

## Voltage/Frequency Control of an Induction Motor Using FPGA

**R.Thejaswini**

Assistant Professor  
Electrical and Electronics Engineering  
VVCE, Mysore  
thejaswini7@gmail.com

**R.Sindhuja**

Assistant Professor  
Electronics and Communication Engineering  
NIE Institute of Tech, Mysore  
sindhujarajesh5@gmail.com

**A.P Yoganandini**

Assistant Professor  
Electrical and Electronics Engineering  
AIT, Bangalore  
yoganandini.parrie@gmail.com

**Abstract:** Three phase induction motors are the most common motors used in industrial motion control system due to its low-cost, simple and rugged design. Squirrel cage is the widely used type of induction motors. In this paper a V/f controller on a FPGA kit and test the speed control of the 3 phase squirrel cage induction motors. FPGA based variable frequency drive has been designed. The speed control program has been developed in Xilinx 13.1 software using VHDL language. This implementation provides wide range of speed control for the induction motors.

**Keywords:** Induction motors, V/f controller, speed control, FPGA.

### 1. INTRODUCTION

The most popular algorithm for the control of a three-phase induction motor is the open loop V/f control approach. The operation of induction motors in the so-called constant volts/hertz (V/f) mode has been known for many decades and its principle is well understood. With the introduction of solid state inverters the constant V/f control became popular and the great majority of variable speed drives in operation today are of this type. Microprocessor based controllers are more economical, but often face difficulties in dealing with control systems that require high processing and input/output handling speeds. Rapid advances in digital technologies have given designers the option of implementing a controller on a variety of Programmable Logic Device (PLD), Field Programmable Gate Array (FPGA), etc. FPGA is suitable for fast implementation controller and can be programmed to do any type of digital functions. There are three main advantages of an FPGA over a microprocessor chip for controller designing:

- An FPGA has the ability to operate faster than a microprocessor chip.

- The new FPGAs that are on the market will support hardware that is upwards of one million gates, which increase program capacity.
- Because of the flexibility of the FPGA, additional functionality and user interface controls can be incorporated into the FPGA minimizing the requirement for additional external components.

FPGAs are programmed using Very High Speed Integrated Circuit hardware description language (VHDL) and a download cable connected to a host computer. Once they are programmed, they can be disconnected from the computer, and it will be running as standalone device. The FPGAs can be programmed while they run, because they can be reprogrammed in the order of microseconds. This short time means that the system will not even sense that the chip was reprogrammed. Applications of FPGAs include industrial motor drivers, real time systems, digital signal processing, aerospace and defense systems, medical imaging, computer vision, speech recognition, cryptography, computer hardware emulation and a growing range of other areas greater understanding of the design and the characteristics of these motors.

### 2. DESIGNING CONTROLLER USING FPGA FOR THREE-PHASE INDUCTION MOTOR

The process used to design the open loop V/F controller for the three phase induction motor using FPGA technique is presented. The block diagram explains the concept of this system is shown in

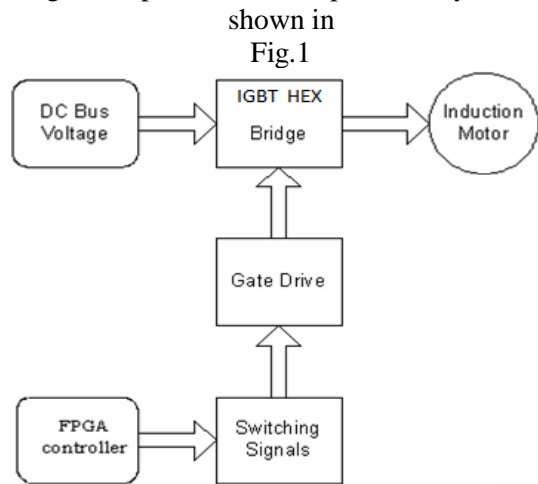


Fig1. System block diagram

### 2.1 DC Link Voltage Generation

The rectifier is a three-phase bridge rectifier as shown in Fig.2 with six diodes. Capacitor is used to remove the ripples in the output of the rectifier. The input and output voltage for three phase rectifier is shown in Fig. 3.

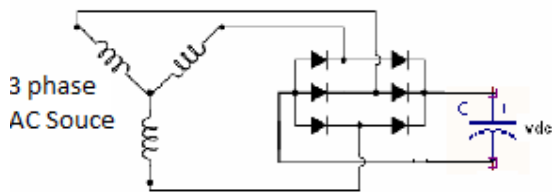


Fig2. Three-phase bridge rectifier with capacitor.

### 2.2 Inverter and Gate Driver Circuit

A voltage source power inverter is used to convert the DC bus to the required AC voltages and frequency. The power section consists of a power rectifier, filter capacitor, and power inverter. The motor is connected to the inverter as shown in Fig. 4. The power inverter has 6 switches that are controlled in order to generate an AC output from the DC input. Pulse width modulation PWM signals generated from the FPGA controller will control these 6 switches. The phase voltage is determined by the duty cycle of the PWM signals. In time, a maximum of three switches will be on, either one upper and two lower switches, or two upper and one lower switch. When switch turns on, current flows from the DC bus to the motor winding. Because the motor windings are highly inductive

in nature, they hold electric energy in the form of current. This current needs to be dissipated while switches are off. Diodes connected across the switches give a path for the current to dissipate when the switches are off. These diodes are also called freewheeling diodes. Upper and lower switches of the same limb should not be switched on at the same time. This will prevent the DC bus supply from being shorted. A dead time is given between switching off the upper switch and switching on the lower switch and vice versa, as will be explained in next section.

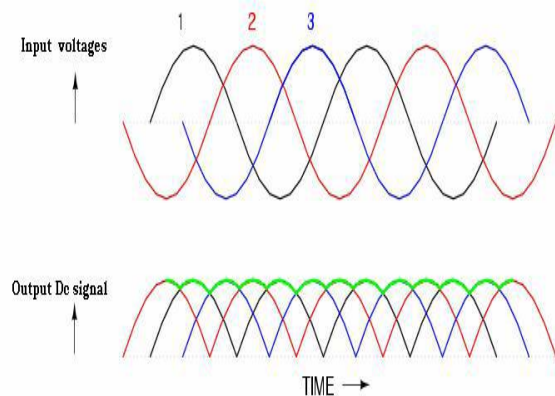


Fig3. Input and output signal for the three phase bridge rectifier

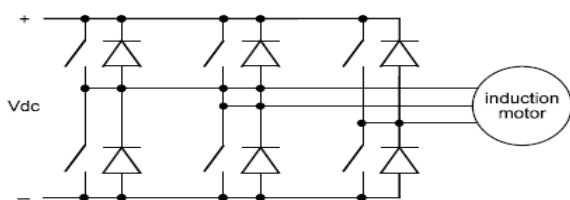


Fig4. Six switches and free-wheeling diodes used in the inverter.

Up to switching powers of a few kW MOSFET (Metal Oxide Silicon Field-Effect Transistors) are often used as switching devices. The MOSFET can be switched on and off by a low level signal requiring virtually no current, is robust and has excellent conduction when 'Switched on'. For powers up to about 10 kW, IGBTs (Insulated Gate Bipolar Transistors) are the economical power control devices shown in Fig. 5. For higher powers the only available devices are thyristors. The signal coming from the FPGA controller can't feed directly to the IGBT gate pin due to the weakness of its voltage (3.3v). So it's required to build a power driver circuit to provide the required voltage.

### 2.3 Firmware Implementation

Sparten 6 development board has 20MHz clock signal by default i.e. each clock cycle is 50ns.

The firmware code is developed based on this frequency. VHDL language is used to program the Spartan-6 FPGA. Xilinx ISE Design Suite 13.1 is used as software to write coding. ModelSim SE 6.3f is used for simulation purpose.

Following are the different processes involved in generation of sine triangle PWM.

2.3.1. Generation of triangular carrier signal of 10 KHz.

Up Down counters are used to generate triangle signal counter counts for each clock cycle of 50ns. To get the frequency of 10 KHz each cycle

$$\frac{1}{10000} = 100\mu s = 100000ns$$

Therefore up/down counter is for

$$\frac{100000ns}{2} = 50000ns$$

The count value  $\frac{50000ns}{50ns} = 1000$  counts

2.3.2. Generation of Sine Wave

Sine wave is generated by lookup table method. Sine data for 360 degrees is stored in 256 locations each of size 9 bits including sign bit  $\sin 0 = 0$

Positive peak value is  $\sin 90 = 1$  for 9 bits it is equal to 255

Negative peak value is  $\sin 270 = -1$  for 9 bits it is equal to -255

A Ramp signal is generated by the up counter to take the samples of sine data from look up table. Samples are taken based on the frequency step applied.

2.3.3. Frequency Step Calculation

Example for 50Hz sine wave with Sampling frequency = 10 KHz. It means that 100000 samples are taken in one second. For 50 Hz the number of samples required for one full cycle of sine wave is equal to  $\frac{10000}{50} = 200$  samples. After 200 samples the signal has to repeat for this we need another ramp signal (i, y and b of program) of 0 to 255 to point 256 locations.

Therefore frequency step

$$\frac{256 \cdot 256}{200} = 327$$

To get 3 phases of 120 degree phase displacement.

Ramp signal is from 0 to 255  $255/3=85$

Ramp signal i starts from 0

Ramp signal y starts from 85

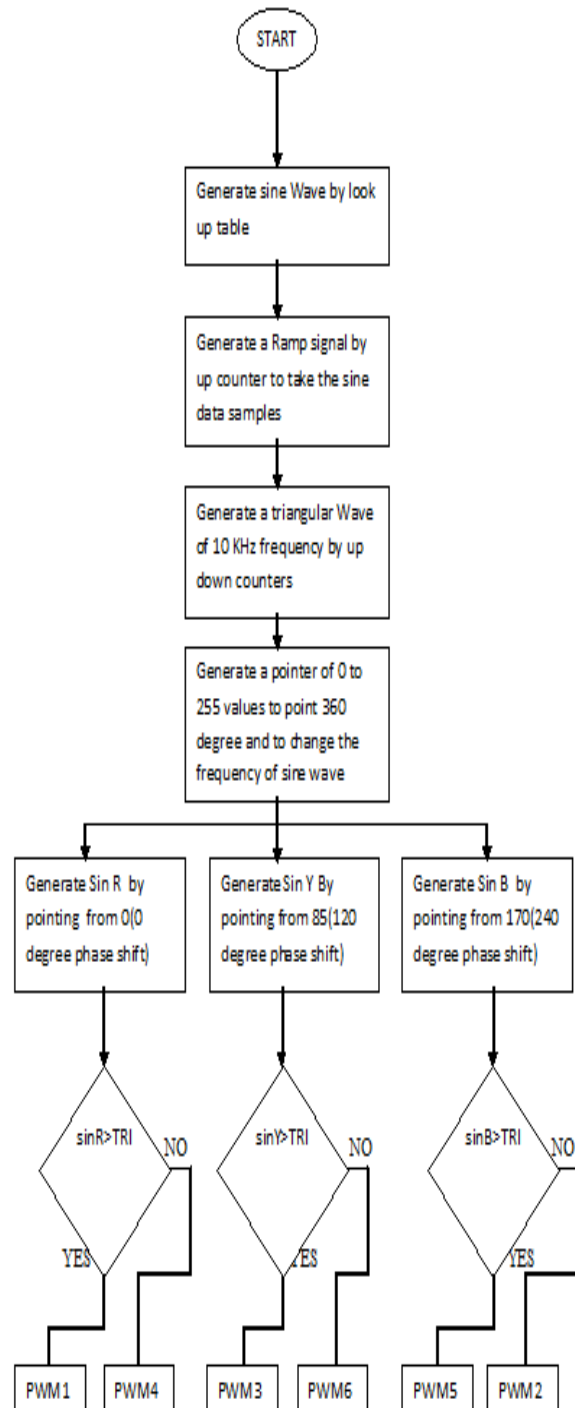
Ramp signal b starts from 170

Magnitude can be varied by multiplying with an integer value.

3. PWM GENERATION

PWM signals are generated by comparing magnitudes of each sine wave form with carrier signal.

3.1 Flow Chart



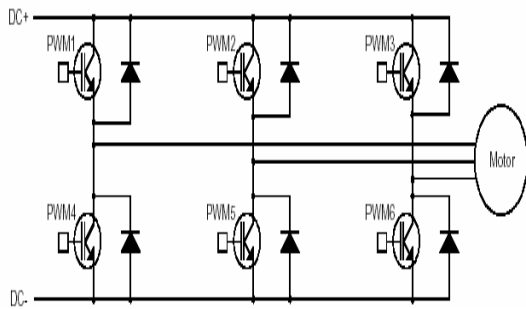


Fig5. IGBTs works as switches in the inverter.

### 3.2 Results and Discussions

Here simulation and the experimental results are discussed. The program is developed in VHDL language in Xilinx ISE Design Suite 13.1 to produce the PWM pulses. The program is verified using ModelSim SE 6.3f simulator for various amplitude and frequency modulation ratios. After programming the FPGA the practical wave forms are observed by using CRO. These signals are fed into gate driver circuit and at the terminals of IGBT's gate, the signals are observed using CRO. Finally the motor is connected and the readings are taken for different values of modulation ratios.

VHDL is designed to fulfill a number of needs in the design process. Firstly, it allows description of the structure of a design that is how it is decomposed into sub-designs, and how those sub-designs are interconnected. Secondly, it allows the specification of the function of designs using familiar programming language forms. Thirdly, as a result, it allows a design to be simulated before being manufactured, so that the designers can quickly compare alternatives and test for correctness without the delay and expense of hardware prototyping.

#### 2.3.4. Simulation Results

The speed control program is developed on Xilinx 13.1 software in VHDL language. The program is verified with Modelsim simulator. Model sim has the feature of forcing clock and other inputs. The following are the graphs obtained for different values of inputs (amplitude and frequency) The above figures show the Modelsim results for the different modulation ratios of amplitude and frequency. As the technique used is open loop V/f control its necessary to vary both amplitude and frequency. The results show the generated three phase variable amplitude and frequency signal.

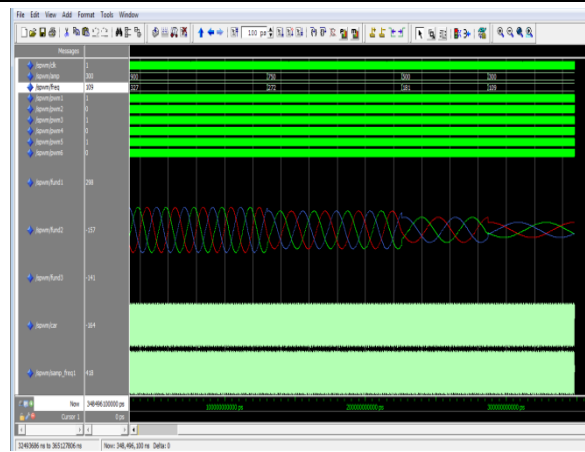


Fig6. Wave forms of control signal for different frequency and amplitude

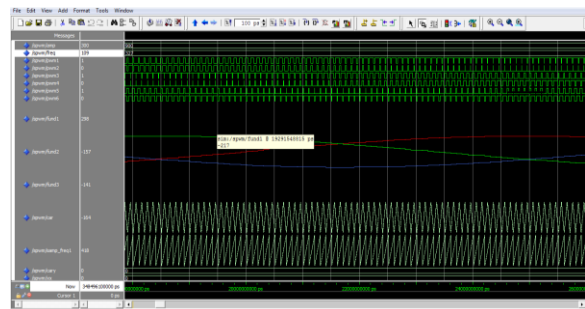


Fig7. Wave forms for  $m_a = 0.9$  and  $m_f = 200$

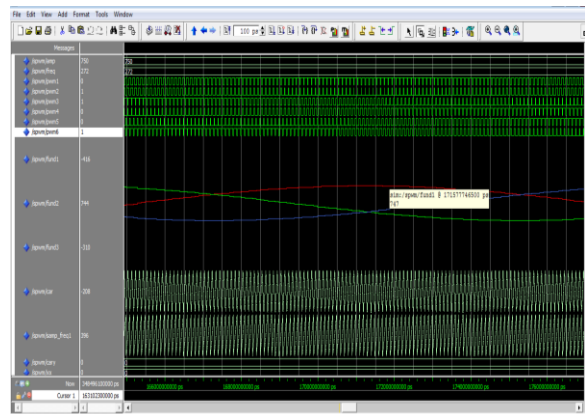


Fig8. Wave forms of for  $m_a = 0.75$  and  $m_f = 244$

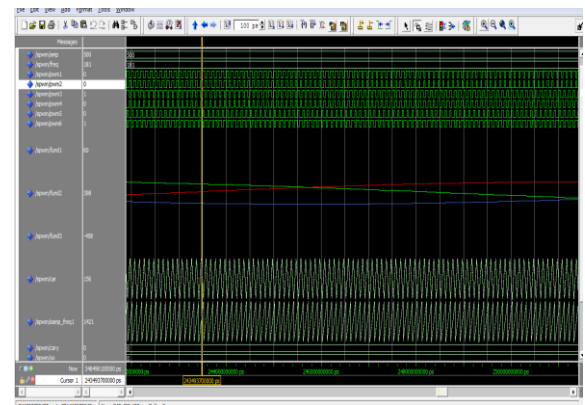


Fig9. Wave forms for  $m_a = 0.5$  and  $m_f = 358$

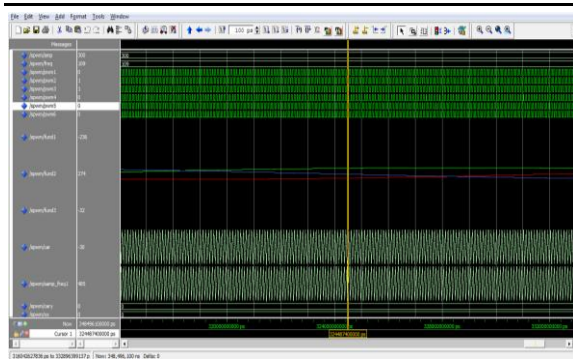


Fig10. Wave forms for  $m_a = 0.25$  and  $m_f = 714$

With reference to Fig. 6 it is clear that both the amplitude and frequency are varying in three phase sine waveform which is also called modulating signal. Modulating signal is compared with the triangular carrier wave of 10 KHz. The PWM signals are generated by comparison of both signals. Hence the width of the PWM signals also changes with the amplitude and frequency variation of the modulating signal.

Fig. 7 to Fig. 10 represents the different simulation results for different frequency and amplitude modulation ratios. From the Modelsim results it is evident that we are getting the required pulse width modulation from the program.

### 3.3 Experimental Results

To conduct the experiment the necessary DC voltage is taken through auto transformer and the bridge rectifier with the filter capacitor. The programmed FPGA kit is connected to the gate driver circuit through buffer to increase the voltage from 3.5v to 12v. The PWM signals from the kit are first verified with the CRO for different frequency and different amplitudes of the modulating signal. The dead band required is allotted in the program itself.

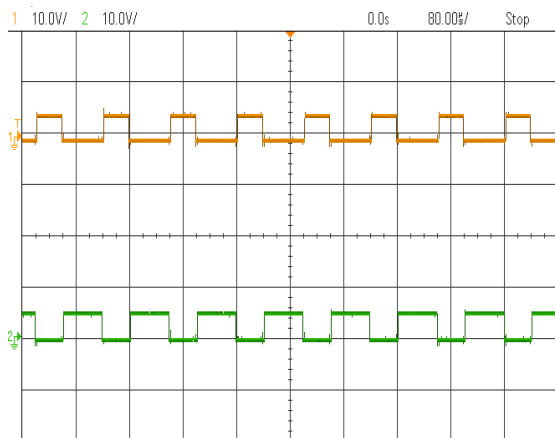


Fig11. CRO results obtained at FPGA pin outs.

Fig. 11 shows the pulses obtained after programming the kit. These pulses are fed to top and bottom IGBT's of one leg of inverter through the gate driver circuit. In Fig. 11 both signals are inverted of each other i.e. top and bottom IGBT's will not turn ON at the same time.

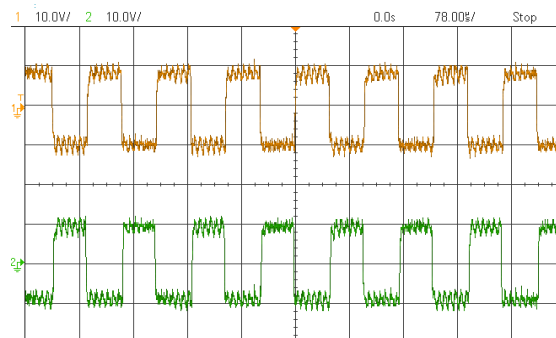


Fig12. CRO results obtained at the gate terminals of IGBT's

Fig 12 shows the CRO results captured at the gate terminals of IGBT's of first leg. The signal is processed in the driver circuit for the required voltage level and isolation then it is fed to the gates of IGBT's. This shows increase in voltage to the required level of around 11V can be seen. To avoid the short circuit of IGBT's a dead time of 2.5µSec is included in the program itself. The Fig.13 shows the dead time between two signals.

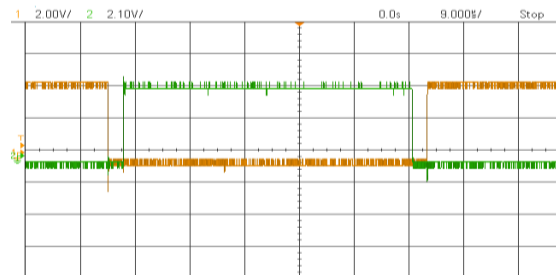


Fig13. Dead time between two signals

### 3.4 Test Results

Motor Specification: 3 Phase, 3.5HP, 415V, 1350RPM, Squirrel cage Induction Motor. Readings taken at  $V_{dc} = 300V$  from the above table it is clear that the speed is changing according to the set speed. As the system is open loop we are getting the results near to the set speed. The speed control is effective from 300 RPM to 1350 RPM. If the set speed is less than 300 RPM the magnitude of the modulating signal is very less and it is insufficient to drive the motor. The set speed is varied in the steps of 100 RPM speed. The value of voltage and frequency changes in accordance with the set speed which is written in the program in such a way that v/f ratio is constant. The inference from

the above results is that the smooth control of speed is possible with FPGA and the results are closer to the set speed.

**Table1.** Results Table

S.No	Set Speed (RPM)	Actual Speed(RPM)
1	300	291
2	400	390
3	500	494
4	600	590
5	700	694
6	800	792
7	900	890
8	1000	992
9	1100	1090
10	1200	1185
11	1300	1282
12	1350	1340

#### 4. CONCLUSION AND FUTURE WORK

From the results it is clear that it is possible to implement the v/f control for induction motor using FPGA as controller. The FPGA is programmed to produce sine triangle PWM. The firmware code is verified by using Modelsim simulator and the results are noticed. The pulses are verified from CRO before giving it to inverter. As the FPGA's has several advantages, the controller will be more advanced and easy to reprogram. It is possible to implement vector control strategies in future on FPGA's which can give better results compared to ASIC's or DSP's. .FPGA is not only for induction motor speed control we can control BLDC motors also by writing the required control algorithms inside the FPGA. Once it is programmed it can work as a standalone system.

#### REFERENCES

- [1] Modern Power Electronics and AC Drives, by Bimal K. Bose. Prentice Hall Publishers, 2001
- [2] Power Electronics by Dr. P.S. Bimbhra. Khanna Publishers, New Delhi, 2003. 3rd Edition.
- [3] Eftichios Koutroulis, Apostolos Dollas, Kostas Kalaitzakis "High-frequency pulse width modulation implementation using FPGA and CPLD ICs" Journal of Systems Architecture 52 (2006) 332–344., 25 October 2005
- [4] Ms. Deepali C. Shimpi. "FPGA Based PWM controller for Three Phase Inverter" International Journal of Science and

Advanced Technology (ISSN 2221-8386), Volume 1 No 8 October 2011

- [5] Ali. M. Eltamaly " FPGA Based Speed Control Of Three-Phase Induction Motor Using Stator Voltage Regulator" Electrical Eng. Dept., College of Eng., Elmansoura University, ElDakahlya, Egypt.
- [6] Gustavo G. Parma, and Venkata Dinavahi, "Real-Time Digital Hardware Simulation of Power Electronics and Drives" IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 22, NO. 2, APRIL 2007
- [7] Marcian N. Cirstea, and Andrei Dinu, "A VHDL Holistic Modeling Approach and FPGA Implementation of a Digital Sensorless Induction Motor Control Scheme" IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 4, AUGUST 2007
- [8] Bahram Rashidi and Mehran Sabahi "High Performance FPGA Based Digital Space Vector PWM Three Phase Voltage Source Inverter" IJ.Modern Education and Computer Science, 2013, 1, 62-71 Published Online January 2013 in MECS
- [9] Modern Power Electronics and AC Drives, by Bimal K. Bose. Prentice Hall Publishers, 2001