NTNU – Trondheim
Norwegian University of
Science and Technology

Low switching frequency modulation scheme for high power three level converters

FPGA based implementation

Biruk Bekele Yenore

Master of Science in Electric Power Engineering
Submission date: June 2013
Supervisor: Tore Marvin Undeland, ELKRAFT

Norwegian University of Science and Technology
Department of Electric Power Engineering

NTNU – Trondheim
Norwegian University of
Science and Technology

Low switching frequency modulation scheme for high power three level converters: FPGA based implementation

Biruk Bekele Yenore

Master of Science in Electric Power Engineering

Submission date: June 2013

Supervisor: Tore M. Undeland (Professor), ELKRAFT

Co- supervisor: Roy Nilsen (Professor), Wärtsilä Norway AS

Norwegian University of Science and Technology

Department of Electric Power Engineering

Low switching frequency modulation scheme for high power three level converters: FPGA based implementation

by

Biruk Bekele Yenore

A thesis submitted to

Department of Electric Power Engineering
Norwegian University of Science and Technology

in partial fulfillment of the requirements for the degree
Master of Science in Electric Power Engineering

June 2013

Problem Description

Increased power rating and higher efficiency are some of the qualities a modern application requires from the electric motors it employs. Fulfilling these requirements needs a different approach in the PWM techniques, and a new modulation technique should be proposed.

In this master's thesis the student should:

- Work out a specification of the FPGA firmware for the proposed modulator together with a fellow master's student
- Decide which method is to be used for dc-bus balancing for the 3-level converter
- Calculate the optimal switching angles as functions of modulation index and pulse number N pr. half period.
- Implement the software for this optimal synchronous modulator.
- Test the modulator in the Lab with the help of a 3-level 3-phase inverter operating an Induction Machine.

Assignment given: 23. Jan 2013

Supervisor: Tore M. Undeland, Elkraft

Co-supervisor: Roy Nilsen, Wärtsilä Norway AS

Preface

I would like to thank almighty God for his protection and providence throughout my life. My sincere thanks go to Prof. Tore M. Undeland for his invaluable support not only on this thesis but also through other academic and professional advices.

Roy Nilsen deserves an honor for his incredible support throughout the whole project starting from the conceptualization of the project idea. He treated me as one of his sons; guided me through my entire journey as a student here at NTNU, and he was always there for me when I needed his say.

I would also like to thank Mr. K. Ljøkelsøy who has never given up answering my questions concerning practical matters. He even used some of his precious time in the lab figuring out the ways to solve problems that I faced. So, I don't take his immense support for granted. I am also thankful to Sverre Skalleberg Gjerde for his support in the laboratory and at many critical times when I desperately was in need of his inputs.

Mannat, my fellow student in this master's thesis, showed a great deal of courage and passion to get things done. We have had good communication to have solved this technically challenging task. She deserves my deepest thanks. The technical staffs in the department of Electric Power Engineering have also contributed for the completion of this thesis work, and I owe thanks to them too. I want to thank my friends in Trondheim especially Gizaw, Taddese, Tadewos and Tesfaye for the unforgettable social time we have had together.

Finally, I want to say thank you to my family for their love and encouragement.

Biruk Bekele Yenore

Trondheim

Norway

Abstract

The demand for electrical motors with high power ratings is increasing in different applications, especially in the offshore industry. The power rating of these motors can be raised by increasing the voltage at the motor terminals to a higher value. This method has the drawback of increasing the losses due to harmonic currents although it fulfills the requirement of increasing the power. An equally important reason to look for a more efficient modulation strategy is that it is not efficient to operate the medium voltage converter at the same switching frequency as the conventional low voltage converters because of high switching losses. To tackle these problems, the PWM technique should be implemented in such a way that it produces the lowest possible harmonic content in the load current. A low switching frequency implementation of the proposed PWM strategy is also an important requirement to alleviate the mentioned problems. One of the ways to do this is programming the pulse patterns so that the total harmonic content fulfills the maximum tolerable harmonic content.

In this master's thesis, techniques to produce the optimal pulse patterns for the motors were studied. First, the mathematical relations that govern the relation between optimal patterns and the required output voltage level were dealt with.

Second, a brief study of the Digital control technologies for the implementation of the modulation principle was made with focus on Xilinx FPGA. The flexibility of FPGA based system together with its ability to execute instructions at hardware level made it an ideal choice for the implementation of the proposed modulation scheme. After having mastered the overall development environment of Xilinx FPGA through the studies, the project design, both software and hardware, for the problem in hand was done using the tools.

Last, the software program that produces the optimal patterns for a certain variation of number of pulses and modulation indices was successfully developed on the FGPA processor.

Finally, an investigation was done on the output waveforms produced by the optimal patterns. It was observed that the pattern change happens as desired, and the optimal patterns produce the required sinusoidal waveform with harmonics less than that which would occur in conventional PWM techniques.

Table of Contents

List of figures

Acronyms and symbols

WTHD	Weighted Total Harmonic Distortion
WTHD0	Normalized Weighted Total Harmonic Distortion
DC	Direct Current
PWM	Pulse Width Modulation
FPGA	Field Programmable Gate Array
N	Number of pulses per half of fundamental period
M	Number of pulses per full fundamental period
ASIC	Application Specific Integrated Circuit
EDK	Embedded Development Kit
DSP	Digital Signal Processors
FOC	Field Oriented Control
CLB	Configurable Logic Block
RAM	Random Access Memory
I/0	Input Output
SDK	Software Development Kit
XPS	Xilinx Platform Studio
IP	Intellectual Property
VHDL	Very Large Scale Integrated Circuit Hardware Development Lang
BSP	Board Support Package
PC	Personal Computer
DRAM	Dynamic Random Access Memory
DDR2	Double Data Rate
VSC	Voltage Source Converters
EEPROM	Electrically Erasable Programmable Read Only Memory
MSPS	Mega Samples Per Second
GTO	Gate Turnoff Thyristors
SHE	Selective Harmonic Elimination
EMI	Electromagnetic Interference
SVM	Space Vector Modulation
NPC	Neutral Point Clamped
AD	Analog to Digital

DA	Digital to Analog
APU	Auxiliary Processor Unit
IGBT	Insulated Gate Bipolar Transistors
MHS	Microprocessor Hardware Specification
JTAG	Joint Test Action Group
SPLB	Slave Processor Local Bus
MPLB	Master’s Processor Local Bus
KHz	Kilo Hertz
Ucf	user constrained file
m	Amplitude modulation index
f_m	frequency modulation index
V	Voltage

1. Introduction

In this chapter, a study of the harmonic losses in an induction machine is presented. The need for a new modulation strategy for medium voltage drives is analyzed as well. A brief theoretical background for the problem under study is presented together with some relevant literature review work.

1.1. Background

Inverter fed ac machines are increasingly becoming common in variable speed drive market, mainly due to the recent improvements in the power semiconductor devices and fast digital signal processing hardware [6]. Based on the requirement of their control methods, ac drives can fall in to one of two basic categories:

- Those that need reduced dynamic requirements such as standard industrial drives, or pump and fan drives operating at a maximum efficiency point. These are usually low cost solutions which operate on constant volts per hertz ($^{V}/_{f}$) control [6].
- High dynamic performance drives on the other hand, those that incorporate more complex signal processing structure which permits an independent control of the machine torque and flux, are used in positioning systems, elevators and robot drives [6].

Variable speed ac drives for high power applications usually employed current source inverters in the earlier days. In medium voltage drives, which are widely used in ac traction systems and electric ship propulsion, the introduction of high power voltage source inverters is relatively eased due to the favorable properties of GTOs such as highest power handling capability. Despite the above mentioned merits of employing GTOs in high power ac drives, the switching loss of these devices is generally high, and consequently the switching frequency must be low, preferably lower than few hundred hertz. Due to this limitation of the switching frequency, the main objective of the present ac drives studies is to minimize the load current distortions and the associated harmonic losses, and to improve the dynamic performance while still having smaller switching frequency.

Finding solution to the above contradicting requirements namely, reduced switching frequency and lower current harmonic level, requires looking at the different components of the drive system in general. One of the interesting solutions that can minimize the problem is employing a high performance Pulsewidth modulation scheme.

A high performance Pulsewidth modulation scheme is a modulation scheme that, in addition to being able to address the above contradicting requirements, also possesses the following properties:

- Having a feedforward structure
- Synchronization between the switching frequency and fundamental frequency
- A large bandwidth current control system
- Fast response in the pulsewidth modulation

This master's thesis focused on design and test of a novel PWM method that considerably decreases harmonic losses in inverter fed ac machines.

1.1.1. Pulse Width Modulation techniques

Pulsewidth modulation (PWM) techniques are used in forced commutated inverters feeding variable-speed ac motor drive systems to control the amplitude of the first harmonic of the output voltage. It is also possible to reduce the harmful effects of other higher order harmonics on the motor by manipulating the PWM patterns. [18]. There are different ways of implementing the PWM techniques in order to achieve the objective of reducing the harmonic effects. Those mostly used are:

- The suboscillation method: This method, where the commutation angles are set at the crossings of two waveforms, is characterized by the simple implementation of the inverter control apparatus. Several commutations, however, are required to keep down the harmonic effects [18].
- The Elimination method: In this technique, the commutation angles are chosen in a straightforward manner, according to the criterion of eliminating a number of harmonics. The control equipment is complex, but satisfactory results are obtained even with limited number of commutations [18].
- A third approach, which this master's thesis based its studies on, incorporates defining a performance index and achieving its minimization by a suitable choice of the switching patterns. The performance index can be as general as a cost function that is

related to the dynamic behavior of the state variables of the load supplied by the inverter, or as specific as the rms value of current harmonics on the motor.

1.1.2. Harmonics in induction machines

In these days where most of the electricity produced in the world is consumed by motor loads out of which induction motors consist the bulk, study of harmonic losses in these loads is an interesting task both to increase the efficiency of our loads and also to reduce the extra burdens that come due to the unwanted harmonics such as torque oscillation, overcurrent and need for extra heat sinks. The main source of these harmonics in voltage source Inverter fed motors is the non-sinusoidal output of the inverters where the semiconductor switches of the converter do not switch sufficiently high to produce a sinusoidal output.

The harmonic components of a converter are not evenly distributed over the entire frequency spectrum [1]. i.e., the lower frequencies have the highest harmonic components. Therefore, a performance indicator that gives the lower frequency spectra of the harmonic components more weight is required to analyze the total harmonic distortion. Weighted Total harmonic Distortion (WTHD) is one measure that can satisfy the above need.

WTHD can be calculated by normalizing the total current harmonic distortion by the maximum inrush current [9] as it can be seen from equation (1.1).

$$WTHD = \frac{\sqrt{\sum_{n=2}^{\infty}(\frac{Un}{n})^2}}{U1} \qquad (1.1)$$

From equation (1.1), it is clear that the lower order harmonics account the highest proportion due to their high $\frac{1}{n^2}$ factor compared to that of higher order ones. According to [7] and [9], the total harmonic losses (both stator and rotor) incurred by a machine are estimated, assuming constant machine parameters, to be:

$$3(WTHD0)^2\left({I_{1,inrush}}^2 r_1 + {I_{2,inrush}}^2 r_2\right) \qquad (1.2)$$

where WTHD0 is the same as WTHD but with U_1 replaced with the DC bus voltage to avoid the WTHD dependency of the modulation index, $I_{1,inrush}$ is the fundamental inrush stator current, $I_{2,inrush}$ is the fundamental inrush rotor current, r_1 and r_2 represent the stator and rotor per phase resistances of the machine respectively. Therefore, reduction of WTHD0 to the lowest tolerable value is the measure of the efficiency of modulation strategies with respect to minimizing harmonic losses.

In addition, harmonic currents also introduce unwanted torque pulsations in induction machines due to the interaction of the harmonic currents and the magnetic field of fundamental frequency component. The additional torque created by the harmonic currents has a magnitude proportional to the amplitude of the harmonic current producing it. These torque pulsations cause mechanical oscillations and eventual wear of the machine.

1.1.3. Switching losses in Voltage Source Inverters

The switching power loss of a VSC with a sinusoidal ac line current and with IGBT switches is estimated using the following relation [2]:

$$\frac{6}{\pi} f_s\left(E_{ON,1} + E_{OFF,1} + E_{OFF,D}\right).\frac{V_{dc}}{V_{ref}}.\frac{I_L}{I_{ref}} \qquad (1.3)$$

where f_s is the switching frequency, $E_{ON,1}$ and $E_{OFF,1}$ are the turn-on and the turn-off energies of the IGBT respectively, $E_{OFF,D}$ is the turn-off energy in the power modules' diode due to reverse recovery charge current, V_{dc} is the dc link voltage, I_L is the peak value of the ac line current assumed to be sinusoidal, and V_{ref} and I_{ref} are the reference voltage and current where the switching energies provided by data sheets are given.

As it can be clearly noticed from equation (1.3) above, the switching losses of a VSC are proportional to switching frequency. Moreover, as the operating voltage of the machine increases, which is the case in medium voltage drives, the switching losses also linearly increase. Hence, to reduce the switching loss, we have to either reduce switching frequency or decrease the voltage on the motor terminals. Due to the fact that the use of medium voltage converters is one of the few ways to increase the power rating of the induction machines, the switching losses can only be minimized by reducing the switching frequency. However, reducing the switching frequency would lead us to the old problem of harmonic distortion in the fundamental current component if any of the usual modulation techniques is to be employed for medium voltage drives. Hence, a new modulation approach that satisfies the requirements of both lower switching frequency and minimum WTHD0 should be developed and implemented.

1.1.4. Multilevel inverters

A large number of applications, an example being ship propulsion, are requiring higher power components in recent years. The main driving force for this is the objective of increasing the efficiency and reducing production cost. As it was pointed out in the background section above, the power rating of the components is raised by increasing the operating voltage of the system, and hence applications involving these components are called medium voltage applications. Due to the limited voltage handling capability of the semiconductor switches, it is troublesome to connect only one semiconductor switch to medium voltages. It is this motivation that led to the concept of multilevel inverters.

The concept behind multilevel converters is to use a series of power semiconductor switches with several lower voltage dc sources to perform the power conversion by synthesizing a staircase voltage waveform [20]. The high voltage at the output of the converter is an aggregate of these multiple dc sources while the rated voltage of the power semiconductor device depends only upon the rating of the dc voltage sources to which it is connected.

Some of the advantages offered by multilevel converters, according to [20] and [26], are:

- Staircase waveform quality: This feature ensures
    - An output voltage of lower distortion
    - Reduced $\frac{dv}{dt}$ stresses which reduces electromagnetic compatibility problem
- Smaller Common Mode (CM) voltage.
- Input current with low distortion
- Lower switching frequency resulting in lower switching losses, which in turn means higher efficiency

One particular disadvantage of employing multilevel converters is the number of power semiconductor switches needed and hence as a result the complexity of the drive circuit. There are different topologies for multilevel inverters some of which are: cascaded H-bridge converter, diode clamped converter and flying capacitors (capacitor clamped) converter.

In this master's thesis, a diode clamped three level inverter is used as a basis for the development and test of the modulation strategy, and all the software and hardware developed in the project are mainly based on this topology.

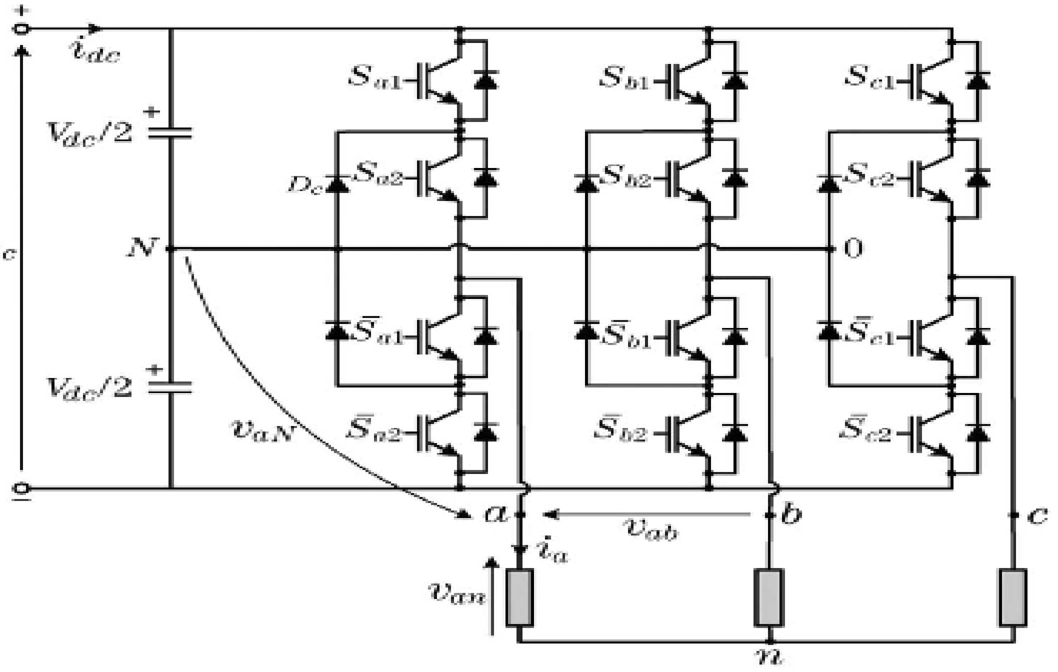


Figure 1-1: Power circuit of the three-level diode-clamped inverter (NPC) [21]

In Neutral Point Clamped (NPC) converters, DC link is built from two capacitors connected in series, thus dividing the grid voltage in to two, and these voltage levels can be switched to the output point. As it can be seen from the circuit in figure 1-1 above, there are three voltage levels that can appear at the output of the inverter: V_{dc} , $\frac{V_{dc}}{2}$ and 0(N). Considering phase a for example, when S_{a1} and S_{a2} are ON, the output voltage becomes V_{dc}, it becomes 0 when $\overline{S_{a1}}$ and $\overline{S_{a2}}$ are ON and $\frac{V_{dc}}{2}$ when $\overline{S_{a1}}$ and S_{a2} are ON. Therefore, unlike two level inverters where there are only 2 levels on the output voltage of the inverter, it is possible to have three levels in three level inverters which in turn reduces the harmonic distortion of the output voltage. It is also worth noting that the driving signals for the pairs of complementary switches $\overline{S_{a1}}$ and S_{a1} or $\overline{S_{a2}}$ and S_{a2} are a negation of one another to avoid short circuiting the dc-bus which would occur if any pair of the complementary switches is turned ON simultaneously.

There are different modulation techniques that can be employed to multilevel inverters each with its merits and drawbacks. The most widely used are:

- *Carrier-Based Three-Level PWM Modulation*: this method is based on comparison of a reference voltage signal with two sinusoidal carriers. Being simpler to implement, the technique is the most widely used one but has a drawback of difficulty to implement DC-bus balancing algorithm.

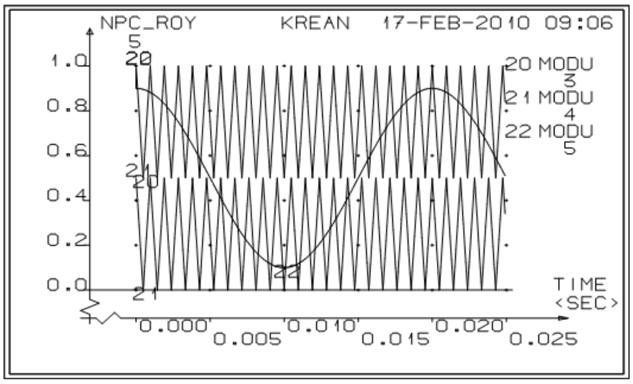


Figure 1-2: Naturally Sampled PWM For 3 Level Converter [17]

- *Space Vector modulation*: A three phase three-level inverter has $3^3 = 27$ different switching state combinations of all three phases which generate different voltage vectors. Some of these voltage vectors are redundant. That is, different switch combinations end up in the same voltage vectors, which is an attractive feature for DC-bus balancing even though the technique is complex to implement, compared to the previous one.

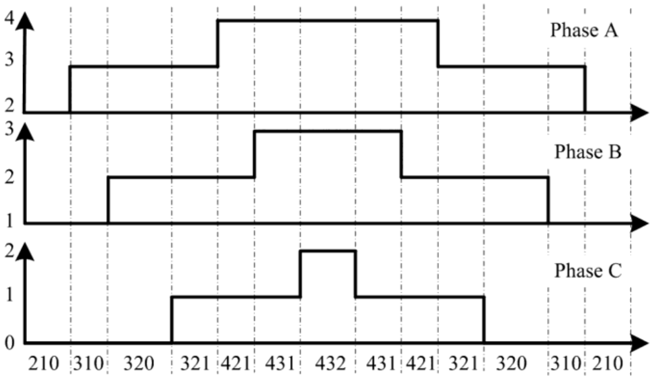


Figure 1-3: Space vector modulation

- *Programmed modulation*: In this method the switching time is calculated from the pre-programmed angle patterns stored in a look up table so that certain harmonics are eliminated or the total harmonic level is below a specific limit. Programmed modulation is particularly attractive for three level inverters because the equipment needs to run at low switching frequency to reduce the semiconductor losses. This method, due to the above mentioned advantage, is becoming increasingly popular as the voltage rating of the inverters increases. Even if the hardware to implement this method is more complex than the other methods, the increasing development of digital electronics makes it still feasible. This method is the subject of investigation in this thesis.

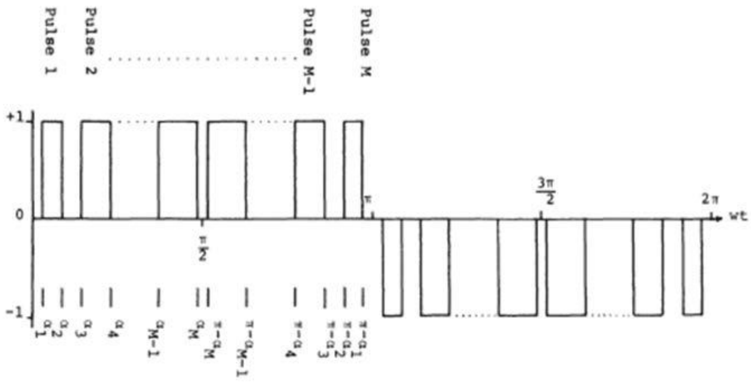


Figure 1-4: Programmed modulation

Harmonic content of 3-level Voltage Source inverters

Almost all dc-to-ac converters possess a certain amount of harmonics at their output due to a finite switching frequency of the semiconductor switches. The harmonic content of the output voltage can be analyzed using Fourier series decomposition of the voltage at the output of the converter [1] as shown below in equation (1.4).

$$V(t) = V_0 + \sum_{h=1}^{\infty} V_h(t) \qquad (1.4)$$

The output of Voltage Source Inverters has usually a zero average value so that the dc part of the above equation is usually ignored. Moreover, if the converters are connected to a balanced three-phase load such as three phase ac machines, the proportion of the harmonics that appears in the load is greatly reduced because there are some harmonics that appear on the leg voltage, U_0 but do not produce motor harmonic current. This will be explained shortly.

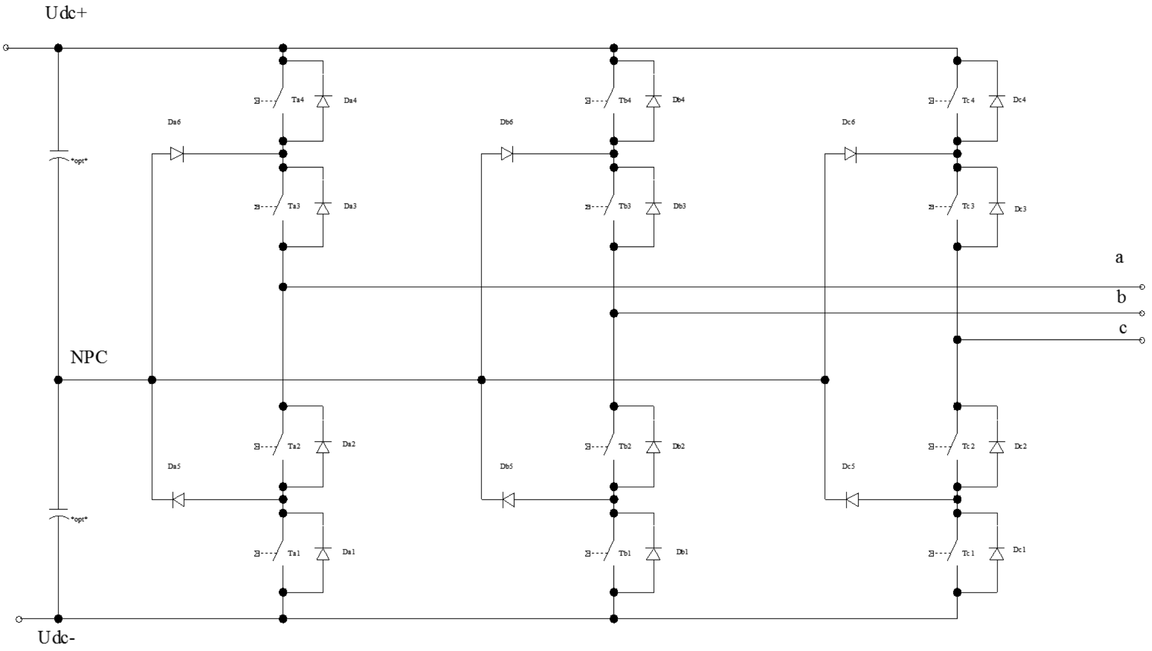


Figure 1-5: Neutral Point Clamped three phase inverter

In Figure 1-5 above, the line voltages appearing on the input terminals of the load are U_{ab},U_{ac} and U_{bc} where,

U_{ab}=U_{a0}- U_{b0} (1.5)

U_{bc}=U_{b0}- U_{c0} (1.6)

U_{ca}=U_{c0}- U_{a0} (1.7)

while U_{a0},U_{b0} and U_{c0} are the bridge leg voltages of the converter with respect to the mid-point.

Equations (1.5)-(1.7) above can be expressed as follows noting that the neutral point voltage is the same for all the three legs and cancels out during subtraction.

U_{ab}=U_{a}- U_{b} (1.8)

U_{bc}=U_{b}- U_{c} (1.9)

U_{ca}=U_{c}- U_{a} (1.10)

Where U_{a}, U_{b} and U_{c} are the phase voltages of the motor load.

If the PWM patterns of the three bridge legs are 120 degrees phase shifted to each other, the three bridge leg voltages sum up to zero. That is,

$$U_{a0} + U_{b0} + U_{c0} = 0 \qquad (1.11)$$

Using the components of these voltages,

U_a- U_0+U_b- U_0+U_c- U_0 =0 and this leads to

$$U_0 = (U_a + U_b + U_c)/3 \qquad (1.12)$$

Substituting this expression to the mid-point voltage gives the following for the load phase voltages

$$U_a(t) = \frac{1}{3}\left(2U_{a0}(t) - U_{bo}(t) - U_{co}(t)\right) + U_0 \qquad (1.13)$$

$$U_b(t) = \frac{1}{3}\left(2U_{b0}(t) - U_{co}(t) - U_{ao}(t)\right) + U_0 \qquad (1.14)$$

$$U_c(t) = \frac{1}{3}\left(2U_{c0}(t) - U_{ao}(t) - U_{bo}(t)\right) + U_0 \qquad (1.15)$$

The above set of equations, together with the fact that the zero system voltage in symmetrical three phase loads is zero, leads to the following conclusion:

- Even harmonics in the bridge leg voltages are cancelled in the line voltages
- nth Harmonic voltages, where n is a multiple of 3 do not produce any load current in symmetrical load.

One of the implications of the above result is that the modulation technique that can be employed to reduce the harmonics is not supposed to deal with the third harmonic component which has the highest magnitude.

1.1.5. Synchronous modulation

In the usual sinusoidal pulse width modulation, the number of pulses in the positive half period is not always the same as the number of pulses in the negative half period of the voltage. That is, the ratio between the switching frequency and the fundamental current frequency (frequency modulation index, $m_f = \frac{f_s}{f_1}$) is not an integer. This kind of modulation technique is called Asynchronous modulation. If Asynchronous modulation is implemented for low values of m_f, the resulting output current of the converter will have sub harmonic current components, i.e. current components with frequencies lower than the fundamental frequency [17].

In low switching frequency applications, to avoid the sub harmonic problem, Synchronous modulation is applied. This type of modulation is the one where m_f a positive integer, and hence it has the same number of pulses in the positive and negative half cycles of the fundamental period of the voltage. Programmed modulation, a topic of study in this project, is one of the techniques that implement synchronous modulation. In programmed modulation the frequency modulation index (m_f) is the same as the number of pulses per fundamental period.

In synchronous modulation, a change in the fundamental frequency is achieved by changing the frequency modulation index within the given switching frequency limit as shown in figure 1-6. That is, if the load needs a higher frequency voltage, the number of pulses per fundamental period is reduced to keep the switching frequency below the maximum allowable limit.

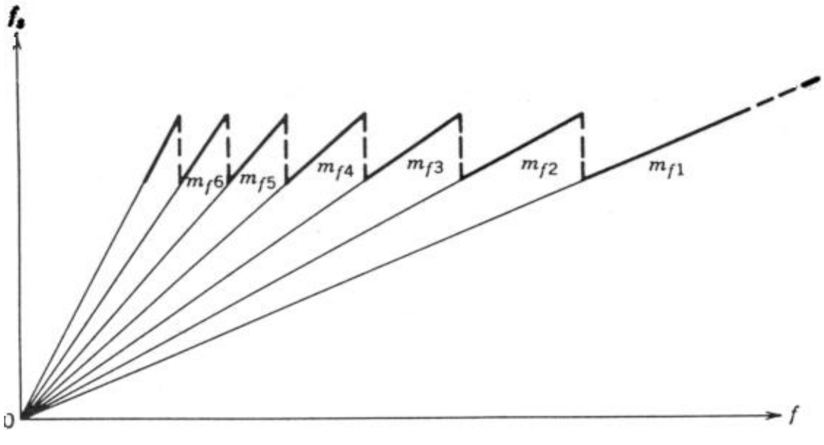


Figure 1-6: Number of pulses versus fundamental frequency for Synchronous modulation [17]

1.2. Literature review

The concept of programmed modulation as a means to eliminate harmonic content of converter outputs has been studied by many researchers. In 1973, Hasmukh S. Patel and Richard G. Hoft [3] studied the theoretical problem of eliminating harmonics in inverter-output waveforms. They developed a generalized method for eliminating fixed number of harmonics, and they also calculated the solutions for up to five harmonics. Although their analysis is based on two level inverters, the result is that it is feasible to obtain practically sinusoidal waveforms which are needed for sinusoidal loads like electrical motors. In 1980, Giuseppe S. Buja [5] developed a general method for optimizing the output waveform of PWM inverters for any load type. Buja also developed the mathematical tools and the numerical techniques to calculate the optimum switching patterns in the space of PWM inputs. In 1990, P. Enjeti and W. Shireen [4] proposed a new method for generating programmed PWM waveforms that simultaneously eliminate harmonics and reject dc-link ripple.

A more direct application of programmed pulse patterns for use in Electric Motor drives was developed by Joachim Holtz [6]. According to Holtz, feed-forward PWM modulators generate volt-second differences between the controlling reference signal and the switched output waveform during transient operation, and this dynamic modulation error is particularly pronounced at low switching frequency. As a means to avoid this problem, Holtz hence developed a trajectory tracking approach where an optimum trajectory of harmonic current computed from the steady state trajectories of the current vector based on the optimal switching sequences is imposed on any transient condition to minimize the harmonic currents. The concept of Holtz is studied thoroughly by Roger Enes [7] and simulated for a 3 phase 3 level inverter fed induction machine, and the result is that fast dynamic control is achievable with programmed modulation through manipulation of the switching-patterns to control the torque.

1.3. Scope and Limitation of the thesis

The scope of the thesis encompasses:

- Studying the effect of harmonics and switching losses on medium voltage drives
- Calculating optimal angles for selected values of number of pulses (N) and modulation indices
- Developing an algorithm for programmed modulation scheme
- Interfacing the software and hardware developed by a fellow student on Xilinx Platform Studio (XPS)
- Testing the modulation scheme on induction motor
- Discussion of the experimental result

In any given research, there are always limitations. In view of the foregoing, I hereby list some of the limitations of this thesis:

- Optimal angles are calculated to the closet possible value, and hence the angles are not the ultimate optimal
- The number of pulses (N) is limited to only four values and therefore the modulator has been tested only for certain frequency range
- Dc-bus voltage balancing and common mode voltage rejection routines are not part of the software

1.4. Structure of the report

In chapter 1, a presentation of the need for a new modulation strategy for medium voltage drives is done along with a brief theoretical background for the problem under study. Literature review is also included in this chapter. Chapter 2 contains a description of the mathematical basis for the calculation of the optimal switching angles. Discussion about the Xilinx EDK and its basic components is the focus of chapter 3. Chapter 4 presents both the hardware and software components of the experimental setup of the thesis, while in chapter 5, the main findings of the study, both from the simulation studies and the experiment, and the implications of the results obtained are clarified. Finally, Chapter 6 brings the thesis to a close with a conclusion and recommendations for further research.

2. Programmed modulation

In this chapter the mathematical basis for the calculation of the optimal switching angles is presented.

2.1. Selective Harmonic Elimination (SHE)

In steady state, a PWM modulated converter leg voltage can be expressed by its sine and cosine components using Fourier transformation as shown in equation (2.1) below [1].

$$\mathrm{V(wt)} = \sum_{h=1}^{\infty}(a_h \sin(hwt) + b_h \cos(hwt)) \qquad (2.1)$$

As it was mentioned in section 1.3 in the previous chapter, the dc term does not exist for steady state because the equilibrium line is zero. Additionally, if the converter output signals are assumed to be quarter wave symmetric, the following expressions hold true.

$$a_h = 4\frac{U_{dc}}{h\pi}\left[\sum_{k=1}^{N}(-1)^k \cos(h\alpha_k)\right] \qquad (2.2)$$

$$b_h = 0 \qquad (2.3)$$

Therefore, the harmonic components are sinusoidal functions, and mathematical tools can be used to eliminate as many harmonics as needed.

Selective harmonic elimination (SHE) is a technique based on the calculation of preprogrammed switching patterns for power inverters to obtain output voltages with zero content in specific harmonics [21]. One of the advantages of Programmed modulation is that the switching instants can take place at any time in the fundamental period of the voltage, as shown in Figure 2-1 below, since there is no any carrier that determines these instants.

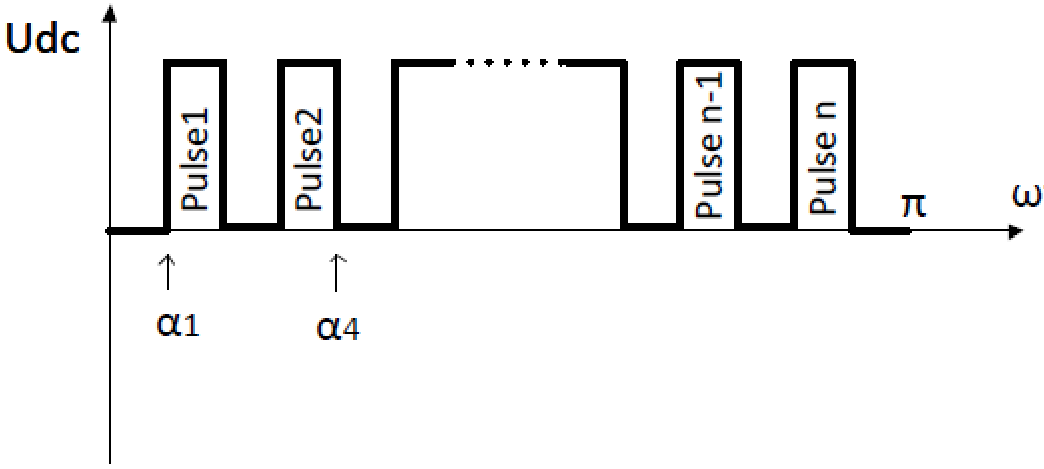


Figure 2-1: Outputs of a converter for n pulses per half of the fundamental period

From Equation (2.2) and (2.3) above, if n harmonics are needed to be eliminated, n+1 equations are solved, because one of the equations controls the magnitude of the fundamental voltage. The facts which are mentioned in section 1.1.4 above are utilized when selecting which of the harmonics are to be eliminated by the Programmed modulation technique. That is, the third harmonics and harmonics with orders of multiple of three are not problems for a symmetrical load such as motors, and harmonics with even order also get cancelled.

This leads to the conclusion that the first harmonic that needs to be eliminated is the 5th harmonic and in addition, for example if 9 harmonic components are to be eliminated, then the 7th, 11th, 13th, 17th, 19th, 23rd and 25th are those components for which an optimal pattern is required that makes their amplitude zero.

Therefore, a possible SHE solution is a set of $\alpha_i \in \left(0, \frac{\pi}{2}\right)$, $i = 0, \dots \dots n$, where $\alpha_i \leq \alpha_{i+1}$, with each α_i representing the switching angle where the power converter has to change the output voltage [21]. An implication of this is that a solution $\alpha_i \in \left(0, \frac{\pi}{2}\right)$ must satisfy the following set of $n + 1$ equations to eliminate n harmonics with $n = 5,7, \dots \dots ..$ q:

$$m = \frac{4}{\pi}[sin(\alpha_0) - sin(\alpha_1) + sin(\alpha_2) - \cdots + (-1)^n sin(\alpha_n)]$$

$$0 = \frac{4}{5\pi}[sin(5\alpha_0) - sin(5\alpha_1) + sin(5\alpha_2) - \cdots + (-1)^n sin(5\alpha_n)]$$

$$0 = \frac{4}{7\pi}[sin(7\alpha_0) - sin(7\alpha_1) + sin(7\alpha_2) - \cdots + (-1)^n sin(7\alpha_n)]$$

.

.

.

.

$$0 = \frac{4}{q\pi}[sin(q\alpha_0) - sin(q\alpha_1) + sin(q\alpha_2) - \cdots + (-1)^n sin(q\alpha_n)] \qquad (2.4)$$

where a valid solution must satisfy the following constraint:

$$0 < \alpha_1 < \alpha_2 < \alpha_3 < \cdots \ldots \ldots \ldots . < \alpha_N < \frac{\pi}{2} \qquad (2.5)$$

In the above set of equations, m is the desired modulation index and n is the number of odd harmonics that are not triple multiples of the fundamental frequency

Solving these equations is not straight forward but if Matlab *fsolve* is used, a set of angles can be found that satisfy the above set of n+1 equations. However, since there are several local minima as a solution for the equations, the main challenge is finding the optimal initial conditions, and hence several cut and try iteration is used to estimate the initial conditions. Moreover, all the optimal angles that the *f solve* produces do not necessarily fulfill the criteria of the sequence of the switching angles. Hence, manual check must be done to the outputs so that the constraint in equation (2.5) above is fulfilled.

The figures below show the optimum angles as a function of the modulation index (Figure 2-2) and the magnitude of the harmonic contents after eliminating 8 harmonic components (Figure 2-3) for N=8 where N is the number of switching instants in a quarter period of the fundamental voltage. These results are based on the *f solve* function developed by Roy Nilsen at Wärtsilä Norway AS. An important observation here from Figure 2-2 is that the optimum angles are linear functions of the modulation index for a wider modulation index range which makes the use of linear interpolation feasible during programming.

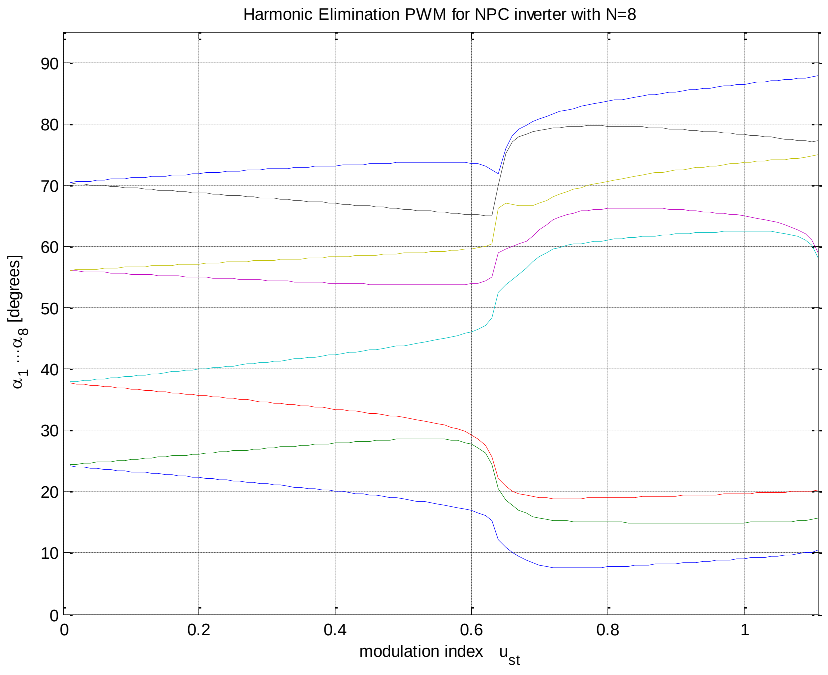


Figure 2-2: Switching angles as a function of the modulation index for N=8[7]

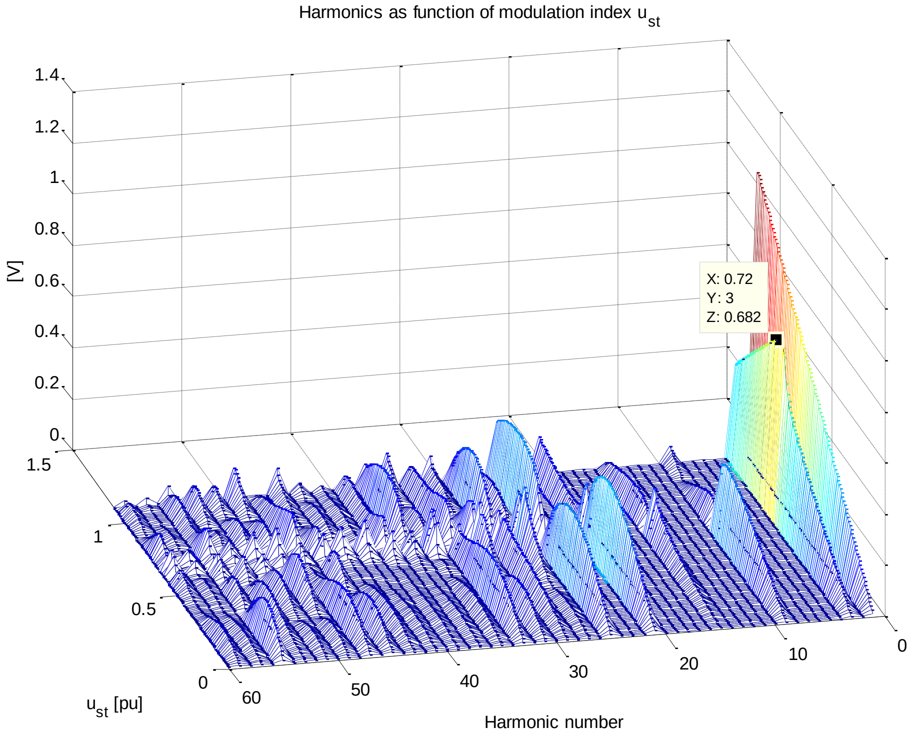


Figure 2-3: Harmonic magnitudes as a function of modulation index for N=8[7]

2.2. Minimum WTHD0

Another way to implement programmed modulation is by developing an algorithm that calculates the optimal angles which minimize the total harmonic content of the output of the converter rather than eliminating specific harmonics. The criterion that weighs this total harmonics is defined in section 1.1.2 as WTHD0. According to [8] and [9], taking in to consideration that the triple harmonics and even harmonics do not produce any harmonic output current in symmetrical three phase loads, the WTHD is expressed in equation (2.6) below.

$$WTHD = \frac{1}{U_{a0,1}}\left(\sum_{h=2}^{\infty}\left(\frac{U_{a0,h}}{h}\right)^2\right) = \frac{1}{U_{a0,1}}\left(\sqrt{\left(\sum_{i=1}^{\infty}\left(\frac{U_{a0,6i-1}}{6i-1}\right)^2 + \left(\frac{U_{a0,6i-1}}{6i+1}\right)^2\right)}\right) \qquad (2.6)$$

As it was mentioned in section 1.1.2, to avoid the frequency dependence of the first harmonic component, the dc voltage is used as a scaling factor. A consequence of this modification is then that the expression for the normalized WTHD, WTHD0, has the following form [9].

$$\sum_{h=2}^{\infty}\left(\frac{U_{a0,h}}{h}\right)^2 = \left(\sqrt{\left(\sum_{i=1}^{\infty}\left(\frac{U_{a0,6i-1}}{6i-1}\right)^2 + \left(\frac{U_{a0,6i-1}}{6i+1}\right)^2\right)}\right) \qquad (2.7)$$

Whereas the individual harmonic magnitudes are calculated using

$$\hat{U}_{a0,h} = \frac{4 \cdot U_{dc1}}{h \cdot \pi} \cdot \sum_{k=1}^{N} (-1)^{k+1} \cdot \cos(h \cdot \alpha_k) \qquad 0 < \alpha_1 < \alpha_2 .... < \alpha_N < \frac{\pi}{2}$$
$$h = 6 \cdot \mathrm{i} \pm 1, \quad \mathrm{i} = 1,2,3,4.... \qquad (2.8)$$

Equation (2.7) describes the total harmonic content of the outputs of the bridge leg voltages but in section 1.1.4, it was also shown that minimizing the harmonic content of the bridge leg voltages effectively minimizes the harmonic content of the phase voltages too.

There are several ways to find the optimal patterns that satisfy the above equation, but as it was the case for the selective harmonic elimination method, the main problem is finding the optimal initial angles to start solving the nonlinear equations. The Matlab-routine *fmincon* which finds the minimum of a constrained nonlinear multivariable functions is used for optimization by Roy Nilsen at Wärtsilä Norway AS. This function was tested for different values of N, and the result, as depicted in figure 2-4 and 2-5 below for N=4, shows that it is feasible to reduce the WTHD0 of the current drawn by the load.

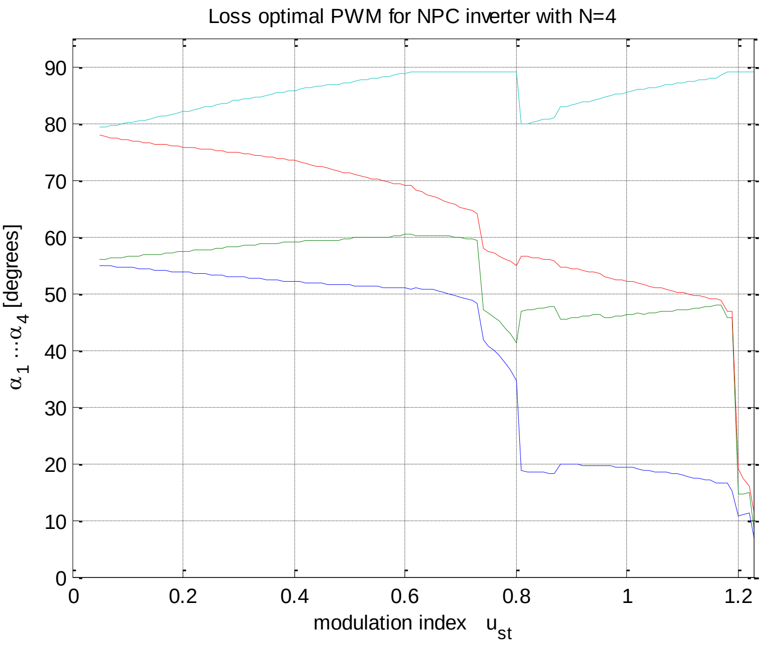


Figure 2-4: Optimal switching angles as a function of the modulation index for N=4 [7]

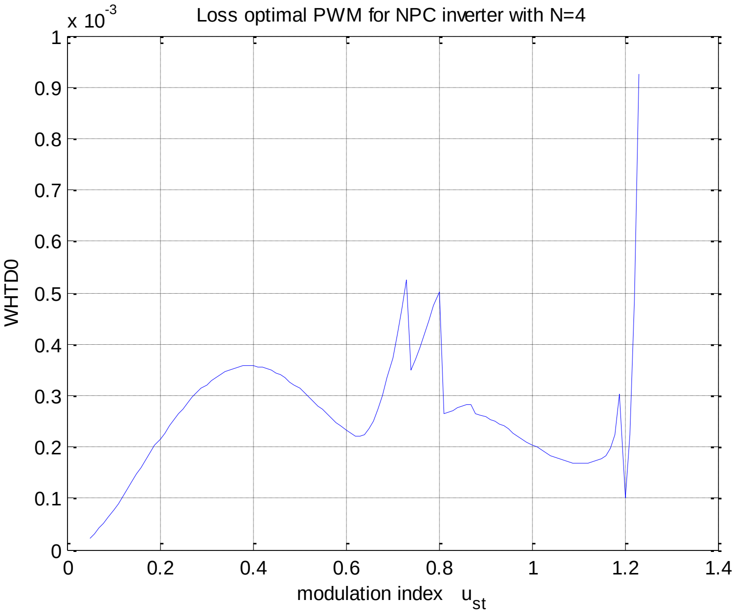


Figure 2-5: WTHD0 as a function of the modulation index for N=4 [7]

In this report, the optimal angles calculated based on minimum WTHD0 by Roger Enes [7] based on the optimization program developed by Roy Nilsen at Wartsila Norway AS are used as the basis to test the modulation algorithm.

2.3. Comparison of programmed modulation with other techniques of PWM

In the previous sections, it was shown that programmed modulation reduces harmonic content of the output voltage, which paves the way for operating PWM in lower switching frequencies. To see the improvement in the harmonic content, a performance indicator was taken and compared for programmed modulation and other PWM techniques, specially sub oscillation method. The performance indicator here is the total harmonic content of the load current (I_σ) rather than WTHD, because the latter only shows the effect of lower order harmonics, which in turn makes it difficult to get a complete comparison.

A periodic waveform consisting of a succession of pulses of amplitude varying between $\frac{V_B}{2}$ and $\frac{-V_B}{2}$ is assumed for phase voltage $V(\alpha)$ of an inverter as shown in figure 2-6 below:

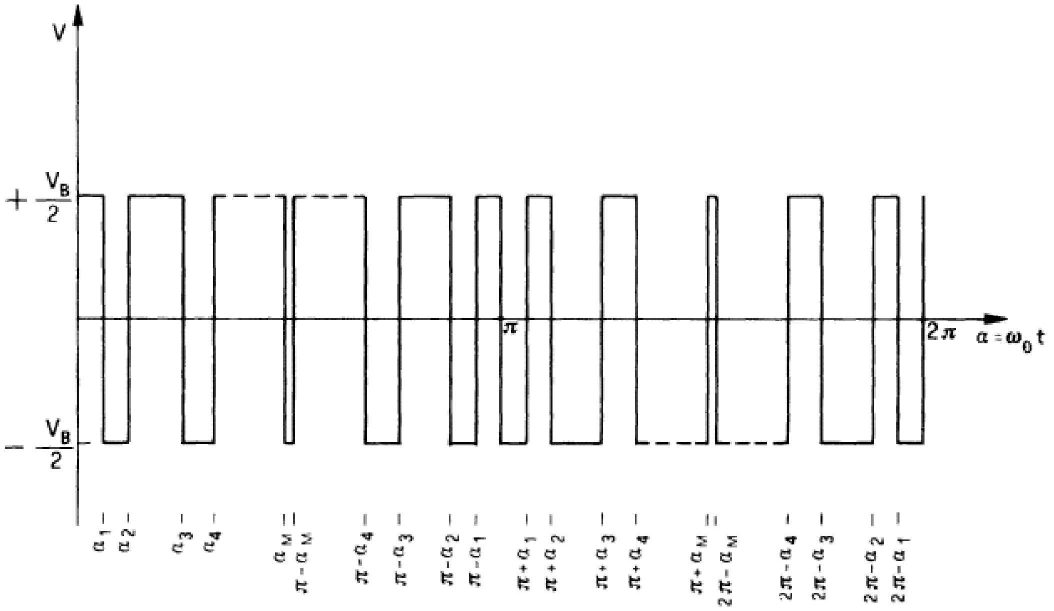


Figure 2-6: A half wave symmetric Programmed pulse pattern

The waveform in figure 2-6 above has obviously odd harmonics, owing to its half wave symmetry, and the peak of the harmonics can be given as:

$$V_n = s(-1)^M \frac{2V_b}{n\pi} \left[1 - 2\sum_{i=1}^{M}(-1)^{i+1} cos(n\alpha_i)\right] \qquad (2.9)$$

where s assumes $+1\ or - 1$ value if $V(\alpha)$ has respectively $\frac{V_B}{2}$, or $-\frac{V_B}{2}$, value between α_M and $\frac{\pi}{2}$ and M is the number of pulses in a fundamental period of $V(\alpha)$ [5].

The total harmonic current in one phase of the motor, assuming only leakage inductances determine it, can be given by:

$$I_\sigma = \sqrt{\left(\frac{1}{2}\sum_{n=3}^{\infty}\left(\frac{V_n}{n\omega_0 L}\right)^2\right)}\,, \qquad (2.10)$$

where V_n,peak value of the harmonic n of the output voltage,

L, leakage inductance of the motor,

ω_0, fundamental angular frequency,

n, harmonic number

By letting $K = \frac{V_B/2}{\omega 0}$ and $\sigma = \left(\frac{2}{V_B}\sum_{n=3}^{\infty}\frac{1}{2}\left(\frac{V_n}{n}\right)^2\right)$, equation (2.10) can be simplified into:

$$I_\sigma = K\sigma \text{ [18]} \qquad (2.11)$$

From the above expressions, it can be observed that minimizing σ effectively minimizes the total harmonic current. In addition, σ is a function of the switching angles and values of $\alpha_1, \alpha_2, \alpha_3,,,, \alpha_M$ that are obtained using the criteria $\sigma = min$ represent optimal switching patterns [18].

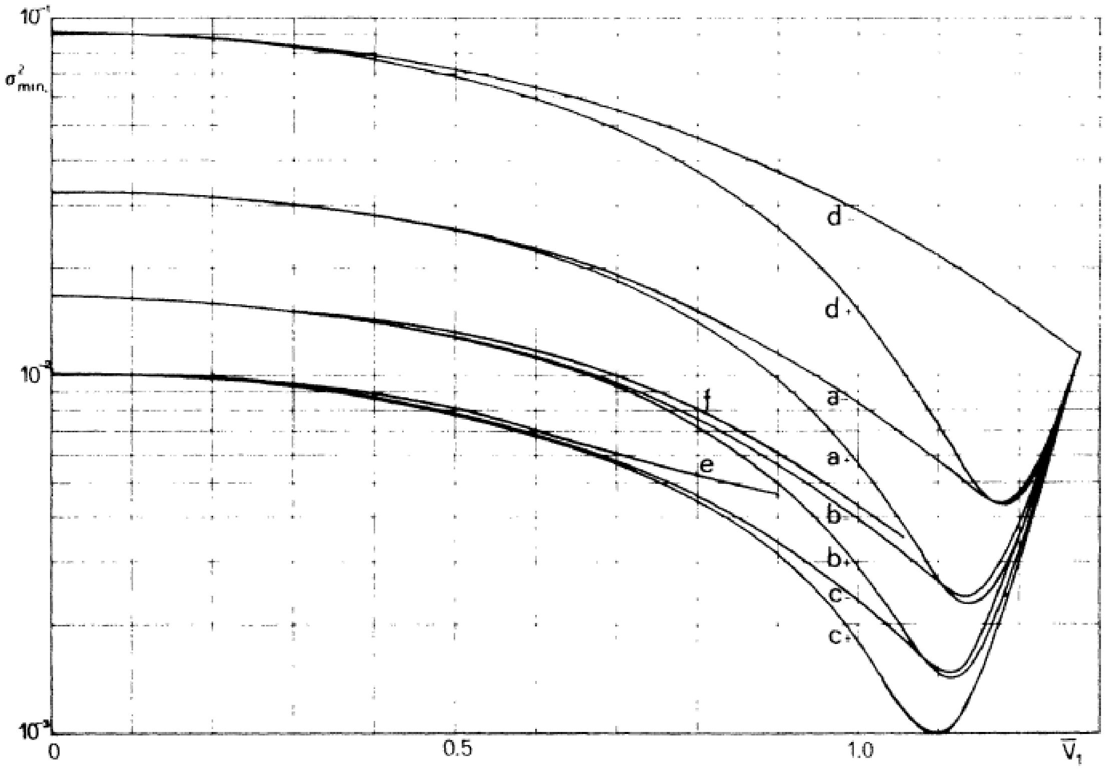


Figure 2-7: Comparison between programmed modulation and sub oscillation technique [5]

In figure 2-7 above, curve (e) shows a sub oscillation technique where a sinusoidal signal is compared to a triangular signal having nine times more switching frequency than the sinusoidal signal. On the other side, curve (c) shows a programmed modulation with M=4 and comparing the two curves reveals that the total harmonics is lower for the programmed modulation case. The same phenomenon happens if curve (f) and curve (b), both of which have M=4 are compared.

Taken together, the comparison shows that the improvement of σ^2 is significant if programmed modulation is used, particularly for higher values of V_1.

2.4. Common mode voltage elimination and DC-bus voltage balancing in programmed modulation

2.4.1. Common mode voltage

In the analysis so far, there is an important issue that has been overlooked. That is, when we considered programming pulse patterns, it was only aimed at eliminating those harmonics that cause huge distortion to the load current. The total harmonics has been the only objective function, in case of minimum WTHD0 method also. Nevertheless, even if harmonics of multiples of 3, which are not part of the objective function in the calculation so far, do not produce load current, they may cause a common mode voltage.

In a balanced three phase system, the sum of the phase voltages add up to zero so that the star point voltage does not have potential difference with respect to ground. This is no more true when PWM-generated voltages, where the star point voltage is different from zero, and this non-zero component induces shaft voltage and hence bearing current.

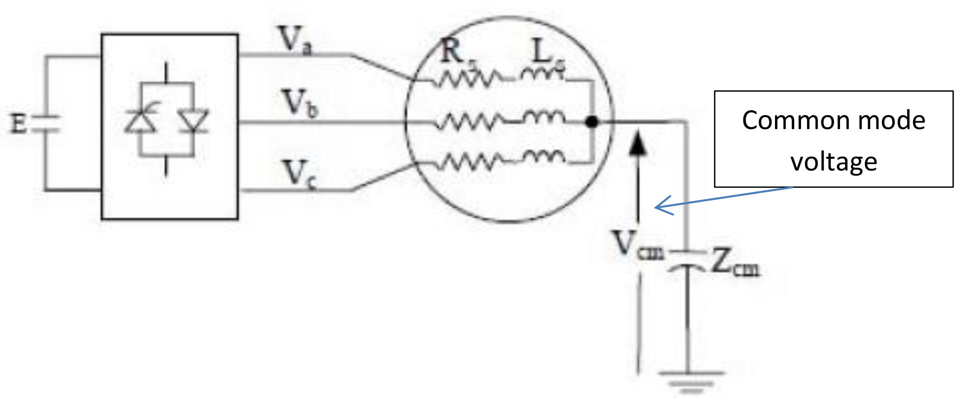


Figure 2-8: Common mode voltage at the star point of the motor

Figure 2-8 above depicts the common mode voltage that appears at the star point of the motor. When PWM is employed, $\frac{dv}{dt}$ of this voltage is high, which means it induces voltage in the motor shaft. If this induced shaft voltage exceeds the breakdown voltage of the lubricant in the bearings, it causes large capacitive current to flow in the bearings. The effect of this current is that it results in premature failure of the motor bearings and also poses EMI issues.

Hence, it appears that minimizing this common mode voltage to an acceptable value should be a criterion when effective modulation strategy is selected. In programmed modulation strategy, which is a subject of detail investigation in this thesis, reducing common mode voltage can be realized in two ways:

1) By including the expression $V_{n,cm} = \frac{(V_{n,a}+V_{n,b}+V_{n,c})}{3} = 0$ in the objective function to calculate the optimal angles where $V_{n,a}$, $V_{n,b}$., $V_{n,c}$ are the n^{th} harmonic voltage components of phase a, b and c respectively. This method, even if it can eliminate the common mode voltage for as many harmonics as possible, is computationally demanding because of the large number of equations to be solved. Nevertheless, once the optimal angles are calculated offline and stored in the memory of the modulator

system, the online manipulation becomes an easy task. This in turn reduces the execution time in the processor and therefore smaller sampling times can be implemented.

2) Using redundant switching states eliminates the common mode voltage to a certain extent. In three level inverters, for lower modulation indices it is possible to synthesize the same voltage space vector by two different combinations of phase a, b and c voltages, a phenomenon that is the basis for space vector modulation (SVM). Exploiting this feature helps us to select switching combinations that give the lowest common mode voltage. A drawback of this method is that the processor has to execute more number of instructions per sampling period and this elongates the sampling time.

Normally, one of the above two methods can be implemented depending on the need of the application which proves that programmed modulation gives the flexibility to include different objective functions to achieve desired output voltage waveform quality.

2.4.2. DC-bus voltage balancing

An important drawback of NPC converters is the unequal voltage that appears across the two capacitors in the dc link, which results in an unstable neutral point. Most capacitor voltage imbalances are caused by the use of medium vectors which clamp one phase to neutral point of DC-bus and introduce additional current flowing from, or to, neutral point. As a result inequalities occur in charging and discharging of capacitors as shown in figure 2-9 below.

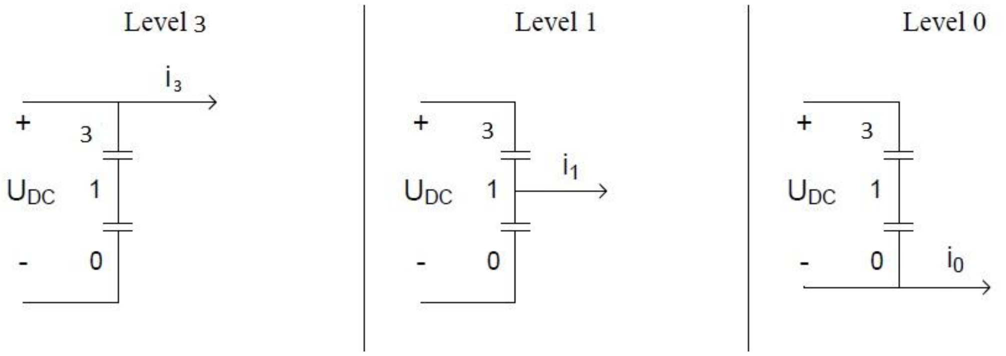


Figure 2-9: Position of current at different level [27]

Other source of imbalances in capacitor voltages are asymmetries in the system where non equal delays during transistors switching introduce differences in capacitors usage and increasing difference between its voltages [24]. The effect of this is [23]:

- The dc-link capacitors and the devices of the converter are stressed and

- The output voltages will contain low-frequency distortion.

Therefore, any modulation strategy that can be employed to NPC converters must also incorporate a technique for controlling this phenomenon. There are different ways of dealing with the imbalance, but only voltage space vector method is briefly discussed in this.

Voltage space vector method is based on the representation of the three phase quantities as vectors in a two dimensional plane. In NPC converters there are different combinations of phase voltages that give the same voltage space vectors. An interesting feature here is that the pair of the vectors that give the same vector state have different direction of the neutral current as depicted in table 2-1 below.

Small vectors (abc) that give positive current	Neutral point current, i_{NP}	Small vectors (abc) that give negative current	Neutral point current, i_{NP}	Medium Vectors (abc)	i_{NP}
100	i_a	211	$-i_a$	210	i_b
221	i_c	110	$-i_c$	120	i_a
010	i_b	121	$-i_b$	021	i_c
122	i_a	011	$-i_a$	012	i_b
001	i_c	112	$-i_c$	102	i_a
212	i_b	101	$-i_b$	201	i_c

Table 2.1: Space Vectors participating in DC-bus balancing and the associated neutral point current

This feature can be exploited to control the neutral point current and in turn the capacitor voltages in such a way that the neutral point is stable. In NPC converters, there are 27 switch combinations that give 19 voltage space vectors. Out of these 19 vectors, the zero state vectors and the three bigger vectors (vectors with none of the phases connected to the neutral point) do not influence the capacitor voltage balance.

Figure 2-10 below shows the 12 states which participate in dc bus balancing in NPC, and the direction of the neutral current is also specified for the two different vectors.

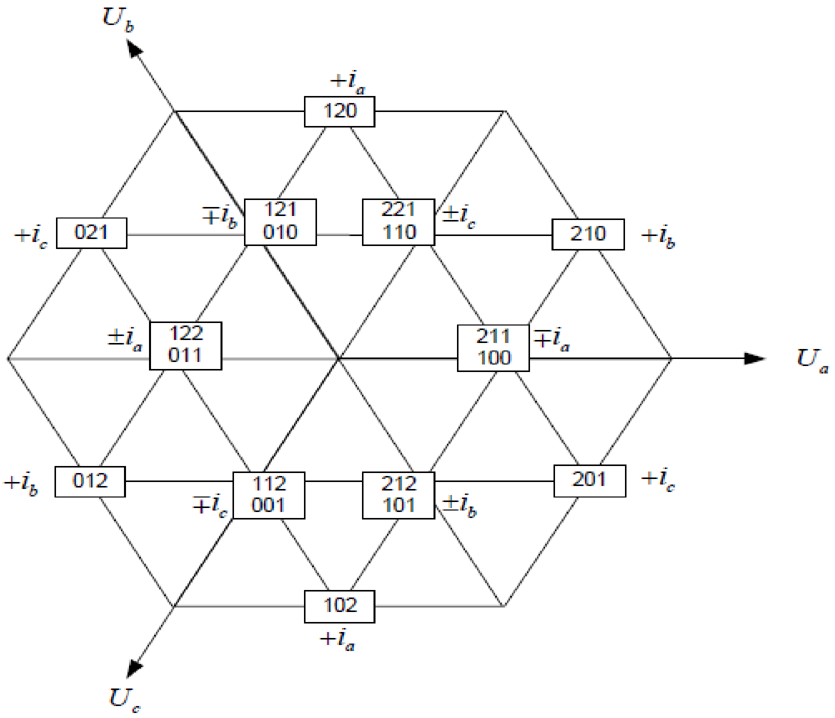


Figure 2-10: Space vectors and the corresponding neutral point current

There are different ways to control the dc-bus voltage by controlling which of the smaller voltage vectors to use.

In programmed modulation, dc-bus voltage balancing method which is applied to Space Vector modulation can be implemented. This can be achieved by using a single register to store the switching states of all the three phases instead of three independent registers for each phase. Assuming there are three switching states in a sampling period for each phase, there will be all together nine switching states for all the three phases in the given sampling period.

Phase a Switching states			Phase b Switching states			Phase c Switching states		
State 1	State 2	State 3	State 1	State2	State 3	State 1	State 2	State 3
01	01	10	10	00	01	10	10	10

Table 2-2: 9 different registers for all the 9 states in a sampling time

The above table can be rearranged as follows in such a way that only one register holds the states of all the three phases.

Switching State 1	01 10 10
Switching state 2	01 00 10
Switching state 3	10 01 10

Table 2-3: 3 registers for all the 9 states in a sampling time

From tables 2-2 and 2-3 above, it can be seen that all the three states in the chosen sampling period are deliberately small vectors to show how the dc-bus voltage balancing works in this modulation scheme. Since there is a complementary combination of switches that gives the same voltage space vector, the controller decides whether to pick these vectors or the complementary ones based on the status of the dc-link capacitor voltage. That is, if for example, the upper capacitor has more voltage than the lower and the phase current is negative, then it means the upper capacitor should be discharged and the lower charged. In this case, if the needed space vector can be manufactured using 122 or 011, then the controller decides to send 011 to the modulator. Therefore, by effectively utilizing the advantages offered by these redundant switching states, it is possible to avoid the use of extra hardware to reduce dc-bus capacitor voltage imbalance. One drawback of this method is that we cannot apply it for higher modulation indices because the bigger space vectors, which generate higher output voltage, do not have redundant states. The detail control strategy and implementation of this method is not the scope of this thesis.

2.5. Dynamic modulation error and the trajectory tracking approach

It has been stated that programmed modulation enables us to implement modulation at reduced switching frequency in such a way that the harmonic contents of the current is diminished in a significant way. Nevertheless, this advantage is limited only to steady state conditions as transient change in the modulation index or change in the number of pulses (N) disrupts the optimal pattern which introduces significant harmonics which is called Dynamic modulation error (D(t)).

The steady state trajectory at any time t can be expressed as: [16]

$$\mathrm{I(t)} = \mathrm{i}_{\mathrm{s1}}(\mathrm{t}) + \mathrm{i}_{\mathrm{tr}}(\mathrm{t}) + \mathrm{i}_{\mathrm{h,ss}}(\mathrm{t}) \qquad (2.12)$$

Where $\mathrm{i_{s1}(t)}$ is the fundamental stator current, $\mathrm{i_{tr}(t)}$ is a transient current component, and $\mathrm{i_{h,ss}(t)}$ is the steady-state harmonic current. The transient current component is zero in steady state. If at $\mathrm{t_1}$ a change in optimal patterns is going to happen, then an additional current component is needed in the above expression to account for the offset in space that comes from the fact that two optimized steady-state trajectories rarely have the same instantaneous current values at any given point in time.

Then the new relation becomes:

$$\mathrm{I(t) = i_{s1}(t) + i_{tr}(t) + i_{h,ss}(t) - D(t)} \qquad (2.13)$$

From the above analysis, it is then almost impossible to apply fast current control in inverters with programmed modulation. However, the trajectory tracking approach is found to be an adequate solution for the elimination of the dynamic modulation error. In this method, the steady state trajectories of the current vector computed from the optimal switching patterns are used as templates and the controller responds when the actual stator current trajectory deviates from this template at transient operation [6].

The waveforms of the fundamental voltages, represented by the voltage space vector U_1, the fundamental frequency ω_1 and the pulse number N are required as input to the modulator. The dynamic modulation error, $\mathrm{D(t)}$, calculated from equation 2.11 above carries an important information that is needed to implement pulsewidth modulators for high performance drives. However, it is difficult to calculate this error from the measured current because of two main reasons:

- It is an advantage of pulsewidth modulators having a feedforward structure that they can be operated and tested without the load being connected. This advantage is lost if the modulation error, determined from measured current signals, is used as a feedback signal [19].
- The extraction of specific current components, like the fundamental current space vector, $\mathrm{i_{s1}(t)}$, or the transient current, $\mathrm{i_{tr}(t)}$, as instantaneous signals from a measured current waveform I(t) is difficult [19].

Hence, if estimated currents are processed rather than the measured currents, the above problem is solved. The estimation is done on the basis of the pulsewidth input (U^*) and output voltage (U_s) signals as shown in figure 2-11 below.

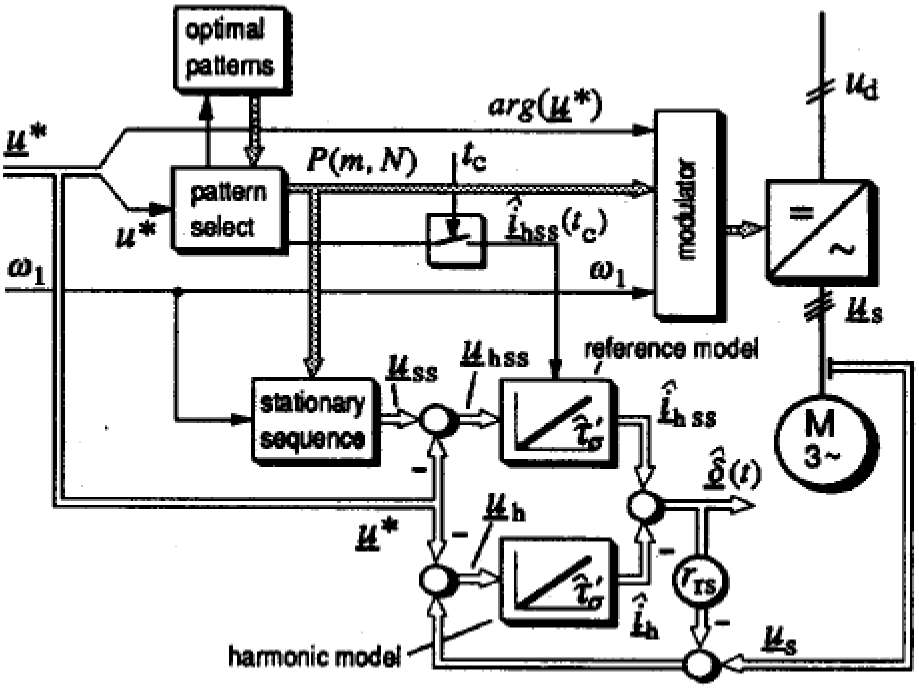


Figure 2-11: Estimation of the dynamic modulation error: signal flow diagram [19].

The modulator output voltages (U_k) and the motor stator voltages (U_s) is the same neglecting the inverter switching delay and it can be expressed as:

$$U_s(t) = U_1(t) + U_{hss}(t) \qquad (2.14)$$

where $U_{hss}(t)$ is a steady state harmonics.

However, during transient operation, there is an additional voltage component ($U_{tr}(t)$) that is imposed by the controller to give rise to the needed change in the current. Consequently, hence equation 2.12 can be expressed as:

$$U_s(t) = U_1(t) + U_{hss}(t) + U_{tr}(t) \qquad (2.15)$$

From equation 2.13 above, the modulation error can be calculated by taking the difference between the input signal $U^*(t)$ and the output $U_s(t)$. This quantity describes any undesired signal in the modulator performance.

3. Xilinx Embedded Development Kit (Xilinx EDk)

In this chapter, a brief discussion about Xilinx EDK is presented. The basic components, its advantages and drawbacks are also described.

3.1. Overview

With the increased development of digital controllers, control of Power Electronic Converters is becoming more and more sophisticated and desired control goals are being achieved. Different types of digital devices have been employed for this purpose. Arguably the most common device for this application is Digital Signal Processors (DSPs) because of their low cost and good performance features [10]. DSPs can be defined as Microprocessors designed to perform digital signal processing [10]. In Power Electronic applications, DSPs that have controller abilities are required to replace hardware component by performing the equivalent operation in software. DSPs also have a significant contribution for improving the efficiency and reliability of motor control applications by executing complex control routines like Field Oriented Control (FOC) which otherwise become cumbersome if ordinary processors are used.

Nevertheless, DSPs have significant drawbacks compared to their counterparts that implement hardware functionalities like FPGA based processors and ASICs. As almost all the functions of a DSP are implemented using software functions and software operations are executed sequentially, time critical applications are seldom designed using DSPs. In addition, the hardware reprogrammable nature of FPGA based processors outpasses the limited flexibility of DSPs.

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects [12]. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. The following are some of the benefits of FPGA based designs:

It is possible to emulate a circuit to be manufactured. This gives the circuit under test to be diagnosed at hardware level before actual production.

FPGAs also provide a single hardware block to be used for implementing multiple functions, a feature that is not found in Application Specific Integrated Circuits (ASICs).

FPGAs also incorporate blocks of commonly used functionalities such as RAM; clock management, and DSP. The following are the basic components in a Xilinx FPGA:

- ✓ ***Configureable Logic Blocks (CLBs).***

    - The CLB is the basic logic unit in an FPGA that consists of a configureable switch matrix with 4 or 6 inputs, some selection circuits and flip-flops [12]. It is by the programmed interconnection of these blocks that different hardware modules are built.

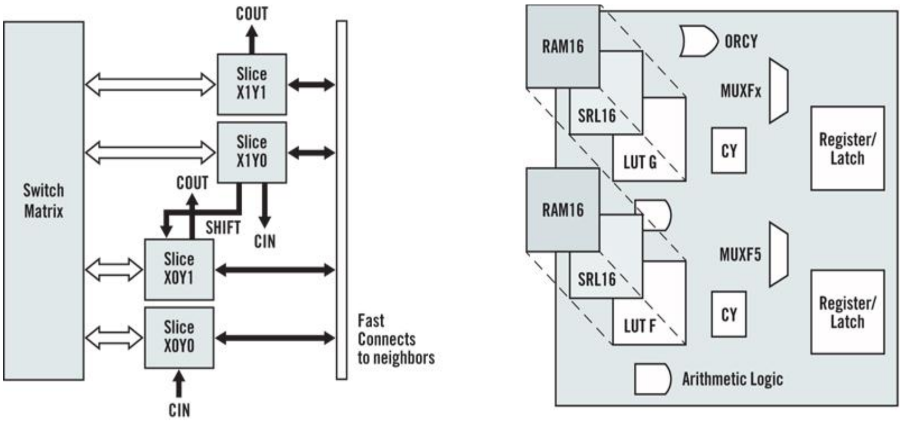


Figure 3-1: Basic CLB structure [12]

- ✓ ***Interconnect***

    - These flexible routings route the signals between CLBS and to and from I/Os [12].

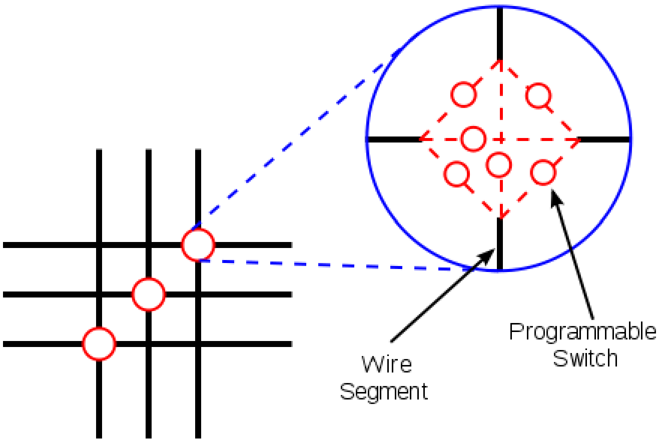


Figure 3-2. Programmable interconnects

As it is evident from figure 3-2, the switches can be programmed to connect any of the possible lines and it is also possible to reprogram the switches.

✓ *SelectIO (IOBs)*

- IOBs provide flexible I/O support. These blocks allow the user to communicate data from and to the FPGA based system. The I/O blocks are also reprogrammable, and hence it is possible to have multiple input output applications on a single pin.

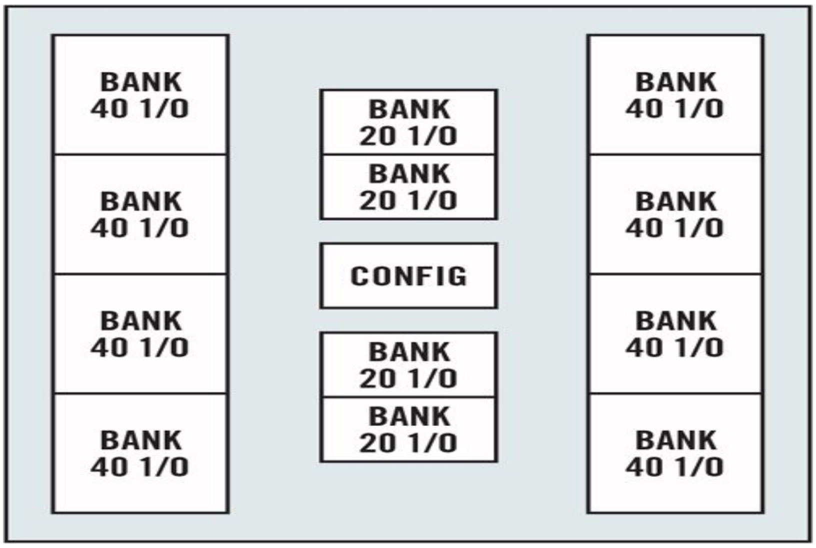


Figure 3-3: Basic IOBs structure [12]

- ✓ ***Memory***

    - This Embedded Block Memory allows on-chip memory in designs.

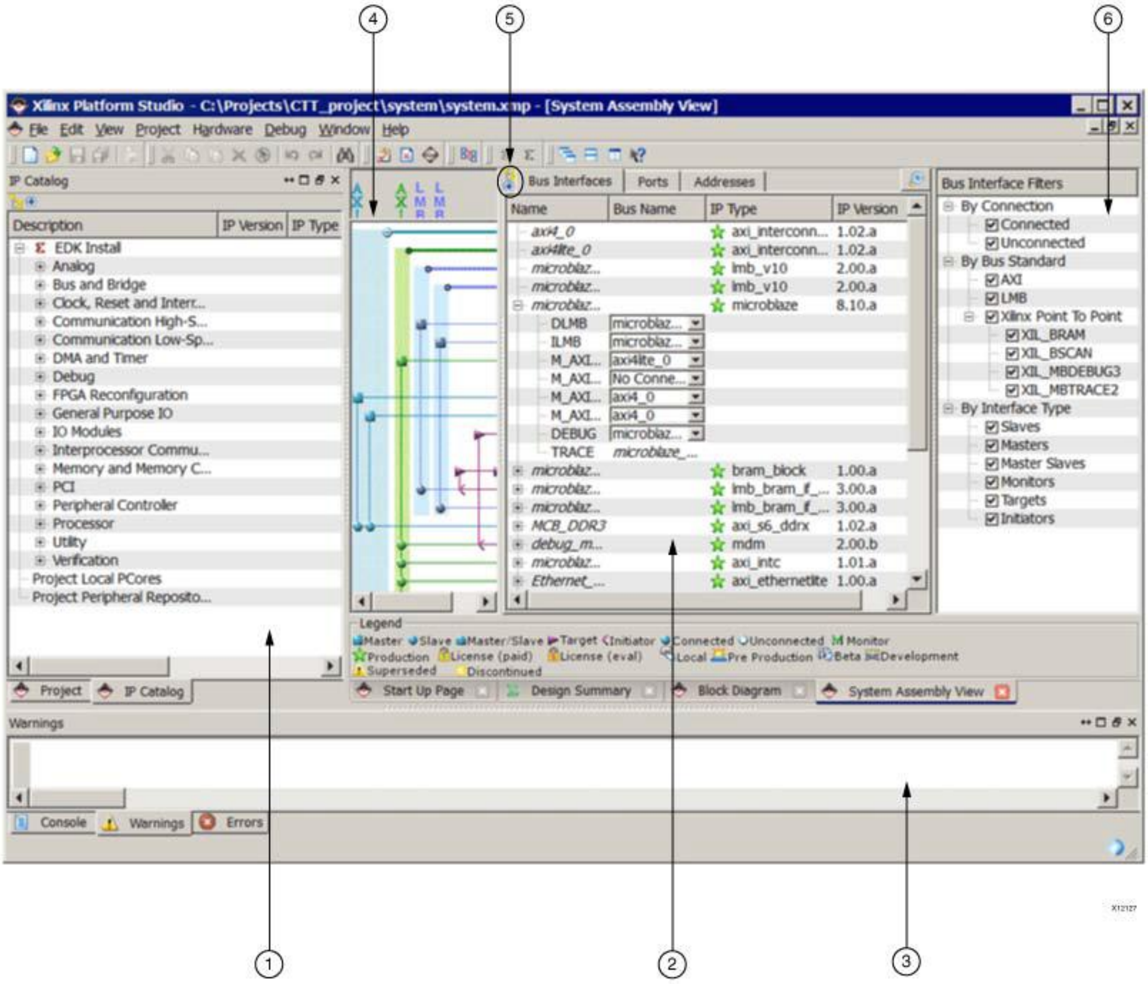
- ✓ ***Complete Clock Management***

    - The most advanced FPGAs from Xilinx offer both digital clock management and phase-looped locking that provide precision clock synthesis [12].

One of the interesting features of working with Xilinx FPGA is the Embedded Development Kit that helps to ease the design phase of an FPGA based systems. The Xilinx Development Kit is a suite of tools and Intellectual Property (IP) that enable one to design a complete embedded processor system for implementation in an FPGA device [13]. Two of these tools which are used in this master's thesis are: Xilinx Platform Studio (XPS) and Software Development Kit (SDK). Since the thesis is focused on designing and testing a new modulator for the programmed PWM modulation, the project consists of 2 phases, software design and hardware design, both of which can be done on Xilinx EDK but using different tools. Below is a brief description of the two tools used to achieve the objective of the thesis.

3.2. Xilinx Platform Studio (XPS)

XPS is primarily used for embedded processor hardware system development [13]. Specification of the microprocessor, peripherals, and the interconnection of these components, along with their respective detailed configurations is also done in XPS.



Name	Bus Name	IP Type	IP Version
axi4_0		axi_interconn...	1.02.a
axi4lite_0		axi_interconn...	1.02.a
microblaz...		lmb_v10	2.00.a
microblaz...		lmb_v10	2.00.a
microblaz...		microblaze	8.10.a
DLMB	microblaz...		
ILMB	microblaz...		
M_AXI...	axi4lite_0		
M_AXI...	No Conne...		
M_AXI...	axi4_0		
M_AXI...	axi4_0		
DEBUG	microblaz...		
TRACE	microblaze_...		
microblaz...		bram_block	1.00.a
microblaz...		lmb_bram_if_...	3.00.a
microblaz...		lmb_bram_if_...	3.00.a
MCB_DDR3		axi_s6_ddrx	1.02.a
debug_m...		mdm	2.00.b
microblaz...		axi_intc	1.01.a
Ethernet_...		axi_ethernetlite	1.00.a

Figure 3-4: XPS project window [13].

Figure 3-4 above shows the different components of an XPS main window [13]:

1) Project information area: offers control of and information about the project on design.
2) System assembly view: allows users to view and configure system block elements
3) Console window: provides feedback from the tools invoked during run time
4) Connectivity panel: is a graphical representation of the hardware platform connections
5) View buttons: enables users to sort and revise their designs
6) Filters pane: lists the filters where the bus interfaces or ports tabs are selected

XPS has a tremendous potential to configure and integrate plug and play Intellectual Property (IP) cores from the Xilinx Embedded IP catalog, with customs or third part Verilog and VHDL designs. It also possesses a feature that allows different modules to be designed and synthesized using Verilog or VHDL that would later be used as components of the whole project. In this thesis, XPS was used to design the modulator hardware. Hence, a brief description of this design is presented in section 4.1.1.

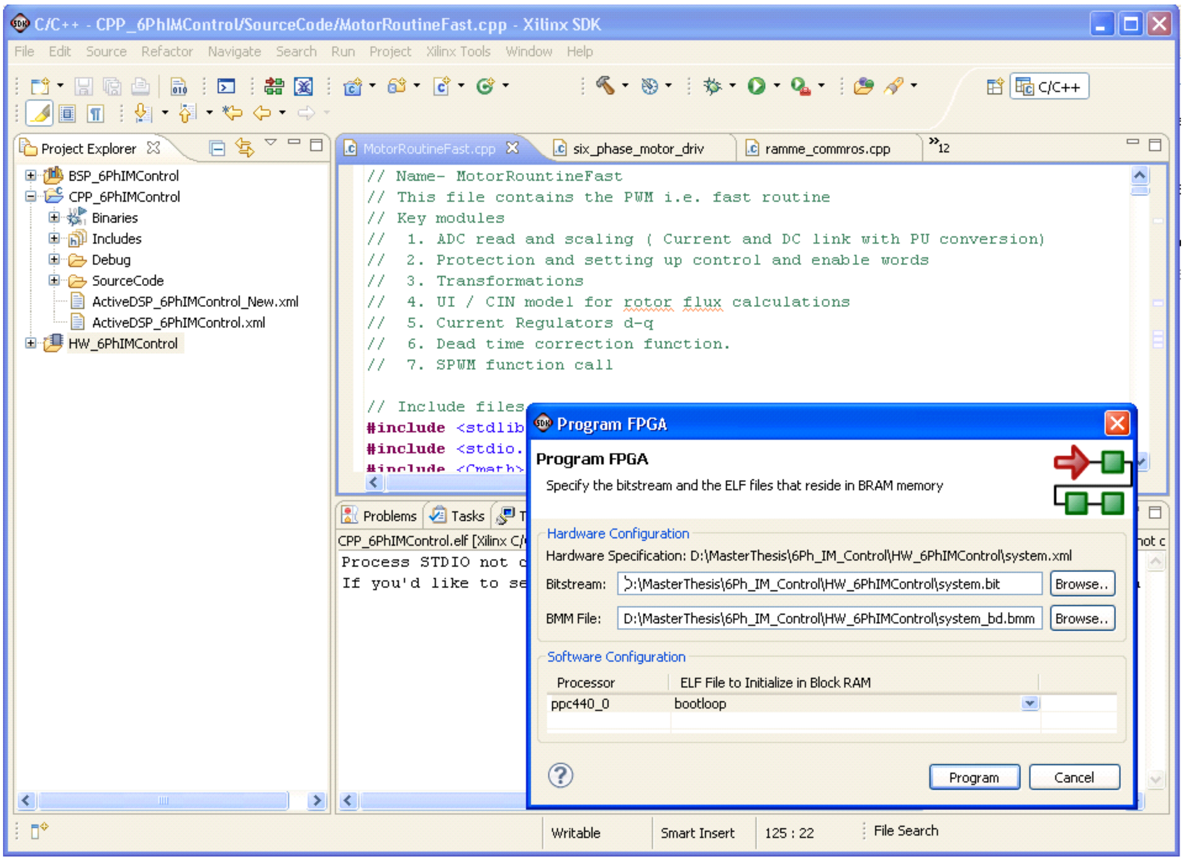
3.3. Software Development Kit (SDK)

The SDK is an integrated development environment, complementary to XPS, that is used for C/C++ embedded software application creation and verification [13]. This part of the kit is mainly used to write, load and debug software program on the processor of the FPGA system. While writing the program, one can also access the different modules on the board so that the program is able to communicate with the hardware design on the FPGA.

Each and every software project that is developed on SDK must have its own workspace where the SDK stores the project data and metadata. A software project contains one or more source files, together with the necessary header files, to allow compilation and generation of a binary output (.elf) file [13].

In order for the SDK to access and configure the different modules on the FPGA board, there must be a hardware platform for the software design. This hardware platform is the embedded hardware design that is developed in XPS and exported to SDK. In addition to the hardware platform, the SDK also needs a collection of libraries and drivers to be able to communicate with the different modules on the hardware platform of the FPGA board. This set of drivers and libraries is called Board Support Package (BSP) [13].

SDK is not just a tool for creating a software application; it also performs functions such as editing of software projects, building software projects and debugging the project on the target hardware.



C/C++ - CPP_6PhIMControl/SourceCode/MotorRoutineFast.cpp - Xilinx SDK

```
// Name- MotorRountineFast
// This file contains the PWM i.e. fast routine
// Key modules
//  1. ADC read and scaling ( Current and DC link with PU conversion)
//  2. Protection and setting up control and enable words
//  3. Transformations
//  4. UI / CIN model for rotor flux calculations
//  5. Current Regulators d-q
//  6. Dead time correction function.
//  7. SPWM function call

// Include files
#include <stdlib
#include <stdio.
#include <Cmath>
```

```
CPP_6PhIMControl.elf [Xilinx C/
Process STDIO not c
If you'd like to se
```

Program FPGA

Program FPGA

Specify the bitstream and the ELF files that reside in BRAM memory

Hardware Configuration

Hardware Specification: D:\MasterThesis\6Ph_IM_Control\HW_6PhIMControl\system.xml

Bitstream: D:\MasterThesis\6Ph_IM_Control\HW_6PhIMControl\system.bit Browse..

BMM File: D:\MasterThesis\6Ph_IM_Control\HW_6PhIMControl\system_bd.bmm Browse..

Software Configuration

Processor	ELF File to Initialize in Block RAM
ppc440_0	bootloop

Figure 3-5: A screen shot of an SDK window

The following flow chart summarizes the process flow of Xilinx FPGA based design.

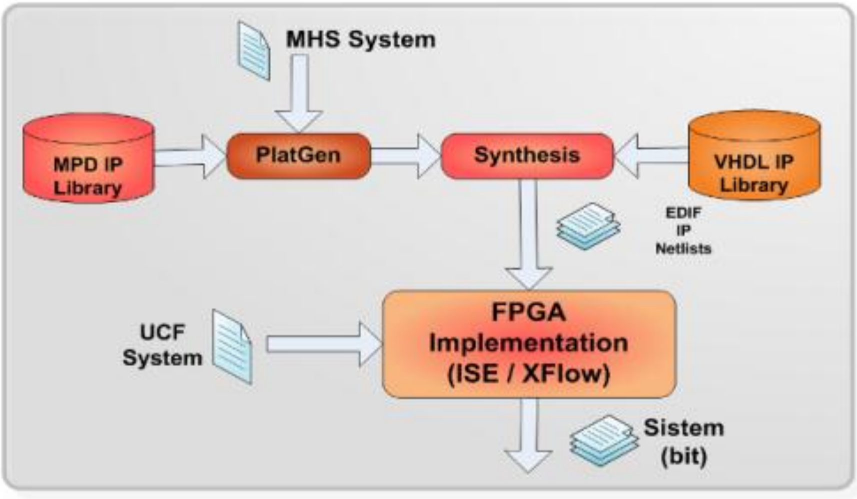


Figure 3-6: Flow chart of FPGA based design [12]

4. Experimental setup

In this chapter, the laboratory setup on which the project was done is described in a brief detail.

4.1. Hardware Setup

The general hardware setup used in this project is the one that was being used by Nebrom Berihu for his master's thesis in the academic year 2011/12. The hardware setup has the following components:

- Six-phase induction machine
- A three phase three level inverter to supply the six-phase machine.
- A dc machine that is loaded with water-cooled resistors: the dc machine functions as a load to the induction machine
- A diode rectifier that functions as a dc bus
- A three phase variable Transformer
- Current and voltage sensors and rotary pulse encoder for the control application
- An FPGA board and
- A personal computer

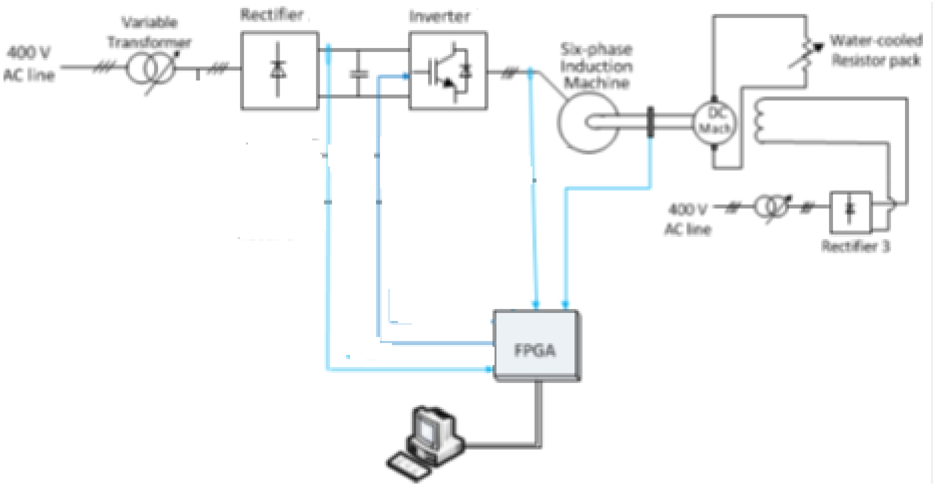
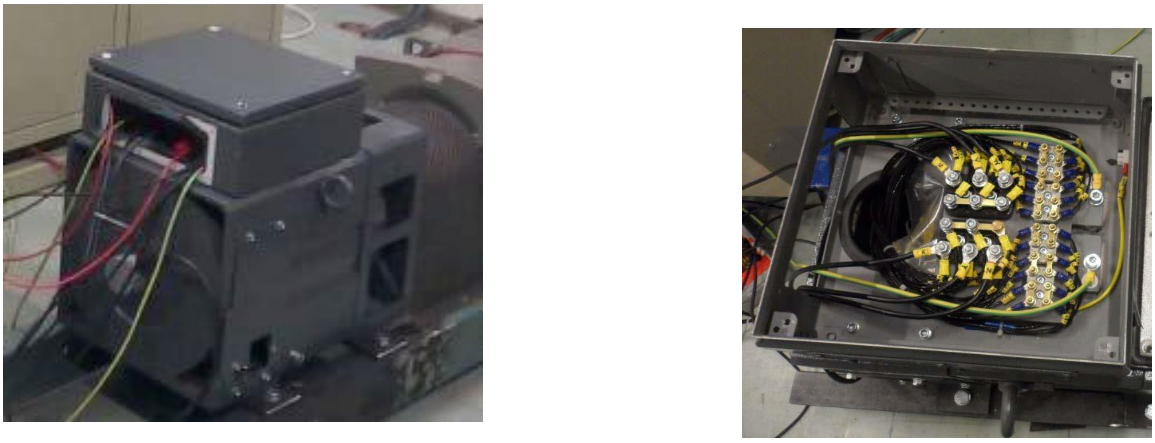


Figure 4-1: Schematic of six-phase motor drive lab setup [14]

4.1.1. Six phase induction machine

The six phase induction machine used in the experiment is a squirrel cage rotor type and has two 3 phase stator winding groups. The two groups of windings are separated by 30 degrees from each other. The six phase machine can also be run as a three phase machine by supplying only one group of the three phase windings with voltage. The details of the machine parameters are given in Appendix A.



(a) (b)

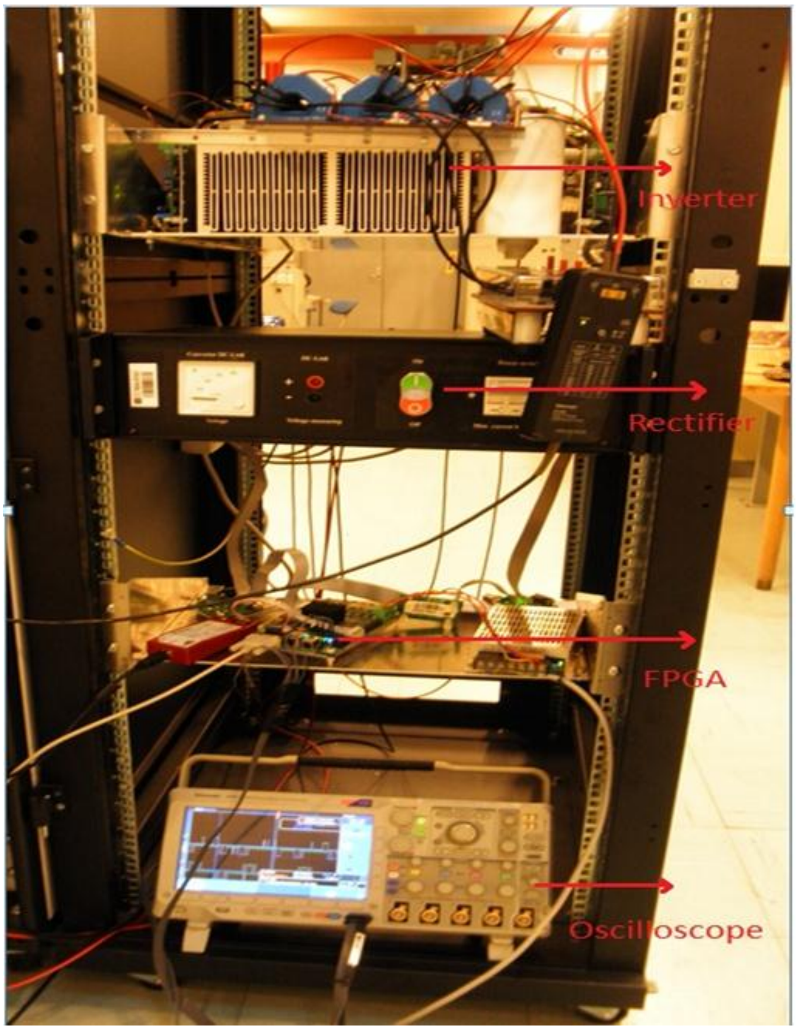
Figure 4-2 : Six phase induction machine: (a) External view

(b) Input terminals

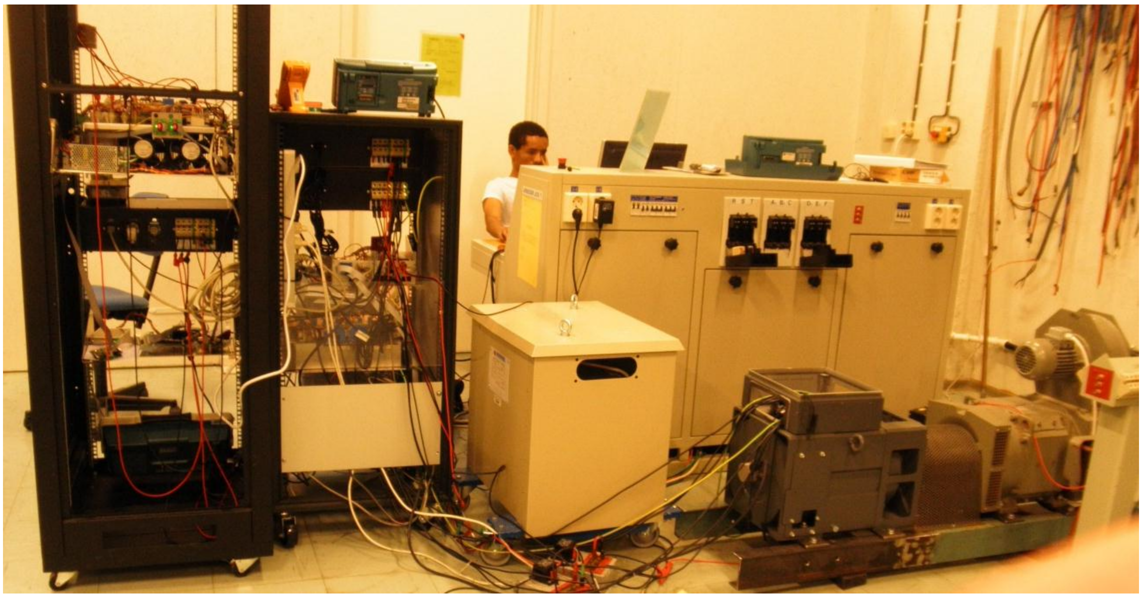
4.1.2. Converters

A three phase diode rectifier, connected to a variable transformer, was used to supply the inverter. The input voltage to the rectifier can vary from 0 to 400 V while the output varies from 0 to 540 V. 63 A is the maximum current rating of the rectifier.

The six phase induction machine under test was supplied with a three phase three level IGBT inverter with a rating of 40KW. The input to the inverter comes from the rectifier, and the inverter can carry a maximum input voltage of 200A, while it can be switched with a frequency of up to 3 KHz. The module also consists of a capacitor bank and capacitive and RC snubbers. IGBTs on the inverter have 1200V, 200A rating.



(a)



(b)

Figure 4-3: (a) Hardware setup

(b) Complete hardware setup including motor

4.2. Hardware design on the FPGA board

The FPGA board that was used for developing the modulator is Xilinx Virtex 5 FXT FPGA. The main components of this board according to [15] are:

- PowerPC processor
- Memory: 256 MB DRAM (DDR2), 128 MB Flash and 64 KB EEPROM
- Analog to Digital Converter(ADC): 8 x 12 bit, 40 MSPS
- Digital to Analog Converter(DAC): 4 X 12 bit, 20 MSPS
- Digital Input-Output: 3 x 16 bit TTL, 3 x 2 LVDS signal pair
- Communication ports: RS232, USB, Ethernet and CAN.
- Interface IP modules (AD converter serial signal interface, DA converter interface, pulse encoder interface) and
- Signal Processing IP modules (accumulator filter block, PI regulator, angle transformation, pulse width modulator and comparator limiter)

Figure 4-5 below shows schematics of the board in use.

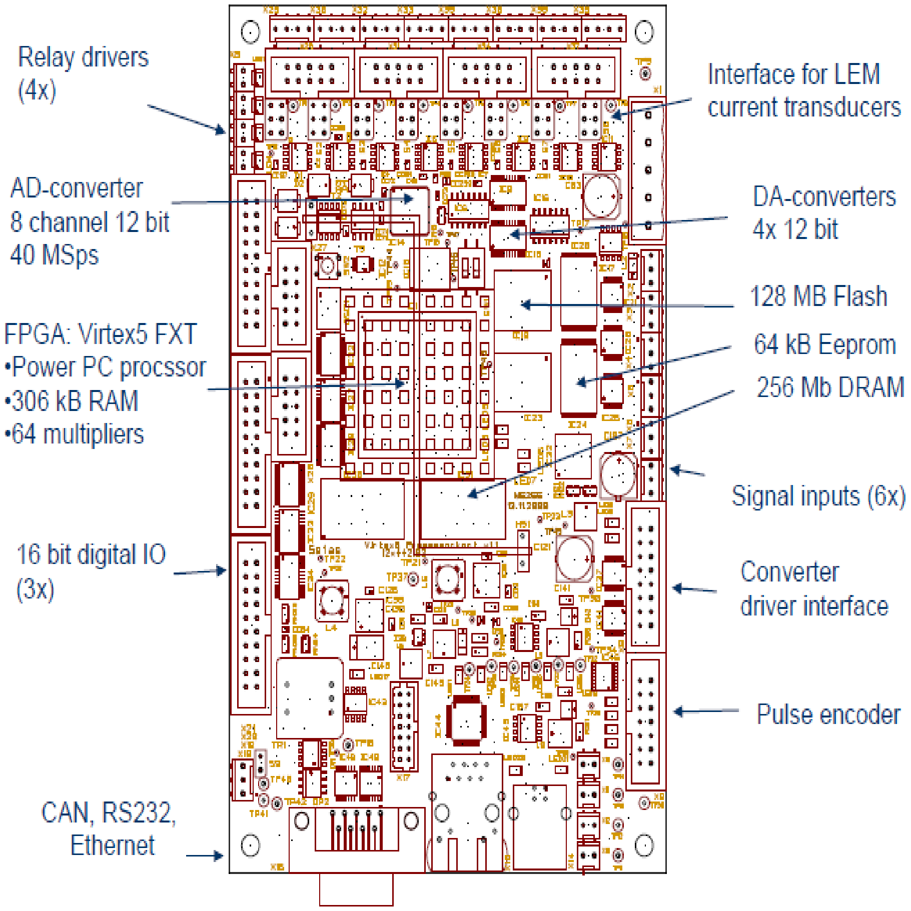


Figure 4-4: Xilinx FPGA board [15]

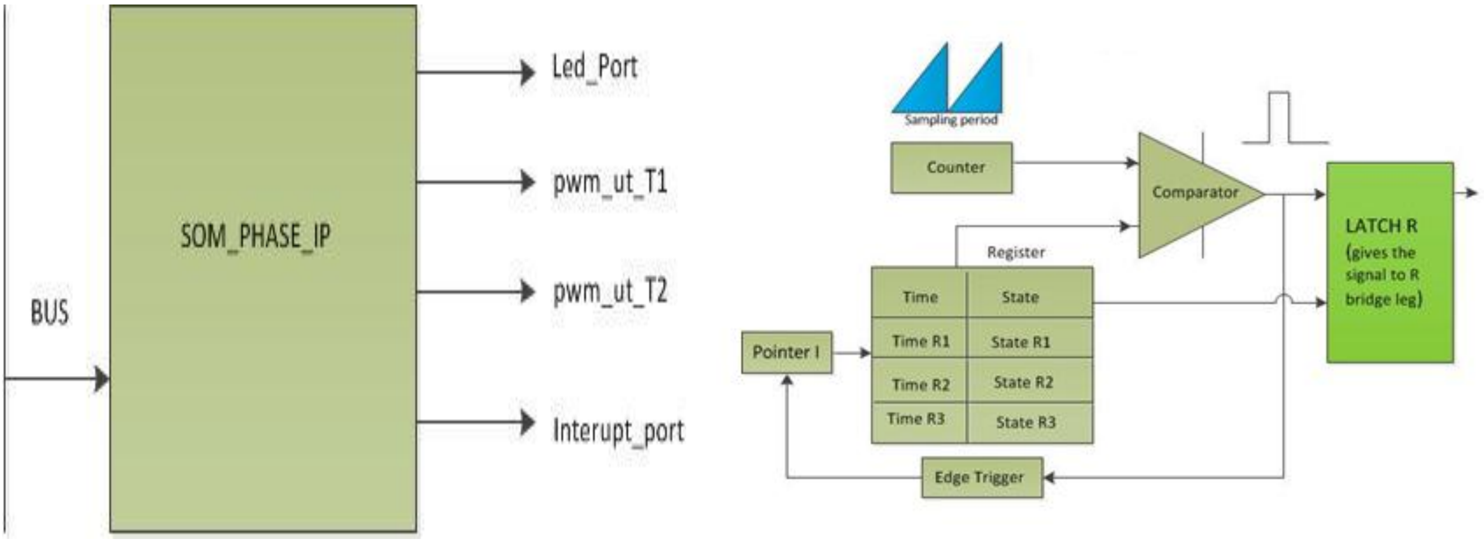
4.2.1. Modules in the hardware design

In this section, a discussion of some of the components used in this thesis is presented while their detail interconnection in XPS and the functional description are presented in section 4.2.2.

- **PowerPC processor:** The processor used in this hardware setup is a 32 bit PowerPC processor that is embedded on Virtex 5 FPGA board. The PowerPC 440 embedded processor contains a dual-issue (capability to execute two instructions per clock cycle), superscalar, pipelined processing unit, along with functional elements required to implement memory management, cache control, timers, and debug facilities.

- **Memory Block:** The most important part of the modulation scheme is the memory system where the optimal angles are stored for later online manipulation. The static memory blocks available on the FPGA are one with a 64K and another with 128K and the board has also an external dynamic memory (DRAM).

- **Modulator:** The modulator, together with the extended memory capability, is the part of the drive control system where a vast modification is made compared to the previous modulation techniques. The modulator was designed on the FPGA board using VHDL code. Composed of a number of registers, counter and comparator, the modulator is a fully digital circuit that generates the switching states by accepting the switching time and switch state values from the processor bus. There are different ways of implementing these based on the need of the modulation scheme but in this thesis, only per phase implementation was done. The state and the time registers fetch the corresponding state and time values from the processor bus at the start of each sampling period. Then, the comparator compares the contents of the time register with the counter value, and if the counter value is equal to any of the time values on the time register, it latches the output to the state of that particular time.



(a) (b)

Figure 4-5: (a) Modulator module

(b) Components inside the module

In addition, there are different auxiliary components that are needed for the whole modulation system to function properly. Some of these components are:

1) *Virtex®-5 Auxiliary Processor Unit (APU) Floating-Point Unit*:
   One of the main challenging phenomena in programmed modulation is that there is a lot of data manipulation because of the huge number of angles stored and the arithmetic related to converting the angles to switching states. Therefore, it takes longer time for the ppc processor to execute these operations, which means that the needed switching state and time calculations would not be completed in one sampling period if the processor is left to handle it. Hence, a need for additional processor arises for the proper operation of the modulation circuit, and the APU is best for this purpose.

2) Clock generator:

   It is clear that the different components in the FPGA board need clock signal for their proper operation. The clock generator provides the necessary clocks for these components.

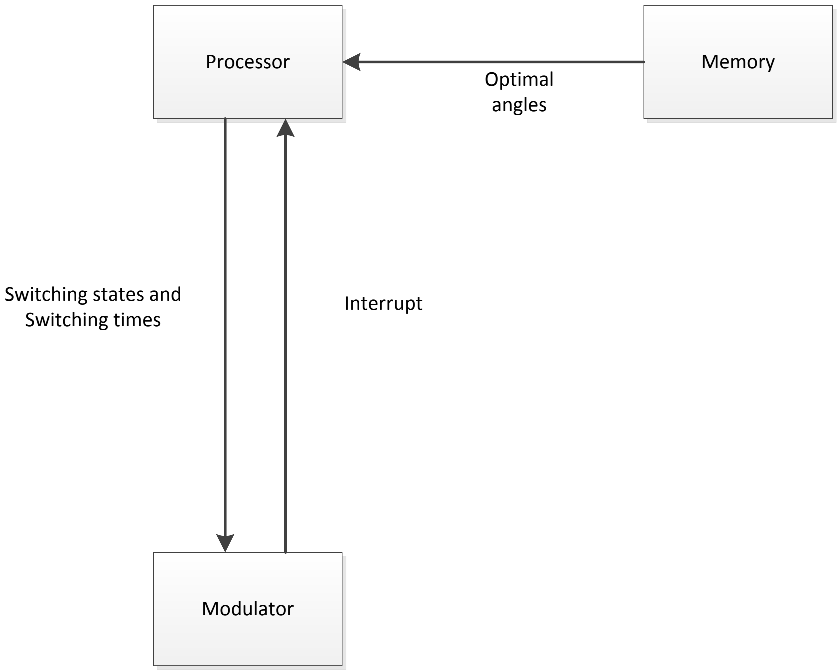


Figure 4-6: Signal flow between the processor, memory and the modulator

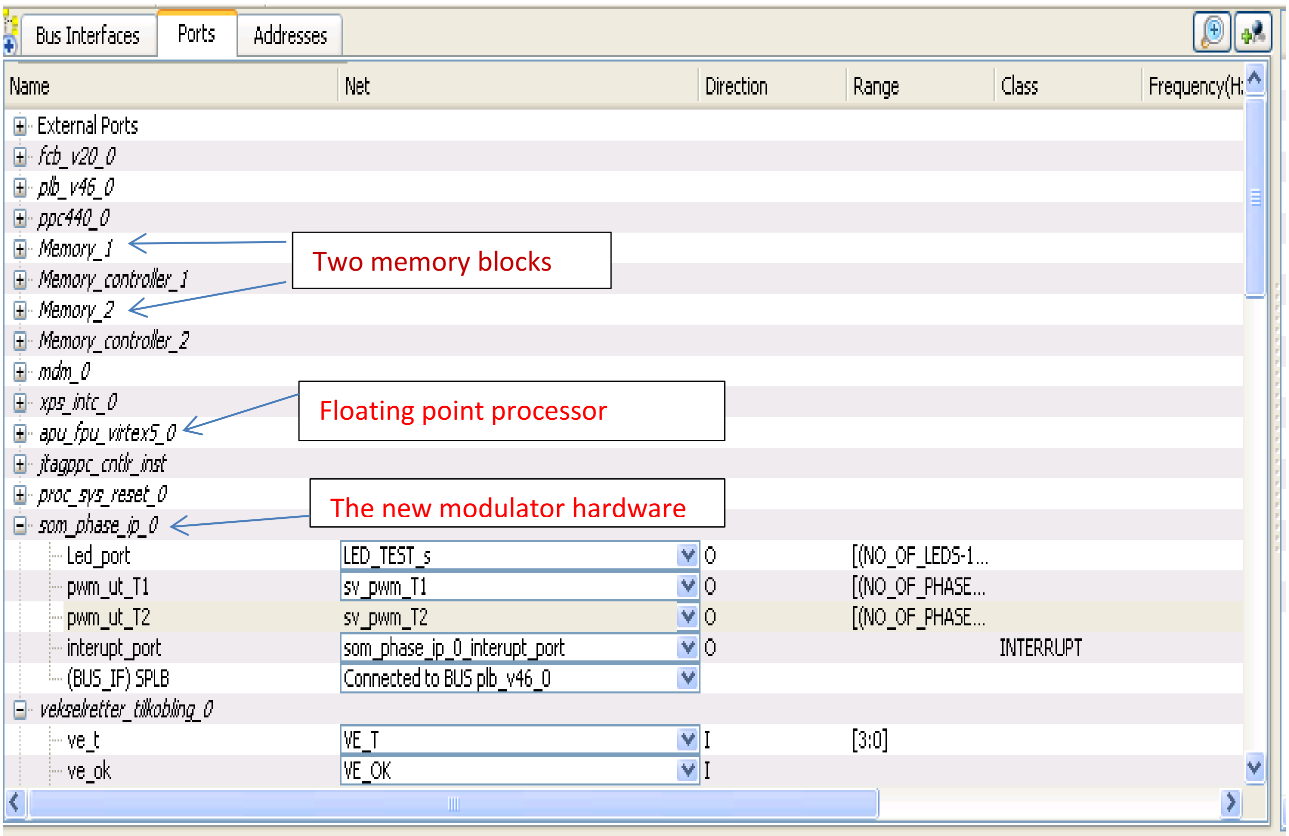
Figure 4-6 above shows the signal flow among the three main parts of the modulation hardware. Most of the operations are performed in the processor using the data from the memory. The processor finishes all these operations in the given sampling time and puts the switching states and time on the processor bus to be fetched by the modulator when the next interrupt starts. Detail of the activities the processor does is described in the software setup section of the report.

4.2.2. XPS design of the hardware

In chapter 3, an overview of Xilinx tools was made with a brief description of XPS and SDK. It has also been mentioned that XPS is used to build the hardware platform by connecting custom IPs and Xilinx built in components. In this section, the hardware design is explained with a special focus on the IP modules in the design. Some of these modules were mentioned in the previous section but not with a design detail.

XPS hardware design can be done in two different ways:

1) Using the plug and play functions in the tool: In this method, the different hardware components needed for the design are connected manually by dragging and dropping them in the system assembly view panel of the tool. This method is very easy, and hence it is most popular one.



Name	Net	Direction	Range	Class	Frequency(H:
External Ports					
fcb_v20_0					
plb_v46_0					
ppc440_0					
Memory_1					
Memory_controller_1					
Memory_2					
Memory_controller_2					
mdm_0					
xps_intc_0					
apu_fpu_virtex5_0					
jtagppc_cntlr_inst					
proc_sys_reset_0					
som_phase_ip_0					
Led_port	LED_TEST_s	O	[(NO_OF_LEDS-1...		
pwm_ut_T1	sv_pwm_T1	O	[(NO_OF_PHASE...		
pwm_ut_T2	sv_pwm_T2	O	[(NO_OF_PHASE...		
interupt_port	som_phase_ip_0_interupt_port	O		INTERRUPT	
(BUS_IF) SPLB	Connected to BUS plb_v46_0				
vekselretter_tilkobling_0					
ve_t	VE_T	I	[3:0]		
ve_ok	VE_OK	I			

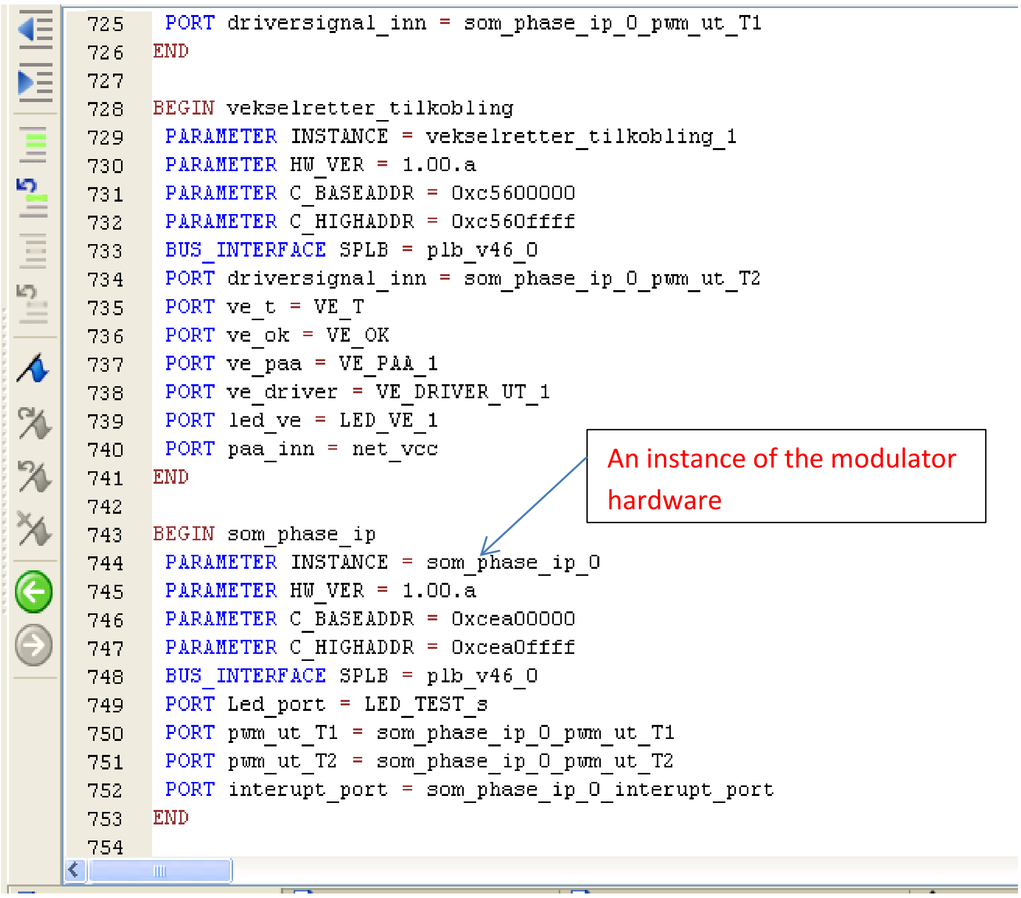
Two memory blocks

Floating point processor

The new modulator hardware

Figure 4-7: System assembly view of the hardware design

2) Using Microprocessor Hardware Specification (MHS) tool: This method is a text based editor where a user can write the different modules, their instances, port names and sizes, addresses and their connections.



```
 PORT driversignal_inn = som_phase_ip_0_pwm_ut_T1
END

BEGIN vekselretter_tilkobling
 PARAMETER INSTANCE = vekselretter_tilkobling_1
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_BASEADDR = 0xc5600000
 PARAMETER C_HIGHADDR = 0xc560ffff
 BUS_INTERFACE SPLB = plb_v46_0
 PORT driversignal_inn = som_phase_ip_0_pwm_ut_T2
 PORT ve_t = VE_T
 PORT ve_ok = VE_OK
 PORT ve_paa = VE_PAA_1
 PORT ve_driver = VE_DRIVER_UT_1
 PORT led_ve = LED_VE_1
 PORT paa_inn = net_vcc
END

BEGIN som_phase_ip
 PARAMETER INSTANCE = som_phase_ip_0
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_BASEADDR = 0xcea00000
 PARAMETER C_HIGHADDR = 0xcea0ffff
 BUS_INTERFACE SPLB = plb_v46_0
 PORT Led_port = LED_TEST_s
 PORT pwm_ut_T1 = som_phase_ip_0_pwm_ut_T1
 PORT pwm_ut_T2 = som_phase_ip_0_pwm_ut_T2
 PORT interupt_port = som_phase_ip_0_interupt_port
END

```

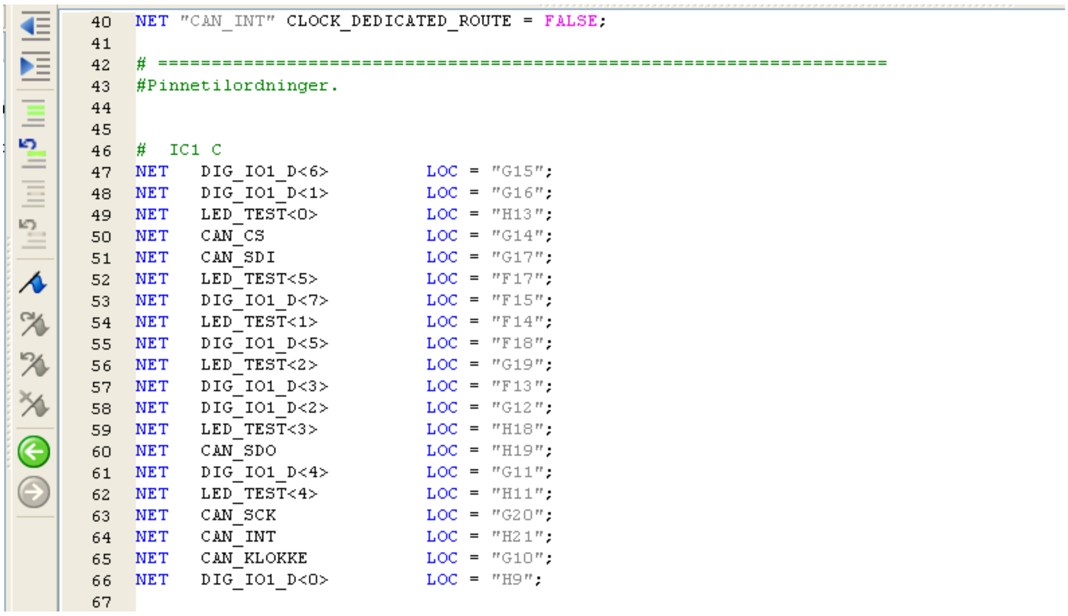
An instance of the modulator hardware

Figure 4-8: Sample MHS file of the hardware design

Even if it is very easy to customize the connections and all the other parameters in the design if *mhs file method* is used compared to the drag and drop tool, it needs mastering the tool which otherwise makes the design prone to error.

As it can be seen from figure 4-8, each component (for example in this case: *som_phase_ip_0*) is an instance of the blue print of the component (in our case: *som_phase_ip*) .Hence, this enables the designer to have as many components of a certain type as needed only using a blue print of the component.

In addition, a *user constrained file (ucf)* was included in the design in order for the compiler to successfully map the pins on the FPGA board to the output ports in the design.

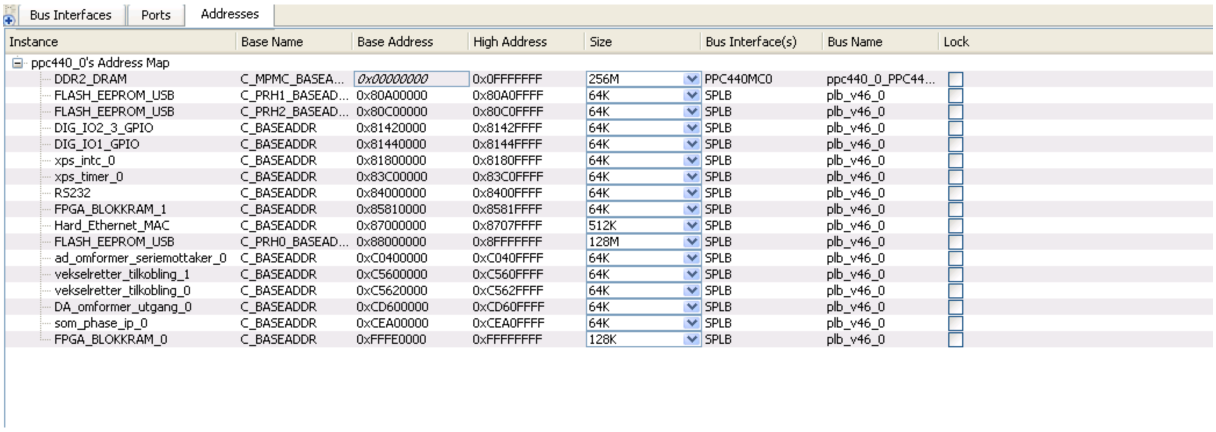


```
NET "CAN_INT" CLOCK_DEDICATED_ROUTE = FALSE;

# ====================================================================
#Pinnetilordninger.


#  IC1 C
NET    DIG_IO1_D<6>          LOC = "G15";
NET    DIG_IO1_D<1>          LOC = "G16";
NET    LED_TEST<0>           LOC = "H13";
NET    CAN_CS                LOC = "G14";
NET    CAN_SDI               LOC = "G17";
NET    LED_TEST<5>           LOC = "F17";
NET    DIG_IO1_D<7>          LOC = "F15";
NET    LED_TEST<1>           LOC = "F14";
NET    DIG_IO1_D<5>          LOC = "F18";
NET    LED_TEST<2>           LOC = "G19";
NET    DIG_IO1_D<3>          LOC = "F13";
NET    DIG_IO1_D<2>          LOC = "G12";
NET    LED_TEST<3>           LOC = "H18";
NET    CAN_SDO               LOC = "H19";
NET    DIG_IO1_D<4>          LOC = "G11";
NET    LED_TEST<4>           LOC = "H11";
NET    CAN_SCK               LOC = "G20";
NET    CAN_INT               LOC = "H21";
NET    CAN_KLOKKE            LOC = "G10";
NET    DIG_IO1_D<0>          LOC = "H9";

```

Figure 4-9: A screen shot of user constraints file (ucf) for the design

In this thesis, the drag and drop method was used owing both to its simplicity and the unfamiliarity of the author with the mhs file method. However, both methods are complementary, and any change made in one of the windows (mhs or system assembly view) is automatically transferred to the other.

Finally the address of the different components connected to the bus is generated, and the resulting file is exported to the software platform.



Instance	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name	Lock
ppc440_0's Address Map							
DDR2_DRAM	C_MPMC_BASEA...	0x00000000	0x0FFFFFFF	256M	PPC440MC0	ppc440_0_PPC44...	□
FLASH_EEPROM_USB	C_PRH1_BASEAD...	0x80A00000	0x80A0FFFF	64K	SPLB	plb_v46_0	□
FLASH_EEPROM_USB	C_PRH2_BASEAD...	0x80C00000	0x80C0FFFF	64K	SPLB	plb_v46_0	□
DIG_IO2_3_GPIO	C_BASEADDR	0x81420000	0x8142FFFF	64K	SPLB	plb_v46_0	□
DIG_IO1_GPIO	C_BASEADDR	0x81440000	0x8144FFFF	64K	SPLB	plb_v46_0	□
xps_intc_0	C_BASEADDR	0x81800000	0x8180FFFF	64K	SPLB	plb_v46_0	□
xps_timer_0	C_BASEADDR	0x83C00000	0x83C0FFFF	64K	SPLB	plb_v46_0	□
RS232	C_BASEADDR	0x84000000	0x8400FFFF	64K	SPLB	plb_v46_0	□
FPGA_BLOKKRAM_1	C_BASEADDR	0x85810000	0x8581FFFF	64K	SPLB	plb_v46_0	□
Hard_Ethernet_MAC	C_BASEADDR	0x87000000	0x8707FFFF	512K	SPLB	plb_v46_0	□
FLASH_EEPROM_USB	C_PRH0_BASEAD...	0x88000000	0x8FFFFFFF	128M	SPLB	plb_v46_0	□
ad_omformer_seriemottaker_0	C_BASEADDR	0xC0400000	0xC040FFFF	64K	SPLB	plb_v46_0	□
vekselretter_tilkobling_1	C_BASEADDR	0xC5600000	0xC560FFFF	64K	SPLB	plb_v46_0	□
vekselretter_tilkobling_0	C_BASEADDR	0xC5620000	0xC562FFFF	64K	SPLB	plb_v46_0	□
DA_omformer_utgang_0	C_BASEADDR	0xCD600000	0xCD60FFFF	64K	SPLB	plb_v46_0	□
som_phase_ip_0	C_BASEADDR	0xCEA00000	0xCEA0FFFF	64K	SPLB	plb_v46_0	□
FPGA_BLOKKRAM_0	C_BASEADDR	0xFFFE0000	0xFFFFFFFF	128K	SPLB	plb_v46_0	□

Figure 4-10: Address map of the hardware design.

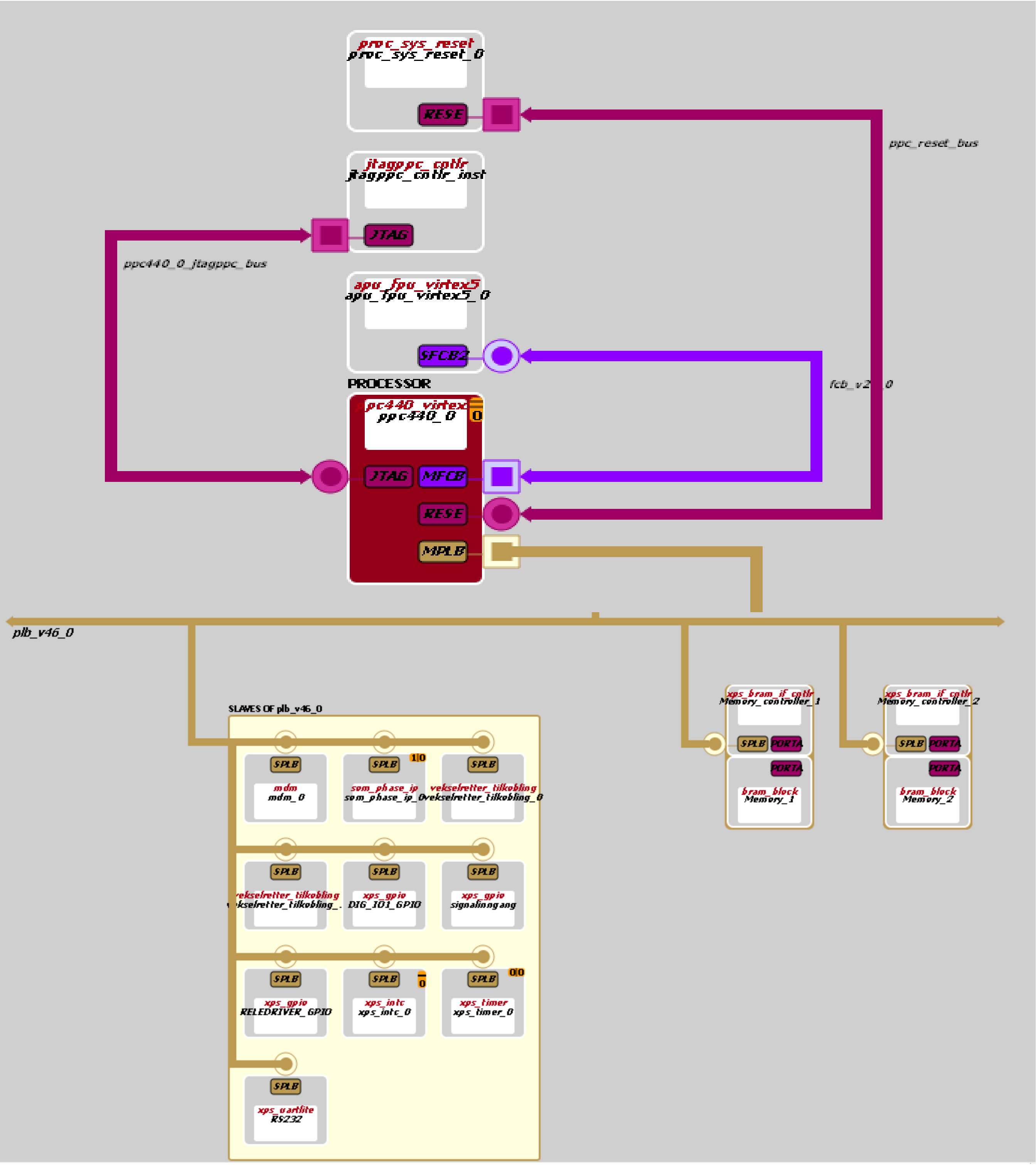


Figure 4-11: General view of the XPS hardware design

Figure 4-11 above shows the detail interconnection of the different modules in the FPGA board. The *proc_sys_reset* at the top of the diagram is used to enable/disable certain features of the ppc processor so that a user can customize the design according to the application in hand. The *jtag_ppc_ctrl* connects the JTAG chain of the FPGA, a component that is used for loading the program to the FPGA and debugging it, to the processor. As it was clarified above in section 4.2.1, there is a need for a separate processor that handles the Floating Point calculations, and *apu_fpu_virtex5*, in the above diagram, represents this component. The PROCESSOR is the central part of the design; hence it has connection to all the other components via the *plb_v46_0* bus (connection to IP modules) or through specified connections like:

- ✓ *ppc_reset_bus* to *proc_sys_reset*
- ✓ *ppc_440_jtagppc_bus* to JTAG
- ✓ *fcb_v2_0 to FPU*

It can be noticed that the different components in the design share the bus, but only the ppc440 processor has a master's connection (*MPLB*) to the bus, while all the others are connected with a slave connection (*SPLB*). The diagram also illustrates the IP modules used in this thesis, some of which are described in section 4.2.1.

Therefore, the interaction between the components can be summarized as follows:

> The modulator sends an interrupt signal to the processor at each sampling period. The processor executes the code, and during execution, the processor fetches the optimal angles from the memory. After the completion of the execution, the processor copies the switching states and times to the modulator. This process repeats itself until a stop command is given to the processor.

4.3. Software setup

The main task of this thesis was to develop a full modulation circuit and a ppc440 program for the implementation of the Programmed modulation technique on FPGA based system. Hence, the detail of the software part of the project is presented here. As it was mentioned in section 3.3, the software language used by Xilinx SDK is C++, and it is also the language used in this thesis to program the processor.

Apart from the main functions that are implemented in the SDK tool, the software also includes *board support package (bsp)* which is generated from the hardware exported from the XPS design. The main function of bsp is to enable the software locate the addresses of the different components in the hardware design and the drivers for these components. In addition, it contains different supportive files for the proper operation of the software.

4.3.1. Software Process Flow Chart

In section 4.1 above, it has been described that programmed modulation technique is time intensive because of the different functions that need to be called and executed for the proper operation of the modulation scheme. The software is divided in to two interrupt routines namely “*fast interrupt routine*” and “*slow interrupt routine*” and one main routine called “Background routine”. The background routine does not have any functions within it but it runs until an interrupt routine is called. Since the objective of the thesis was to test the modulation scheme in an open loop system, the slow interrupt routine is not functioning because all the necessary functions for this kind of test were included in the fast interrupt routine. The fast interrupt is ten times faster than the slower and it has a frequency of 2000 Hz. In fact, this frequency is the same as the carrier signal frequency implemented on the sinusoidal PWM hardware in previous experiments on the same lab setup.

The whole process that takes place in the processor is summarized below in figure 4-12 using flow chart, and this algorithm is implemented using C++ language in the SDK tool.

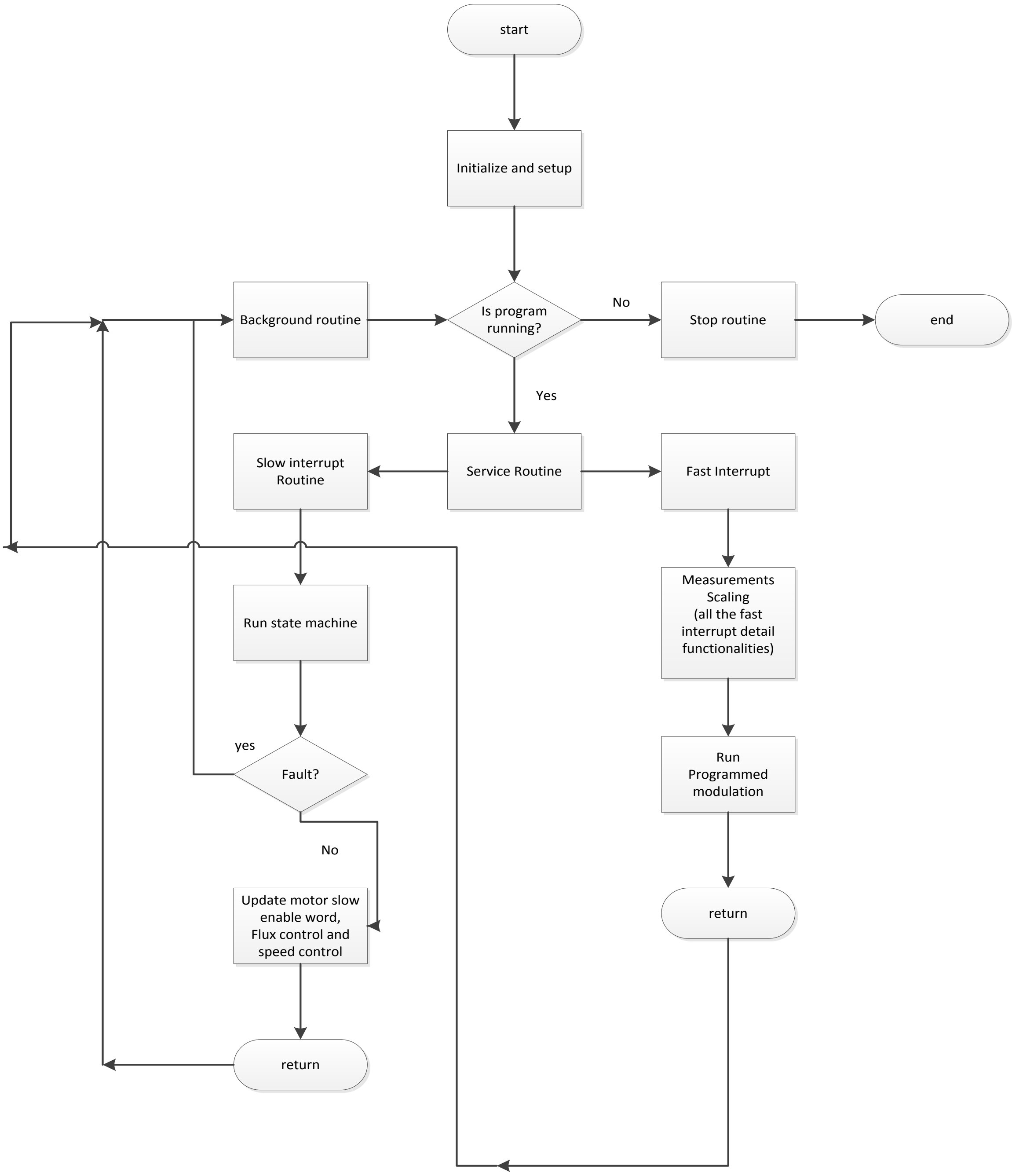


Figure 4-12: Algorithm of the control system implemented on the processor

4.3.2. Programmed Modulation Algorithm (Fast interrupt routine)

Despite its significant advantages for improving the efficiency of the converter, programmed modulation is a demanding task to implement on a finite memory processor system. This challenge comes due to the large number of optimal angles to be stored on the memory of the processor. In this thesis, a set of optimal angles was stored in the memory block for some modulation index (m) values and few variations of the pulse numbers (N).

A hysteresis system was applied to calculate the number of pulses based on the required stator voltage frequency. The hysteresis helps to avoid unwanted variations of the number of pulses that comes from small change in the stator voltage frequency. A description of this phenomenon is shown in table 4.1 and figure 4-13.

The optimal angle selection was done in two ways:

i. All the angles for specific number of modulation indices (m) and a certain number of pulses (N) were stored on the processor memory.
    - ❖ This method has a significant drawback, because the processor memory has not enough space to store all the optimal angles that are needed for a precise and flexible switching pattern.

ii. An online linear interpolation was applied to calculate most of the angles from a selected number of stored optimal angles. The stored angles are those where there is no linear relationship between the angles and the modulation indices.
    - ❖ Despite the fact that this method reduces the amount of memory needed to store all the optimum angles, it has a significant disadvantage of increasing the processing time needed to calculate the optimum angles. The linear interpolation is also not a reliable solution because of the nonlinear relationship of the optimal angles and the modulation indices. This nonlinearity can be seen from figure 2-4.

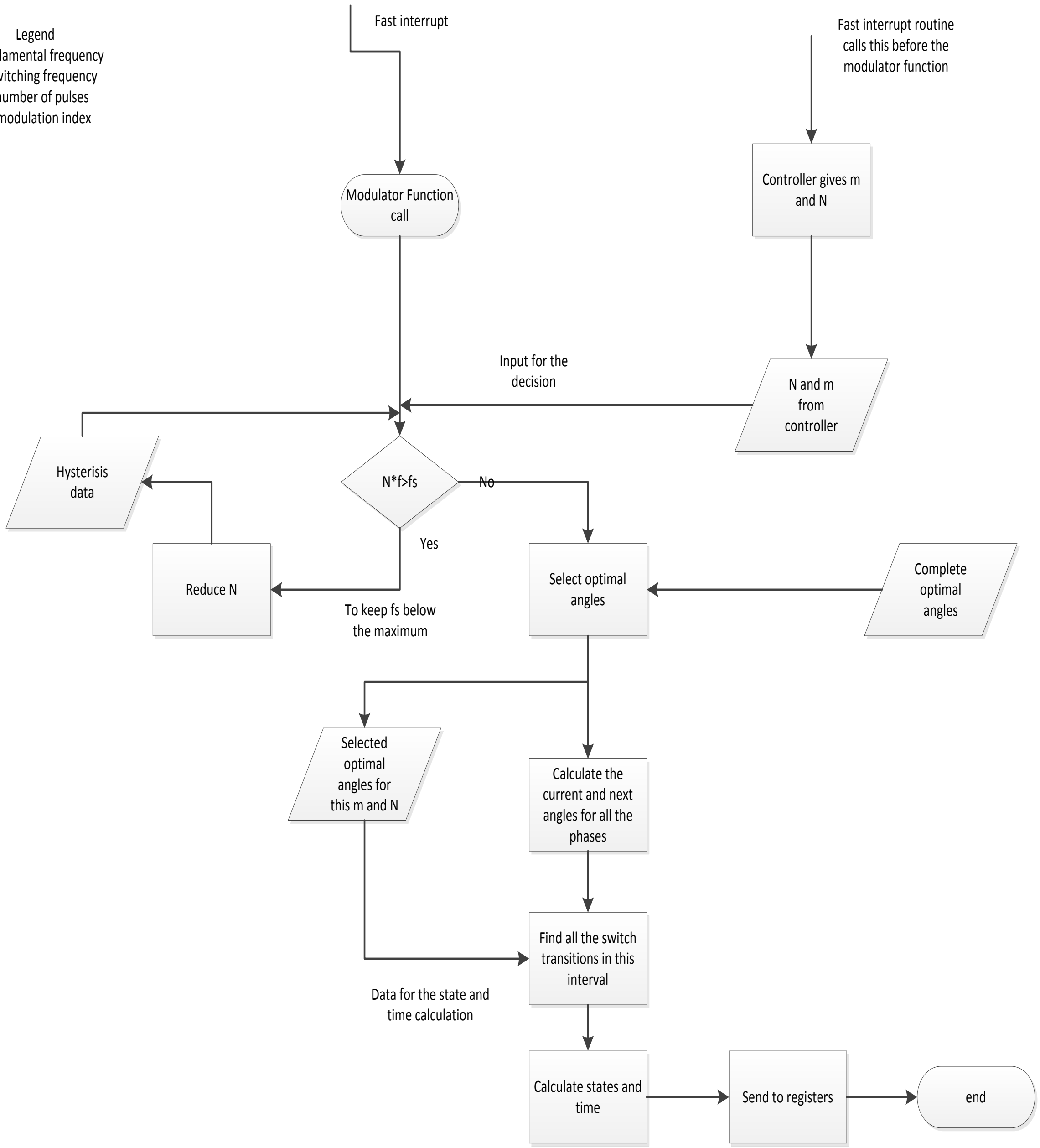

Legend
f=fundamental frequency
fs=switching frequency
n=number of pulses
m=modulation index
Fast interrupt
Fast interrupt routine calls this before the modulator function
Modulator Function call
Controller gives m and N
Input for the decision
N and m from controller
Hysterisis data
N*f>fs
No
Yes
Reduce N
To keep fs below the maximum
Select optimal angles
Complete optimal angles
Selected optimal angles for this m and N
Calculate the current and next angles for all the phases
Find all the switch transitions in this interval
Data for the state and time calculation
Calculate states and time
Send to registers
end

Figure 4-13: Flow chart for the modulation algorithm

Below is a brief explanation of each of the process blocks:

Once the fast interrupt is called the following processes take place:

The modulation function takes two arguments (required fundamental frequency (f) of the voltage and the required amplitude modulation index of the fundamental voltage (m)). The number of pulses (N) is calculated from the switching frequency and the fundamental frequency using a hysteresis method.

Number of pulses N	For N>20 Asynchronous Modulation For higher switching frequency and/or lower motor speed	10	9	8	7	6	5	4	3
Range of fundamental frequency (in Hertz) for each N		27.27-30	30-33.3	33.3-37.5	37.5-43	43-50	50-60	60-75	75-100

Table 4.1. Number of pulse (N) calculation for maximum switching frequency of 300 Hz: Hysteresis method

The modulator then selects N from table 4.1 above, and using the modulation index from the controller output, the program goes to finding the optimal angle pattern for this (m,N) value. The optimal angles for the selected (m,N) are then stored in a temporary variable.

Due to lack of adequate memory on the FPGA board, all the possible number of pulses were not implemented in this thesis. Only N=3,4,5 and 6 were programmed and the curve in figure 4-14 below shows how N varies with increasing stator frequency for the four different values of N.

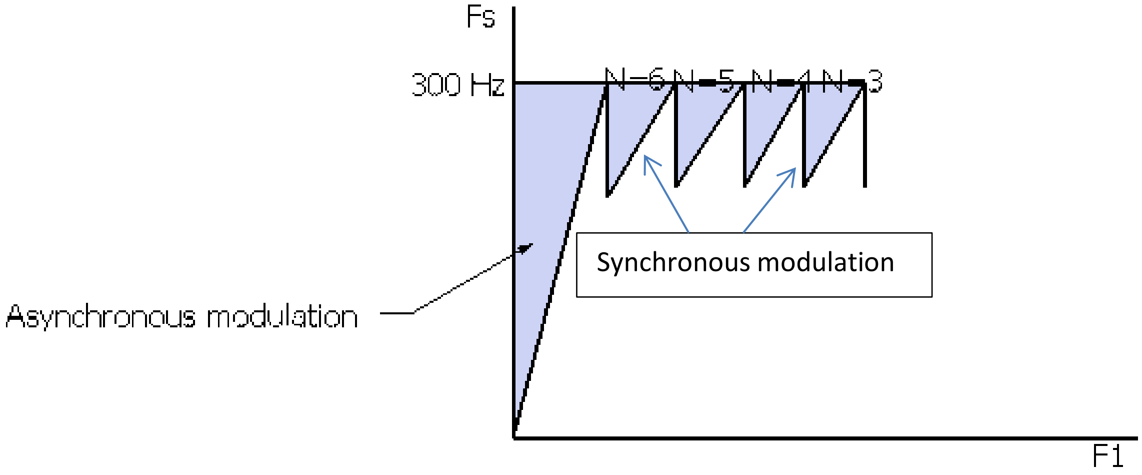


Figure 4-14: Switching frequency versus fundamental frequency

In each sampling period, the starting ($\alpha_{s,2}$) and ending (α_e) voltage angles of the next sampling period are calculated using the fundamental frequency (f), the sampling period (T_s) and the voltage angle at the current sampling period ($\alpha_{s,1}$) as follows:

$$\alpha_{s,2} = \alpha_{s,1} + 2\pi f T_s \qquad (4.1)$$

$$\alpha_e = \alpha_{s,2} + 2\pi f T_s \qquad (4.2)$$

Figure 5-1 in section 5.1 illustrates the above expressions.

Using the starting and ending voltage angles of the next sampling period, all the voltage angle values in the optima pattern which lie between these two angles is selected and stored in another variable. These angle values are then changed to switching time instants using the following relation:

$$t_n = \frac{\alpha_n - \alpha_e}{360 f} \qquad (4.3)$$

Where $t_n$is the n^{th} switching time in the next sampling period, α_n is the n^{th}optimal angle that lies in the starting and ending angles of the next sampling time.

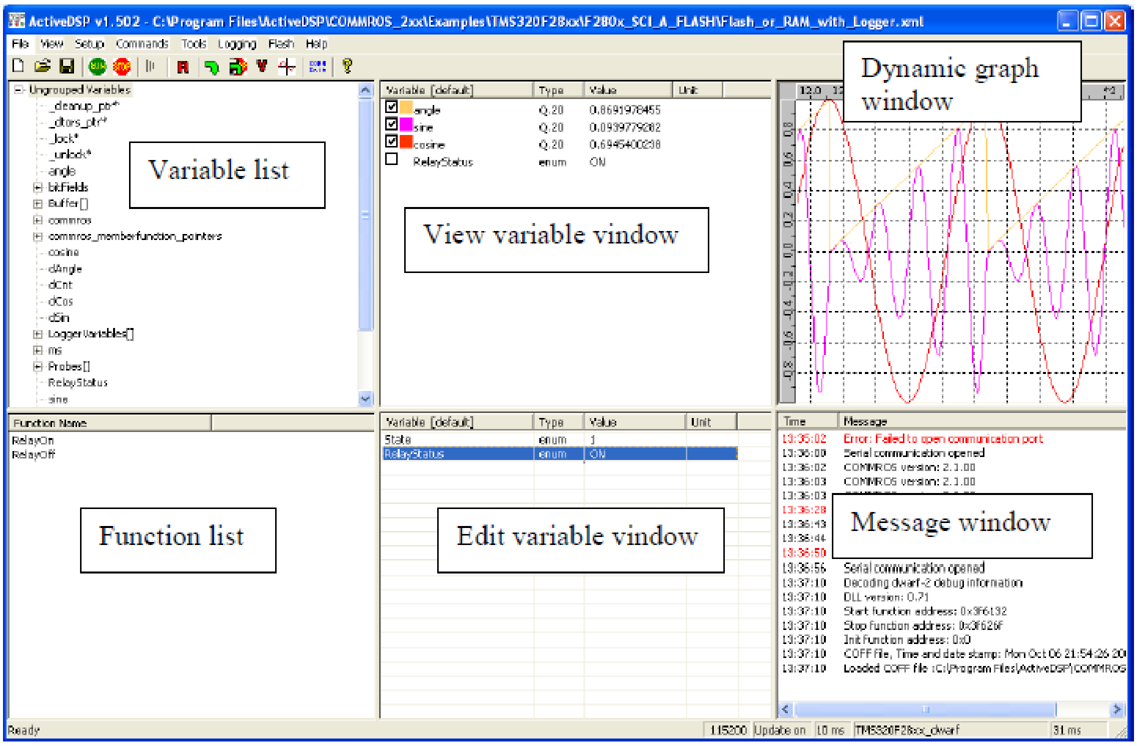
The above calculated time is the actual time the switch is going to change state within the next sampling period. However, the modulator has a digital timer in the form of a counter that is reset at the end of each sampling period and the above calculated time is changed in to this counter values (t_d) as follows:

$t_d = t_n * maximum\ count.$

The $maximum\ count$ is decided by the clock frequency of the processor bus which in turn is fed by the clock generator on the FPGA board and for the FPGA board used in this thesis, the frequency of the processor is 100 MHz. Therefore, for a sampling frequency of 1KHz, $maximum\ count$ becomes 100,000.

So far only the time instants where the switches change state have been calculated. A separate routine is built in the software that specifies which of the 3 levels (+, - or Neutral) the converter changes state to, and once the time instants and the switch states are calculated, the values are sent to the respective registers. There are certain sampling periods where there is no transition of the switches, and in these intervals, the switch states from the previous sampling period are maintained.

In order to be able to interact with the processor and monitor the variables of the program, ActiveDSP software is used. A screen shot of ActiveDSP software is shown below in figure 4-15. The screen has different windows which are used for monitoring and updating the variables in the software, and it also allows us to debug our program by providing real time feedback from the processor. It communicates with the processor using RS232 cable in addition to its capability to log data so that analysis of step change in some of the variables in the processor is possible.



ActiveDSP v1.502 - C:\Program Files\ActiveDSP\COMMROS_2xx\Examples\TMS320F28xx\F280x_SCI_A_FLASH\Flash_or_RAM_with_Logger.xml

```
⊟ Ungrouped Variables
    _cleanup_ptr*
    _dtors_ptr*
    _lock*
    _unlock*
    angle
  ⊞ bitFields
  ⊞ Buffer[]
  ⊞ commros
  ⊞ commros_memberfunction_pointers
    cosine
    dAngle
    dCnt
    dCos
    dSin
  ⊞ LoggerVariables[]
  ⊞ ms
  ⊞ Probes[]
    RelayStatus
    sine
```

Variable [default]	Type	Value	Unit
☑ angle	Q.20	0.8691978455	
☑ sine	Q.20	0.0939779282	
☑ cosine	Q.20	0.6945400238	
☐ RelayStatus	enum	ON	

Function Name	
RelayOn	
RelayOff	

Variable [default]	Type	Value	Unit
State	enum	1	
RelayStatus	enum	ON	

```
Time      Message
13:36:02  Error: Failed to open communication port
13:36:00  Serial communication opened
13:36:02  COMMROS version: 2.1.00
13:36:03  COMMROS version: 2.1.00
13:36:03
13:36:28
13:36:43
13:36:44
13:36:50
13:36:56  Serial communication opened
13:37:10  Decoding dwarf-2 debug information
13:37:10  DLL version: 0.71
13:37:10  Start function address: 0x3F6132
13:37:10  Stop function address: 0x3F626F
13:37:10  Init function address: 0x0
13:37:10  COFF file, Time and date stamp: Mon Oct 06 21:54:26 20
13:37:10  Loaded COFF file :C:\Program Files\ActiveDSP\COMMROS
```

Ready | 115200 | Update on | 10 ms | TMS320F28xx_dwarf | 31 ms

Figure 4-15: ActiveDSP window screen shot

5. Results and discussion

This chapter presents the main findings of the master's work. The implication of the results is also discussed and analyzed.

5.1. Introduction

The thesis work was mainly concerned with the study, design and implementation of programmed modulation on FPGA board. The work also included a brief study of the theory behind harmonic content of a three level inverter output, switching losses, dc-bus balancing methods in programmed modulation and synchronous modulation. The problem under study consists two parts: software design and hardware design and implementation. The detail hardware (modulator) studies were studied by a fellow student [28].

Based on the theoretical studies made, the following results were obtained:

- For medium voltage drives where the voltage level of the converters is higher, a new method of modulation is needed to avoid two problems:
    - Higher switching losses which are found to be proportional to the voltage
    - Significantly increased harmonic losses which also increase with the voltage
- The task of solving these two problems is not straight forward because if the switching frequency is reduced to minimize the switching losses, the harmonic content of the output also increases which is a problem by itself.
- Synchronous Modulation was studied which was found to be theoretically able to satisfy both reduced switching losses and minimum harmonic losses by reducing the switching frequency.

The problem under study was thoroughly investigated with the help of Matlab/Simulink models by Roger Enes [7]. The results obtained show that programmed pulse pattern generation, if implemented using synchronous modulation, is an ideal solution for alleviating the problem of high harmonics and larger switching losses in steady state operation of the drives. This interesting advantage of programmed modulation can also be easily extended to dynamic operations of drives by implementing a fast trajectory control. Therefore, designing a programmed modulation is the very first step to harness this unique advantage. In addition, testing the feasibility of the system and exploring the challenges of the scheme are included in the experimental work part of this master's thesis.

Having been motivated by the above findings, programmed modulation scheme was developed and switching patterns were generated for different number of pulses under different modulation indices. The generated switching patterns were then applied to a three level inverter, and the resulting wave form was analyzed. After careful analysis of these waveforms, a six phase induction machine was connected to the inverter, and the resulting load current waveform was thoroughly investigated to see the harmonic content of the current.

5.2. Simulation results and discussion

In this subsection, the software implemented on the processor is tested if it produces the necessary and required outputs for the modulator. ActiveDSP was used to monitor and control the variables on the processor, and some variables were also logged and plotted on Matlab.

The calculation of the switching instants for an n^{th} sampling period is done in the $(n-1)^{th}$ sampling interval before it as discussed in section 4.3.2. In order to accomplish this, the stator angle in the beginning ($\alpha_{s,2}$) and the end (α_{e}) of the interval for which the patterns are going to be applied is calculated. Figure 5-1 below shows the two extreme (starting and ending) angle values for 25 sampling periods.

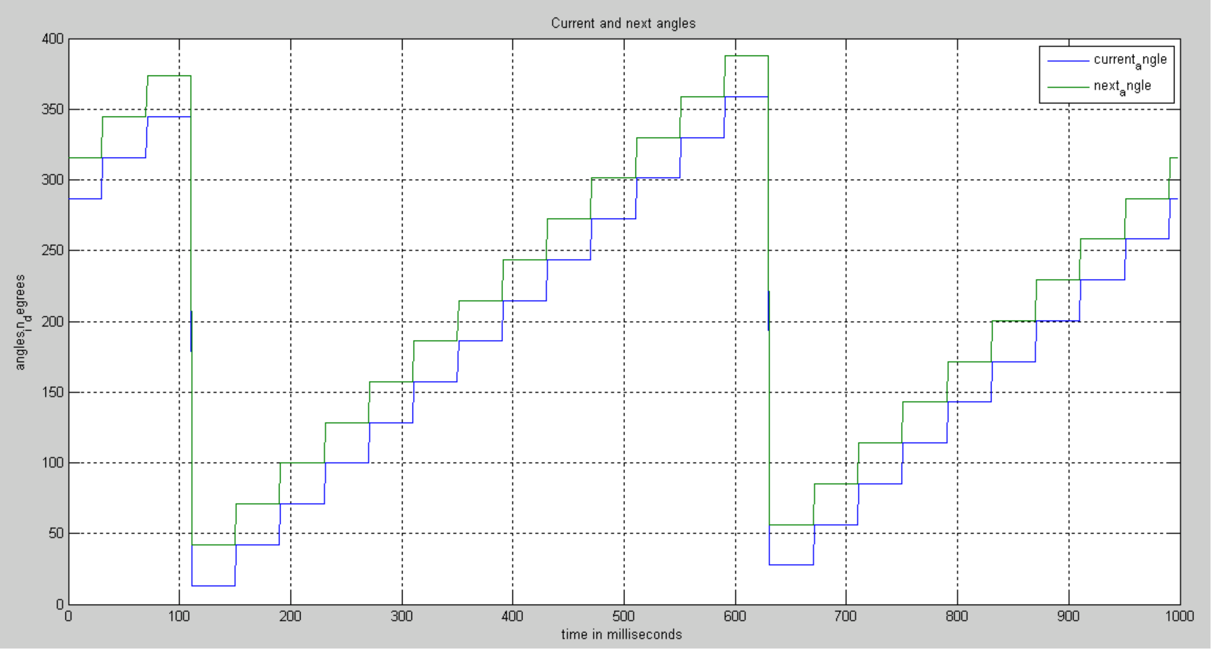


Figure 5-1: The starting and ending angle of the next sampling period for phase a stator voltage frequency of 80 Hz.

When we calculate the two boundaries in the next sampling period, we use the stator voltage frequency received in the current sampling interval. This is because the sampling frequency is high compared to the stator voltage frequency, and hence it is assumed that the stator voltage (both in magnitude and frequency) change is not significant.

Figure 5-2 below shows the voltage angles of all the three phases as they increment from minimum to maximum, and when the maximum exceeds 360 degrees, it is decremented by 360 degrees to start the next fundamental stator voltage period. It is also clear from the figure that the three phases lag each other by 120 degrees.

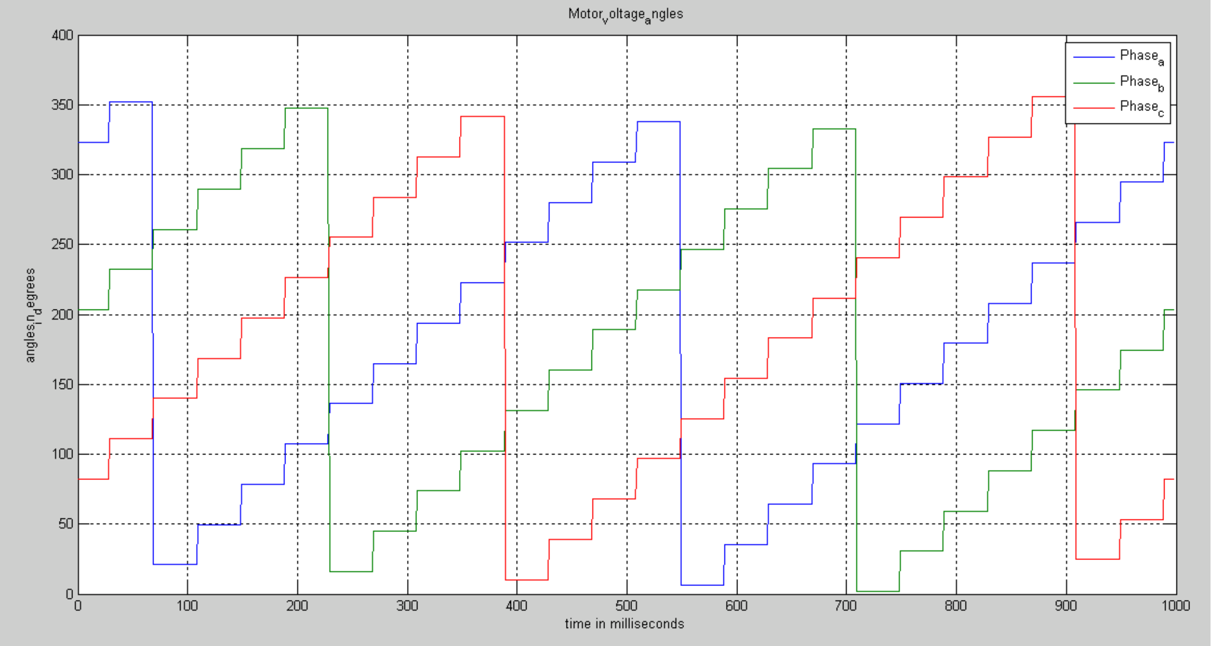


Figure 5-2: The starting and ending angle of the next sampling period for the three phases at a stator voltage frequency of 80 Hz.

After finding the starting and ending angles of the next sampling interval, the program then scans and finds out those angles in the optimal angles pattern selected for that particular modulation index and pulse number which lie in these two extreme angle values using the relation in section 4.2.2. The switching times are calculated for each angle, and then together with the switching states, are sent to the register to be stored until the modulator hardware fetches them.

Figure 5-3 shows the two voltage angles (current and next) for phase a in each sampling period together with the switching patterns that take place in the sampling period for $N = 3$ and $m = 0.5$.

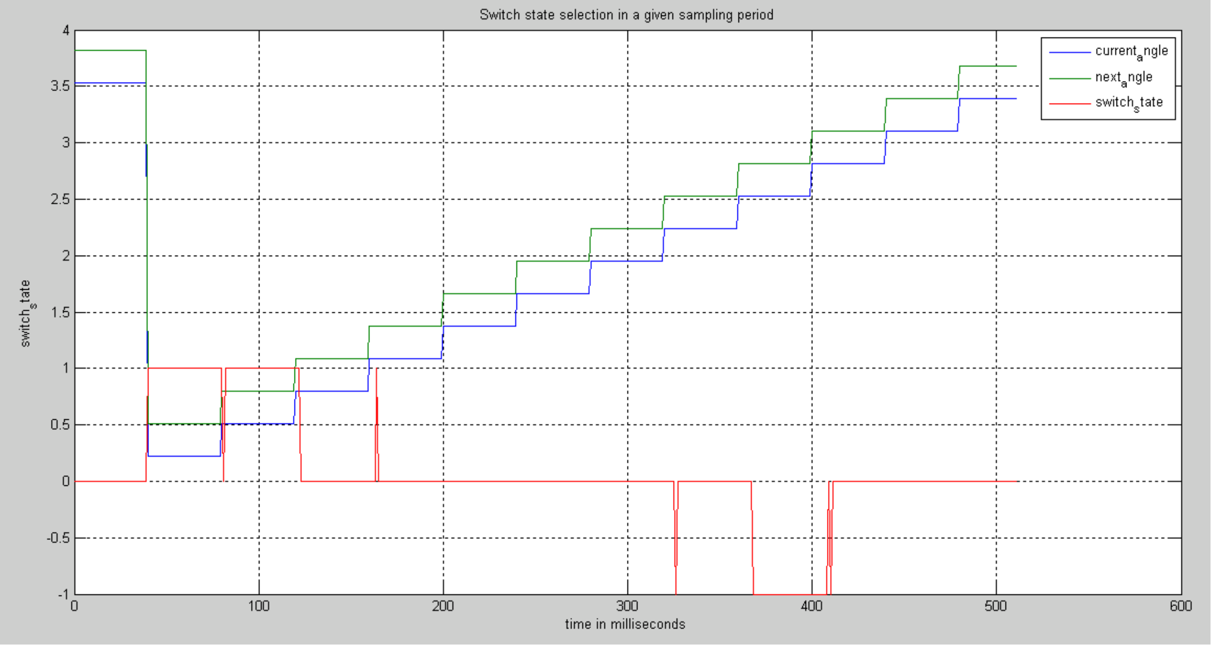
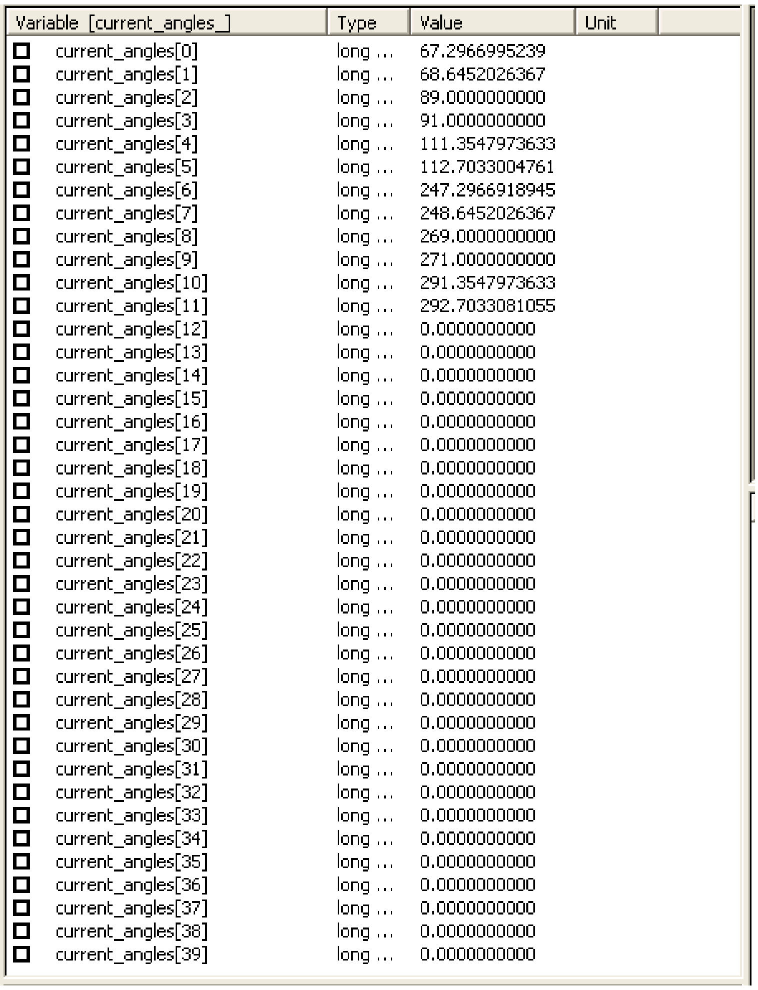


Figure 5-3: Current and next sampling angles and the switching patterns for 12 switching instants per fundamental period of the stator voltage. The angle values are normalized by 100 for the sake of showing the switching states on the same curve.

In the following section, a brief discussion of the switching time calculation for pulse number $N = 3$ is presented where N is the number of switching instants per quarter period of the stator voltage frequency.

A variable that holds the temporary optimal angles for the sampling period under consideration is an array called current_angles. The size of the array is 40, and 40 was selected because the maximum number of switching instants per quarter of the fundamental period is 10. That is $N = 10$ in this thesis, and hence the maximum switching angles per one complete fundamental period is $4 * 10 = 40$.

A screen shot of this variable for $N = 3$ is shown below in figure 5-4. It can be understood from the figure that there are 12 switching instants per fundamental period, and the optimal angles produce a quarter wave symmetric sinusoidal wave. All the array elements are not populated because the array is dimensioned for the maximum optimal angles, i.e. 40. The optimal angles in the C++ code that correspond to this modulation index and number of pulses are found in Appendix B and can be cross-checked with the processor output in the figure.



Variable [current_angles_]	Type	Value	Unit
□ current_angles[0]	long ...	67.2966995239	
□ current_angles[1]	long ...	68.6452026367	
□ current_angles[2]	long ...	89.0000000000	
□ current_angles[3]	long ...	91.0000000000	
□ current_angles[4]	long ...	111.3547973633	
□ current_angles[5]	long ...	112.7033004761	
□ current_angles[6]	long ...	247.2966918945	
□ current_angles[7]	long ...	248.6452026367	
□ current_angles[8]	long ...	269.0000000000	
□ current_angles[9]	long ...	271.0000000000	
□ current_angles[10]	long ...	291.3547973633	
□ current_angles[11]	long ...	292.7033081055	
□ current_angles[12]	long ...	0.0000000000	
□ current_angles[13]	long ...	0.0000000000	
□ current_angles[14]	long ...	0.0000000000	
□ current_angles[15]	long ...	0.0000000000	
□ current_angles[16]	long ...	0.0000000000	
□ current_angles[17]	long ...	0.0000000000	
□ current_angles[18]	long ...	0.0000000000	
□ current_angles[19]	long ...	0.0000000000	
□ current_angles[20]	long ...	0.0000000000	
□ current_angles[21]	long ...	0.0000000000	
□ current_angles[22]	long ...	0.0000000000	
□ current_angles[23]	long ...	0.0000000000	
□ current_angles[24]	long ...	0.0000000000	
□ current_angles[25]	long ...	0.0000000000	
□ current_angles[26]	long ...	0.0000000000	
□ current_angles[27]	long ...	0.0000000000	
□ current_angles[28]	long ...	0.0000000000	
□ current_angles[29]	long ...	0.0000000000	
□ current_angles[30]	long ...	0.0000000000	
□ current_angles[31]	long ...	0.0000000000	
□ current_angles[32]	long ...	0.0000000000	
□ current_angles[33]	long ...	0.0000000000	
□ current_angles[34]	long ...	0.0000000000	
□ current_angles[35]	long ...	0.0000000000	
□ current_angles[36]	long ...	0.0000000000	
□ current_angles[37]	long ...	0.0000000000	
□ current_angles[38]	long ...	0.0000000000	
□ current_angles[39]	long ...	0.0000000000	

Figure 5-4: Content of the current_angles variable for N=3 and m=0.05

From figure 5.4 above, the optimal angles for this (m,N) combination are: 67.2967, 68.045, 89, 91, 111.3545, 112.7033, 247.2967, 248.6452, 269.00, 271.00, 291.3545 and 292.7033. These optimal angles produce a switching pattern shown below in figure 5.5. It is worth noting that figure 5.5 shows the sampling period in which the optimal angles are found, not the switching times, and it is for this reason that there is no symmetry in the waveform. The switching times are calculated from these angles, and the result is presented in the experimental section.

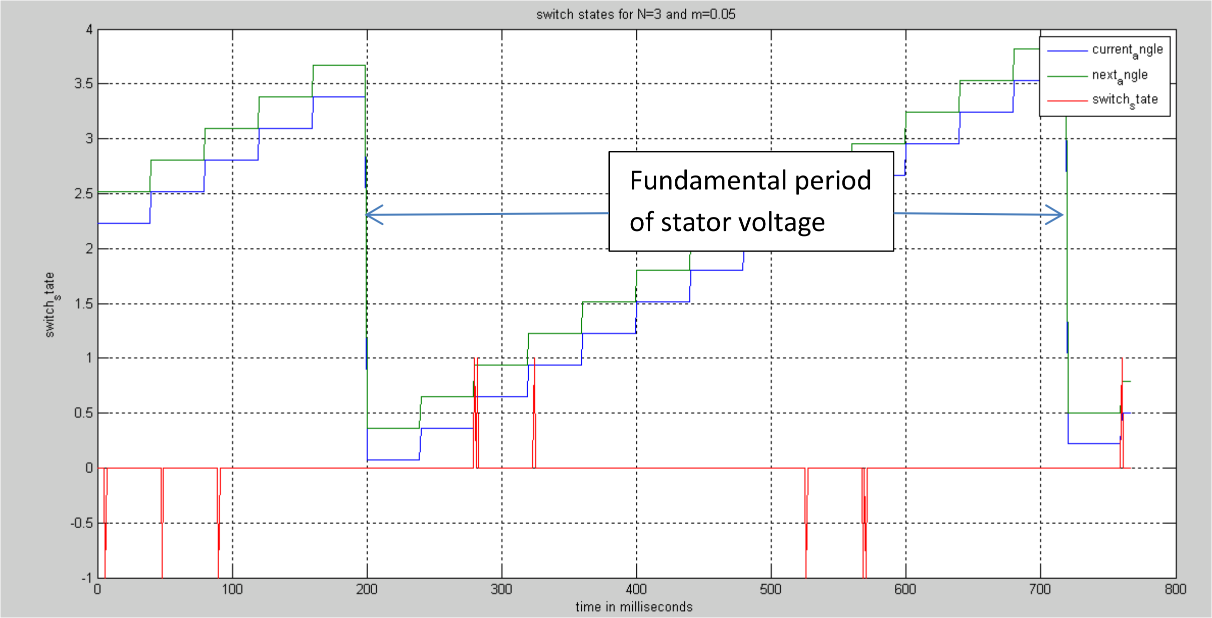


Figure 5-5: Switching pulses and the voltage angles. The angle values are normalized by 100 for the sake of showing the switching states on the same curve.

As we can see from the above curve, there are 12 switching instants in the whole fundamental period for the given maximum switching frequency (300 Hz in this case) and modulation index (0.05 in this case).

A detail look at the figure also enables one to notice that the number of pulses in each quarter period is equal, which is the feature of synchronous modulation. Quarter wave symmetry of the pulses will be shown in section 5. 3 after the angles are converted in to switching times. Below in figure 5-6 is a clearer view of the switching angles for half period and quarter period.

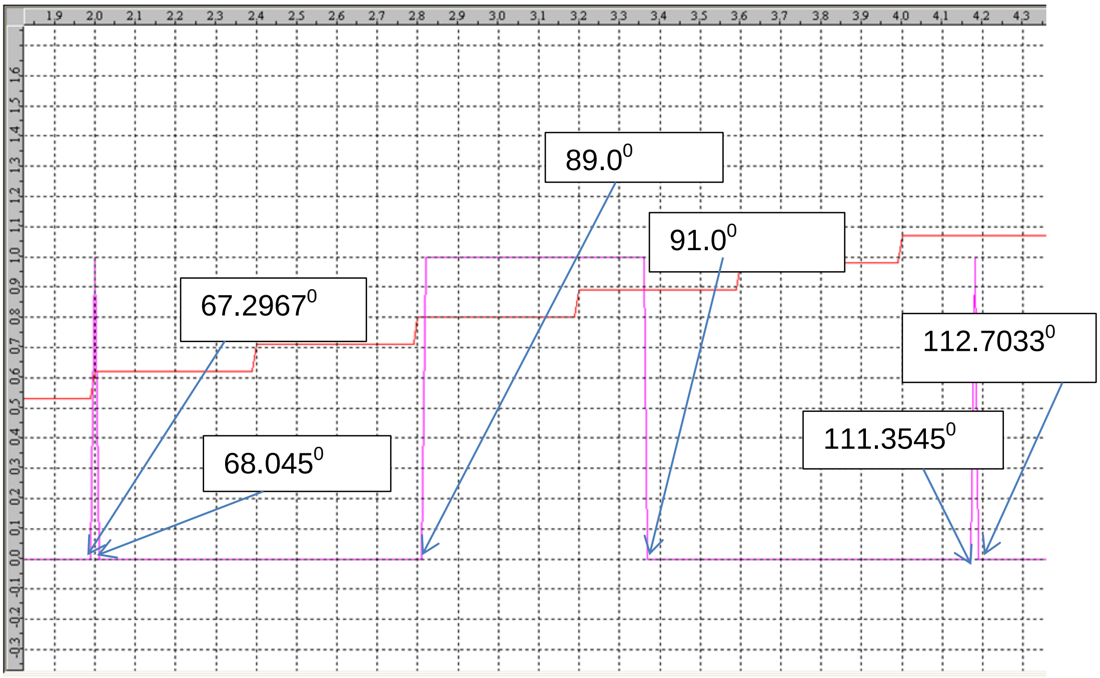


Figure 5-6: Switching pulses (purple) and the voltage angle (red) for a half a period. The angle values are normalized by 100 for the sake of showing the switching states on the same curve.

From figure 5-6 above, it can be noted that for half of the fundamental period of the voltage, there are six switching instants and the switching angles are symmetric with respect to 90^0.

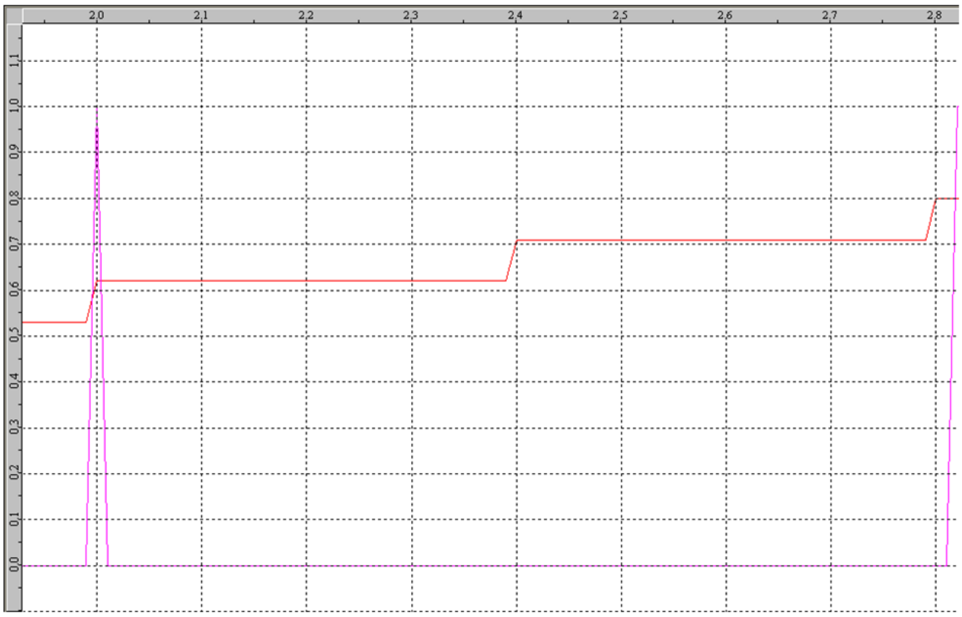


Figure 5-7: Switching pulses (purple) and the voltage angle (red) in degrees in a quarter period. The angle values are normalized by 100 for the sake of showing the switching states on the same curve.

Figure 5-7 on the other hand demonstrates the switching instants that take place in a quarter period. There are three switching instants where the switches change states which correspond to the three optimal angles: 67.2967, 68.045, and 89. A detail look at the figure shows that the switching pulses are produced based on the sampling interval extreme (starting and ending) angle values. That is, for example, the first pulse in the figure that corresponds to the optimal angles 67.2967, and 68.045 is produced during the sampling interval 62 to 71.

The figure also illustrates that there are certain sampling intervals where there is no any switching instant. In these kinds of intervals, the switches do not change their state. This then leads to the conclusion that the switching frequency is different for different sampling intervals, but the average switching frequency remains below the maximum for the whole stator voltage fundamental period.

The effect of a changing modulation index was also tested and the result shows that the program responds for any change in modulation indices. As it can be noticed from figure 5-8, when the modulation index increases, the time duration for the upper (+) and lower (-) switches is longer compared to that of the middle (Neutral or 0) switch. Here a more unrealistic and exaggerated step change of modulation index from 0.04 to 1 is used just to show the response.

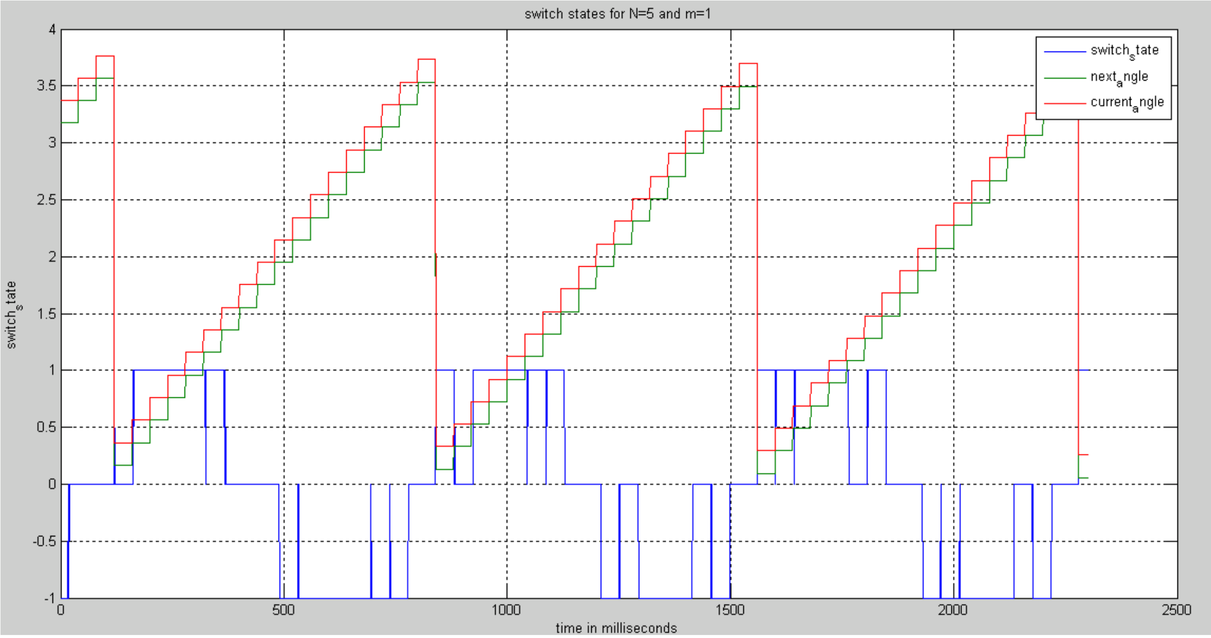


Figure 5-8: Switching pulses when the modulation index (blue) changes from 0.04 to 1 pu. The angle values are normalized by 100 for the sake of showing the switching states on the same curve.

5.3. Experimental results and discussion

In this sub-section, the main findings of the experimental work are discussed. The modulation technique is fully implemented on hardware, and different tests are made to observe the performance of the scheme. The specifications of the hardware modules used for the experiment is given in section 4.1.

5.3.1. Gate drive signals

The overall connection of the inverter to the FPGA board is as shown in figure 5-9 below. It is known that a three level inverter can be considered as a combination of two 2-level inverters connected in such a way that it produces a three level waveform. Accordingly, switches S_{a1} and S_{a3} in leg 1 enable the inverter output voltage to bounce between 0 and V_{dc}, while switches S_{a2} and S_{a4} enable the inverter to switch between 0 and $-V_{dc}$.

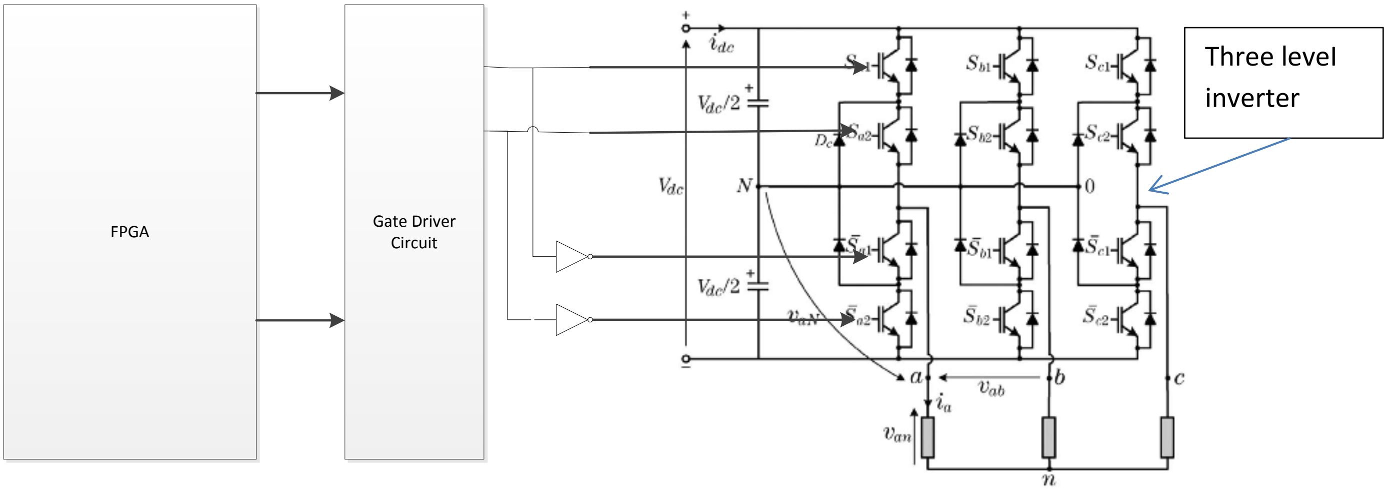
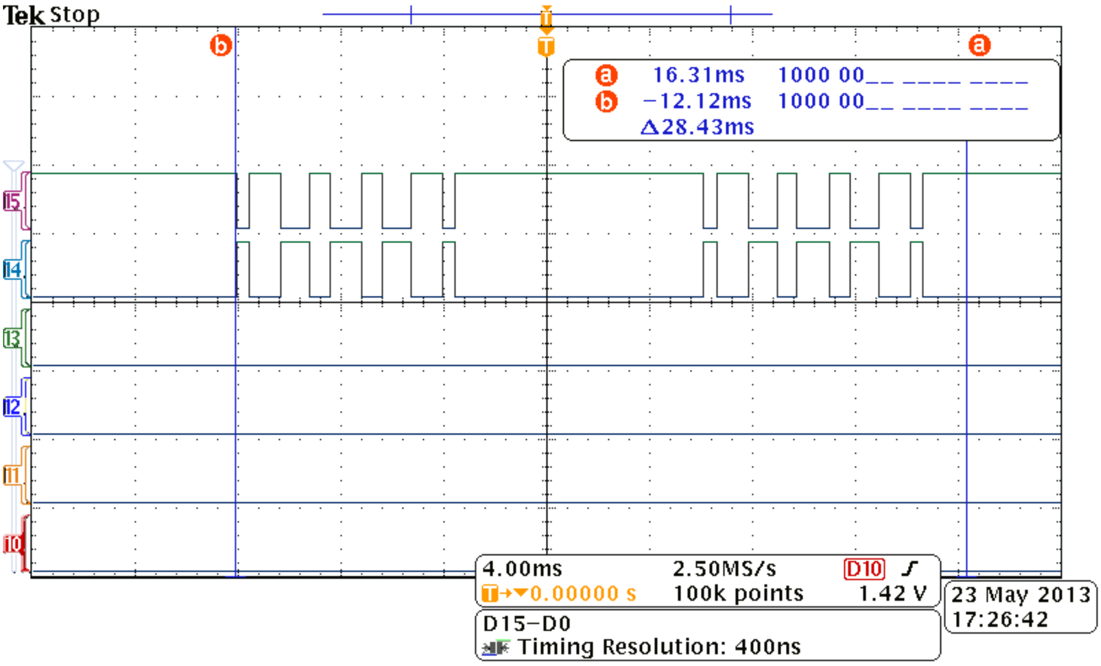


Figure 5-9: Connection between the FPGA, gate driver and the inverter

It is also important to make sure that the switches that eventually can short circuit the dc bus should not be ON at the same, which for example may happen if switches 1 and 3 or switches 2 and 4 are turned ON simultaneously. Therefore, these pairs of switches that have a danger of shorting the dc bus are fed with a pair of signals, one an inverted version of the other, by the FPGA board. Figure 5-10 below shows the signals that go to these switches.



```
a   16.31ms   1000 00__ ____ ____
b  −12.12ms   1000 00__ ____ ____
   Δ28.43ms
4.00ms        2.50MS/s      D10  ∫
T→▼0.00000 s  100k points   1.42 V
D15−D0                              23 May 2013
Timing Resolution: 400ns            17:26:42
```

Figure 5-10: Input signals to the two switches in the bridge

As it can clearly be seen from figure 5-10 above, the pairs of switches receive 2 signals, one a negation of the other, and this ensures that the dc-bus is not short-circuited. The figure also shows that gate drive signal is perfectly produced with the required frequency and required pulse width (in which case the pulse width signifies the magnitude of the voltage needed).

An investigation was also made to make sure that the three gate drive signals for the different phases are 120^0 spaced from each other.

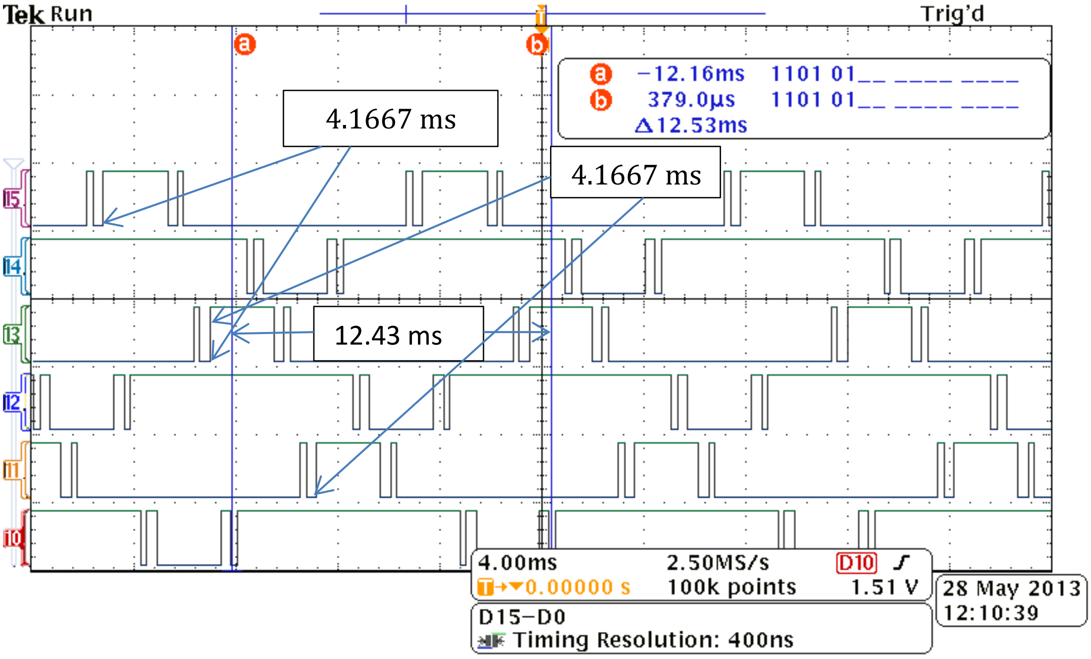


Figure 5-11: Three phase gate drive signals

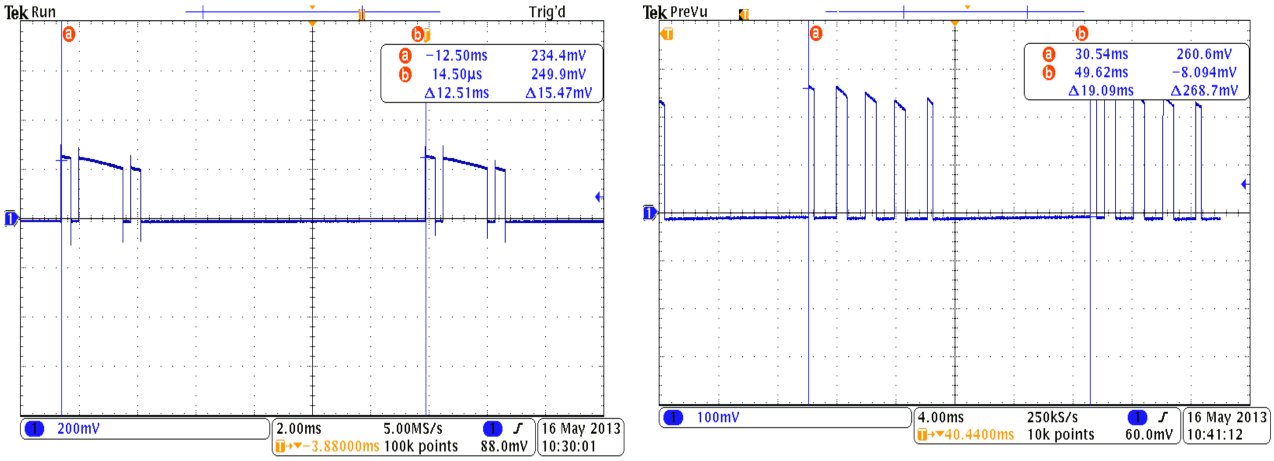
Figure 5-11 above demonstrates two important facts about the modulation strategy. First, it can be seen that the gate drive signals for the three different phases are 120^0 apart from each other. The fundamental frequency that was given as input for the modulator to produce the above waveform is 80 Hz. Therefore, a phase shift of 120^0 is equivalent to a time shift of:

$t = {}^{\alpha}/_{2\pi f} = {}^{(2\pi/3)}/_{(2\pi * 80)} = 4.1667$ ms. A careful look at the above waveform reveals that the three phases are shifted by the calculated time.

Second, the fundamental frequency at the output should be the same as the desired frequency that is given as input for the modulator. The waveform in figure 5-11 above shows that the period of the gate signals is 12.53 ms (this can be seen from the time difference between the two vertical lines labeled a and b), which is in clear agreement with the calculated period. That is, the input frequency is 80 Hz and the period is $T = \frac{1}{f} = \frac{1}{80} = 12.5ms$

5.3.2. 2-level inverter output phase voltage

In section 5.2.1 above, it was clarified that a 3-level inverter can be seen as a module consisting of two 2-level inverters. To demonstrate this, the gate drive signal generated for a 3-level inverter was fed to a 2-level inverter, and the following voltage waveform was captured at the terminal of the 2-level inverter.



(a) (b)

Figure 5-12: 2-level converter output voltage: (a) m=0.65,f=80Hz (N=3)

(b) m=0.5,f=55Hz (N=5)

The waveforms in figure 5-12 above demonstrate that the proposed modulation scheme is designed and implemented perfectly in such a way that the magnitude and the frequency of an inverter output voltage are controlled smoothly. It can also be deduced that the modulation scheme designed for a 3-level converter can easily be interfaced to a 2-level inverter without any additional hardware requirement.

5.3.3. 3-level inverter output phase voltage

So far it has been shown that the signals produced at the gate drives are accurately produced and these signals are tested in a 2-level inverter. However, the modulation algorithm was developed for a three level inverter, and hence the produced gate drive signals were applied to a three level inverter and the output voltage of the inverter is investigated.

All the phase voltages presented here after are measured with respect to the neutral point of the dc-bus.

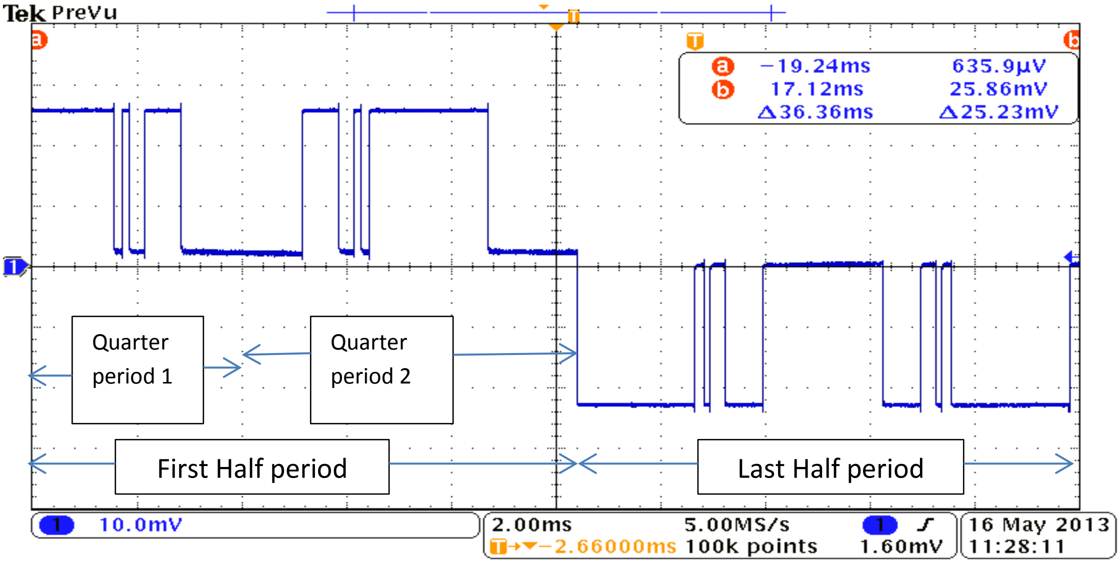


Figure 5-13: 3-level inverter phase a output voltage for m=0.7 and f=45Hz (N=6)

Figure 5-13 above elaborates that the modulation strategy works perfectly well for a 3-level inverter too. It is also clear from the figure that the converter has three output levels measured with respect to the neutral point of the dc-bus. The desired qualities of the voltage waveform can also be clearly observed in the output signals. Some of these characteristics are:

- Quarter wave symmetry is one of the main criterias used during calculating the optimal angles, and the calculated optimal angles should produce a quarter wave symmetric output voltage. This goal was achieved as it can be seen from the figure above. The number of voltage pulses in each quarter period (N, in this thesis) and the position of the pulses with respect to each other is the same for all the quarter periods. In addition, the magnitude of the voltage pulses (+V,0,-V) is the same for the two quarter periods in half period. This in general shows that the number of pulses in the positive half period and negative half period is the same, a feature that is the main aim of synchronous modulation.
- The output frequency should be the same as the one desired by the motor and hence to that given by the controller or the user (in this master’s thesis). The figure elaborates that the period of the output voltage is 22.22 ms, which is equivalent to the input frequency given by the user ($45Hz = \frac{1}{22*10^{-3}s}$)

The output voltage can be varied both in magnitude and frequency by varying the corresponding parameters on the control board (on ActiveDSP in this master's thesis). Figure 5-14 below is the output voltage waveform obtained for a new combination of m and N given by the user.

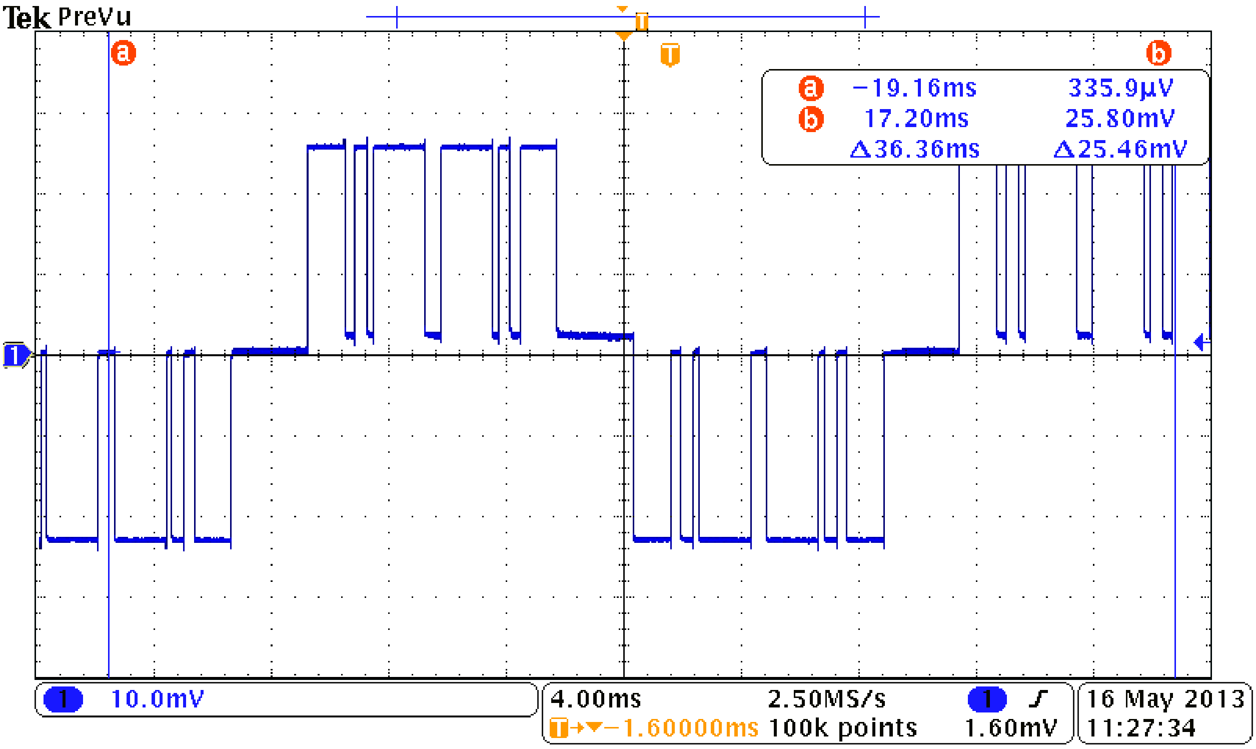
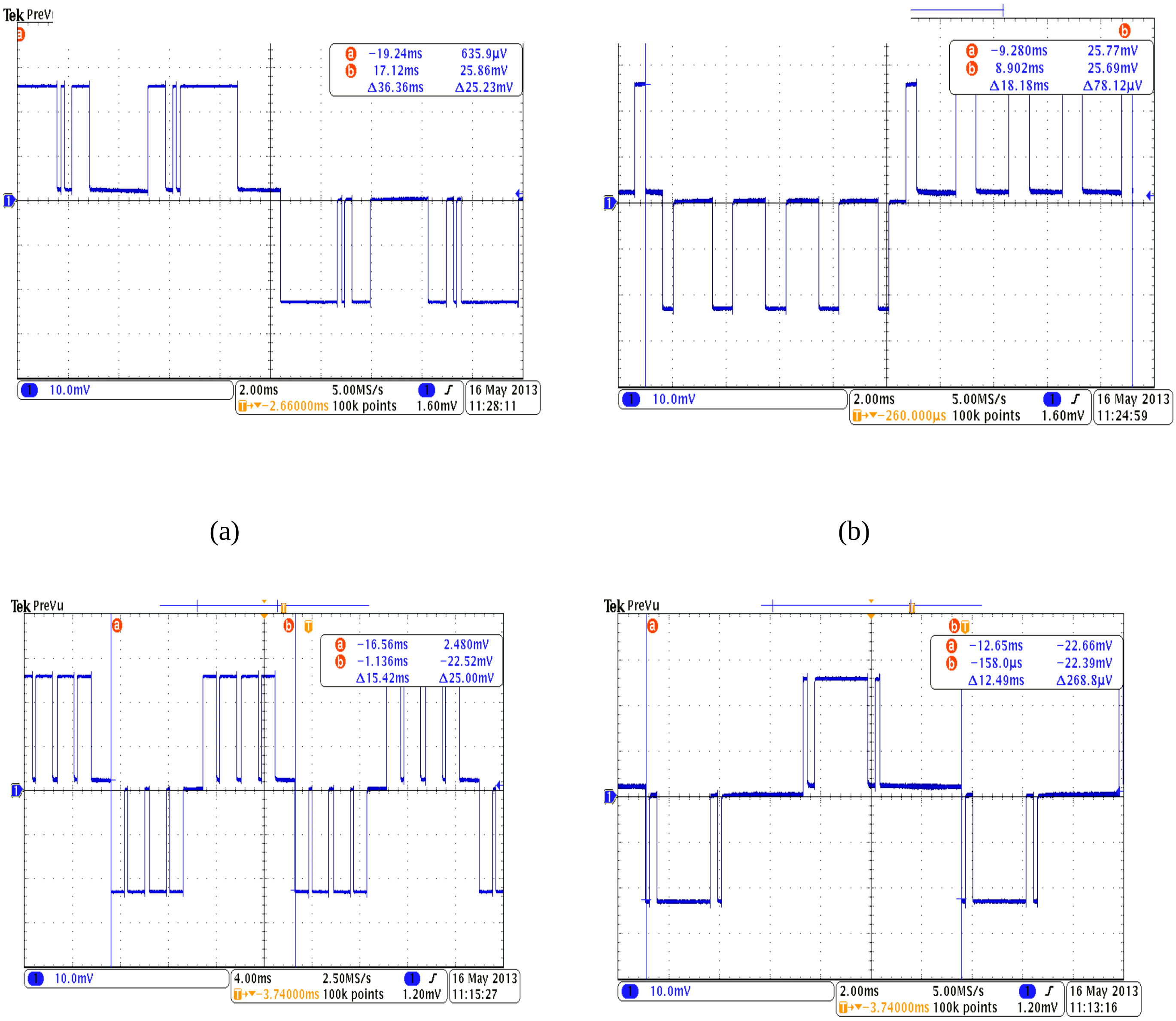


Figure 5-14: Inverter output voltage for phase a when m=0.98 and f=45(N=6)

The functionality of the hysteresis control can also be noticed here. That is, the number of pulses is kept constant for a certain fundamental frequency range by changing the switching frequency. This enables the modulator not to respond for minor frequency and modulation index changes, which otherwise disturbs the optimal pattern.

However, the switching frequency is also maintained below the maximum allowable (300 Hz, in this thesis) in which case if increased fundamental frequency is needed, then it is realized at the expense of a lower number of pulses. A detail description of this phenomenon was presented in section 4.2.2 and the result of the implementation is briefly discussed using the waveforms in figure 5-15 below.



(a)

(b)

(c)

(d)

Figure 5-15: Inverter output voltage for phase a: (a) N=6,f=45 (b) N=5,f=55 (c) N=4, f=65 (d) N=3, f=80

The four waveforms presented in figure 5-15 above show how the number of pulses changes when we change the fundamental frequency from a higher value to a lower one. In waveform (a), the motor is running at 45 Hz and the number of pulses needed for this is $N = 6$, which means the average switching frequency is $f_s = f * N = 45 * 6 = 270Hz$. This switching frequency is well below the 300 Hz switching frequency, which is used as the maximum in this master's thesis. When the motor needs a frequency of 55 Hz, as it is shown in waveform (b) in the above figure, if the same N=6 is used in this case, the switching frequency would be $f_s = f * N = 55 * 6 = 330Hz$, and the switching frequency here is higher than the maximum. The hysteresis routine then reduces the number of pulses to N=5 to decrease the switching frequency to $f_s = f * N = 55 * 5 = 275Hz$. The same phenomenon happens when we go from 55 Hz to 65 Hz and then 80 Hz. Hence, it is possible to keep the switching frequency below a certain desired maximum by changing the number of pulses. Graphical illustration of this phenomenon is given below in figure 5-16.

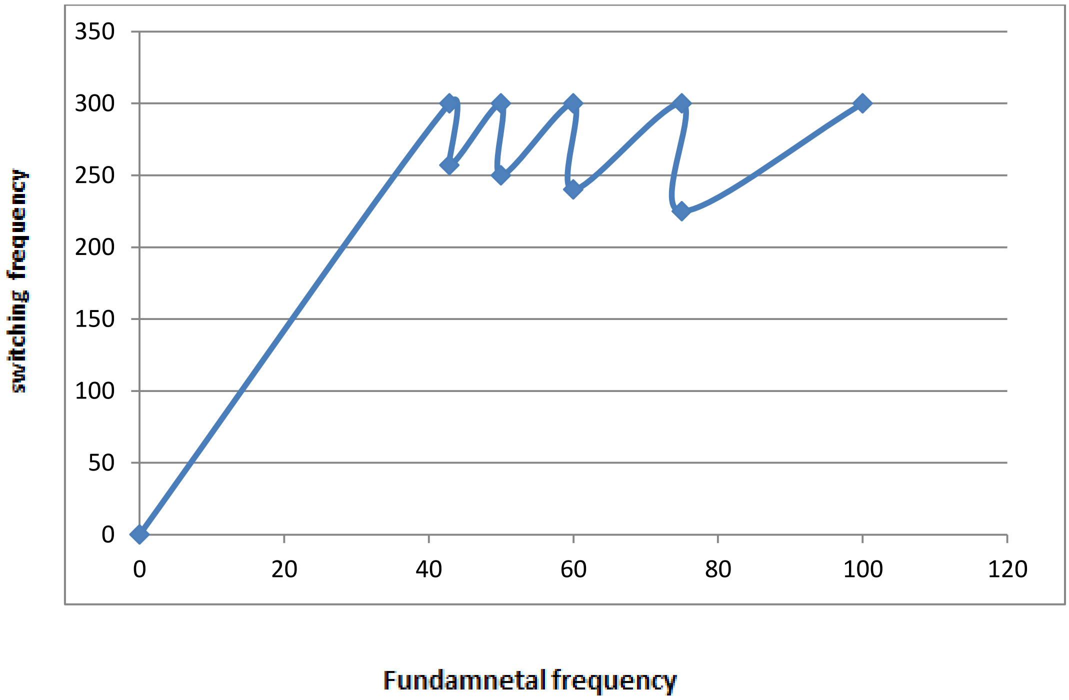


Figure 5-16: Implemented hysteresis number of pulses selector

The three phase voltage in the figure below shows that the modulation strategy can be easily extended to any number of phases by duplicating the components in the hardware module and giving the necessary phase lag between the phases in the software.

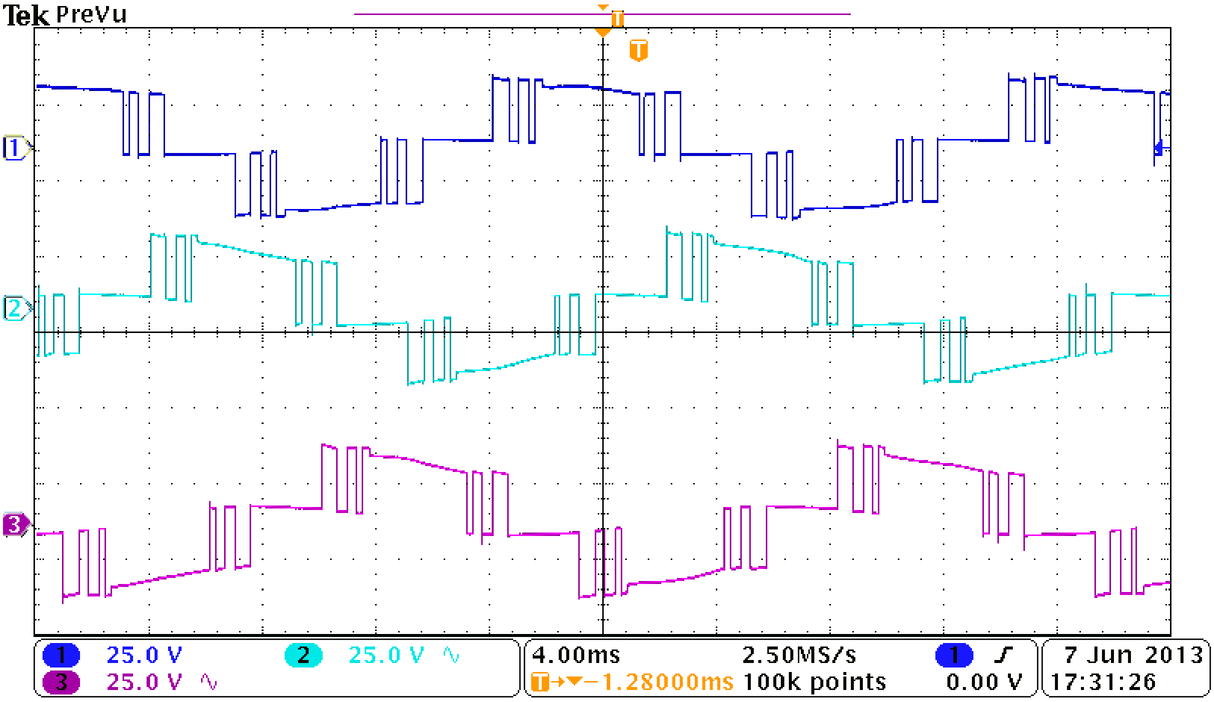
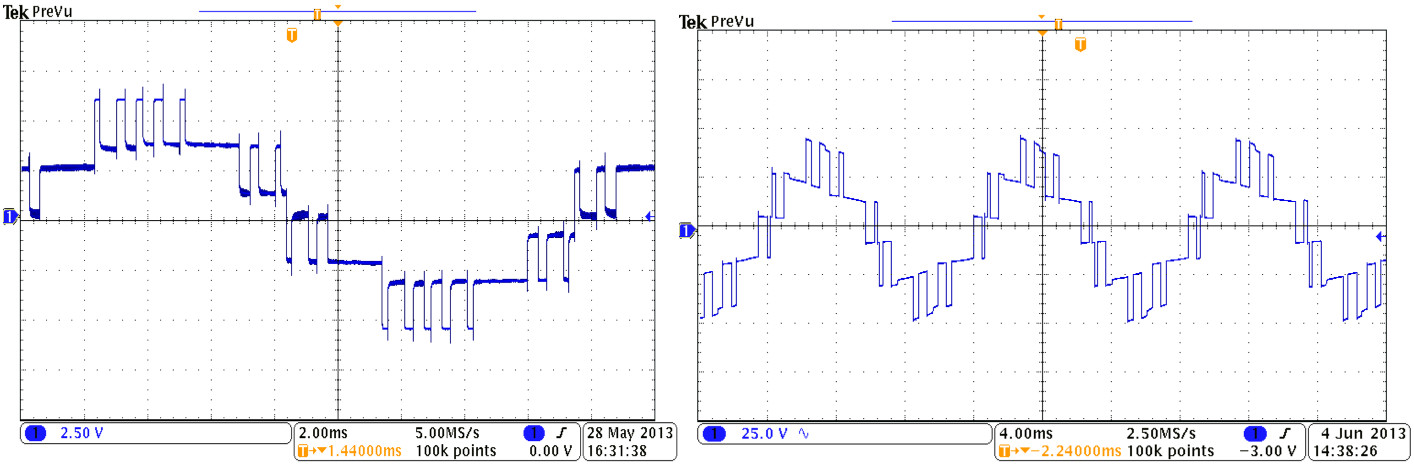


Figure 5-17: Three phase voltage waveform at the inverter terminal for m=1 and f=55 (N=5)

It can be noticed from figure 5-17 above that the waveforms for the three phase voltages are similar but with a time shift equivalent to the phase shift implemented on the software. That is, the same optimal patterns are used to produce switching angles for all the phases but the time the selected optimal angles are changed in to switching time depends on the two extreme angles of the respective phase.

5.3.4. 3-level inverter output Line to line voltage

One of the most interesting features of multilevel inverters is the reduction in harmonics in the line to line quantities. This can clearly be depicted in the different voltage levels in the line to line output voltage of the inverters. In programmed modulation also, this requirement should be met for the modulation strategy to satisfy the objective of reduced harmonics. Figure 5-18 below shows that the line to line voltage has a stair case waveform with multi levels, a phenomenon that does not exist in 2-level inverters.



(a) (b)

Figure 5-18: Line to line voltage at the inverter terminal: (a) on no load condition (b) When load connected

It is plainly visible in the above figure that there are 5 levels in the line to line voltage of a three level inverter output. This is in a perfect coherence with the theory, and hence the total harmonics seen by the motor windings is significantly reduced.

Comparing the two waveforms in figure 5.18 above reveals that there is a clear difference in the output voltage magnitudes between the case when the load is connected and when it is not. When the inverter is connected to a load, the voltage magnitude fluctuates which is most likely due to the imbalance between the two dc-link capacitors, a problem called dc-bus voltage imbalance. A brief discussion about this feature is done in section 5.3.5.

5.3.5. Current waveform

In the previous section, a brief analysis of the gate drive signals and output voltage waveform has been presented. It has also been observed that the produced voltage waveforms are quarter wave symmetric which is an important criterion for the current waveform to have a better sinusoidal nature. These produced voltage waveforms were applied to a six phase induction machine and current measurements were taken. As it has been mentioned in section 4.1.1, the six phase induction machine can be run as a three phase machine by supplying only one set of the three phase windings.

First, the relation between the applied voltage and the resulting motor phase current was investigated. Figure 5-19 below shows that the current in the motor windings is a sinusoidal wave having very little distortion. It is worth noting that the switching frequency of the switches in the inverter is below 300 Hz, which is much lower than the switching frequency that would have been needed for conventional PWM methods to produce a waveform with the same quality. Another observation from the figure is that the current lags the voltage waveform, in consistent with the inductive nature of the induction motor load.

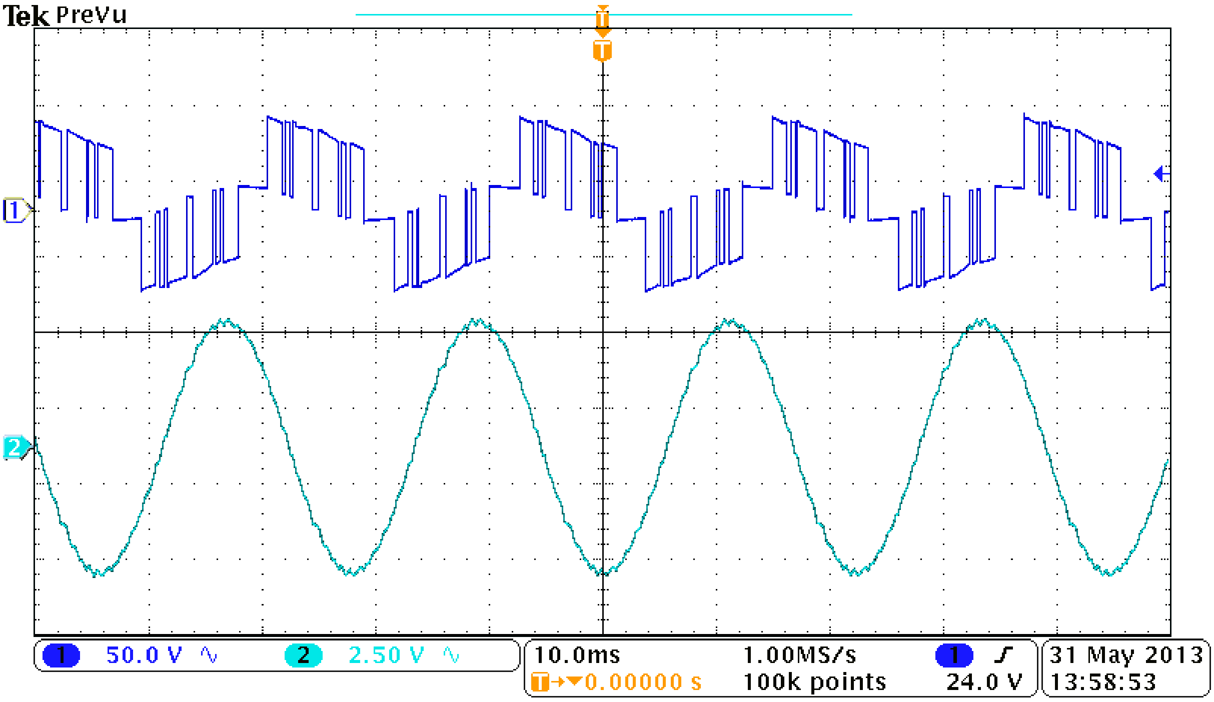


Figure 5-19: Phase voltage versus motor phase current for m=0.98, f=45Hz (N=6).

An equally important feature that should be observed from the waveforms in the figure is the unequal dc-bus voltage that exists across the two capacitors when a load is connected to the inverter terminal. This is the effect of the imbalanced current that flows through the capacitors in a given sampling period. This phenomenon is clearly visible when the load is connected to the neutral point where the two dc-bus capacitors get current of opposite direction. That is, for lower modulation indices, those switches that connect the load to the middle point conduct for longer time allowing charging of the two capacitors and discharging the other, and hence the distortion is also higher at these modulation indices. It appears therefore that for medium and lower modulation indices, the dc-bus voltage imbalance is a main problem for the operation of the converter. Figure 5-20 below better presents this feature.

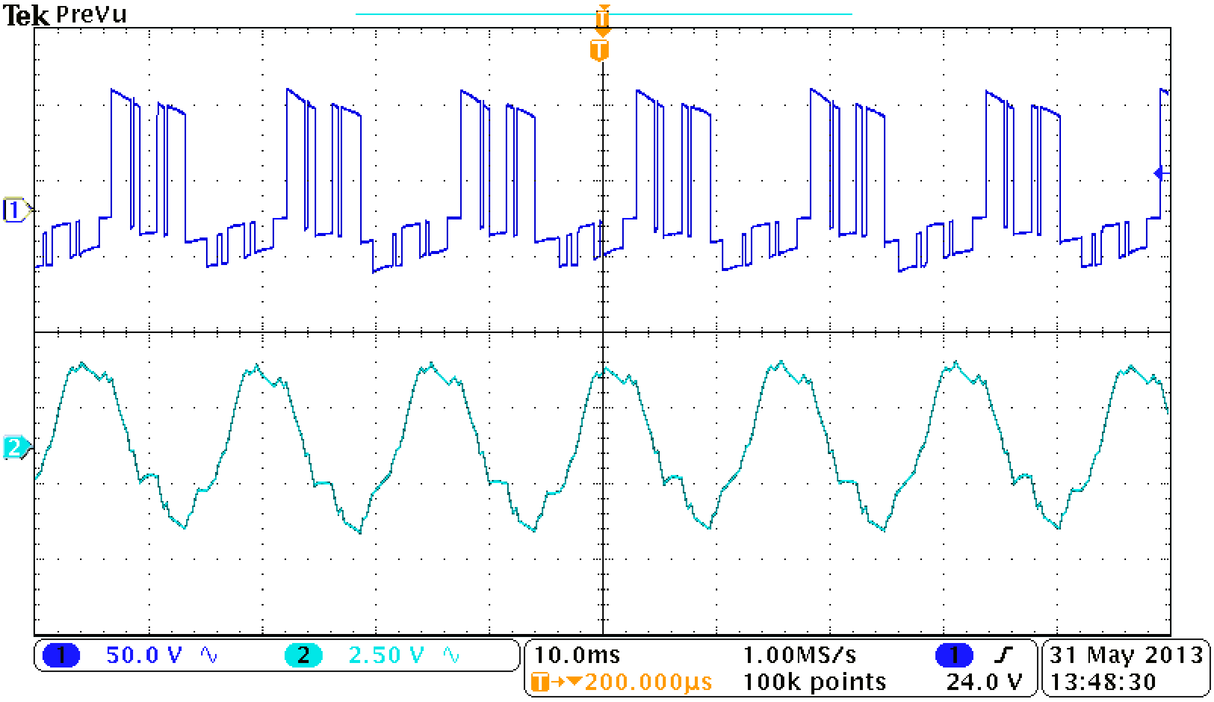


Figure 5-20: Effect of neutral point current on the dc-bus voltage for m=07,f=65Hz (N=4)

It can easily be noticed that the harmonic distortion of the current waveform in figure 5-20 is higher than that in figure 5-19. This is due to the larger dc-bus voltage imbalance in the case of figure 5-19 owing to the lower modulation index implemented in the latter.

As it was mentioned in section 2.4.2, the problem of dc-bus imbalance can be solved by implementing a dc-bus voltage balancing routine in the modulator software. This routine was not included in this thesis work due to the longer execution time it would take for the processor to handle if the routine were part of the program.

5.3.6. Total Harmonic Distortion (THD)

In order to quantitatively see the improvement offered by the proposed modulation scheme, THD of the inverter output voltage waveform was measured for different modulation indices and different fundamental frequency at different switching frequencies. A comparison of these values with the THD measurements from conventional modulation techniques shows that programmed modulation has the lowest THD value among all the current modulation technologies. Table 5-1 below illustrates this.

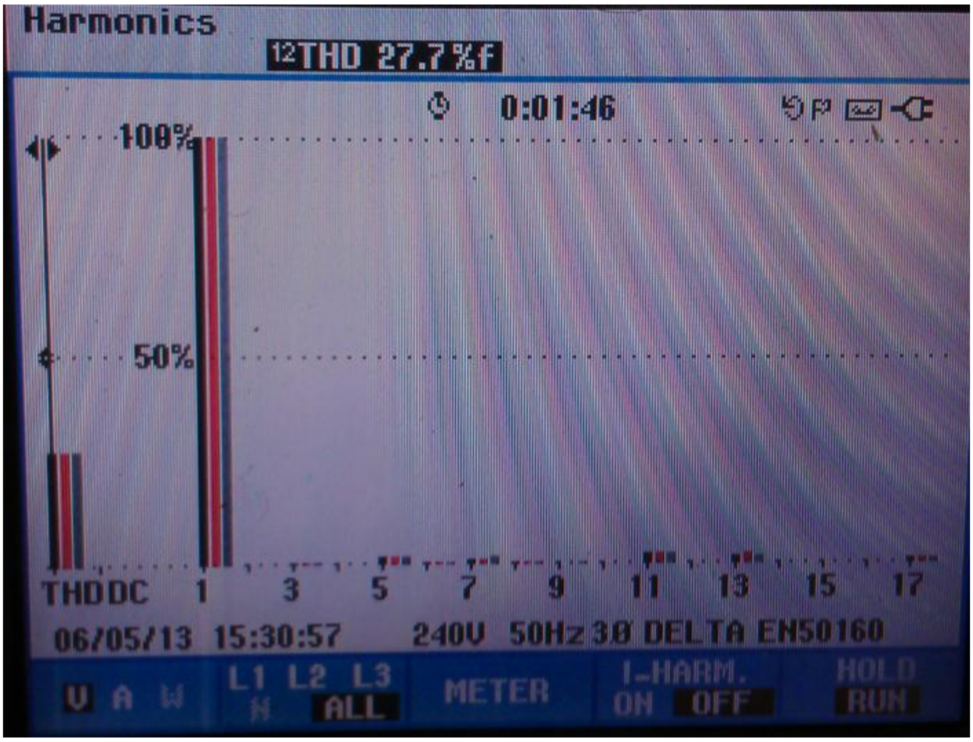


Figure 5-21: THD of the output voltage for m=1 and f=45Hz (N=6)

Modulation technique	THD
SPWM	53.88%
Trapezoidal	52.7%
SVPWM	52.7%
Programmed modulation	24.4%

Table 5-1: Comparison Result of %THD for output Line voltage of inverter for 150 Hz average switching frequency and modulation index of 1.

From the table above, it appears that it is possible to reduce the THD of an inverter output voltage to less than 25% by implementing programmed modulation technique. This is a significant improvement over the existing techniques that produce THD of more than 50% when implemented at the same switching frequency.

The THD content for a switching frequency of 300 Hz was also measured and the result confirms the theoretical hypothesis that the THD can be maintained below a certain value by employing programmed modulation technique.

Fundamental frequency	THD
45 Hz	27.6%
50 Hz	29.1%
55 Hz	29.3%

Table 5-2: THD of voltage waveform for different frequencies at 300 Hz average switching frequency and modulation index of 1.

Finally, the current through all the three phases was measured, and the result shows that the current waveform is symmetric. That is, the phase shift among the three phases is the same as expected with minor irregularities and the harmonic content of the current waveform is very low for all the three phases.

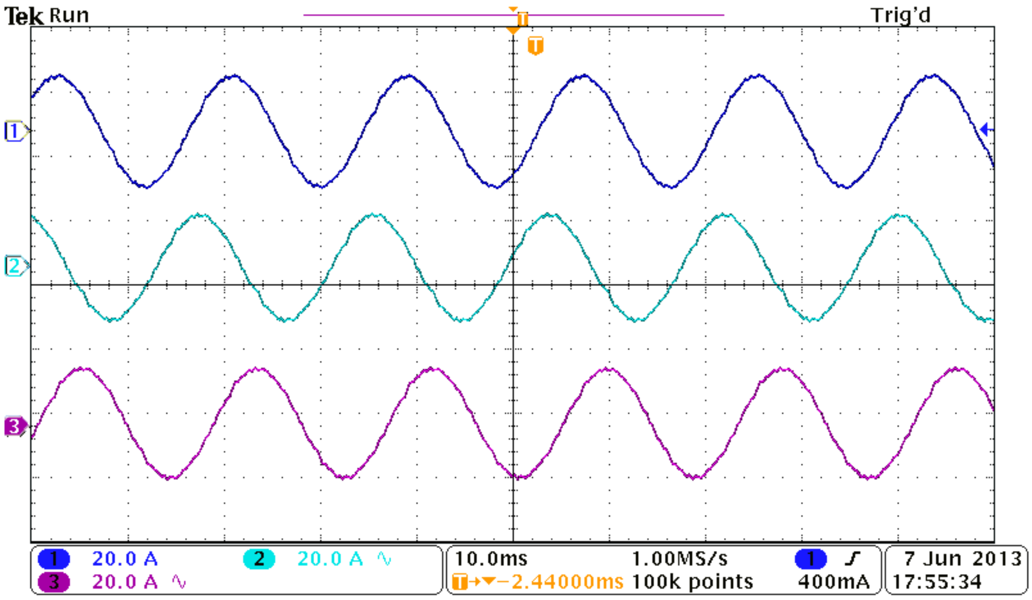


Figure 5-22: Three phase currents (curve 3: phase a, curve 1 : phase b, curve 2: phase c) in the induction motor for m=0.8, f=75 Hz

Generally, the results show that the proposed modulation scheme outpasses all the conventional modulation techniques in terms of its significantly reduced harmonic content.

However, the modulation technique has not yet been implemented fully because there are some loop holes during certain operations.

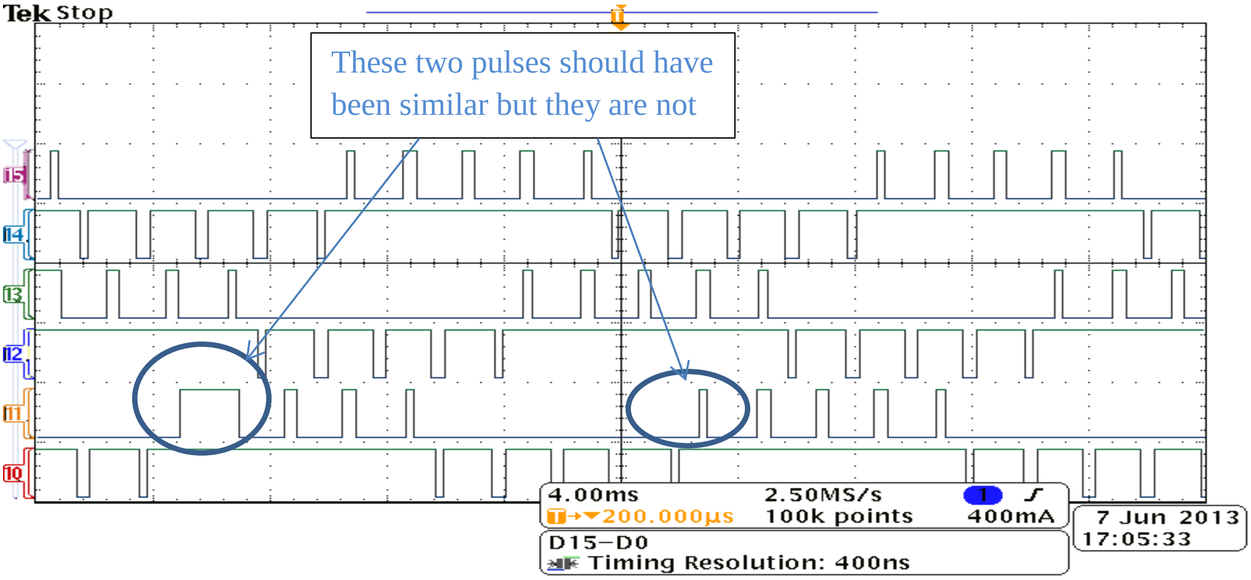

These two pulses should have been similar but they are not

Figure 5-23: Non-symmetric pulses

It can be seen from figure 5-23 above that the pulses pattern is quarter wave symmetric for two of the phases, while one pulse is disturbing the symmetry in the third phase. This problem was not noticed for the majority of the operation cases. The potential source of this discrepancy is the fact that the counter implemented on the hardware is reset at each sampling period which may lead certain pulses to be exempted in the interface between the software and hardware. The attempt to solve this problem was not successful because of the complex nature of the interface between the hardware and software parts. In addition, the problem was discovered very late in the thesis time frame, and as a result there was shortage of time to try a new modulator hardware.

A possible solution that was proposed but not implemented due to the above mentioned reasons was using a counter that is not reset. In this type of counter, the counter value is not zero at the start of each sampling time, and therefore there is low probability of pulses to be missed out.

More curves for different combination of modulation indices and number of pulses are given in appendix E.

6. Conclusion and further works

6.1. Conclusion

Increasing the output power of electrical motors is accomplished by supplying them with a higher voltage level. To solve the resulting problem of increased switching losses, which is proportional to the voltage magnitude, the switching frequency of the switches should be reduced. In this master's thesis, design and implementation of programmed modulation scheme was done. The work included study of related literature and simulation results from previous works done on similar topic, investigation of the current digital control technologies and hardware and software development of the system.

A theoretical study of programmed modulation was done first. Simulation results from previous work were also investigated as part of the study. The results show that, if synchronous modulation is implemented at lower switching frequencies, it effectively reduces the harmonic content of the current in the motor windings, because the optimal angles are selected in such a way that the harmonic level is below the maximum tolerable amount.

An investigation of available technologies to implement the modulation scheme was also done and a detail study was made on FPGA based implementation, particularly Xilinx FPGA. FPGA based systems are found to be better suited for this application because of their reprogrammable hardware which is needed to switch between synchronous modulation and asynchronous modulation at lower motor speed. FPGA system was also selected for its speed of operation as most of the processes take place at hardware level.

A programmed modulation system was designed and implemented on the FPGA board. The result shows that with fast sampling frequency, it is possible to produce optimal switching patterns using programmed modulation. Nevertheless, it can also be deduced that the proposed modulation scheme is memory intensive due to the large number of optimal angles to be stored. An alternative approach of using linear interpolation is used but this method does not produce optimal angles since the relationship between optimal angles and modulation index is not linear except at some intervals.

Measurements of the voltage and current waveforms produced by the proposed modulation system indicate that the proposed modulation scheme gives a very good quality output signal in terms of harmonic content. The total harmonic distortion of the output current is well below that of the conventional sub-oscillation modulation method. This is significantly visible even when the modulator runs at very low switching frequency.

An important observation from the modulation system is also that the dc-bus voltage is heavily affected by the neutral point current and as a result, the output voltage exhibits unequal voltage magnitudes in the positive and negative dc-bus terminals. This feature is well prominent in medium and lower modulation indices due to the longer time the load is connected to the neutral point of the inverter.

6.2. Further works

The thesis focused on the study and design of programmed modulation on FPGA, and hence the theoretical and experimental findings of the thesis are interesting but limited. Therefore, the work can be continued with the following focuses:

- Further studies on the optimal angles for more number of pulses
- Including common mode voltage elimination in the objective function
- Integrating the dc-bus voltage balancing algorithm
- Designing feedback loop for the control system
- Including the trajectory controller to avoid harmonics due to pulse pattern shift during transient operations

References

[1] Ned Mohan Tore M. Undeland, William P. Robbins, "*Power Electronics: Converters, Applications, and Design*", 3rd ed. Wiley, 2003

[2] M.H. Bierhoff, F.W. Fuchs, "*Semiconductor Losses in Voltage Source and Current Source IGBT Converters Based on Analytical Derivation*", 35th Annual IEEE Power Electronics Specialists Conference, pp. 2836-2837, 2004.

[3] Hasmukh S. Patel and Richard G. Hoft, "*Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters, Part I-Harmonic Elimination*", IEEE Transactions on Industry applications, Vol. IA-9, No. 3, pp. 310-311, May/June 1973

[4] P. Enjeti and W. Shireen, "*An Advance Programmed PWM modulator for Inverters Which Simultaneously Eliminates Harmonics and Rejects DC link Voltage Ripple*", Power Electronics Laboratory, Department of Electrical Engineering, Texas A and M University, pp. 681-683, 1990

[5] Giuseppe S. Buja, "*Optimum Output Waveforms in PWM Inverters*" IEEE Transactions on Industry applications, Vol. IA-16, No. 6. November/December 1980, pp. 830

[6] Joachim Holtz, "*The Trajectory Tracking Approach-A New Method for Minimum Distortion PWM in Dynamic High-Power Drives*", IEEE Transactions on Industry applications, Vol. 30, No. 4, July / August 1994, pp. 331-333

[7] Roger Enes, "*Modelling and Control of High Performance Medium Voltage drives*", Norwegian University of Science and Technology, Trondheim, Norway, 2012

[8] Murphy J.D., Egan M.C, "*A Comparison of PWM Strategies for Inverter-Fed Induction Motors*", IEEE Transactions on Industry Applications, Vol. IA-19, No. 3 , May/June 1983, pp. 363-369

[9] Holmes D.G., Lipo T.A, "*Pulse Width Modulation for Power Converters-Principles and practice*", John Wiley & Sons, IEEE Press, 2003,ISBN 0-471-20814-0

[10] Daniel Martinsson and Magnus Lind, "*DSP Controller for Power Electronic Converter Applications*", Dept. of Industrial Electrical Engineering and Automation, Lund, Sweden.

[11] Berkeley Design Technology Inc., "*Choosing a DSP processor*", www.bdti.com.(accessed 10 December, 2012)

[12] http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htmwww.bdti.com.(accessed 19 October, 2012)

[13] "*EDK Concepts, Tools, and Techniques A Hands-On Guide to Effective Embedded System Design EDK 13.1*", UG683, April 13, 2011

[14] Nebrom Berihu Araya, "*Modelling and control of Six-Phase induction Motor Drive*", Norwegian University of Science and Technology, Trondheim, Norway, 2012.

[15] K. Ljøkelsøy, "*Library of IP modules, building blocks for FPGA based converter control system*" Project Memo, AN 10.12.66, SINTEF Energy Research, Trondheim, Norway, Jun. 2010

[16] Joachim Holtz and Bernd Beyer, "*Fast Current Trajectory Tracking Control Based on Synchronous Optimal Pulsewidth Modulation*", IEEE Transactions on industry applications, Vol. 31, No. 5, pp. 1-6, September/October 1995

[17] Roy Nilsen, "*Modulation Methods for 3 Level Inverter*", Technical Report, Wärtsilä Norway AS, 2010, Trondheim, Norway.

[18] Giuseppe S. Buja and Giovanni B Indri, "*Optimal Pulsewidth Modulation for Feeding AC Motors*", IEEE Transaction on Industrial applications, Vol. A1-13, No. 1, pp. 38-44, January/February 1977

[19] Joachim Holtz, Senior Member, IEEE, and Bernd Beyer, "*Optimal Synchronous Pulsewidth Modulation with a Trajectory Tracking Scheme for High Dynamic Performance*", University of Wuppertal, Wuppertal, Germany, pp. 147-154.

[20] Surin Khomfoi and Leon M. Tolbert, "*Multilevel Power Converters*", The University of Tennessee

[21] Leopoldo Garcia Franquelo, *Fellow, IEEE*, Javier Nápoles, Ramón C. Portillo Guisado, Student Member, IEEE, José Ignacio León, Member, IEEE, and Miguel A. Aguirre, Member, IEEE, "*A Flexible Selective Harmonic Mitigation Technique to Meet Grid Codes in Three-Level PWM Converters*".

[22] Jose Rodriguez, Senior Member, IEEE Steffen Bernet, Member, IEEE, Peter K. Steimer, Fellow, IEEE, and Ignacio E. Lizama, "*A Survey on Neutral-Point-Clamped Inverters*", IEEE Transactions on Industrial Electronics, Vol. 57, No. 7, pp. 2219-2223, July 2010.

[23] Josep Pou, Member, IEEE, Jordi Zaragoza, Pedro Rodríguez, Member, IEEE, Salvador Ceballos, Vicenç M. Sala, Rolando P. Burgos, Member, IEEE, and Dushan Boroyevich, Fellow, IEEE, "*Fast-Processing Modulation Strategy for the Neutral-*

Point-Clamped Converter with Total Elimination of Low-Frequency Voltage Oscillations in the Neutral Point", IEEE Transactions on industrial electronics, Vol. 54, No. 4, pp. 2288-2289, AUGUST 2007

[24] Wojciech Kołomyjski M.Sc., "*Modulation Strategies for Three-level PWM Converter-fed Induction Machine Drives*", Warsaw University of Technology, Faculty of Electrical Engineering Ph.D. thesis, Warsaw, 2009

[25] H. Kolstad and K. Ljøkelsøy, "*20 kW IGBT omformer. Beskrivelse*" Arbeidsnotat: an 01.12.12, SINTEF Energiforskning AS, Trondheim, Norway, Sept. 2002.

[26] Sheron Figarado, "*Multilevel inverter topologies with reduced power circuit complexity for medium voltage high power induction motor drives by cascading conventional two-level and three-level inverters*", Centre for Electronics Design and Technology, Bangalore, India

[27] S. Floten and T. S. Haug, "*Modulation Methods for Neutral-Point-Clamped Three-Level Inverter*" N. U. o. S. a. Technology, Ed., ed. Trondhiem, 2010.

[28] Mamta Maharjan, "*Synchronous Optimal Modulator for Multi-phase Machines: Implementation of three-level modulators In Field Programmable Gate Arrays*", NTNU, Trondheim, Norway, 2013

Appendices

A: Six-Phase Induction Machine data

i. Nameplate Data

Parameter	Value
U_N(Nominal Line to line voltage) in Vrms	400V
I_N(Nominal line current) in Arms	11.8A
f_N(Nominal frequency) in Hz	75Hz
p(Number of pole pairs)	2
n_N(Nominal speed) in mechanical in rpm	2235rpm
M_n(Nominal output torque) in Nm	50Nm
P_N(Nominal Power output) in kW	11.7Kw
$\cos\varphi_N$(Nominal power factor)	0.77
n_{max}(mechanical rpm)	5000

Table: A-1: Induction machine nameplate data

ii. Equivalent Circuit parameters:

Parameter	Value [pu]
r_S	0.031
r_R	0.0068
x_S	2.086
x_σ	0.2175
x_H	1.8685
σ_r	0.0566

Table A-2: Induction machine equivalent circuit per unit quantities

B: Sample C++ code of the modulation software

```
/*
 * initialize.cpp//This routine contains the optimal quarter wave angles for N=3,4,5 and 6 and
in addition it also initializes the full wave optimal angles.
 *
 *  Created on: 11. mars 2013
 *      Author: admin
 */
float
optimal_angles[4][32][40]={{{67.2967,68.6452,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{66.6115,69.3395,88.4785,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{66.0052,69.9800,87.8339,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{65.2751,70.7820,87.0037,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{64.7006,71.4309,86.3370,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{63.9851,72.2475,85.5058,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{63.4078,72.9001,84.8437,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{62.6696,73.7087,84.0221,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{62.0548,74.3359,83.3658,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{61.2608,75.0896,82.5354,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.4938,60.1442,74.7988,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.4486,60.2011,73.4829,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.5186,60.3013,72.1042,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.5938,60.3907,70.7004,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.6366,60.4934,69.3236,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.5197,60.5408,68.0157,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{50.1160,60.4220,66.7778,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{46.5091,51.1967,59.7419,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{44.9165,50.3329,58.7188,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{42.0962,47.9145,57.2599,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{39.8719,46.4167,56.0754,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{35.6968,42.6934,53.8100,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{28.2477,36.8607,51.2411,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{27.0124,37.3835,50.0622,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{23.9996,38.8429,47.6017,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{22.2809,39.9078,46.3721,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{20.8054,40.8623,45.4816,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{18.7960,42.5366,44.8391,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0},
{11.4887,15.8474,19.9125,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0}},
{{54.9375,56.0631,79.8309,81.1717,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0},
{54.1858,56.8316,78.6475,82.3686,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{53.7114,57.3219,77.9071,83.1044,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{53.3399,57.7055,77.2975,83.6906,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{52.8850,58.1709,76.5039,84.4134,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{52.5652,58.4664,75.6949,84.9243,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{52.1963,58.7493,74.2206,85.2180,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{51.9594,58.9869,73.4898,85.9278,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{51.7042,59.2899,72.1510,86.4392,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{51.5111,59.4985,71.6608,86.9934,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{51.3465,59.7280,70.9319,87.3266,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{51.2727,59.8062,70.1940,87.8561,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{17.2592,47.4363,59.9085,70.5935,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{16.7397,48.2304,59.7049,71.0526,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{14.1174,51.2767,59.4556,69.0445,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{13.5702,51.8226,59.3484,69.7915,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{13.2378,52.3299,59.2397,70.6196,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{13.6747,52.4529,59.3878,72.1830,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{15.5253,50.1744,58.3291,74.9863,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{17.0367,48.6258,57.5715,77.3627,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{18.5840,47.0724,56.6849,79.6387,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{18.2769,47.5748,56.0851,80.4854,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{19.8476,45.6447,54.3289,83.2368,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{19.5628,46.1643,53.6850,84.0516,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{19.3624,46.0333,52.3096,85.2361,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{18.8825,46.3695,51.5079,86.0075,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{18.1675,46.8396,50.4302,86.8532,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{17.4261,47.3157,49.6192,87.4573,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{16.4489,47.8849,48.8849,88.4466,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{11.0859,14.6499,17.4587,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0},
{11.2622,14.8441,15.9462,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0}},
{{8.9964,9.9964,50.8048,52.1914,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{8.5792,10.2874,49.9466,52.8693,88.5141,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{8.0408,10.6497,49.0380,53.3127,87.9027,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{7.6668,11.0898,48.1976,54.0443,87.0059,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{7.3124,11.6701,47.2817,54.5406,86.4465,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{7.2582,12.6072,46.0120,54.9538,85.6592,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{7.0204,13.1672,45.3882,55.6867,85.0608,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.8347,13.5555,44.6553,56.4916,84.1051,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.6481,13.9135,44.0857,57.1441,83.3519,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.4205,14.3552,43.3775,57.9616,82.4079,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.2948,14.6092,42.9566,58.4485,81.8316,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.1805,14.8521,42.5415,58.9318,81.2440,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{6.0797,15.0824,42.1344,59.4118,80.6432,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.9940,15.2992,41.7376,59.8892,80.0274,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.9250,15.5019,41.3533,60.3656,79.3963,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.8745,15.6888,40.9876,60.8540,78.7564,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.8447,15.8587,40.6419,61.3624,78.1136,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.8388,16.0083,40.3032,61.8810,77.4667,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.8581,16.1381,39.9605,62.4041,76.8178,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{5.0219,12.0846,42.9948,62.1294,72.1343,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0},
{37.2017,40.3265,48.7427,53.6089,59.1815,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{31.7045,35.0147,45.5192,50.9783,57.7678,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{28.5189,33.4390,44.6964,51.0992,57.2802,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{27.1592,34.1127,43.8915,50.6698,56.0695,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{26.0642,34.8198,43.4058,50.2300,54.9615,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{24.7278,35.7450,42.5851,49.5434,53.5969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{23.5556,36.6401,42.2409,49.0532,52.4307,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{14.0422,26.2310,29.8671,77.2392,80.7789,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{14.1100,24.3837,27.1877,79.7778,81.4983,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{10.6058,13.9047,17.1567,84.6102,85.6102,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0},
{10.8874,14.0940,15.5335,84.6957,85.6957,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0}},
{{8.9904,9.9904,50.7981,52.1888,84.9938,85.9938,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0},
{8.7178,10.1579,50.1480,52.7559,84.5461,86.3262,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{8.3355,10.4003,49.5144,53.2809,84.0334,86.6131,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{7.7025,10.7953,48.6290,53.9939,83.2379,86.6686,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{7.3684,11.0944,48.0549,54.6085,82.6005,86.8076,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{6.9085,11.5026,47.3131,55.4291,81.6524,86.7604,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{6.6742,11.7276,46.8343,55.9752,80.9290,86.9709,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{6.3971,12.0006,46.2448,56.7094,80.1049,87.2818,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{6.1612,12.2197,45.7703,57.3191,79.4666,87.5293,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.9002,12.4714,45.1923,58.0763,78.7216,87.9148,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.8123,12.5799,44.8671,58.4898,78.3029,88.2404,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.6298,12.7523,44.5074,58.9904,77.8563,88.4165,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.5561,12.8543,44.1923,59.3840,77.4392,88.7712,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.4785,12.9688,43.8475,59.8455,76.9774,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.3716,13.1547,43.4617,60.3222,76.3418,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.3030,13.3220,43.0787,60.7665,75.6658,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{14.1186,50.1155,53.1827,54.9715,60.0641,70.4579,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{13.7539,50.3573,53.1237,55.1784,59.9418,71.2555,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{15.9693,48.2313,53.6572,56.8658,60.6117,74.5417,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{17.0395,47.3148,53.6448,56.8736,60.1935,76.5581,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{20.2922,43.1603,51.2951,56.4329,60.4698,80.7256,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{22.0311,40.5747,47.9003,52.6869,58.4745,83.2014,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{21.7639,40.8666,47.5944,53.0514,58.0784,83.8616,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{21.4265,41.1829,47.2982,53.4349,57.5548,84.8159,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{21.1275,41.4322,47.0689,53.6993,57.1305,85.6074,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{20.7142,41.8055,46.7127,54.0649,56.5997,86.4641,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{19.6805,42.8537,46.5124,53.3694,55.4952,86.7007,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{18.6224,43.5667,46.2025,53.4999,54.7851,87.2043,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{10.5056,16.4393,20.8657,71.9617,74.3517,87.7667,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{11.1391,16.1948,20.6868,72.8924,73.8924,88.8123,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0},
{8.7382,10.5102,11.5448,78.8937,79.8937,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0},
{5.6336,6.6336,7.7389,78.2709,79.2709,89.0000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0}}};

void Initialize(void)//this function calculates the remaining 3*n optimal angles from the n
angles and it is run at the start of the program
{

        for (int pulse_number_index = 0; pulse_number_index<4; pulse_number_index++) //
a loop that changes the quarter wave symmetric angles in to the full angles up tp 360 degrees
                        {

                        for(int
modulation_index_index=0;modulation_index_index<=32;modulation_index_index++)
                        {

                        for (int postion_of_angle_in_the_pattern = 0;
postion_of_angle_in_the_pattern <= 39; postion_of_angle_in_the_pattern++)
                        {


if(postion_of_angle_in_the_pattern<=9)

optimal_angles[pulse_number_index][modulation_index_index][postion_of_angle_in_the_pa
ttern+10]=180-optimal_angles[pulse_number_index][modulation_index_index][9-
postion_of_angle_in_the_pattern];
                                else

if(postion_of_angle_in_the_pattern>=20)

optimal_angles[pulse_number_index][modulation_index_index][postion_of_angle_in_the_pa
ttern]=180+optimal_angles[pulse_number_index][modulation_index_index][postion_of_ang
le_in_the_pattern-20];
                                else
                                goto nothing;//this means that the
already occupied angles are left there



                                nothing:
                                {
                                }



                        }

                        }
```

```
                                   }
}
```

D: Sample VHDL code of the modulator hardware design

```
-- Filename:          som_phase_ip.vhd
-- Version:           1.00.a
-- Description:       Top level design, instantiates library components and user logic.
-- Date:              Fri Apr 12 13:00:53 2013 (by Create and Import Peripheral Wizard)
-- VHDL Standard:     VHDL'93
------------------------------------------------------------------------------
-- Naming Conventions:
--   active low signals:                    "*_n"
--   clock signals:                         "clk", "clk_div#", "clk_#x"
--   reset signals:                         "rst", "rst_n"
--   generics:                              "C_*"
--   user defined types:                    "*_TYPE"
--   state machine next state:              "*_ns"
--   state machine current state:           "*_cs"
--   combinatorial signals:                 "*_com"
--   pipelined or register delay signals:   "*_d#"
--   counter signals:                       "*cnt*"
--   clock enable signals:                  "*_ce"
--   internal version of output port:       "*_i"
--   device pins:                           "*_pin"
--   ports:                                 "- Names begin with Uppercase"
--   processes:                             "*_PROCESS"
--   component instantiations:              "<ENTITY_>I_<#|FUNC>"
------------------------------------------------------------------------------
```

```
library ieee;

use ieee.std_logic_1164.all;

use ieee.std_logic_arith.all;

use ieee.std_logic_unsigned.all;


library proc_common_v3_00_a;

use proc_common_v3_00_a.proc_common_pkg.all;

use proc_common_v3_00_a.ipif_pkg.all;


library plbv46_slave_single_v1_01_a;

use plbv46_slave_single_v1_01_a.plbv46_slave_single;


library som_phase_ip_v1_00_a;

use som_phase_ip_v1_00_a.user_logic;


------------------------------------------------------------------------------

-- Entity section

------------------------------------------------------------------------------

-- Definition of Generics:

--   C_BASEADDR                   -- PLBv46 slave: base address

--   C_HIGHADDR                   -- PLBv46 slave: high address

--   C_SPLB_AWIDTH                -- PLBv46 slave: address bus width

--   C_SPLB_DWIDTH                -- PLBv46 slave: data bus width

--   C_SPLB_NUM_MASTER’SS         -- PLBv46 slave: Number of master’ss

--   C_SPLB_MID_WIDTH             -- PLBv46 slave: master’s ID bus width

--   C_SPLB_NATIVE_DWIDTH         -- PLBv46 slave: internal native data bus width

--   C_SPLB_P2P                   -- PLBv46 slave: point to point interconnect scheme
--   C_SPLB_SUPPORT_BURSTS        -- PLBv46 slave: support bursts
--   C_SPLB_SMALLEST_MASTER'S        -- PLBv46 slave: width of the smallest master's
--   C_SPLB_CLK_PERIOD_PS          -- PLBv46 slave: bus clock in picoseconds
--   C_INCLUDE_DPHASE_TIMER        -- PLBv46 slave: Data Phase Timer configuration; 0 = exclude timer, 1 = include timer
--   C_FAMILY                     -- Xilinx FPGA family
--
-- Definition of Ports:
--   SPLB_Clk                    -- PLB main bus clock
--   SPLB_Rst                    -- PLB main bus reset
--   PLB_ABus                     -- PLB address bus
--   PLB_UABus                    -- PLB upper address bus
--   PLB_PAValid                  -- PLB primary address valid indicator
--   PLB_SAValid                  -- PLB secondary address valid indicator
--   PLB_rdPrim                   -- PLB secondary to primary read request indicator
--   PLB_wrPrim                   -- PLB secondary to primary write request indicator
--   PLB_master'sID                -- PLB current master's identifier
--   PLB_abort                   -- PLB abort request indicator
--   PLB_busLock                  -- PLB bus lock
--   PLB_RNW                      -- PLB read/not write
--   PLB_BE                      -- PLB byte enables
--   PLB_MSize                    -- PLB master's data bus size
--   PLB_size                    -- PLB transfer size
--   PLB_type                    -- PLB transfer type
--   PLB_lockErr                  -- PLB lock error indicator
--   PLB_wrDBus                    -- PLB write data bus
--   PLB_wrBurst                  -- PLB burst write transfer indicator
--   PLB_rdBurst                   -- PLB burst read transfer indicator
```

E: More waveforms

Gate signals

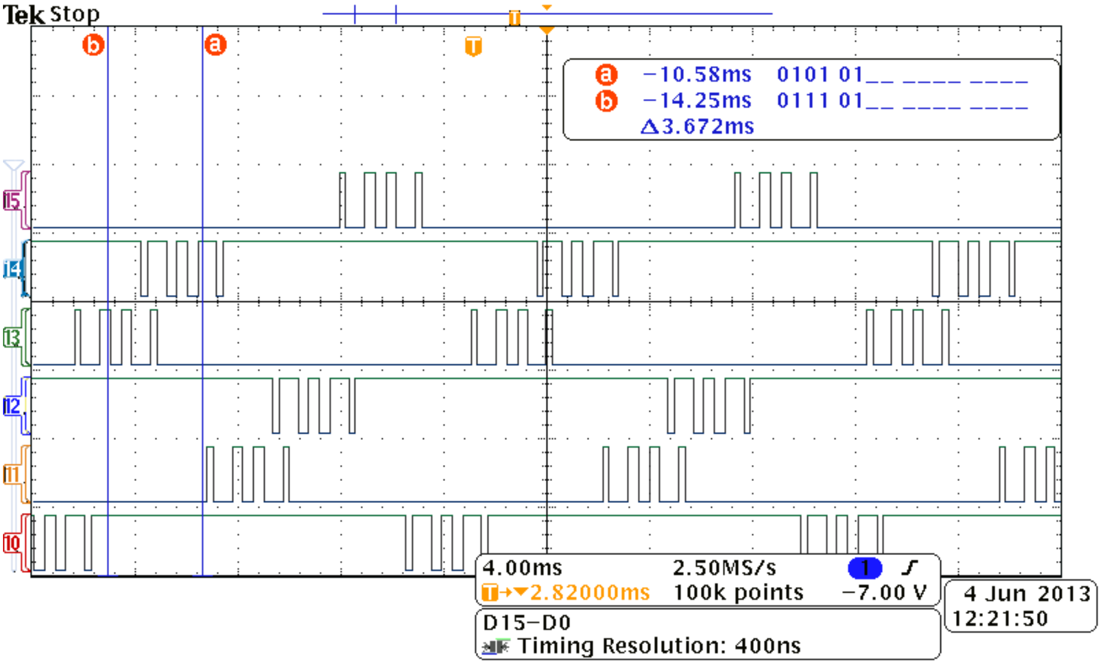


Figure 0-1: Gate signals for m=0.3, N=4

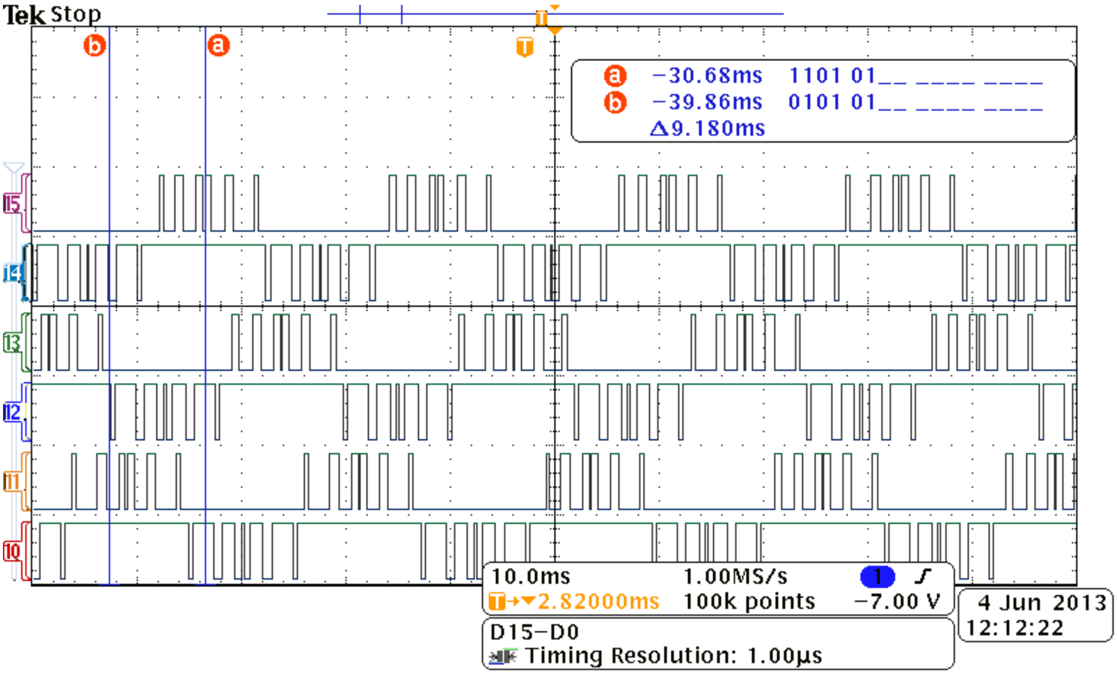

Tek Stop
a −30.68ms 1101 01__ ____ ____
b −39.86ms 0101 01__ ____ ____
Δ9.180ms
D5
D4
D3
D2
D1
D0
10.0ms 1.00MS/s 1 −7.00 V
T→▼2.82000ms 100k points
4 Jun 2013
12:12:22
D15−D0
Timing Resolution: 1.00µs

Figure 0-2: Gate signals for m=0.5, N=6

Line voltages

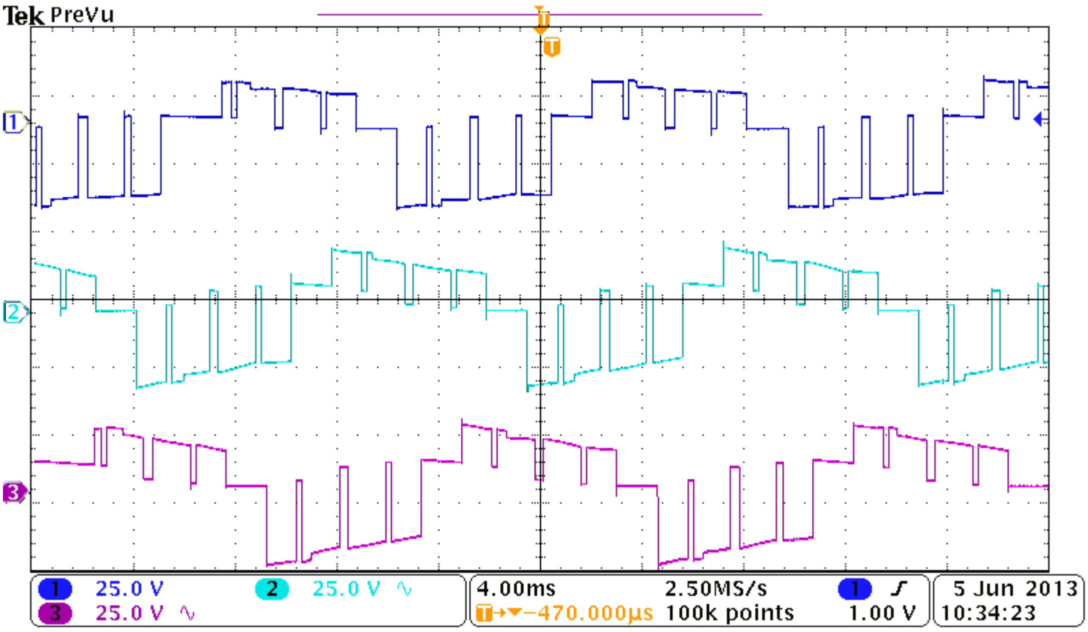


Figure 0-3: 3-phase voltage for m=1 and N=4

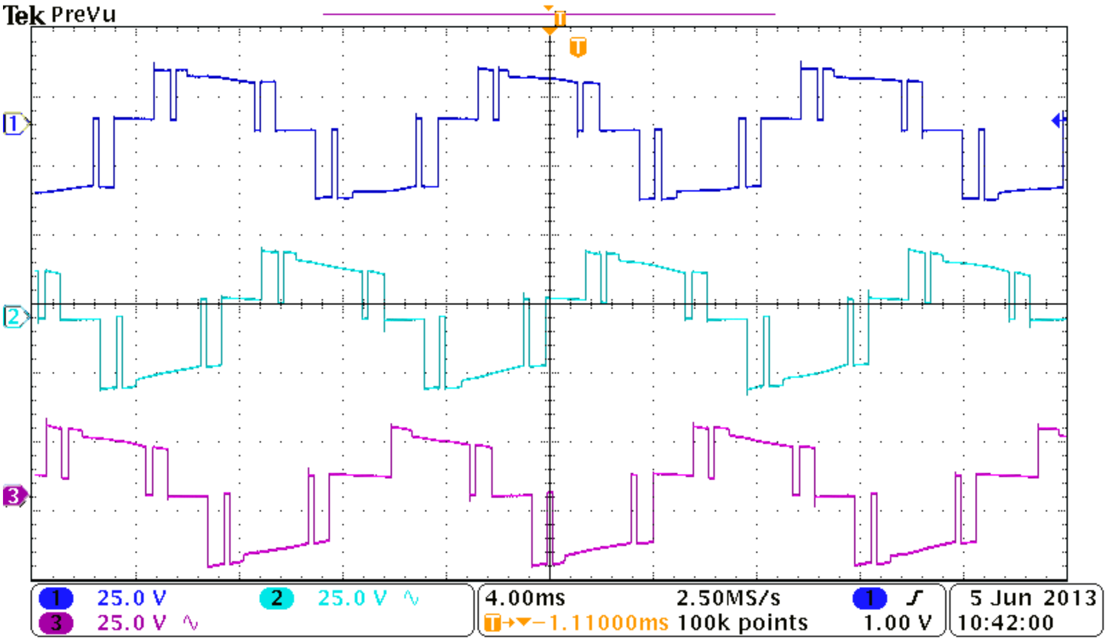


Figure 0-4: 3-phase voltage for m=1 and N=3

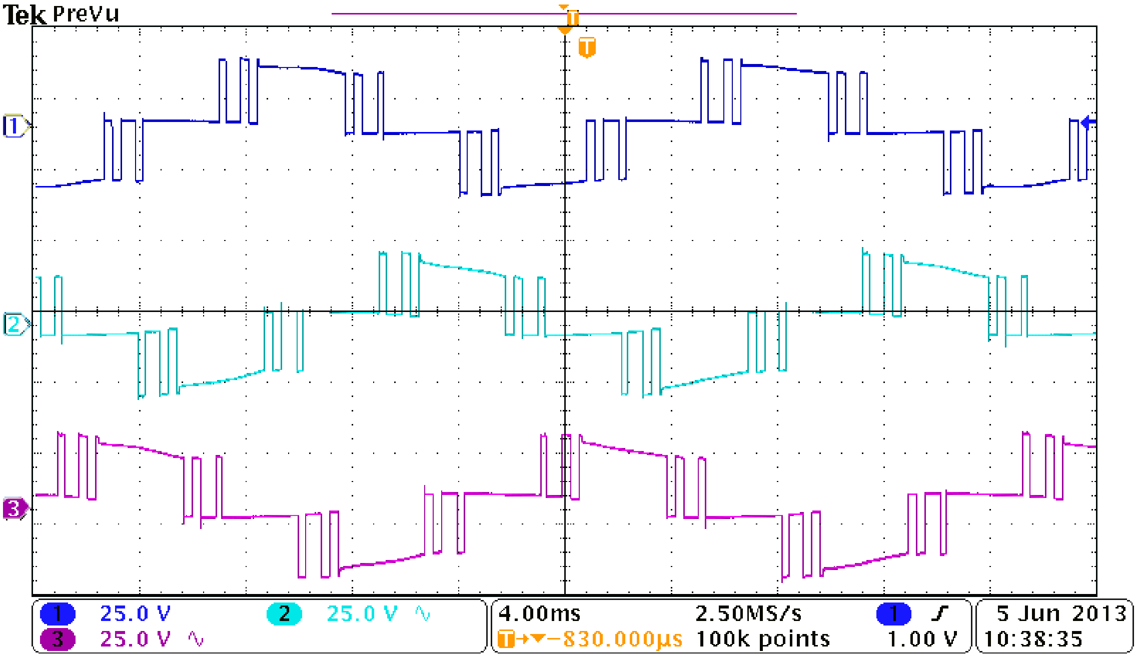


Figure 0-5: 3-phase voltage for m=0.87 and N=5

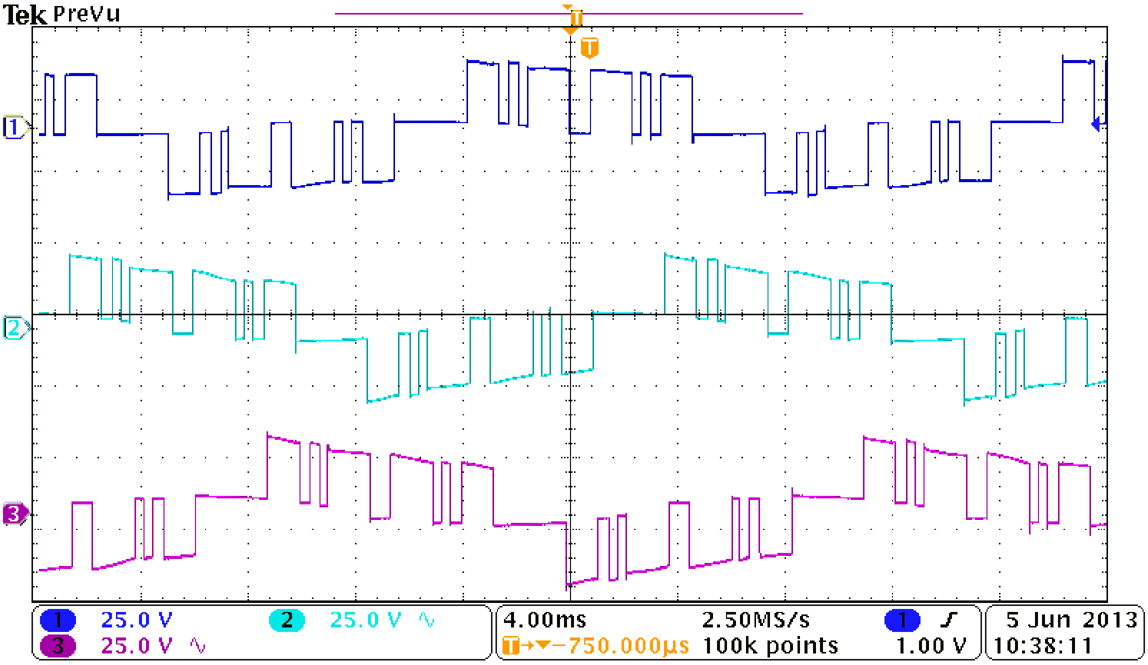


Figure 0-6: 3-phase voltage for m=0.87 and N=6

Line to line voltages

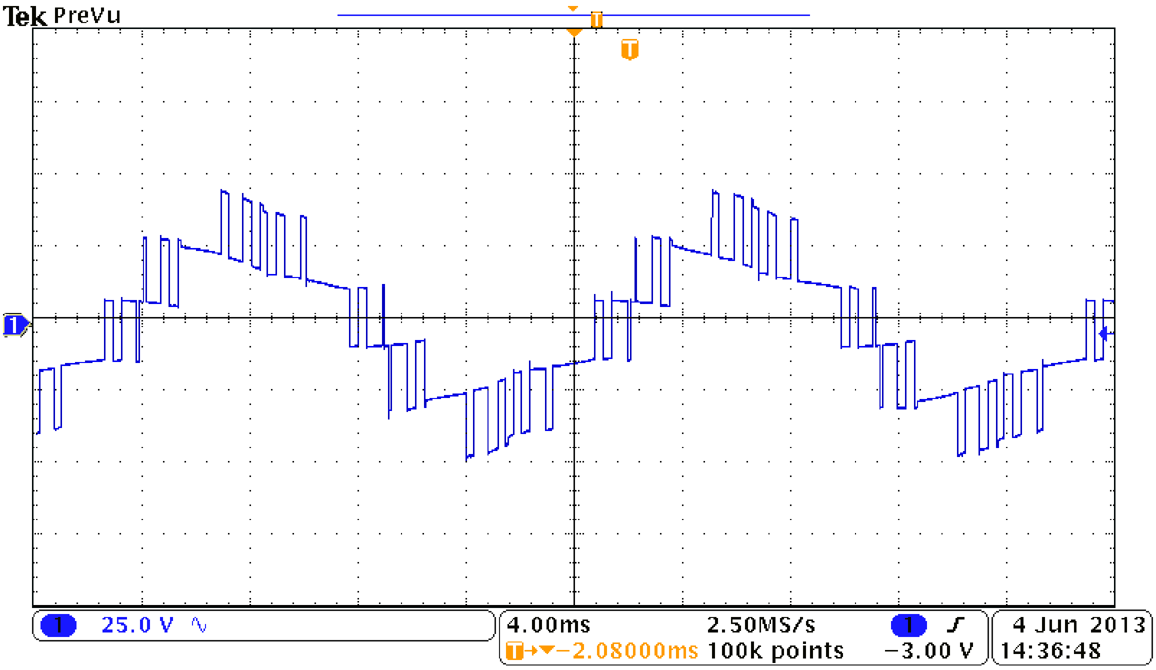


Figure 0-7: Line to line voltage for N=5, m=1

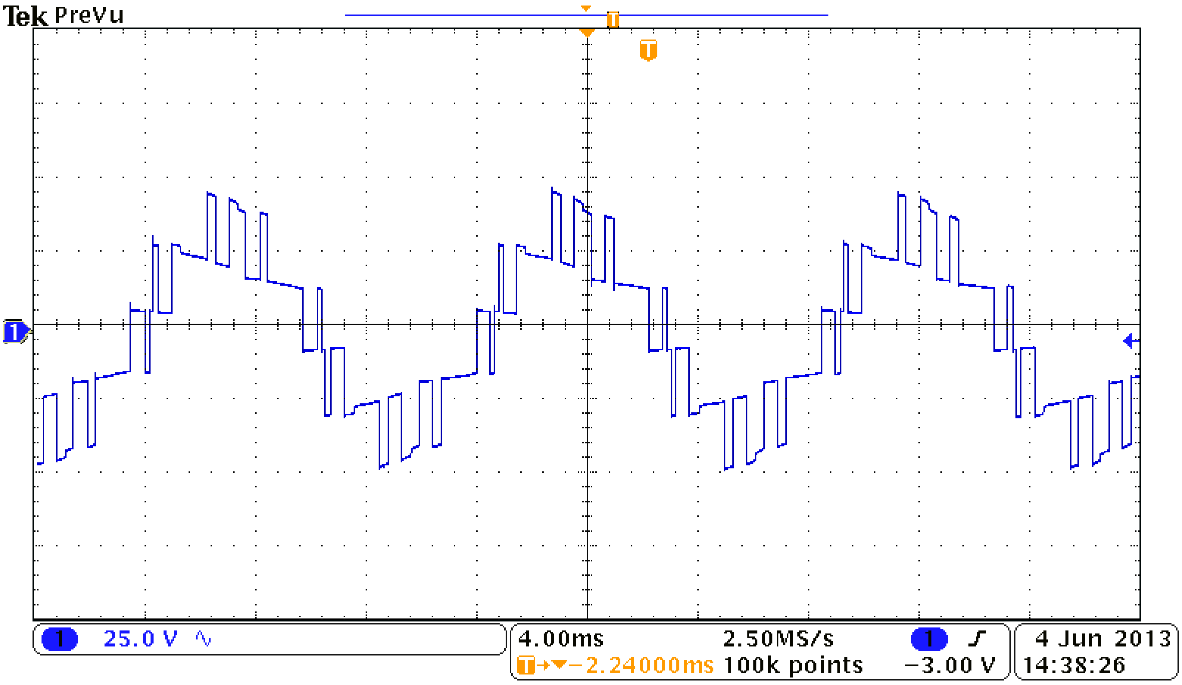


Figure 0-8: Line to line voltage for N=3, m=1

Current waveform

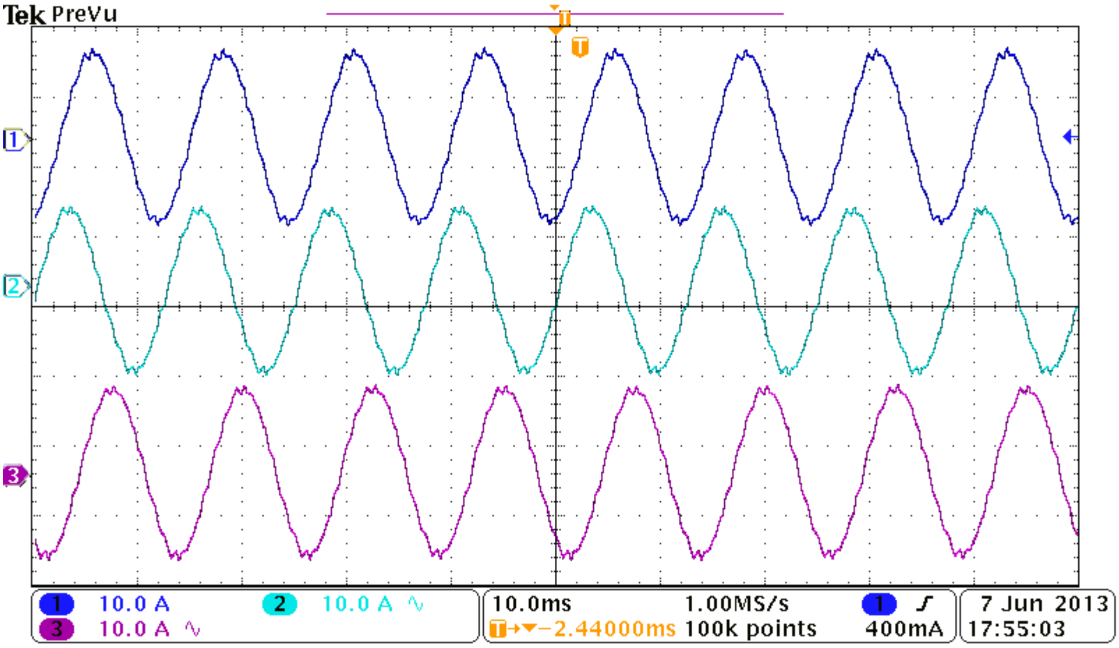


Figure 0-9: Three phase currents for m=0.87 and f=80Hz

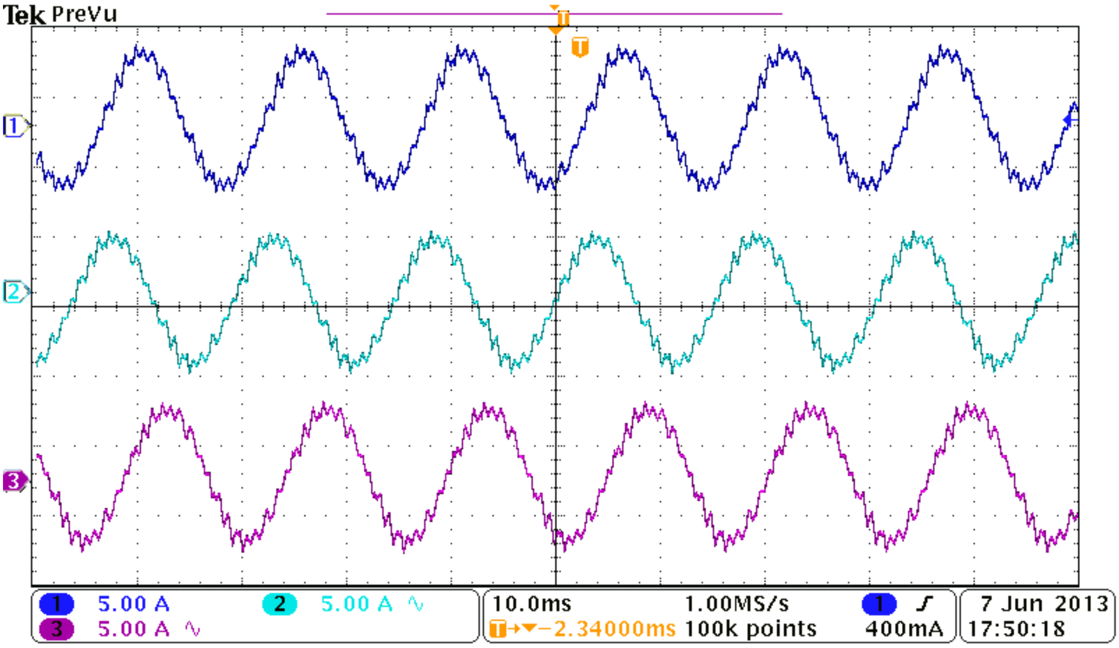


Figure 0-10: Three phase currents for m=0.3 and f=55Hz