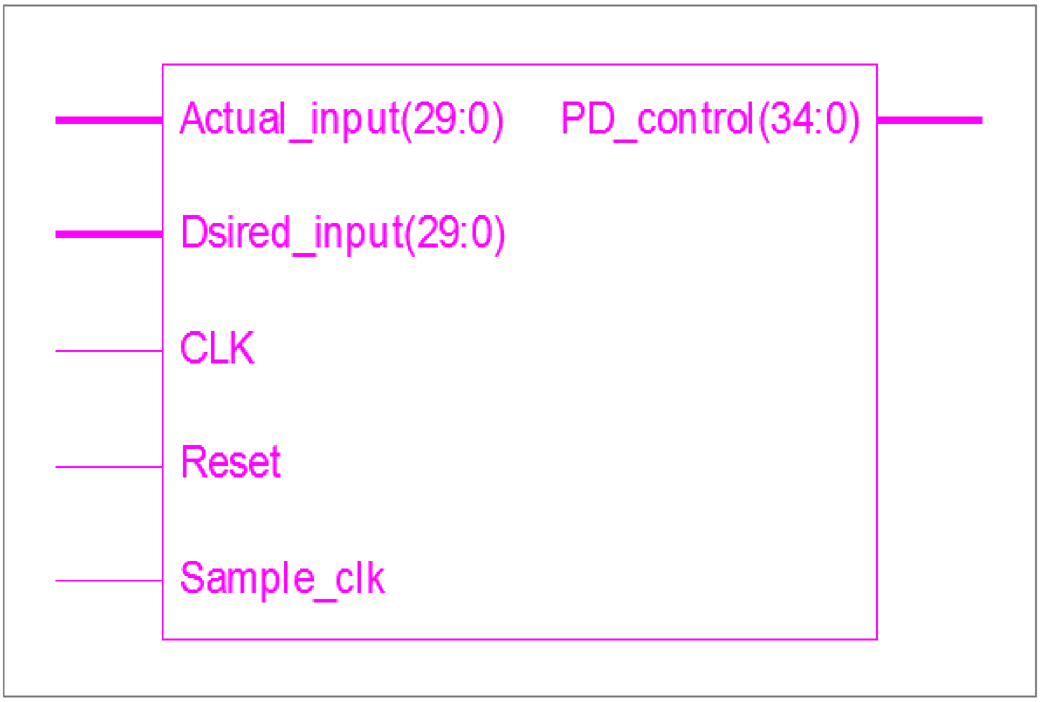

Figure 15. Outline PD Algorithm in HDL Spartan 3E Device

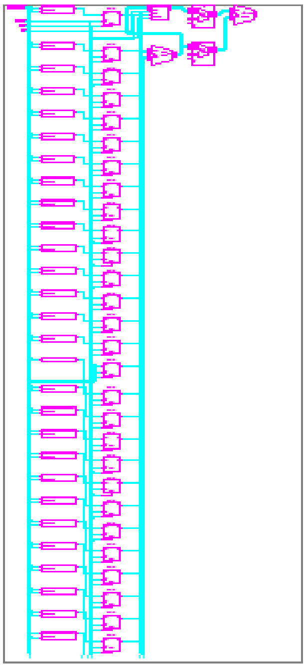

Figure 16. Interior View: PD Algorithm in HDL Spartan 3E Device

The following Figure (Figure 17) shows the PD Program based on VHDL code.

```
entity PD_Controler is
      Generic(Kp : std_logic_vector(7 downto 0)  := B"11111010";
               Kv : std_logic_vector(7 downto 0)  := B"00111100");
      Port(Actual_input : in  STD_LOGIC_VECTOR (29 downto 0);
        Dsired_input : in  STD_LOGIC_VECTOR (29 downto 0);
        PD_control : out  STD_LOGIC_VECTOR (34 downto 0);
        Reset : in  STD_LOGIC;
        CLK : in  STD_LOGIC;
        Sample_clk : in  STD_LOGIC);
end PD_Controler;
-----------------------------------------------------------------------------------------------
architecture Behavioral of PD_Controler is
-----------------------------------------------------------------------------------------------
------Signals definitions
-----------------------------------------------------------------------------------------------
signal Actual_input_buf : std_logic_vector(39 downto 0);
signal Desired_input_buf : std_logic_vector(39 downto 0);
signal Error_gain : std_logic_vector(47 downto 0);
signal Error_diff : std_logic_vector(39 downto 0);
signal Error : std_logic_vector(39 downto 0);
signal Error_diff_gain : std_logic_vector(47 downto 0);
signal PD_control_buf : std_logic_vector(39 downto 0);
-----------------------------------------------------------------------------------------------
------------Numerical Differential calculator
COMPONENT Derivative_control
   PORT(
      error : IN std_logic_vector(39 downto 0);
      Reset : IN std_logic;
      CLK : IN std_logic;
      Sample_clk : IN std_logic;
      D_error : OUT std_logic_vector(39 downto 0)
      );
   END COMPONENT;
```

Figure 17. VHDL Code: PD Algorithm in HDL Spartan 3E Device

Fuzzification is a mathematical procedure to convert an element in the universe of discourse into the membership value of the fuzzy set. This procedure converts controller inputs into information that the inference mechanism can be easily used to activate and apply rules. Suppose that fuzzy set *A* P is defined on [a,b]; that is the universe of discourse is [a,b]; for any *x*_[a,b], the result of fuzzification is simply µAP (*x*). Figure 18 below shows an example in which the fuzzification result for *x* =7 is 0.4.

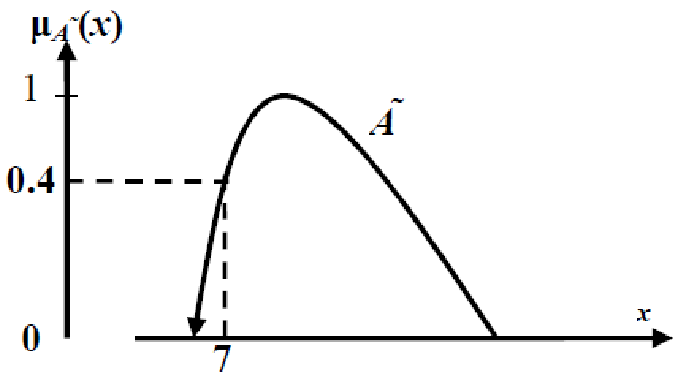

Figure 18. An Example Showing How Fuzzification Works

The overlapping degree (V) in the proposed design is two, which means that at each time instance there are two active, (have nonzero membership values), fuzzy sets for each input variable at maximum. The proposed fuzzification process has been designed using two fuzzifier blocks, one for each input variable. The fuzzifier block implies the fuzzification process by taking the input and produces four output values. These values represent the sequence numbers of the two active fuzzy sets (e1, e2 and de1, de2) and the membership degrees of the variable in each one of them (μe1, μe2 and μde1, μde2). The memory base has designed using ROM, the use of ROM is is better than RAM when the programmability is directly achieved by the implementation techniques (as in the case of FPGA). Figure 19 shows the fuzzifier block.

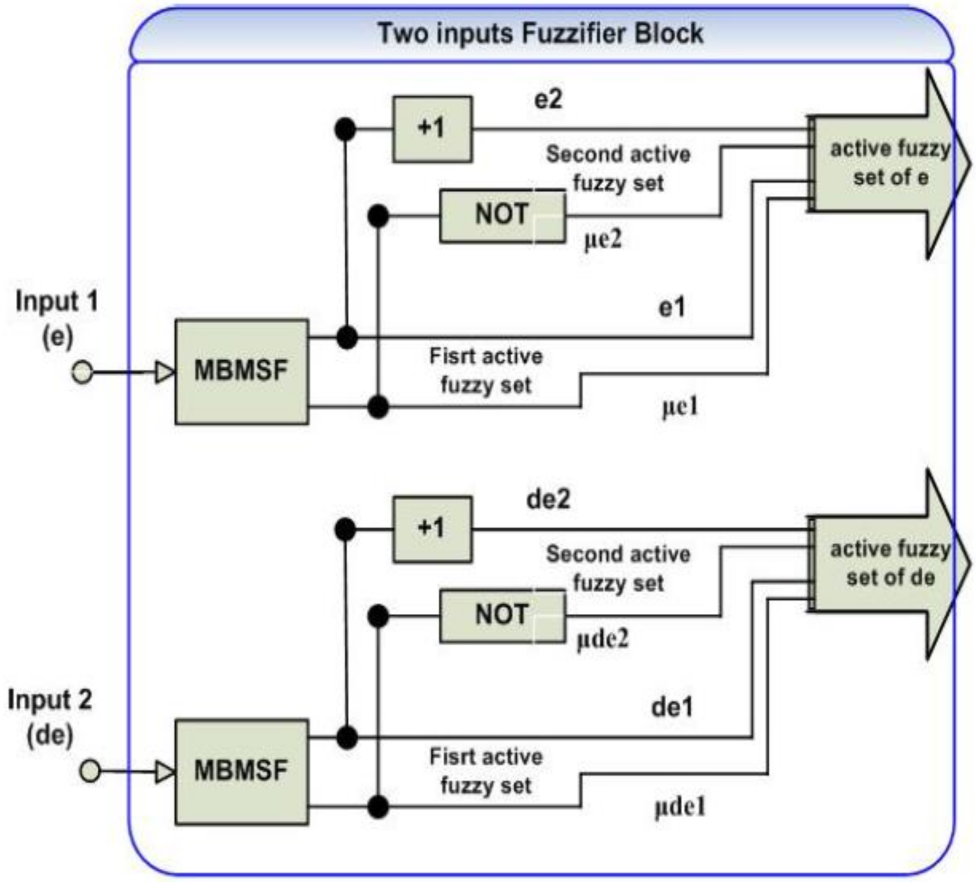

Figure 19. Structure of Fuzzifier Block (NOT=ONE'S Complement)

Using memory based membership functions (MBMSF) allow to use unrestricted fuzzy set shapes. The memory model has impelemented with maximum possible membership value in the proposed design, where the maximum coded in p values is ($2^p - 1$), where p=4 bits in 6-bits version of PIDFC, and p=6 bits in 8-bits version of PIDFC. This dictates that the summation of membership values of two consecutive fuzzy set is always equal to $(2^p - 1)$. Each word in the MBMSF is divided into two parts. The first part is represents the sequence number of the active fuzzy set (3-bits, in both versions). Assigning 3 bits for the sequence number of the fuzzy set, give the controller flexibility to accept for each input variable up-to 8 fuzzy sets. The second part of memory word is p bits data word that represents the membership value of input in the active fuzzy set. The total memory length at each input is equal to (2_q).

4. Result

Timing Report: FPGA-based algorithm reduce the process time as well improve stability and flexibility. Figure 20 shows the actual and desired input, and torque performance in transient state. Regarding to this Figure however actual and desired inputs equal to zero but torque performance has fluctuations in first 10 ns.

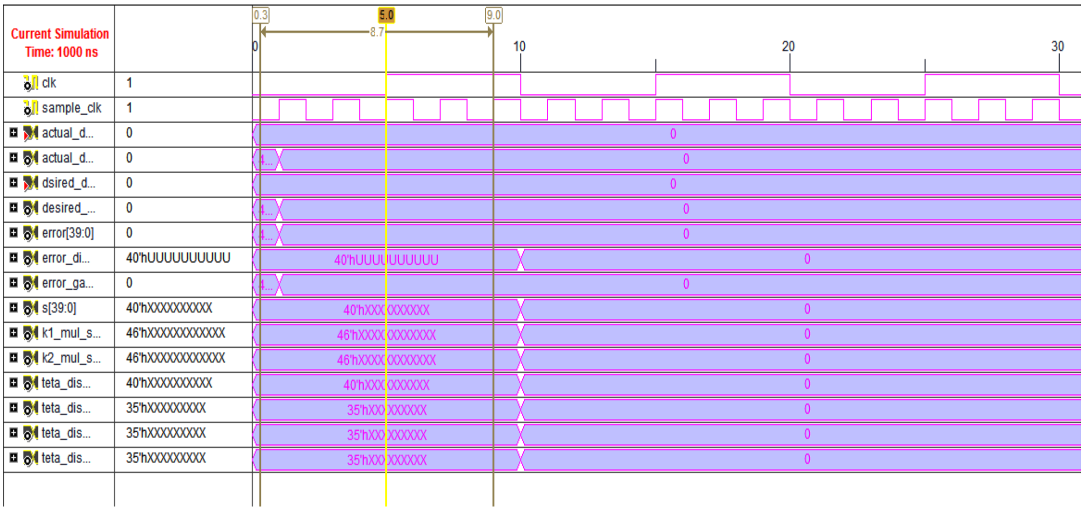

Figure 20. FPGA-Based Algorithm in Fist 10 ns

Regarding to Figure 21, the output performance between 100 to 130 ns is equal to zero. In this time controller is inactive, this time is the controller's delay. The next 20 ns $(130 - 150\ ns)$ illustrate the desired position is 50 degrees and improvement the actual position from 0 $degrees$ to 43 $degrees$. Regarding to the Figure 21 the error is about 7 $degrees$.

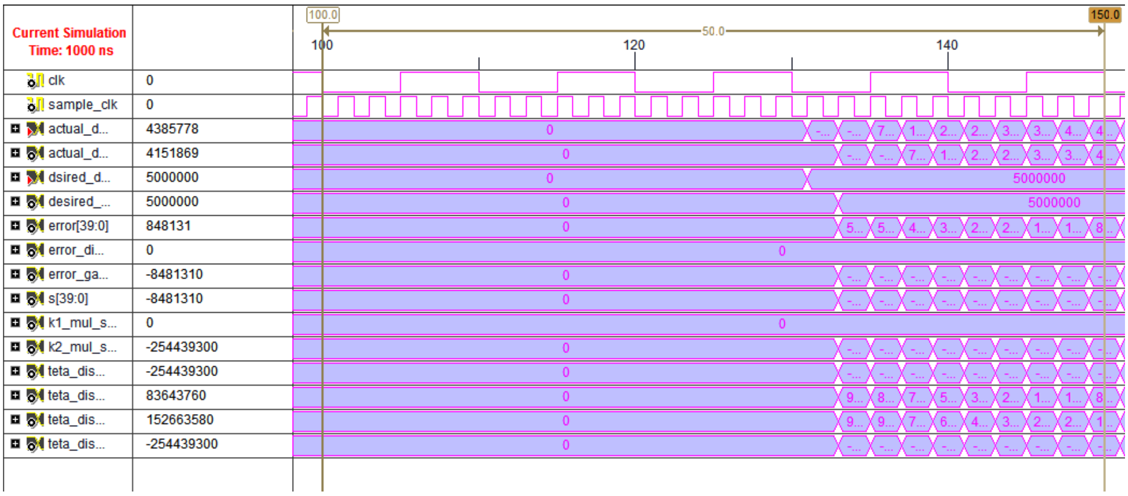

Figure 21. FPGA-Based Algorithm Result Between 100-150 ns

5. Conclusion

Higher execution speed versus small chip size is achieved by designing fuzzifier algorithm with simplified structure as parallel structure, and also by designing PDFC by accumulating the output of the PDFC. These methods reduce the number of rules needed significantly. And also enables the controller to work depending on two external signals to provide high- flexibilities with different applications. The controller needs 16 clock cycles to generate an output with maximum clock frequency of 40 MHz. Therefore, the proposed controller will be able to control a wide range of the systems with high sampling rate.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments. This work was supported by the Iranian Institute of Advanced Science and Technology, under grant no. 2014-Persian Gulf-DIG.

References

[1] L. Reznik, Fuzzy controllers: Butterworth-Heinemann, **(1997)**.

[2] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control", International Journal of Computational Intelligence, vol. 3, no. 4, pp. 303–311, **(2006)**.

[3] N. Sulaiman, Z. A. Obaid, M. Marhaban and M. Hamidon, "Design and Implementation of FPGA-Based Systems-A Review", Australian Journal of Basic and Applied Sciences, vol. 3, no. 4, pp. 3575-3596, **(2009)**.

[4] X. Shao and D. Sun, "Development of an FPGA-based motion control ASIC for robotic manipulators", The Sixth World Congress on Intelligent Control and Automation, **(2006)**, pp. 8221-8225.

[5] S. Namvarrechi, S. Soltani, I. Nazari, A. Roshanzamir and A. Jahed, "Design Single Chip Micro based controller for First Order Delay Systems", International Journal of Hybrid Information Technology,http://dx.doi.org/10.14257/ijhit.2016.9.5.07, vol. 9, no. 5, pp. 71-98, **(2016)**.

[6] **F. Piltan,** O. Avatefipour, S. Soltani, O. Mahmoudi, M. Reza S. Nasrabad, M. Eram, Z. Esmaeili, S. Heidari, K. Heidari and M. Mahidi Ebrahimi, "Design FPGA-Based CL-Minimum Control Unit", International Journal of Hybrid Information Technology,http://dx.doi.org/10.14257/ijhit.2016.9.1.10, vol. 9, no.1, **(2016)**, pp. 101-118.

[7] F. Piltan, N. Sulaiman, M. H. Marhaban, A. Nowzary and M. Tohidian, "Design of FPGA-based Sliding Mode Controller for Robot Manipulator", International Journal of Robotic and Automation, vol. 2, no. 3, **(2011)**, pp. 183-204.

[8] F. Piltan, N. Sulaiman, A. Jalali and K. Aslansefat, "Evolutionary Design of Mathematical tunable FPGA Based MIMO Fuzzy Estimator Sliding Mode Based Lyapunov Algorithm: Applied to Robot Manipulator", International Journal of Robotics and Automation, vol. 2, no. 5, **(2011)**, pp. 317-343.

[9] F. Piltan, I. Nazari, S. Siamak and P. Ferdosali, "Methodology of FPGA-Based Mathematical error-Based Tuning Sliding Mode Controller", International Journal of Control and Automation, vol. 5, no. 1, **(2012)**, pp. 89-118.

Authors

Farzin Piltan, He is an outstanding scientist in the field of Electronics and Control engineering with expertise in the areas of nonlinear systems, robotics, and microelectronic control. Mr. Piltan is an advanced degree holder in his field. Currently, Mr. Piltan is the Head of Mechatronics, Intelligent System, and Robotics Laboratory at the Iranian Institute of Advanced Science and Technology (IRAN SSP). Mr. Piltan led several high impact projects involving more than 150 researchers from countries around the world including Iran, Finland, Italy, Germany, South Korea, Australia, and the United States. Mr. Piltan has authored or co-authored more than 140 papers in academic journals, conference papers and book chapters. His papers have been cited at least 3900 times by independent and dependent researchers from around the world including Iran, Algeria, Pakistan, India, China, Malaysia, Egypt, Columbia, Canada, United Kingdom, Turkey, Taiwan, Japan, South Korea, Italy, France, Thailand, Brazil and more. Moreover, Mr. Piltan has peer-reviewed at least 23 manuscripts for respected international journals in his field. Mr. Piltan will also serve as a technical committee member of the upcoming EECSI 2015 Conference in Indonesia. Mr. Piltan has served as an editorial board member or journal reviewer of several international journals in his field as follows: International Journal Of Control And Automation (IJCA), Australia, ISSN: 2005-4297, International Journal of Intelligent System and Applications (IJISA), Hong Kong, ISSN:2074-9058, IAES International Journal Of Robotics And Automation, Malaysia, ISSN:2089-4856, International Journal of Reconfigurable and Embedded Systems, Malaysia, ISSN:2089-4864.

Mr. Piltan has acquired a formidable repertoire of knowledge and skills and established himself as one of the leading young scientists in his field. Specifically, he has accrued expertise in the design and implementation of intelligent controls in nonlinear systems. Mr. Piltan has employed his remarkable expertise in these areas to make outstanding contributions as detailed follows:Nonlinear control for industrial robot manipulator (2010-IRAN SSP), Intelligent Tuning The Rate Of Fuel Ratio In Internal Combustion Engine (2011-IRANSSP), Design High Precision and Fast Dynamic Controller For Multi-Degrees Of Freedom Actuator (2013-IRANSSP), Research on Full Digital Control for Nonlinear Systems (2011-IRANSSP**),** Micro-Electronic Based Intelligent Nonlinear Controller (2015-IRANSSP), Active Robot Controller for Dental Automation **(**2015-IRANSSP), Design a Micro-Electronic Based Nonlinear Controller for First Order Delay System (2015-IRANSSP).

The above original accomplishments clearly demonstrate that Mr. Piltan has performed original research and that he has gained a distinguished reputation as an outstanding scientist in the field of electronics and control engineering. Mr. Piltan has a tremendous and unique set of skills, knowledge and background for his current and future work. He possesses a rare combination of academic knowledge and practical skills that are highly valuable for his work. In 2011, he published 28 first author papers, which constitute about 30% of papers published by the Department of Electrical and Electronic Engineering at University Putra Malaysia. Additionally, his 28 papers represent about 6.25% and 4.13% of all control and system papers published in Malaysia and Iran, respectively, in 2011.

Maryam Rahmani, She is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. She has been working at “Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “Pre-Master Student Researcher” of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. She has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. She has been working at “Research of Full Digital Control for Nonlinear System for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “research assistant scholar“ of a research team composed of 27 researchers since Feb. 2014 to date. Her current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Omid Mahmoudi, He was research assistant at Institute of Advanced Science and Technology, Research Center, IRAN SSP from Feb. 2014 to March. 2015. He was research assistant of team (9 researchers) to design a Micro-electronic Based nonlinear controller since Feb, 2014 to March, 2015 and research student (34 researchers) to Filtering the hand tremors in flexible surgical robot for Experimental Research and Education since Jan, 2013 to Feb, 2014, and published 5 journal papers since 2014 to date. His current research interests are nonlinear control, artificial control system, Microelectronic Device, and HDL design.

Meysam Esmaeili, He is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at “Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “Pre-PhD Student Researcher“ of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at “Research of Full Digital Control for Nonlinear System for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “research assistant scholar“ of a research team composed of 27 researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Mohammad Ali Tayebi, He is currently research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at “Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “Pre-PhD Student Researcher“of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at “Research of Full Digital Control for Nonlinear System for Experimental Research and Education” project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as “research assistant scholar“ of a research team composed of 27 researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Mahsa Piltan, She is currently a research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. She has been working at "DesignHigh Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-PhD Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. She has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. She has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. Her current research interests are nonlinear control, artificial control system and design FPGA-based controller.

Hamid Cheraghi, He is currently a research assistant at Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. He has been working at "Design High Precision and Fast Dynamic Controller for Multi-Degrees of Freedom Actuator for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "Pre-Master Student Researcher" of a research team composed of 21 researchers since Feb. 2013 to Feb. 2014. He has had the main roles in initiation and development of this project which has resulted thus far in three scientific publications up to now. He has been working at "Research of Full Digital Control for Nonlinear System for Experimental Research and Education" project at Iranian Institute of Advance Science and Technology, Sanaat Kade Sabz Passargad Research Center (IRAN SSP) as "research assistant scholar" of a research team composed of 27 researchers since Feb. 2014 to date. His current research interests are nonlinear control, artificial control system and design FPGA-based controller.