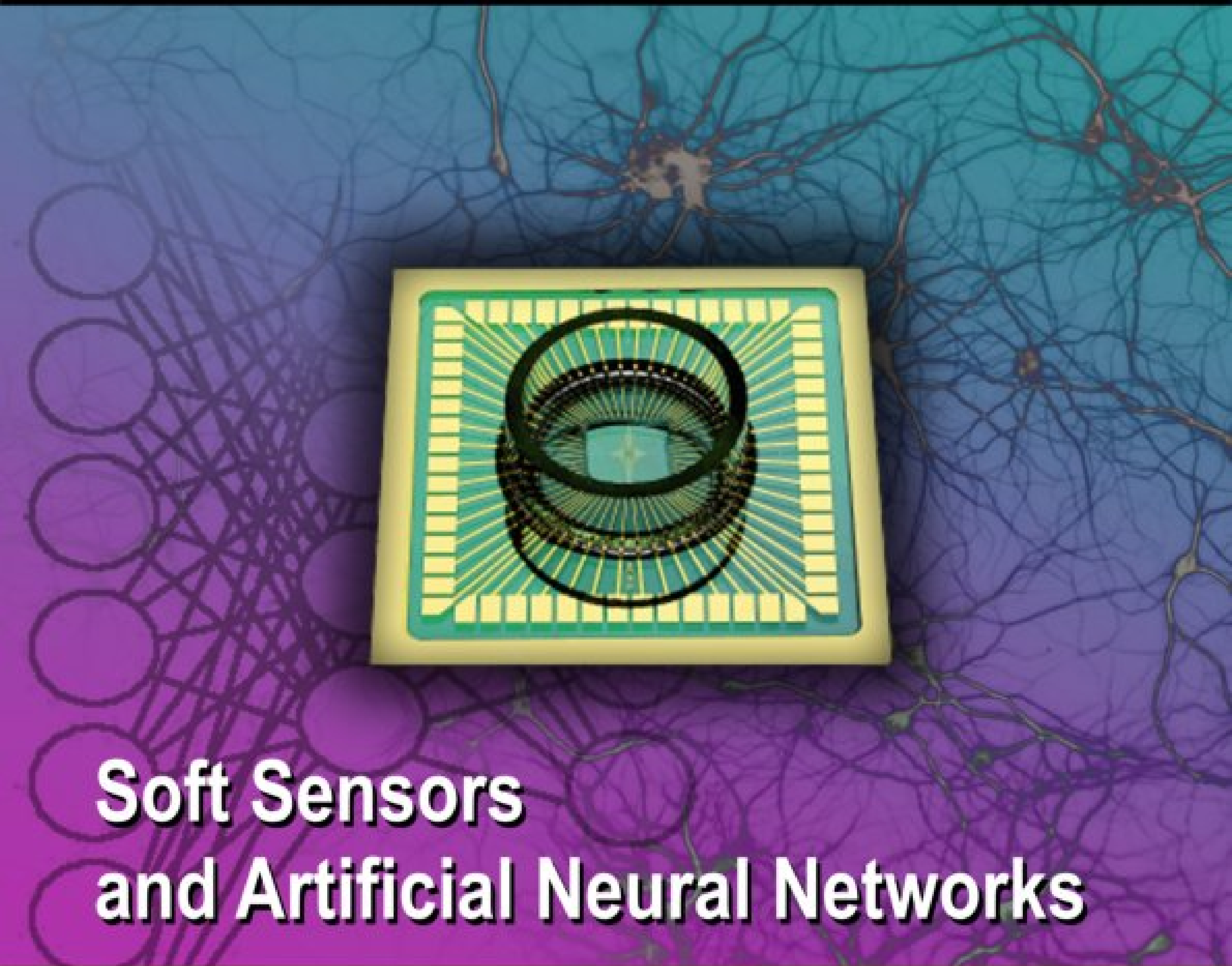
ISSN 1726-5479

SENSORS & TRANSDUCERS

10/10 vol. 121



Soft Sensors and Artificial Neural Networks

International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

Volume 121, Issue 10, September 2010 **www.sensorsportal.com** **ISSN 1726-5479**

Sensors & Transducers Journal

Contents

Volume 121
Issue 10
October 2010

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Computational Sensor Network: Book Review
Sergey Y. Yurish.. I

ANN Modeling of a Chemical Humidity Sensing Mechanism
Souhil Kouda, Zohir Dibi, Fayçal Meddour, Abdelghani Dendouga and Samir Barra........................ 1

Design of Artificial Neural Network-Based pH Estimator
Shebel A. Alsabbah, Maazouz A. Salahat and Mohammad K. Abuzalata.. 10

Improved RBF Neural Network Based Soft Sensor: Application to the Optimal Robust Calibration of a Six Degrees of Freedom Parallel Kinematics Manipulator
Dan Zhang and Zhen Gao.. 18

Real Time Interfacing of a Transducer with a Non-Linear Process using Simulated Annealing
S. M. GirirajKumar, K. Ramkumar, Bodla Rakesh, Sanjay Sarma O. V. and Deepak Jayaraj.......... 29

Visible and Near Infrared (VIS-NIR) Spectroscopy: Measurement and Prediction of Soluble Solid Content of Apple
Herlina Abdul Rahim, Kim Seng Chia and Ruzairi Abdul Rahim. 42

Control System Design for Cylindrical Tank Process Using Neural Model Predictive Control Technique
M. Sridevi, P. Madhavasarma, S. Sundaram 50

Application of Genetic Algorithm for Tuning of a PID Controller for a Real Time Industrial Process
S. M. Giri Rajkumar, Atal. A. Kumar, N. Anantharaman. 56

Modeling and Control of Multivariable Process Using Intelligent Techniques
Subathra Balasubramanian, Radhakrishnan T. K.... 68

Limitations of Feedback, Feedforward and IMC Controller for a First Order Non-Linear Process with Dead Time
Maruthai Suresh and Ranganathan Rani Hemamalini.. 77

Embedded Based DC Motor Speed Control System
Chandrasekhar T., Nagabhushan Raju K., V. V. Ramana C. H., Nagabhushana KATTE and Mani Kumar C.... 94

Real Time Implementation of a DC Motor Speed Control by Fuzzy Logic Controller and PI Controller Using FPGA
G. Sakthivel, T. S. Anandhi, S. P. Natarajan.. 106

IDC Based Battery-free Wireless Pressure Sensor
Jose G. Villalobos, Zhen Xu, and Yi Jia 121

Energy Efficient MAC for Wireless Sensor Networks
Pekka Koskela, Mikko Valta and Tapio Frantti... 133

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal’s webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

International Frequency Sensor Association (IFSA).

Sensors & Transducers Journal, Vol. 121, Issue 10, October 2010, pp. 106-120

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Real Time Implementation of a DC Motor Speed Control by Fuzzy Logic Controller and PI Controller Using FPGA

G. Sakthivel, T. S. Anandhi, S. P. Natarajan
Department of Electronics &Instrumentation Engineering, Annamalai University,
Chidambaram-608002, India
Tel.: +91-9443270714
E-mail: gsauei@gmail.com

Received: 12 September 2010 /Accepted: 18 October 2010 /Published: 26 October 2010

Abstract: Fuzzy logic control has met with growing interest in many motor control applications due to its non-linearity, handling features and independence of plant modelling. The hardware implementation of fuzzy logic controller (FLC) on FPGA is very important because of the increasing number of fuzzy applications requiring highly parallel and high speed fuzzy processing. Implementation of a fuzzy logic controller and conventional PI controller on an FPGA using VHDL for DC motor speed control is presented in this paper. The proposed scheme is to improve tracking performance of D.C. motor as compared to the conventional (PI) control strategy .This paper describes the hardware implementation of two inputs (error and change in error), one output fuzzy logic controller based on PI controller and conventional PI controller using VHDL. Real time implementation FLC and conventional PI controller is made on Spartan-3A DSP FPGA (XC3SD1800A) FPGA for the speed control of DC motor. It is observed that fuzzy logic based controllers give better responses than the conventional PI controller for the speed control of dc motor. *Copyright © 2010 IFSA.*

Keywords: Conventional controller, FLC, VHDL, FPGA.

1. Introduction

Most of the fuzzy logic applications with the physical systems require a real-time operation to interface high speed constraints. The simple and usual way to implement these systems is to realize it as a software program on general purpose computers, these ways cannot be considered as a suitable design solution. Higher density programmable logic device such as Field Programmable Gate Arrays (FPGAs) can be used to integrate large amounts of logic in a single IC. FPGAs are one of the fastest growing parts of the digital integrated circuit market in recent times. They can be configured to implement complex hardware architectures. Dynamically reconfigurable FPGA systems can adapt to various computational tasks through hardware reuse. FPGA becomes one of the most successful of technologies for developing the systems which require a real time operation. FPGAs are more sufficient than the simple way because they can cover a much wider range of operating conditions. Semi-custom and full-custom Application Specific Integrated Circuit (ASIC) devices are also used for this purpose but FPGA provide additional flexibility: they can be used with tighter time-to-market schedules. FPGA places fixed logic cells on the wafer, and the FPGA designer constructs more complex functions from these cells [9]. The term field Programmable highlights the customizing of the IC by the user, rather than by the foundry manufacturing the FPGA. In control applications, in order to get better control responses, [10, 11] Controller is implemented in FPGA. FPGA are two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks. The interconnections consist of electrically programmable switches which is why FPGA differs from Custom ICs, as Custom IC is programmed using integrated circuit fabrication technology to form metal interconnections between logic blocks. FPGA provides its user a way to configure: The intersection between the logic blocks and the function of each logic block. Logic block of an FPGA can be configured in such a way that it can provide functionality as simple as that of transistor or as complex as that of a microprocessor. It can used to implement different combinations of combinational and sequential logic functions.

This paper is organized as follows. Section 2 explains difference between FLC and conventional control Section 3 explains briefly the methodology of Fuzzy Logic Controller. Section 4 explains a hardware implementation on the reconfigurable FPGA system Section 5 explains an overall experimental set up for speed control of DC motor. Section 6 will give a discussion on the servo and regulatory response of PI and Fuzzy Logic controller. Finally, a conclusion is drawn in Section 7.

2. Fuzzy Vs. Conventional Control

In order to design a conventional controller for controlling a physical system, the mathematical model of the system is needed. A common form of the system model is differential equations for continuous-time systems or difference equations for discrete-time systems. Unless physical insight and the laws of physics can be applied, establishing an accurate nonlinear model using measurement data and system identification methods is difficult in practice. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant [12, 13].As an alternative, fuzzy control provides a formal methodology for representing, and implementing a human's heuristic knowledge about how to control a system, which may provide a new paradigm for nonlinear systems. Fuzzy controller is unique in its ability to utilize both qualitative and quantitative information. Qualitative information is gathered not only from the expert operator strategy, but also from the common knowledge [12, 13]. Fuzzy control should not be employed if the system to be controlled is linear, regardless of the availability of its model. PID control and various other types of linear controllers can effectively solve the control problem with significantly less effort, time, and cost.

The benefits of fuzzy controllers could be summarized as follows:
1. Fuzzy controllers are more robust than PID controllers because they can cover a much wider range of operating conditions than PID can, and can operate with noise and disturbances of different nature.
2. Developing a fuzzy controller is cheaper than developing a model-based or other controller to do the same thing.
3. Fuzzy controllers are customizable, since it is easier to understand and modify their rule, which not only use a human operator's strategy, but also are expressed in natural linguistic terms.

4. It is easy to learn how fuzzy controllers operate and how to design and apply them to a concrete application.

2. Structure of Fuzzy Logic Controller

Fuzzy logic has rapidly become one of the most successful of today's technologies for developing sophisticated control systems. With its aid, complex requirements may be implemented in amazingly simple, easily maintained, and inexpensive controllers [20]. Fuzzy control use only a small portion of the fuzzy mathematics that is available, this portion is also mathematically quite simple and conceptually easy to understand.

The fuzzy controller, have four main components:

- The Rule-Base holds the knowledge, in the form of a set of rules, of how best to control the system.
- The Inference Mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.
- The Fuzzification Interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.
- The Defuzzification Interface converts the conclusions reached by the inference mechanism into the inputs to the plant [23], [25].

3. FPGA-Based Fuzzy Logic Controller

There are a number of reasons for using fuzzy logic in the speed control of DC motor, the primary advantage being the flexibility offered by fuzzy logic. The backbone of any FLC is embodied in a set of fuzzy rules, with two implications:

- The fact that the control strategy is represented by a set of rules and not an elaborated set of equations. This allows the designer to change the basic characteristics of the Controller with minimal fuss, simply by redefining the rules.
- The *fuzzy* aspect of the rules, deals with the imprecise definition of the system. This allows vagueness in the design of the control system to be tolerated to a certain degree and eliminates the need for a well-defined mathematical model of the plant.

The following sections describe the development of FLC for speed control of DC motor.

The present design utilizes three types of membership functions – Γ-function, *L*-function and Λ-function. These functions have been proven to produce good results for control applications and can be easily implemented into hardware. The universe of discourse of the input variables is partitioned into five fuzzy sets or *linguistic values* (*B*1 to *B*5), while the output variable can take any of the nine linguistic values (*D*1 to *D*9). Graphical representations of the membership functions are shown in Fig. 1 and Fig. 2.

The crisp values of the input variables are mapped onto the fuzzy plane using the equations above. It gives each input variable a membership function relating to the fuzzy sets, $(B_i^1 \ to \ B_i^5)$. It has to be pointed out that in these equations, B_i^j is used to denote the linguistic value as well as membership function, while membership function using $\mu_{\mathrm{B}i,\,j}(x_i)$.The universe of discourse of the output variable is divided into nine linguistic Values. The membership functions of the output values are intentionally made to be symmetrical, as this will simplify the defuzzification computation. E1 to E9 are the mean of each function and act as the weightings to the weighted average method of defuzzification.

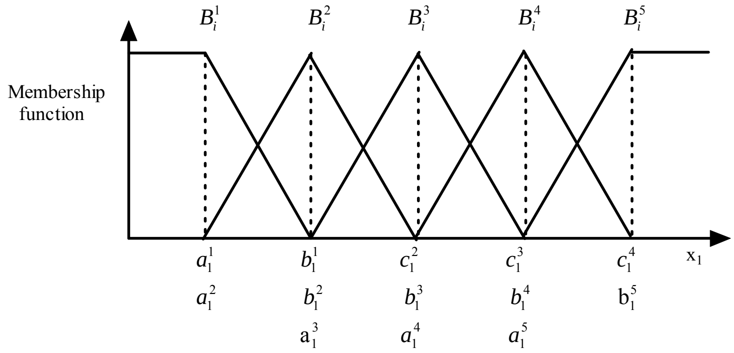


Fig. 1. Memberships function of input variable.

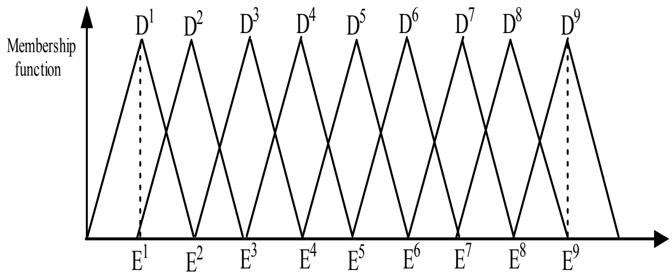


Fig. 2. Memberships function of output variable.

Each input variable can take any of the five linguistic values, therefore 25 (= 5 X 5) rules are formulated. The rules have the typical fuzzy rule structure, using linguistic *variables* in both the antecedent and consequent, and are expressed in IF-THEN manner. They map the input states onto 25 output conditions (*C*1 to *C*25). The fuzzy rules have the general form,

$$R^K : \text{IF } x_1 \text{ is } A_1^k \text{ AND } x_2 \text{ is } A_2^k \text{ , THEN } y \text{ is } C^K \tag{1}$$

Then the rule base can be represented by a fuzzy associative memory (FAM) table (Table 1).

Table 1. FAM table for FLC design.

ΔE(X_2) / E(X_1)	**NB** (B_2^1)	**N** (B_2^2)	**Z** (B_2^3)	**P** (B_2^4)	**PB** (B_2^5)
NB (B_1^1)	NVB R^1	NB R^2	N R^3	NS R^4	Z R^5
N (B_1^2)	NB R^6	N R^7	NS R^8	Z R^9	PS R^{10}
Z (B_1^3)	N R^{11}	NS R^{12}	Z R^{13}	PS R^{14}	P R^{15}
P (B_1^4)	NS R^{16}	ZS R^{17}	PS R^{18}	P R^{19}	PB R^{20}
PB (B_1^5)	Z R^{21}	PS R^{22}	P R^{23}	PB R^{24}	PVB R^{25}

The FLC design in this work incorporates Mamdani's implication method of inference, which is one of the most popular methods in fuzzy control applications. In essence, Mamdani's implication for the fuzzy rule of (4) is given by

$$\mu_c(y) = \max_k \left[\min\left[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2)\right]\right] \quad k = 1,2,....25 \tag{2}$$

The implication has a simple min–max structure which makes it easy to incorporate into hardware. The block diagram in Fig. 3 provides an overview of the controller's internal structure. Two input variables are fuzzified, producing the corresponding linguistic values and membership functions (B_i^j). The first phase of Mamdani's implication involves *min*-operation since the antecedent pairs in the rule structure are connected by a logical 'AND'. All the rules are then aggregated using a *max*-operation.

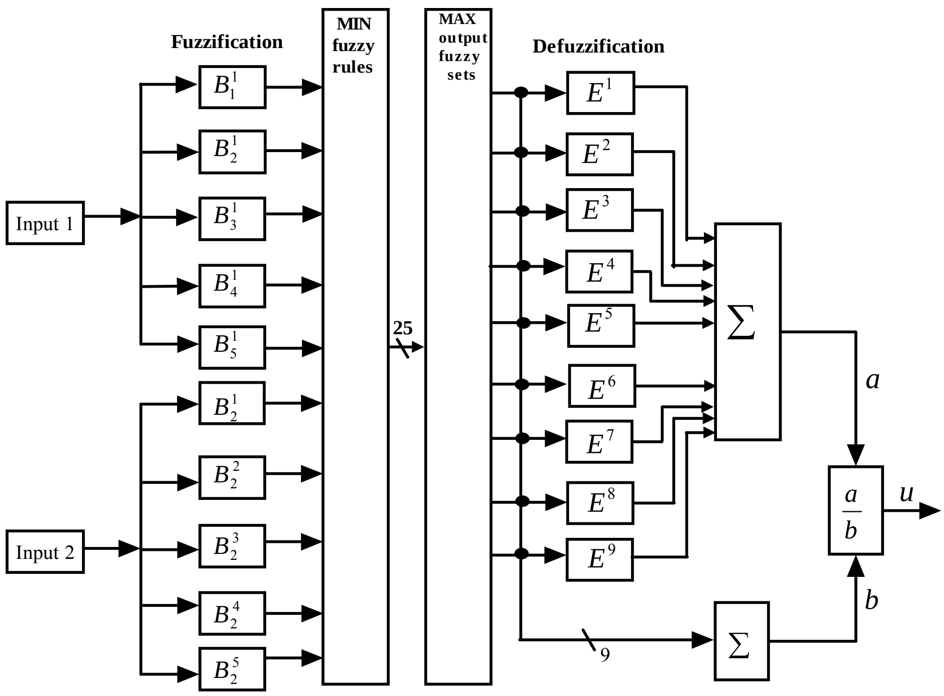


Fig. 3. Block diagram of the operations in a Fuzzy Logic Controller.

The dominant criterion in the selection of defuzzification technique lies in the implementation stage. The weighted average method is viewed to be an appropriate technique for systems involving hardware implementation. Due to the fact that the output membership functions are symmetrical in nature, the mean of the fuzzy sets can be used as weightings for the defuzzification process. This technique requires several multiply-by-a-constant operations and only one division process. Rule base moulds the functionality of an FLC. The rules are most likely to be formulated based on some level of human understanding of the plant. The design of the FLC in this paper is based on PI controllers .The rule base is constructed from the control law of a PI system.

3.1. PI Control

The proportional–integral (PI) controller is a well-known system in control engineering. It is, in essence, a lag compensator characterized by the transfer function

$$G(s) = K\left(1 + \frac{1}{T.s}\right),$$

where *G*(*s*) is the gain; *K* is the control parameter; *T* is the time constant.

The control law is given by the equation

$$u_{PI} = K_p.e + K_I \frac{1}{T} \int_0^t e.dt \tag{3}$$

where *u* is the control signal; *e* is the error, given by *e* = (input value) – (reference value).

Differentiating (3) gives

$$\frac{du}{dt} = K_p \frac{de}{dt} + K_I .e \tag{4}$$

In discrete-time systems, (4) can be written as

$$u(kt) - u(kT - T) = K_p.\{e(KT) - e(kT - T)\} + K_I.e(kT)$$

$$\Delta u = K_p.\Delta e + K_I.e, \tag{5}$$

where Δu is the change in *u* over one sampling period; Δe is the change in *e* over one sampling period.

The values of controller parameters are found from Z-H tuning method. From the values found are K_p=0.12 and K_i= 0.0201. The characteristic of a PI controller can be represented by the phase plane diagram shown in Fig. 4. A diagonal line where Δu = 0 divides the area where Δu is positive and Δu is negative.

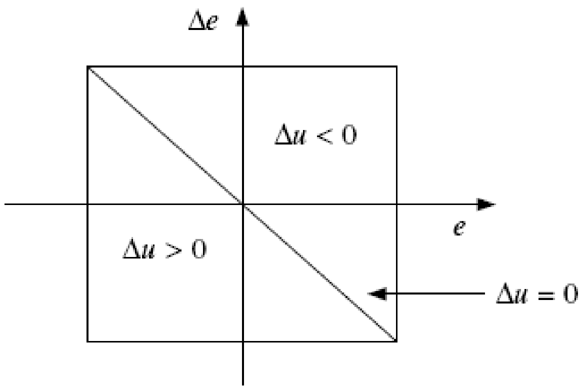


Fig. 4. Characteristic of PI controller.

3.2. PI-like Fuzzy Control

At this stage, the control law in (8) is not in fuzzy terms. In order to design a fuzzy controller based on the PI control structure, the following definitions are made:
Let

E is the linguistic variable for the error e;
ΔE is the linguistic variable for the change of error Δe;
U is the linguistic variable for the control output u.

The maximum range of motor is +/- 1500 rpm. The possible error range is -750 to 750.The universe of Discourse of the change in error is based on experiment data from conventional controller which gives the range of error is +/- 150.Ouput of the controller ranges from +/- 8. Linguistic variables for $E, \Delta E$, U are defined as

LE = {Negative Big(-750rpm), Negative(-500rpm), Zero(0), Positive(500rpm), Positive Big(750 rpm)}

$L\Delta E$ = {Negative Big(-150rpm), Negative(-75 rpm), Zero(0), Positive(75), Positive Big(150)}

LU = { Negative very Big(-8) ,Negative Big(-6), Negative(-4), Negative Small(-2), Zero(0), Positive Small(2), Positive(4), Positive Big(6), Positive very Big(8) }

The corresponding PI control law in IF-THEN rules has the form:

$$R^K : \text{IF } x_1 \text{ is } A_1^K \text{ AND } x_2 \text{ is } A_2^K \text{, THEN } y \text{ is } C^K,$$

where:
A_1^K can take any linguistic value in the set LE;
A_2^K can take any linguistic value in the set $L\Delta E$;
C^K can take any linguistic value in the set LU.

To implement this design into the FLC, let:

- $x1 = E$
- $x2 = \Delta E$
- $\{B_i^1, B_i^2 B_i^3 B_i^4 B_i^5\}$= {Negative Big(NB), Negative(N), Zero(Z), Positive(P), Positive Big(PB)},for i =1, 2
- $\{D^1, D^2, D^3, D^4, D^5, D^6, D^7, D^8, D^9\}$ = { Negative very Big(NVB) ,Negative Big(NB), Negative(N), Negative Small(NS), Zero(Z), Positive Small(PS), Positive(P), Positive Big(PB), Positive very Big (PVB)}.

Fig. 5 shows a block diagram demonstrating the implementation of the FLC in a Speed control of DC motor .In this application, the input interface converts the output of the speed sensor into *error* and *change of error* which are used as the two inputs to the FLC.

Another interface converts the output into the required value for the plant. The characteristics of the interfacing blocks can be described by the following equations:

Input interface:

$$e = V_{REF} - V_{dc};\ x1 = e\ ;\ x2 = x1 - x1z^{-1}$$

Output interface:

$$\Delta u = y\ ;\ \ u = \Delta u + u\, z^{-1}$$

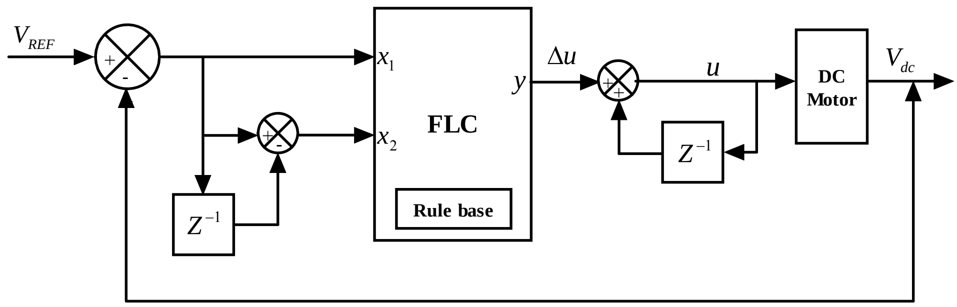


Fig. 5. Overall block diagram of FLC based control system for Dc motor.

4. FPGA Implementation

A Field Programmable Gate Array is a digital integrated circuit that can be programmed to do any type of digital function. When the FPGA-based system is used for implementing the desired FLC, many possible designs can be tried due to the reusability of the FPGA. Often, a hardware implementation on the FPGA-based system is supported by many existing EDA tools for modelling, synthesis, verification, and implementation. The major advantage of using the EDA tools is that the same hardware description language code for modelling can be directly used for synthesis, verification, and implementation. Also, the general architecture of the FLC is invariant except for the change in the number of input and output variables, the number of fuzzy terms, the membership functions, the bit resolutions, and the control rule base according to the different applications. The FPGA consists of three major configurable elements:

1) Configurable Logic Blocks (CLBs) arranged in an array that provides the functional elements and implements most of the logic in an FPGA.
2) Input-output blocks (IOBs) that provide the interface between the package pins and internal signal lines.
3) Programmable interconnect resources that provide routing path to connect inputs and outputs of CLBs and IOBs onto the appropriate network.

The Target device in this case is Spartan®-3A DSP family (XC3SD1800A).The features of Spartan®-3A DSP –2XC3SD1800A is tabulated in Table 2.

Table 2. System resources of Spartan®-3A DSP family.

System gates		1800 k
Equivalent logic cells		37,440
CLB Array (one CLB = four slices)	Rows	88
	Columns	48
	Total CLB	4,160
	Total slices	16,640
Distributed RAM bits		260 k
Block RAM bits		1512 k
DSP48As		84
DCMs		8
Maximum user I/O		519
Maximum Differential I/O pairs		227

The FLC is divided in to 5 VHDL components. Fig. 6 shows a diagram of the FLC architecture. Each component is depicted with its VHDL code. The functionality of components **Interface1**, **Interface2** and part of the component **Infer** will determine the characteristics of the FLC. Input and output variables are designed with a resolution of 8 bits.

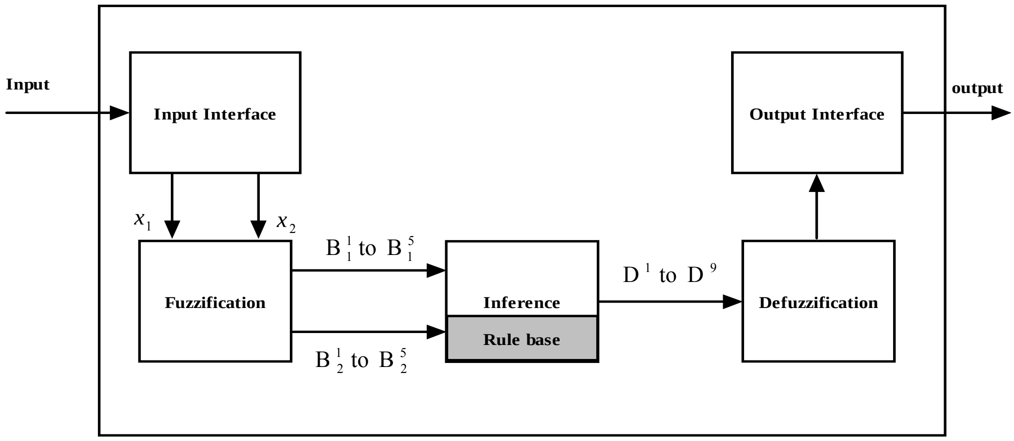


Fig. 6. Components of Fuzzy logic controller in FPGA.

Xilinx XC3SD1800A FPGA is available in several packages and the one used for this design is the PC-512 package which has 512 I/O pins in total. During the generation of the bitstream, the inputs and outputs of the design are mapped to the physical I/O pins of the FPGA. The allocation of pin numbers can either be automatically performed by the implementation tools or explicitly specified by the user. In this design, all the pins are specified by manually assigning the appropriate pin numbers. This enables the designer to have full control over the function of the physical pins in the FPGA. The allocation of pin numbers to the I/O pads is shown by the schematic diagram of the design in Fig. 7. Once the hardware specifications have been confirmed, the netlist is compiled into a bitstream file using the *Implementation* procedure in Xilinx Foundation Project Manager.

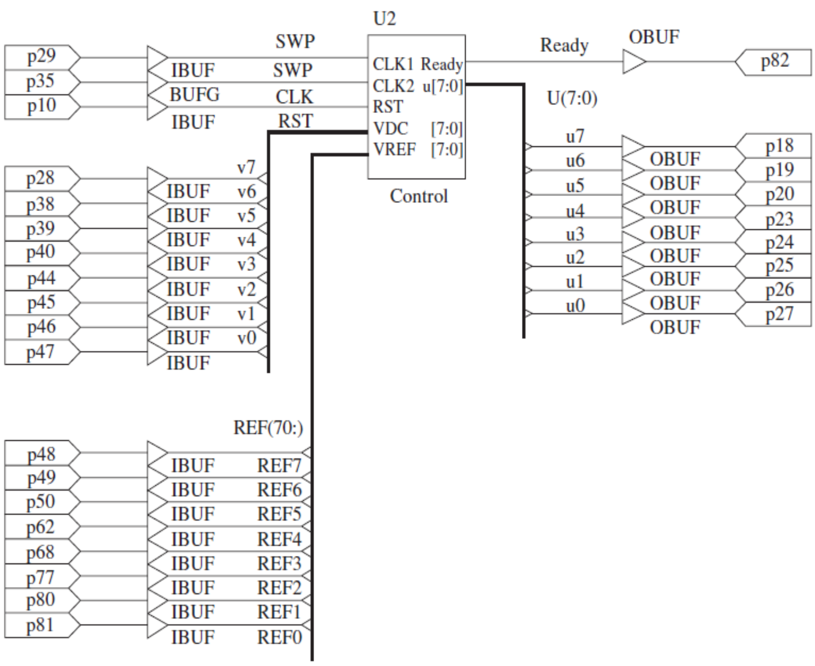


Fig. 7. Schematic diagram of the FPGA design.

5. Experimental Setup

The experimental studies are carried out to evaluate the performance of the controller. Configuration is the Process by which the bit streams of a design, as generated by the development software are loaded into the internal configuration memory of the FPGA. To verify the performance of the controller design on Hardware, the VHDL code (Bit file) is downloaded into the Target FPGA device (Spartan 3 family XC3S400) and the complete system is reset The experimental set up for this system consists of a DC motor , FPGA kit(with on board ADC, DAC, PWM modules), Intelligent power module. The overall block diagram Of Dc motor speed control is shown in Fig. 7. The DC shunt motor is chosen in this work whose parameters are shown in Table 3.

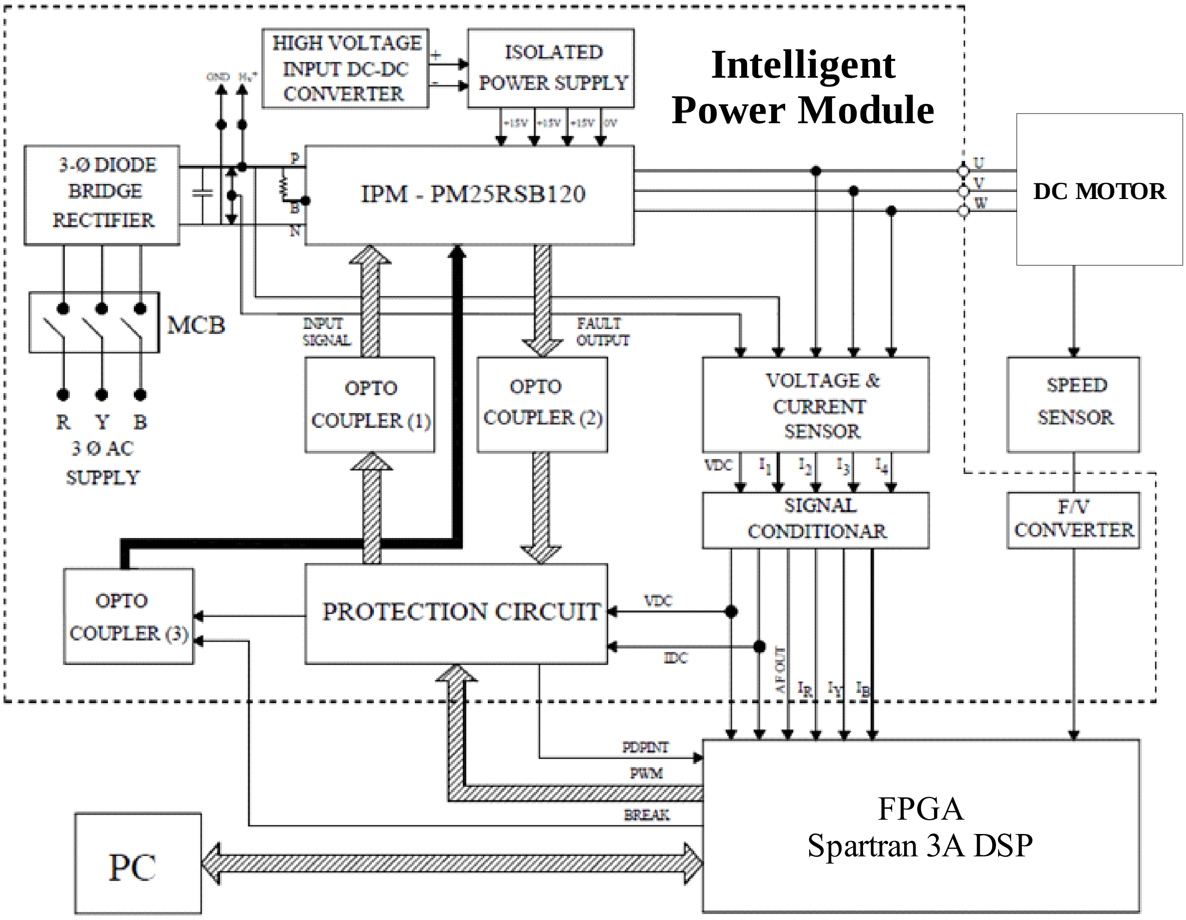


Fig.7. Overall block diagram of speed control of DC motor.

Table 3. Parameters of DC motor.

Type	3 PHASE DC Shunt Motor
SPEED	1500 rpm
Voltage	220 V
Rating	1.0 HP
Current	0.3A
Electrical resistance	1 Ohm
Electrical inductance	0.5H
Moment of Inertia of Motor	$0.01 Kgm^2/s^2$

5.1. Sensor - Optical Encoder

Circular windows around the circular disk mounted on the motor shaft such that it rotates with the shaft. A LED is mounted on the one side of the disk and a phototransistor is mounted on the other side of the disk, opposite to the LED, the following Fig. 8 shows the speed sensor. During rotation when circular window come across the LED, the light passes to the phototransistor. As a result, phototransistor conducts and produces low output at its collector. Each time when light passes through window to the phototransistor, it conducts and output

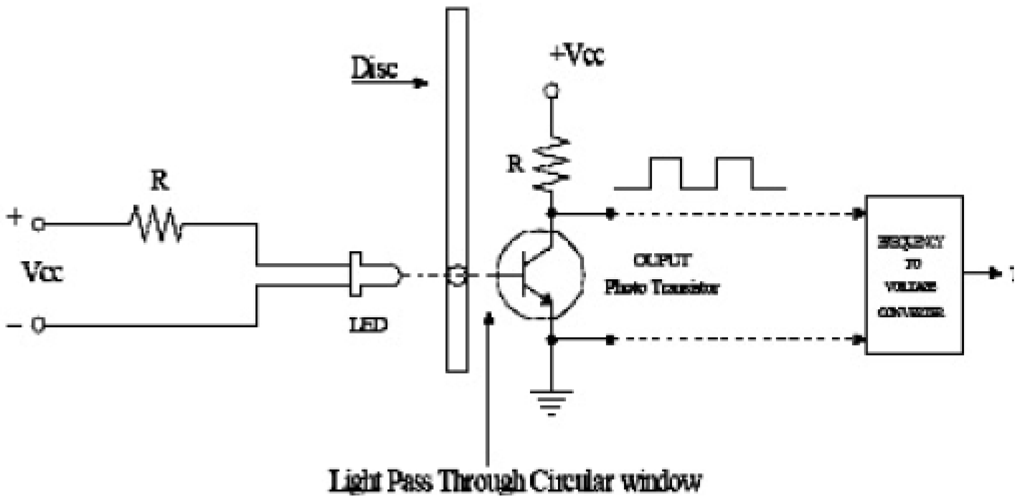


Fig. 8. Optical encoder speed sensor.

goes low, otherwise phototransistor is off and output is high. As disk rotates the train of pulses are generated. The number of pulses in one rotation equals the number of circular windows on the disk. Therefore by counting number of pulses we can decide the position of the shaft as well as number of rotations performed by the shaft. By counting the number of rotations in specific time we can also calculate the speed of rotation. Counting the number of pulses in specific time, these pulses convert frequency to voltage by using frequency to voltage converter.

5.2. Intelligent Power Module

Intelligent power modules (IPMs) are advanced hybrid power devices that combine high speed, low loss IGBTs with optimized gate drive and protection circuitry. The power stage consists of a Variable source converter (VSC) fed drive. It has two stages of power conversion ,a rectifier and converter. The rectifier converters a fixed voltage AC to either fixed or adjustable DC voltage .The converter is constituted of solid state switches(IGBT), that switch the DC power on and off to produce a controllable and desired DC voltage .FPGA generates firing pulses for the power switches in the converter. IPM consists of 1. Intelligent Power Module, 2. Voltage and Current Sensor 3. Signal Conditioner 4. Protection, Circuit 5. Optocoupler, 6. Three phase Diode Bridge Rectifier 7. Speed Sensor 8. Frequency to Voltage converter.

IPM has sophisticated built-in protection circuits that prevent the power devices from being damaged. The protection schemes available in IPM are self protection, over temperature protection, over current protection, short circuit protection. Three phase diode bridge rectifier is used to give rectified DC voltage to IPM. Output voltages and current of IPM are not directly fed to control circuits ,because output voltage of IPM is high, but control circuit is operated in minimum voltage (5v).Hall effect transducers are used for these type of conversion .Signal conditioning circuits are used to give the reference signal of current and voltage to the protection circuit as well as to the ADC of FPGA kit. Optocoupler is used to isolate the control circuit from power circuit. Speed of the Dc motor is sensed by optical encoder which gives the output in terms of frequency proportional to speed. By means of Frequency to voltage converter, sensor output is converted in to voltage.

Fig. 9 shows the FPGA Spartan®-3A DSP kit. The speed of the motor is measured by means of optical encoder and given as input to ADC(AD7266). The ADC used here is 12 bit, high speed, low power, successive approximation ADC and features throughput rates up to 2 MSPS. The set speed is assigned to motor by toggle switches according to the requirement. Once this is done the ADC data will be read and Fuzzy Logic Controller implemented will calculate output value and the output of controller in term changes the duty cycle of PWM to increase or decrease the speed. IPM Module consists of Switching power converters are used in most DC motor drives to deliver the required energy to the motor. The energy that a switching power converter delivers to a DC motor is controlled by Pulse Width Modulated (PWM) signal applied to the gate of a power transistor coming from PWM module FPGA kit. once the current speed equals the set speed, the motor starts running at the set speed. Again to change the set speed, the above procedure is repeated by changing the toggle switch position. As the set speed is varied, the ADC voltage also varies. It is observed that the current speed, which is displayed, on the 'on board' LCD display equals the set speed value. Fig. 10 shows the Photograph of the experimental setup and working model of FPGA based DC motor speed control system

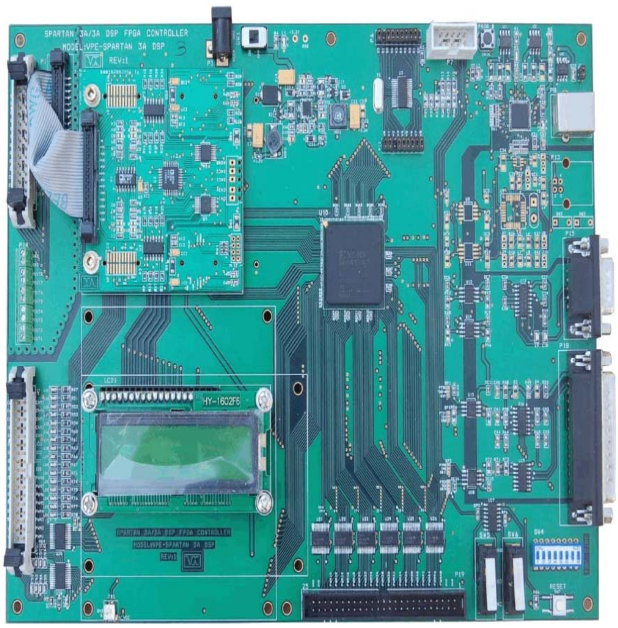


Fig. 9. FPGA Spartan®-3A DSP kit.

Fig. 10. Real time experimental setup.

6. Results and Discussion

The Fuzzy Logic Controller is implemented in FPGA for real time control of speed in DC motor . Resource utilization of Fuzzy Logic Controller is shown in Table 4. It provides the information regarding number slices, Flip Flops, Input output blocks, total number of gates required. The simulation of Fuzzy logic controller is shown in Fig. 10.The Fuzzy logic controller is divided in 5 modules, input interface, fuzzifier, inference , defuzzification, output interface .The output of each module is shown in simulation result.

The servo responses of the PI controller and FLC shown in Fig. 11 and Fig. 12. The set point is varied from 550 rpm to 1100 rpm .For the PI controller (shown in Fig. 10), set point tracking performance is characterized by lack of smooth transition between set point, as well as presence of overshoot and higher rise time. From Fig. 11 we can observe that the FLC having less oscillation , zero overshoots, less rise time. From the Fig. 10 and Fig. 11,it can be seen that FLC performs significantly better than PI controller.

The FLC is used to control the speed of the motor while applying a load change of +10 %. The motor is also run with a PI controller while applying the same load changes. The variations in speed with time for 10% load change for PI controller and FLC is shown in Fig. 13 and Fig. 14.The FLC is able to compensate for the load changes considerably better than PI controller.

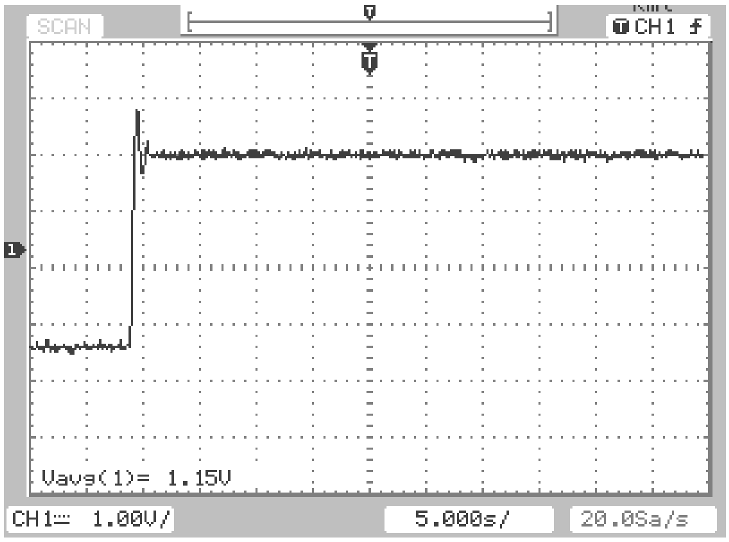


Fig. 11. Servo response of PI controller for a set point change from 550 rpm to 1100 rpm.

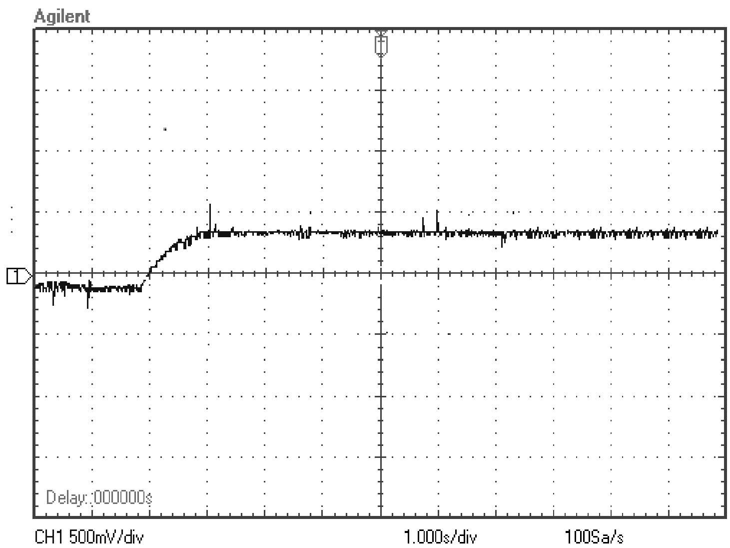


Fig. 12. Servo response of FLC for a set Point change from 550 rpm to 1100 rpm.

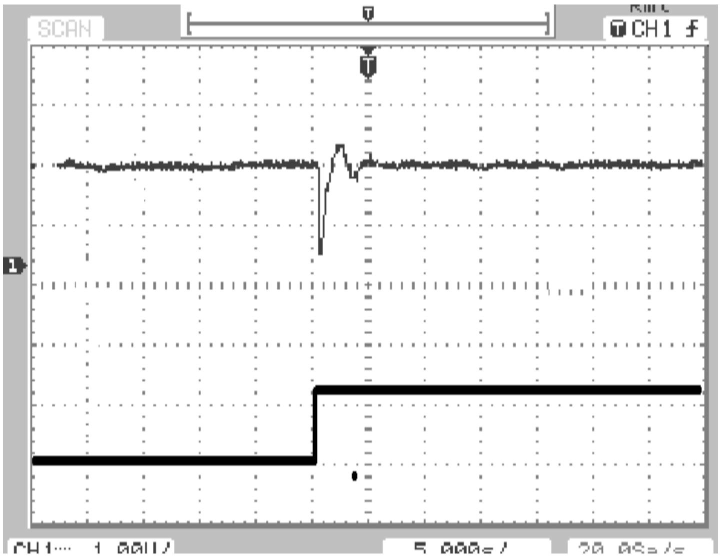


Fig. 13. Regulatory response of PI Controller under +10% change in speed as load.

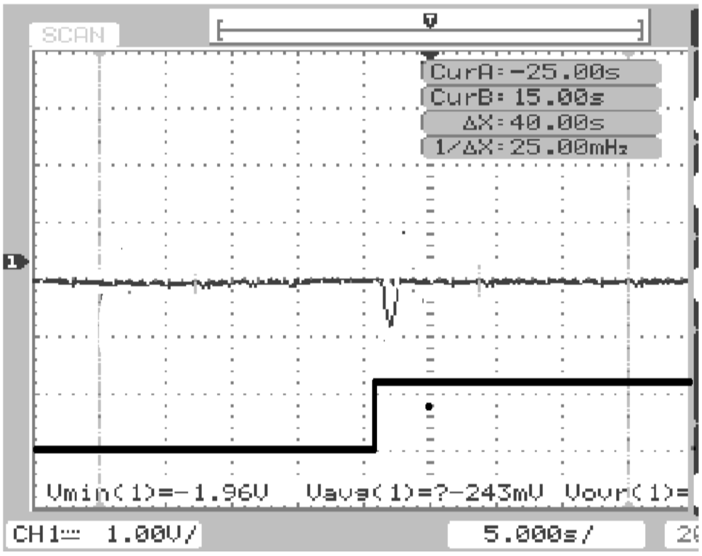


Fig. 14. Regulatory response of FLC Under +10% change in speed as load.

7. Conclusion

This paper presents an approach for the implementation of a fuzzy logic controller for DC motor on an FPGA using VHDL. This paper presents the implementation of a fuzzy logic controller for a DC motor on a Xilinx Spartan III FPGA using VHDL. The implementation of the fuzzy logic controller is very straightforward by coding each component of the fuzzy inference system in VHDL according to the design specifications. The design of the FLC is highly flexible as the membership functions and rule base can be easily changed. Moreover the performance of FLC is compared with PI controller for set point change and load change. The performance FLC was much superior than conventional PI controller and FLC is able to compensate load changes better than PI controller. 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 FLC can also be used for control purposes in other applications.

References

[1]. Jose L. Gonzalez-Vazquez, Oscar Castillo and Luist. Aguilar-Bustos, A Generic Approach to Fuzzy Logic Controller Synthesis on FPGA, *IEEE International Conference on Fuzzy Systems*, 2006, pp. 2317 - 2322.

[2]. S. Poorani, T. V. S. Urmila Priya, K. Udaya Kumar and S. Renganarayanan, FPGA based fuzzy logic controller for electric vehicle, *Journal of The Institution of Engineers,* Singapore, Vol. 45, Issue 5, 2005.

[3]. Barriga, S. Sanchez-Solano, P. Brox, Modeling and implementation of fuzzy systems based on VHDL, *International Journal of Approximate Reasoning*, 41, 2006, pp. 164–178.

[4]. Shabiul Islam, Nowshad Amin, M. S. Bhuyan, Mukter Zaman, Bakri Madon and Masuri Othman, FPGA Realization of Fuzzy Temperature Controller for Industrial Application, *WSEAS Transactions on Systems and Control*, Vol. 2, Issue 10, Sep. 17, 2007, p. 484- 490.

[5]. Eric Monmasson, and Marcian N. Cirstea, FPGA Design Methodology for Industrial Control Systems—A Review, *IEEE Transactions on Industrial Electronics,* Vol. 54, No. 4, august 2007, p. 1824- 1842.

[6]. Masmoudi N., Hachicha M. and Kamoun L., Hardware Design of Programmable Fuzzy Controller on FPGA, *IEEE International Fuzzy Systems Conference Proceedings,* Seoul, Korea, August 22-25, 1999, pp. 1675-1679.

[7]. Daijin Kim, An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System, *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 3, 2000, p. 703-715.

[8]. Michael McKenna and Bogdan M. Wilamowski, Implementing a Fuzzy System on a Field Programmable Gate Array, *IEEE International Joint Conference on Neural Networks,* 2001, Proceedings, IJCNN '01, Vol. 1, 2001, p. 189-194.

[9]. Philip T. Vuong, Asad M. Madni and Jim B. Vuong, VHDL implementation for a fuzzy logic controller, *BEI Technologies, Inc.,* USA, 2006, http://www.wacong.org/wac2006/allpapers/isiac/isiac_248.pdf

[10]. Mohammed Y. Hassan and Waleed F. Sharif, Design of FPGA based PID-like Fuzzy Controller for Industrial Applications, *IAENG International Journal of Computer Science,* 34, 2, 2007.

[11]. Masmoudi N., Hachicha M. and Kamoun L., Hardware Design of Programmable Fuzzy Controller on FPGA, *IEEE International Fuzzy Systems Conference Proceedings*, Seoul, Korea, August 22-25, 1999, pp. 1675-1679.

[12]. H. Ying, Fuzzy Control and Modeling, Analytical Foundations and Applications, *Institute of Electrical and Electronic Engineers Inc.*, USA, 2000.

[13]. K. M. Passino and Stephen Yurkovich, Fuzzy Control, *Addison-Weslwey Longman Inc.*, USA, 1998.

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In additional, some special sponsored and conference issues published annually. *Sensors & Transducers Journal* is indexed and abstracted very quickly by Chemical Abstracts, IndexCopernicus Journals Master List, Open J-Gate, Google Scholar, etc.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 8-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information

Advertising orders and enquires may be sent to sales@sensorsportal.com Please download also our media kit: http://www.sensorsportal.com/DOWNLOADS/Media_Kit_2009.pdf