Volume 4, Issue 12, December 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Speed Control of DC Motor for Robotic Application Using Fuzzy Controller on FPGA Platform

Narendra M. Pathade, Prof. Sania M. Ansari
Department of ExTC & Pune University,
Maharashtra, India

Abstract- Design of a Fuzzy Logic Controller (FLC) requires more design decisions than usual, for example rule base, inference engine, de-fuzzification, and data pre- and post processing. There are three parts to fuzzy controller, the fuzzification of the inputs, the de-fuzzification of the outputs, and the rule base. The controller that is implemented here demonstrates a 1-input, 1-output fuzzy controller with 5-membership functions. Here a VHDL-based design and synthesis approach is used for designing which has considerably reduced the design time. A complete description of the controller A fuzzier, de-fuzzifier and a rule base are written in VHDL and are logically integrated which is further synthesized using logic synthesis tool XILINX ISE9.2i software. The efficiency of the implementation is verified using FPGA. In this paper, the concept of Fuzzy Logic is used to control the speed of a simple DC motor. The hardware descriptive language used for the purpose of design of the Fuzzy Logic Controller is VHSIC Hardware Descriptive Language (VHDL) and is implemented on SPARTAN-3 xc3s500e-4fg320 FPGA board. This method of speed control of a dc motor may represent an ideal application for introducing the concepts of fuzzy logic. Here a sincere effort is made to show, how a fuzzy controller can be used to control the motor speed, which represents a very practical class of engineering problems.

Keywords - FLC, VHDL FPGA, DC motor, Hardware Implementation.

I. INTRODUCTION

The past few years have witnessed a rapid growth in the number and variety of application of fuzzy logic. The application ranges from consumer products such as cameras, washing machines, cars and in industry for medical instrumentation, underground trains and robots. Unlike the conventional controller FLC design is not based on the mathematical model of the plant or system. A FLC is an automatic controller that controls an object in accordance with desire behaviour. For a complex system whose mathematical model is very difficult to define or the transfer function of a plant is undefined, fuzzy logic controllers are very useful in that case [1,3]. The control action of FLC is defined in terms of simple human friendly "if – then rules". These set of rules are describe the system behaviour. These set of rules are called the knowledge base of fuzzy controller. We can easily change the rules accordance with our desire output. So the development time for a new controller can be significantly reduced as compared to conventional one [7].

The motivation behind the implementation of a FLC in VHDL was driven by the need for an inexpensive hardware implementation of a generic fuzzy controller for use in industrial and commercial applications [13]. We have taken a simple FLC for an armature control DC motor speed control. Error and change in error in speed has been used as two inputs to FLC. For both the inputs 5 triangular membership function has been selected and coded in VHDL. An algorithm has been developed in VHDL to fuzzifie the crisp digital values of speed error and rate of change of error. Sugeno type FLC structure has been used to obtain the controlled output. The controller algorithm developed synthesized, simulated and implemented on FPGA Spartan 3E xc3s500e-4fg320 board. The FLC has been design using system generator approach.

II. IMPLEMENTATION OF FLC

The widely used method for Fuzzy Logic is Mamdani style because of its simplicity and its compatibility with human made decision. Although Mamdani Style is widely used, it is not suitable for this implementation. This is because, the Mamdani method requires finding the centroid of a two-dimensional shape by integrating across a continuously varying function. This method is not computationally efficient. It is also not suitable because its output use triangle or trapezoidal membership function to describe the output and its defuzzification calculation is complicated to be done in VHDL despite the results are not necessary effective or better[7]. Sugeno style is much more suitable to be used in this situation. This is because, the output of this method, uses a single spike, singletons to describe the outputs where there is a unity at single particular point and zero elsewhere. This causes all the output of each fuzzy rule to be constant [8].

1. Fuzzification

The first component in the FLC is the fuzzifier that converts crisp inputs into a set of membership values in the interval [0,1] in the corresponding fuzzy sets. In this paper, triangular membership functions (triangular membership function being a special case of the trapezoidal function) are used for two inputs Error and change in error in speed [8]. Each of

inputs is represented by 5 membership functions, which are NB (Negative Big), NM (Negative Medium), ZE (Zero), PM (Positive Medium), PB (Positive Big) as shown in Fig.1

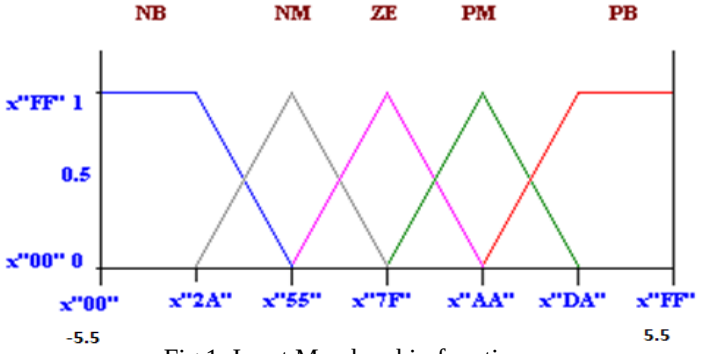

Fig.1. Input Membership functions

In fuzzification, the degree of membership function is determined. This is done by locating the location of the input in the membership function and determines the degree of the membership function. The degree of the membership function is from 0 to 1 where in this paper, it is represented by X00 and XFF ("X" sign indicates hexadecimal number representation).

In VHDL, Each trapezoidal membership function is defined by two points and two slopes values. The entire membership function can be divided into three segments: I, II and III as shown in Fig4. The Y axis shows the degree of membership function (μ) as a value between 0 and 1. The X axis shows the universe of discourse and is divided into three segments. The degree of membership depends on the location of the input value with reference to these segments. Fig2 shows how trapezoidal input membership functions are formed in the fuzzification process [8].

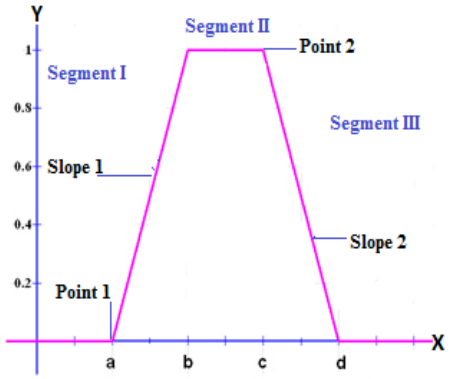

Fig.2. Trapezoidal Type Membership Function

Declaration of membership functions in VHDL as follows:

type input is (term, none);
type mfs is record linguistic: input;
point1: std_logic_vector (7 downto 0);
slope1: std_logic_vector (7 downto 0);
point2: std_logic_vector (7 downto 0);
slope2: std_logic_vector (7 downto 0);
end record;
type mfs_functions is array((natural range <>) of mfs;
constant linguistic_name: mfs_functions:=
((linguistic => term, point1=> x"04", slope1 => x"7F", point2 => x"09", slope2 => x"55"),
(linguistic => none, point1=> x"FF" , slope1=> x"FF", point2 => x"FF", slope2 => x"FF"));

The following algorithm illustrates the procedure of this process:

Fuzzification ();
Set n = number of membership function
μ = array of degree of membership function
m = array of membership function
start loop for i = 1 to n
if input value < m[i].point1 then μ[i]=0;
else if input value < m[i].point2 then μ[i]=(input value- m[i].point1) × m[i].slope1;
else μ[i]= 255- (input value – m[i].point2) ×m[i].slope2;
end if;
end loop;

2. Rule Inference

The degree of membership is determined in the fuzzification stage. The next step is to create rules to decide what action should be taken in response to the given set of degree of membership function. The "AND" and "OR" fuzzy operators are best used for rules with multiple antecedents. The fuzzy operator, "OR" is used to evaluate the disjunction of the rules antecedents and "AND" is used to evaluate the conjunction of the rules antecedents. "AND" fuzzy operator is since it is required to evaluate the conjunction of the rules antecedents. Since "AND" is the minimum operation between multiple antecedents, the minimum function is used. The "OR" fuzzy operator also can be used when more than one rules involved with the same output. The rule base of system is defined in Table 2[12].

Table1 -Fuzzy final rules

ECE	**NB**	**NM**	**ZE**	**PM**	**PB**
PB	NM	NS	NB	PB	PB
PM	NM	NM	NB	PB	PB
ZE	NB	NB	ZE	PB	PB
NM	NB	NB	NB	PM	PM
NB	NB	NB	NB	PS	PM

3. Rule Evaluation

Rule: position(1)<=maximum(minimum (u1(0), u2(1)),
minimum (u1(1), u2(0)));

A total number of rules that should be produced to describe the complete fuzzy control strategy can be calculated by multiplying the input membership function with the output membership function. Although there are number of possible rules, most of them can be discarded as long as the design is able to determine how the fuzzy control system should be operated.

4. Defuzzification

After the output for the each rule has been identified, the next step is to combine all the output into a single value that can used to control the motors. This process is done through defuzzification. The defuzzification technique used in Sugeno method is weighted average. This is done by multiplying fuzzy output obtained from the rules evaluation with its corresponding singleton value, then sum of this value is divided by the sum of all fuzzy output obtained from the rules evaluation. The result from this calculation is the final single output which can be used to control the motor movements. Since there is no division symbol supported by Xilinx ISE Compiler, a divider circuit has to be designed to perform defuzzification.

The following pseudo-code illustrates the procedure of this fuzzification process [8]:

Defuzzification();
Set n = number of output membership function
s = array of singleton of output membership function
p = array of result of all rule evaluation
sum = 0;
For i = 1 to n do begin product = (s(i)×p(i)) + product;
sum = p(i) + sum;
end for loop;
output = product / sum;

III. SIMULATION

1. Simulation in Xilinx ISE

Simulations have been done in Xilinx ISE.Fig.3 and Fig.4 shows that Test Bench Waveform and RTL view of FLC using VHDL respectively. In this figure the outputs have calculated according to input i.e. Error and Change in Error

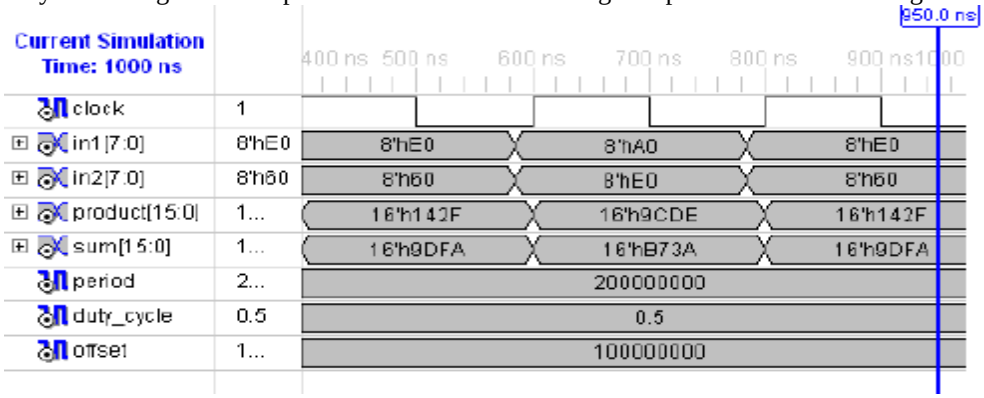

Fig.3. Test Bench Waveform using Xilinx ISE

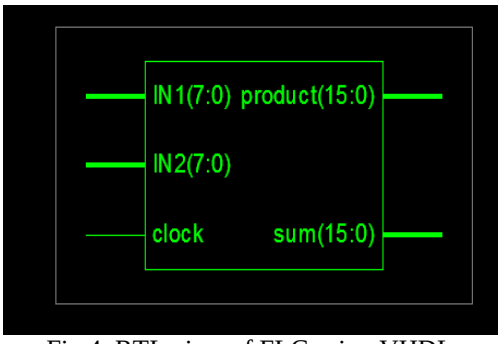

Fig.4. RTL view of FLC using VHDL

Fig.5 shows the Simulink Diagram of FLC and PID using Xilinx system generator. We have taken a same 2nd order system for Dc motor to show the step response performance of FLC. Black box represents the FLC using VHDL code. The two input of FLC i.e. Error and Change in Error are going through the Gateway In to the system generator Black Box, the two outputs are coming through the Black Box. The control output of FLC is finally obtained by dividing one output by another.

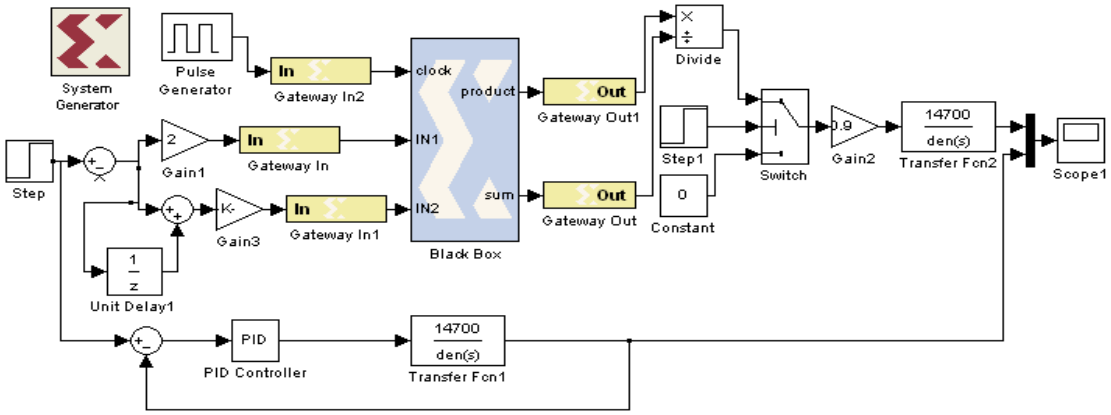

Fig.5. Simulink Diagram of FLC and PID using Xilinx system generator

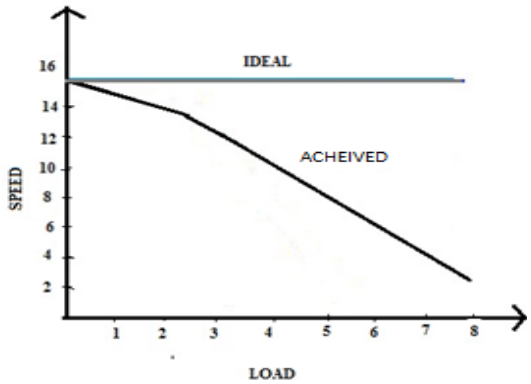

Fig. 6. Speed-Load Characteristics of the DC Motor obtained by implemented FLC Approach

IV. EXPERIMENTAL SETUP

In this section, an experiment is set up to demonstrate the performance of the Fuzzy Logic Controller. Configuration is a process by which the bit streams of the design, as generated by the software are loaded into the internal configuration memory of the FPGA [2]. The performance of the controller design on Hardware is verified by downloading the VHDL code (bit file) into the target FPGA device (Spartan 3 family X3CS200). The experimental setup consists of a DC motor, FPGA kit, a driver circuitry to drive the motor. The specifications of the DC motor used are as given in table 1 below:

Table 2: Parameters of DC Motor

Type	**DC gear motor**
Voltage	12v
Current	3.6A
Torque	172mN.m
Speed	1000rpm

The controller is tested for different speed, tracking performance of the motor both on load and no-load conditions with a given speed. The motor load is generated by applying some amount of tension or strain on the motor rotation. This procedure is equivalent to applying friction load to the motor. The motor speed is detected by using an IR sensor at the input and fed back to the controller to maintain the motor speed.

Fig 6: Experimental setup showing interface between the motor And FPGA

V. CONCLUSION

For designing FLC, a high- level modelling approach in VHDL have to be used. The advantages of this are reducing the design time, evaluation of the design functionality in a short time and quickly exploring of different design choices. Once the basic design of the fuzzy logic control system has been defined, the implementation of the fuzzy logic controller is very straight forward by coding each component of the fuzzy inference system in VHDL according to the design specifications. By simply changing some parameters in the codes and design constraint on the specific synthesis tool, one can experiment with different design circuitry to get the best result in order to satisfy the system requirement. The peak overshoot and settling time both are better than others. The future scope may involve implementation of FLC on system on chip (SOC).

REFERENCES

[1] S. Assilian and E.H. Mamdani, "An Experiment in Linguistic Synthesis with Fuzzy Logic Controller", Int. Journal on Man machine studies. Vol. 7,pp. 1-13,1975.

[2] R. Palm, "Scaling of Fuzzy Controller using the cross correlation", IEEE Trans. Fuzzy Syst., Vol. 3, pp. 116-123, Feb. 1995

[3] H. X. Li and H. B. Gatland, "Conventional Fuzzy Control and its enhancement", IEEE Trans. Syst., Man, Cyber., Vol. 26, pp. 791-797, 1996.

[4] Y. F. Li and C. C. Lau, "Development of Fuzzy Algorithm for Servo System", IEEE Int. conference on Robotics and Automation, April 24-29, 1998.

[5] Andrew Kusiak, "Fuzzy Logic", The University of Iowa, Iowa City 2004.

[6] Daijin Kim, "An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System", IEEE Transaction on Industrial Electronics, Vol. 47, No. 3, pp. 703-715, June 2000.

[7] Sameep and Kuldip S. Rattan, "Implementation of a Fuzzy Controller on an FPGA using VHDL", 22nd International Conference (NAFIPS), pp. 110-115, March 2003.

[8] Philip T. Voung, Asad M. Madni and Jim B. Vuong, "VHDL Implementation for a Fuzzy Logic Controller", World Automation Congress (WAC), July 24-26, 2006.

[9] Clive Max field, Design Warrior Guide to FPGAs, Elsevier publications 2004.

[10] Dr. Peter R. Wilson, Design recipes for FPGAs, Embedded technology series, Elsevier publication 2007.

[11] Moe Shahdad, "An Overview of VHDL Language and Technology". 23rd Conference on Design Automation, pp.320-326, 1986.

[12] Yodyium Tipsuwan and Mo-Yuen Chow, "Fuzzy logic Microcontroller Implementation For DC Motor Speed control", IEEE, pp. 1271-1276, 1999.

[13] Davi Nunes Oliveria, Gustavo Alves de Lima Henn and Otacilio da Mota Almeida, "Design and Implementation of a Mamdani Fuzzy Inference System on an FPGA using VHDL" Annual Meeting of the North American Fuzzy Information Processing Society, 2010.

[14] Marek J. Patyra, Janos L. grantner and Kirby Koster, "Digital Fuzzy Logic Controller : Design and Implementation", IEEE Transaction on Fuzzy System, Vol. 4, No. 4, pp. 439-459, November 1996.

[15] Valentina Salapura and Volker Hamann, "Implementing Fuzzy Control Systems Using VHDL and Statcharts", Design Automation Conference with EURO-VHDL, pp. 53-58,1996.

