FPGA Implementation of Three-Phase Induction Motor Speed Control Using Fuzzy Logic and Logic Based PWM Technique

R.P. Dhobale **Prof. D.M. Chandwadkar**

Abstract **— This paper presents the design and implementation of FPGA based three phase induction motor speed controller using fuzzy logic and logic based PWM technique. Logic based pulse width modulation (PWM) generation method is used to vary the speed of induction motor. Using sine lookup tables and PWM programmed into the FPGA logic, control signals are generated to drive three phase integrated power module commonly used in AC induction motor drives. Fuzzy logic control (FLC) has developed growing interest in many industrial motor control applications due to its nonlinearity handling features and independence of modeling requirement. The hardware implementation of FLC on FPGA is vital because of the increasing number of fuzzy applications requiring highly parallel and high speed fuzzy processing. FLC system is presented with a Xilinx SPARTAN3 XC3S 400E FPGA.**

Key Words **— FLC, FPGA, PWM**

I. INTRODUCTION

Induction motors are the most widely used motors for appliances, industrial control, and automation; hence, they are often called the workhorse of the motion industry. They are robust, reliable, and durable. When power is supplied to an induction motor at the recommended specifications, it runs at its rated speed. However, many applications need variable speed operations. For example, a washing machine may use different speeds for each wash cycle. Many industrial applications like heating, ventilation and air-conditioning, industrial drives, motion control, robotics, automotive control like electric vehicles etc. [1], [2] require variable speed to maintain the desired purpose of an application. Induction motor control is complex due to its nonlinear characteristics. While there are different methods for control, Variable Voltage Variable Frequency (VVVF) or V/*f* is the most common method of speed control used in low cost fairly constant load applications.

Classical control systems like PI control have been used for the speed control of induction motors. The main disadvantage of classical PI controllers is their large overshoot and excessive settling time. To deal with these problems, fuzzy logic control has recently been applied to the control of electrical drive systems [3]. Fuzzy logic, deals with control system challenges such as vagueness, uncertainty and use membership functions with values varying between 0 and 1 [4]. However, induction motors can only run at their rated speed when they are connected to the main power supply. That is why variable frequency drive is needed to vary the rotor speed of an induction motor. Implementation of the fuzzy logic in real time applications with the physical systems constrained with high speed operation requirements. The simple and straightforward path to implement these systems is to realize it as a software program on general purpose computers. Such a methodology is cannot be considered as a suitable design solution for cost sensitive and/or industrial applications. Higher density programmable logic device such as Field Programmable Gate Arrays (FPGA) can be used to integrate large amounts of logic in a single IC. Implementation on FPGA is one of the alternatives to tackle the real time requirements and disadvantages of conventional microcontroller or DSP based control design.

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 the user a way to configure the intersection between the logic blocks and the function of each logic block. It can be used to implement different combinational and sequential logic functions.

Designs using FPGA offers flexibility of re-programming to suit customized application and fast development life cycle to deal tighter time-to-market schedules. This paper presents the efficient and faster solution to implement fuzzy logic controller using field programmable gate arrays. It applies the logic based PWM method to control the speed of three phase induction motor using PI approximated fuzzy control.

II. SYSTEM BLOCK DIAGRAM

Fig. 1 shows system block diagram for the FPGA based fuzzy controller for three phase induction motor. System consists of fuzzy controller, PWM generator, and speed command processing blocks implemented in FPGA.

Set speed command processing and actual speed processing blocks are used to scale speed input signal to required values for comparison and error generation. Integrated power

module is used as power stage to drive the three phase induction motor.

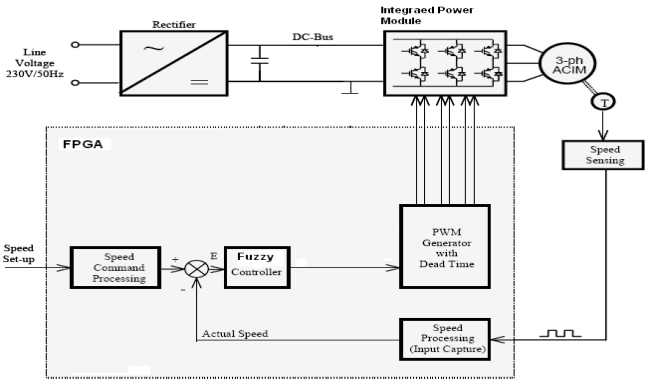

Fig. 1: System Block Diagram

Actual motor speed is sensed by techo-generator and feedback signal is given to fuzzy controller through speed processing unit which counts the no. pulses per minutes and scales the actual speed for comparison.

III. LOGIC BASED PWM TECHNIQUE

In logic based PWM technique, an induction motor is fed with variable frequency signals generated by the PWM control implemented in FPGA. Hence, a control of this type offers low cost and is an easy to implement solution for fairly constant load application with variable speed requirement. In such controls, very little knowledge of the motor is required for frequency control and hence this control is widely used. Sinusoidal PWM generation using lookup table is implemented to generate approximate sine wave. In this method, the sinusoidal weighted values are stored in the FPGA and are made available at the output at user defined intervals. The advantage of this technique is that very little calculation is required. Only one look-up table of the sine wave is required, as all the motor phases are 120 electrical degrees displaced.

Pulse Width Modulation (PWM) signals are required to drive a three phase H-bridge inverter power module. PWM signals are generated by means of up-down counter and three comparators. The maximum count value of the counter is kept as a configurable parameter which also defines the resolution or minimum step increment available to generate the sinusoidal wave forms.

Consider the following; a four bit counter counts from 0 – 15, with 0 being the minimum voltage available at the power inverter and 15 being the maximum voltage available at the power inverter. Now, with the inverter voltage ranges from 0V – 400V, each time the counter increments by 1, the inverter voltage will step by $1/16^{th}$ of the maximum voltage in this case by 25 Volt. The PWM period frequency depends upon the value of the configurable parameter and the speeds of the clock used to increment or decrement the counter count value. The counter counts up from 0 to its maximum value and then count down from its maximum value to 0, as seen in Fig. 2. The exact PWM period frequency is derived by a clock divider which scales down the system clock frequency before it clocks the counter unit.

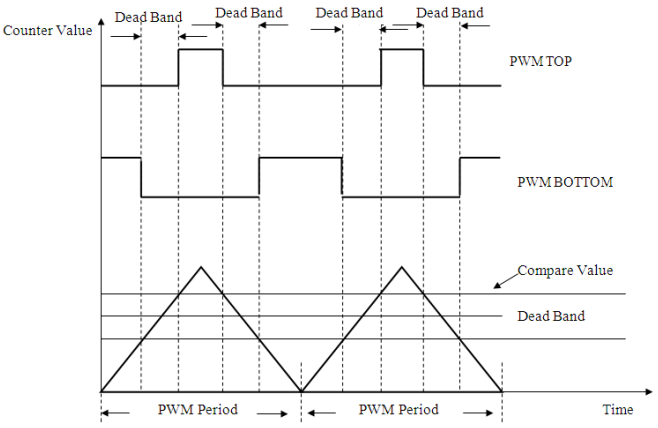

Fig. 2: PWM Output with dead band

The exact frequency of the PWM period can be determined with the following equation:

$$f = CLK / 2 (2^{CNT\_MAX}) / CLOCK\_DIVIDER \qquad (1)$$

For design, CLK = 50 MHz, CNT_MAX = 9, and CLOCK_DIVIDER = 3 is considered. PWM period frequency resulting from above values is about 16 KHz [5].

In practice the power semiconductors do not switch immediately but takes a small time for transition. to handle this and to control the inverter without risk of damaging the power semiconductors it become necessary to consider the transition delay of power semiconductors to switch-on and off. To deal with this problem, a dead-band is inserted between the turning on-off of the PWM Top and Bottom signal. Thus both PWM outputs remain off to allow the power semiconductors time to switch. Dead band value is made configurable in the design to set it according to the power stage requirement.

IV. SINE WAVE GENERATION

In this design sine table skipping method using sine lookup table is employed. Sine lookup table contain 256 entries corresponding to 360° of the sine wave. Each lookup table entry is g 16-bits wide. As three phase induction motor requires three sine waves with each 120° out of phase from the others, three separate indexes are used into the lookup table. By sequentially fetching values from the table and assigning them to the PWMs, sinusoidal output wave form is generated. The process involve integer value based sine lookup table and thus it is an approximation of a sine wave. The frequency of the sine wave can be increased by skipping

entries in the sine lookup table. The sine wave thus generated has a constraint on minimum and maximum frequency and is depend upon the number of entries in the sine lookup table and the PWM period.

This constraint on minimum and maximum frequency can be resolved by increasing the no. of entries in a sine lookup table. But, large lookup tables require additional space required within the FPGA device and is undesirable for low cost applications. One of the solution is to use linear interpolation with a 256 entry lookup table.

V. FLC DESIGN AND IMPLEMENTATION

FLC is adopted for numerous industrial application designs because of its many advantages. Flexibility offered by a fuzzy logic is a major reason. The strength of FLC is embodied in a set of fuzzy rules, with two implications: 1) 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 efforts, simply by redefining the rules. 2) 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.

FLC typically consists of four principal components: 1) fuzzifier which converts a crisp input into a fuzzy term set; 2) fuzzy rule base which stores fuzzy rules describing how the fuzzy system performs; 3) fuzzy inference engine which performs approximate reasoning by associating input variables with fuzzy rules; and 4) defuzzifier which converts the fuzzy output to a crisp value for the actual system input over the range. The control performance is more or less influenced by the selection of fuzzy sets of linguistic variables, the shapes of membership functions, the fuzzy rule base, the inference mechanism, and the defuzzification method.

To design FLC for speed control of induction motor three types of membership functions – Γ-function, L-function and Λ-function are utilized. These functions are known to produce good results for control applications and are relatively easy to implement on FPGA. 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). Fig. 3 shows representations of the membership functions. The crisp values of the input variables are mapped on the fuzzy plane using the equations above. It gives each input variable a membership function relating to the fuzzy sets (B_{1i} to B_{5i}). It has to be pointed out that in these equations; B_{Ji} is used to denote the linguistic value as well as membership function. The universe of discourse of the output variable is divided into nine linguistic Values.

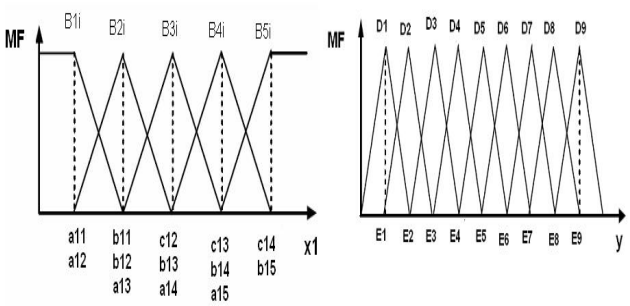

Fig. 3: Input-Output Membership Functions

To simplify the defuzzification computation the membership functions of the output values are intentionally made symmetrical as E_1 to E_9 are the mean of each function and are used in the calculation required for weighted average method of defuzzification. Each input variable can take any of the five linguistic values, therefore 25 rules are derived. The fuzzy rules are expressed in the typical IF-THEN structure, using linguistic variables in both the antecedent and consequent. Fuzzy rules map the input states to 25 output states (C_1 to C_{25}). Thus fuzzy rules are generalized in the following form:

$$R_n: IF\ x_1\ is\ A_{1K}\ AND\ x_2\ is\ A_{2K}\ THEN\ y\ is\ C_K \qquad (2)$$

Thus deriving set of fuzzy rules forms a rule base and can be represented by a fuzzy associative memory (FAM) table (Table 1). The FLC proposed in this design employ Mamdani's implication method of fuzzy inference, which is known as the most accepted methods in fuzzy control applications. Fuzzy rule implication using Mamdani's inference method is given by:

$$\mu_c(y) = max[min[\mu_{A1K}(x_1), \mu_{A2K}(x_2)]] \quad k=1,2,..25 \qquad (3)$$

The implication has a simple max–min structure which makes it easy to incorporate into hardware.

The block diagram in Fig. 4 represents an overview of the FLC's internal operational structure [7]. Here two input variables are fuzzified, producing the corresponding linguistic values and membership functions B_{ik}. As the antecedent pairs in the fuzzy rule structure are combined with a logical 'AND', min-operation on all the fuzzyfied inputs are applied in the first phase. In the second phase all the rules are aggregated using a max-operation. The weighted average method is considered as a suitable technique for hardware design using FPGA implementation. Since the output membership functions are symmetrical in nature, the mean of the fuzzy sets can be used as weights for the defuzzification process. This technique requires several multiplication by-a-constant operations and a single division operation. In the process of designing a fuzzy controller for speed control of induction motor; classical PI control strategy is taken as an initial reference and then essence of PI strategy is transferred to fuzzy domain.

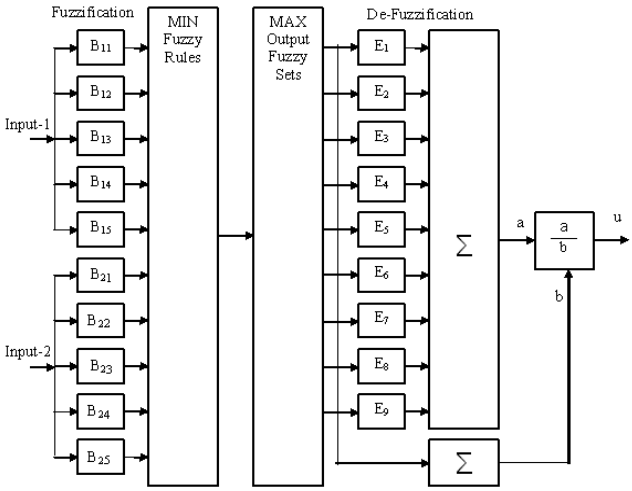

Fig. 4: Block diagram of fuzzy inference system

Following parameters are considered for design [7]:

Motor rated speed is 1420 rpm

Let *E* be the linguistic variable for the error *e*

ΔE be the linguistic variable for the change of error Δe *and*

U be the linguistic variable for the control output *u*.

The maximum range of motor is +/- 1420 rpm. The possible error range is -710 to 710 rpm. The universe of discourse of the change in error is based on experiment data from conventional controller which gives the range of error is +/- 140 rpm.

Controller output ranges from +/-8 rpm. Linguistic variables for error (LE), change in error (LΔE) and output (LU) are defined as:

LE = {Negative Large (-710 rpm), Negative Small (-355rpm), Zero (0), Positive Small (355rpm), Positive Large (710 rpm)}

LΔ*E* = {Negative Large (-140rpm), Negative Small (-70 rpm), Zero (0), Positive Small (70), Positive Large (140)}

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

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

R_K: IF x_1 is A_{1K} AND x_2 is A_{2K} THEN y is C_K

Where: A_{1K} can take any linguistic value in the set *LE*

A_{2K} can take any linguistic value in the set *L*Δ*E*

C_K can take any linguistic value in the set *LU*.

To implement the design using FLC real world input variables error (E) and change in error (ΔE) are mapped to FLC inputs x1 and x_2 such that x_1 = E, x_2 = ΔE and {B_{i1}, B_{i2}, B_{i3}, B_{i4}, B_{i5}}= {Negative Large(NL), Negative Small (NS), Zero(Z), Positive Small (PS), Positive Large (PL)}, for i =1, 2.

{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9} = {Negative Very Large (NVL), Negative Large (NL), Negative N), Negative Small (NS), Zero (Z), Positive Small (PS), Positive (P), Positive Large (PL), Positive Very Large (PVL)}.

Table I: FAM table for FLC

FAM Table		Change in Error x_2 (ΔE)				
		NL (B_{21})	**NS (B_{22})**	**Z (B_{23})**	**PS (B_{24})**	**PL (B_{25})**
Error x_1 (E)	**NL (B_{11})**	NVL (R_1)	NL (R_6)	N (R_{11})	NS (R_{16})	Z (R_{21})
	NS (B_{12})	NL (R_2)	N (R_7)	NS (R_{12})	Z (R_{17})	PS (R_{22})
	Z (B_{13})	N (R_3)	NS (R_8)	Z (R_{13})	PS (R_{18})	P (R_{23})
	PS (B_{14})	NS (R_4)	Z (R_9)	PS (R_{14})	P (R_{19})	PL (R_{24})
	PL (B_{15})	Z (R_5)	PS (R_{10})	P (R_{15})	PL (R_{20})	PVL (R_{25})

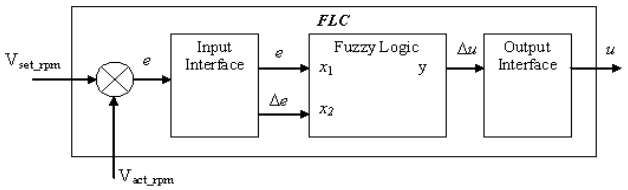

Fig. 5: Block diagram of FLC based control system

Fig. 5 shows a block diagram representing the implementation of the FLC in a speed control of three phase induction motor. In this design, the speed error (e) between set speed (Vset_rpm) and the actual motor speed (Vact_rpm) is calculated. This error is then fed to and processed by an input interface to generate the change in speed error signal (Δe). The two generated signals are used as two inputs x_1 and x_2 to the fuzzy logic processing where fuzzy rules are applied. Result of fuzzy logic processing is then given to output interface which converts the output (y) into the appropriate value (u) required by the PWM generator to drive power module. Function of interfacing blocks can be summarized as follows:

Input interface: $e=V_{set\_rpm} - V_{act\_rpm}$; $x_1 = e$; $x_2 = x_1 - x_1 z^{-1}$,

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

VI. FPGA IMPLEMENTATION

The target device used for implementation of this design is Xilinx SPARTAN3 XC3S 400E FPGA. The design is divided in to 7 VHDL modules viz. 1) Input interface, 2) Fuzzyfication module, 3) Fuzzy Inference Module, 4) De-Fuzzyfication module, 5) Output interface module, 6) Sine wave generation module, 7) PWM generation module. Each module is illustrated with associated VHDL code. Input and output variables are designed with 8 bit resolution. Xilinx ISE 8.1i tools are used for complete development of the design. First, code Functionality is verified using behavioral simulation. The timing is specified by entering constraints that guide the placement and routing of the design by

entering global constraints. The clock period constraint specifies the clock frequency at which the design must operate inside the FPGA. Then the design is implemented and constraints are verified. The inputs and outputs of the design are mapped to the physical I/O pins of the FPGA by explicit specification. Design is then re-implemented and verified using timing simulation. Finally the design is downloaded to the Spartan™-3 board. The experimental set up for this work consists of a three phase induction motor, FPGA kit, integrated power module and DC power supply. Motor used is of three phase induction motor with ratings: 415 Volts, 1.1KW, 1420 rpm.

For development and testing purpose Xilinx SPARTAN3 XC3S 400E FPGA daughter kit is used. After getting desired testing results; the system can be implemented stand alone system.

VII. SIMULATION AND CONCLUSION

The design under consideration is tested with simulation on Xilinx ISE 8.2 software. During testing, sine wave generation and PWM generation modules are tested. The simulation test result is as shown in Fig. 6.

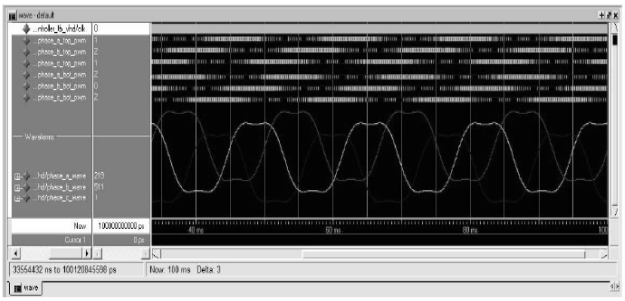

Fig. 6: Simulation result of sine wave & PWM generation

Simulation result shows that logic based PWM generation provides acceptable results and phase signals are never active at the same time, a configurable amount of dead-time is inserted to allow the power semiconductors time to switch. Other modules are designed and under test for hardware/software integration test.

REFERENCES

[1] Bimal K. Bose, "Modern Power Electronics and AC Drives," Pearson Education.
[2] Leonhand.W, "Control of Electrical Drives," Springer Verlag, 1990.
[3] Peter Vas, "Artificial –Intelligence based electrical machines and drives," Oxford university, 1999.
[4] Chuen Chien Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part1,2," IEEE, 1990.
[5] Craig Hackney, "Logic-Based AC Induction Motor Controller, Application Note" from Xilinx Inc.
[6] Vranda Parkhi, "FPGA Implementation of PWM control Technique for three phase Induction motor Drive," First International conference on Emerging trends in Engineering and Technology, DOI 10.11091 ICETET.2008.115, pp 996-1001.
[7] G.Sakthivel et. Al, "Real Time Implementation of a Fuzzy Logic Controller on FPGA Using VHDL for DC Motor Speed Control", International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4511-45.

AUTHOR'S PROFILE

R. P. Dhobale received his Bachelor of Engineering from Walchand Institute of Technology, Shivaji University, Maharashtra, India in Electronics Engineering. He is Oracle Certified Professional Database Administrator from Oracle University, US. His area of interest is VLSI and Embedded Systems and he is currently perusing Master of Engineering in E&TC at K.K. Wagh Institute of Engineering Education & Research Nashik, India.

Prof. D. M. Chandwadkar received his Bachelor of Engineering in Electronics Engineering from Pune University Maharashtra, India. He has completed his Master of Engineering in Electronics and is a Professor and Head of the E&TC Engineering Dept. at K.K. Wagh Institute of Engineering Education and Research, Nashik, India. He has 26 years of experience in the field of teaching, research and industry and is an author of many National and International paper and book publications. He is a member of FIE (I), LMISTE, LMIETE and is a technical consultant for various projects. Currently the author is perusing Ph.D.