NTNU – Trondheim
Norwegian University of
Science and Technology

Control of Medium Voltage Multi-phase machines

Implementation of Minimum loss
Synchronous Optimal Modulation with
Neutral Point Balancing

Krishna Neupane

Master of Science in Electric Power Engineering
Submission date: June 2014
Supervisor: Tore Marvin Undeland, ELKRAFT
Co-supervisor: Roy Nilsen, Wartsila

Norwegian University of Science and Technology
Department of Electric Power Engineering

Abstract

The total loss of an inverter can be reduced by reducing switching losses and harmonics losses. However, the harmonics increases with the reduction of switching frequency and special technique is required to get the overall losses at minimum value. The minimum loss synchronous pulse width modulation described here uses a minimization criterion function to be used with low switching frequency while getting the acceptable harmonics for the medium voltage multiphase drives. The approach uses the fixed set of optimized switching patterns generated by the programmed technique by minimizing the criterion function.

Weighed Total Harmonic Distortion normalized to DC link voltage is used as the minimization parameter as it is directly related to the harmonics in the system. In order to get the optimization angels fast for any value of modulation index, the real time pattern generation method is used instead of storing the values into the memory. This approach has the possibility of adding offsets to overcome the effects such as DC link balancing and transient normalization.

The number of pulses required at low amplitude modulation becomes very high while keeping switching frequency at acceptable range. This makes the complication for the calculation as well as implementation of Synchronous Optimal PWM at low amplitude modulation. It is therefore, asynchronous space vector modulation is prioritized for the low amplitude modulation. The technique of interrupt signal is used to effectively switch from one method of modulation to the other whenever required.

Multilevel inverter has inherent problem of DC link imbalance, which must be addressed in order to get smooth operation. Many alternatives are purposed for the balancing and choice has to be made to get it work for both asynchronous and synchronous modulation. Further, the transient stability needs to be addressed for the system as it might force the system out of the stability and leading to many technical flaws.

The real time operation of the synchronous optimal modulation requires, fast processing and executing processor. As such, Virtex-5 ppc440 FX30T is used which can operate at very high frequency. During this thesis, such an advanced system is simulated using the advanced computing software's and is implemented in the lab. A six-phase induction machine with the rating of 11.7 kW is used in the lab with the help of three-phase three-level inverter. Xilinx ISE suite is used as the medium to communicate between the hardware and software as well as to build the overall embedded system design.

Problem Description

Medium-voltage (MV) drives has the ability to control large industrial loads found in ship propulsion, mines, power station, and metal processing plants, to name a few. As compared to low voltage (LV) drives, Medium-voltage drives operate at lower current for the same power requirements. For the ability to make better sine wave approximation as compared to the conventional, multilevel inverter is being increasingly used in medium-voltage drives. The losses associated with these drives come from switching of IGBTs and harmonics. With the conventional asynchronous pulse width modulation there is a trade-off between the harmonics and IGBTs switching losses. In order to minimize the overall losses, it is desired to keep low switching frequency along with low harmonics. The advanced technique of synchronous modulation can be used for the purpose by using programmed technique.

Synchronous programmed modulation is coupled with advantages and disadvantages. For low amplitude modulation, the synchronous programmed modulation is not effective as the number of pulses required per quarter cycle of the waveform is very high. This gives complications to generate the synchronous optimal patterns even for the high processing computers. On the other hand, the use of limited number of pulses per period in the low modulation amplitude region is avoided as the switching frequency reduces well below the allowed value.

The primary technique implemented for programmed modulation is to store the data in the memory and fetch it to generate the PWM. However, the use of real time implementation is required for faster and smooth operation of the modulation. The use of fast digital computational circuit is required for the purpose.

High harmonics, common mode voltage and start-up failure are some of the problems associated to the Neutral point imbalance. As such, the topic of Neutral point balancing has been very demanding in the multilevel converter recently. Many researches have been conducted aiming the solution, however they are primarily limited to carrier based and space vector modulation. Programmed modulation, being relatively less popular technique is not wide studied field for neutral point balancing. There is a challenge to

find out the best neutral point balancing method to suit the system that is being built without the compromise in switching frequency and harmonics level.

The problems that are addressed in this thesis are

- Development of the Minimum loss synchronous optimal patterns with the switching frequency below 200 Hz.
- Addressing the problem of synchronous optimal modulation at lower level modulation index with the space vector modulation.
- Implementation of Neutral point balancing algorithm.
- Starting of the three phase induction machine with V/F control method.
- Implementation of the hybrid modulation technique with fast processing Field Programmable Gate Array (FPGA).

Supervisor: Prof. Tore M. Undeland, Elkraft, NTNU

Co-supervisor: Roy Nilsen, Wärtsilä Norway AS

Preface

Working with embedded system has always fascinated me and being an Electrical engineer, it is my pleasure to work on my master thesis titled "*Control of Medium Voltage Multi-phase Machines: Implementation of Minimum loss Synchronous Optimal Modulation with Neutral Point Balancing*". This opportunity has flourished my knowledge of Electrical Drives, Power electronics, Digital control, and its implementation with embedded systems.

I would like to thank my supervisor Professor Tore Marvin Undeland and co-supervisor Roy Nilsen for their time, valuable inputs, and support throughout the entire master's specialization project and thesis. I would like to extend my sincere thanks to Kjell Ljøkelsøy from SINTEF for his help in the lab. Several other persons have helped me during the entire master's period. It is therefore, I would like to thank all the technical staffs of Department of Electrical Power Engineering, NTNU for assisting me with the lab equipment's. Finally, I would like to thank my family and friends for their help and support during my studies at NTNU, Norway.

Norwegian University of Science and Technology

June 2014

Krishna Neupane

Table of Contents

List of Figures

List of Tables

Acronym

MV	Medium Voltage
LV	Low Voltage
IGBT	Insulated Gate Bipolar Transistor
PWM	Pulse Width Modulation
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
CMV	Common Mode Voltage
SVM	Space Vector Modulation
PSM	Programmed Synchronous Modulation
WTHD	Weighted Total Harmonic Distortion
SHE	Selective Harmonic Elimination
SOM	Synchronous Optimal Modulation
SPIM	Six phase Induction Machine
ASD	Adjustable Speed Drive
V/F	Voltage/Frequency
EMI	Electromagnetic Interference
EMC	Electromagnetic Capability
NPC	Neutral Point Clamped Inverter
CBPWM	Carrier Based Pulse Width Modulation
SVPWM	Space Vector Pulse Width Modulation
NTV	Nearest Triangular Vector
HWS	Half Wave Symmetry
QWS	Quarter Wave Symmetry
THD	Total Harmonic Distortion
NP	Neutral Point
DSP	Digital Signal Processing
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
IP	Intellectual Property
CLB	Configurable Logic Block
SOC	System on Chip
DCM	Digital Clock Management
XPS	Xilinx Platform Tool
SDK	Software Development Kit

Chapter 1
Introduction

In this chapter, the background and reason for the choice of the topic as master thesis is described. Previous works on the related subjects along with the motivation, objectives and justification for dealing with the subject will be discussed. Further, limitations of the work and different viewpoints on the subject matter will be illustrated.

As the use of power electric drives are increasing, the use of multi-phase system can achieve the high power drives, as they can handle high power in addition of being rugged and energy efficient. There are several advantages of multi-phase drives as compared to conventional three-phase drives, such as reduced amplitude and increased frequency of torque pulsation, reduced rotor harmonic currents, current per phase and dc link current harmonics, high reliability and increased power [1]. Various researches are being carried out aimed to utilize these advantages of multiphase machine. Refer [2], regarding the detailed illustration on multiphase machine. The major problem however is that the multi-phase supply is not readily available.

One of the key areas in the electric drives is the modulation technique and type of converter topologies being used. It is desirable to have high power supply ability while having very low loss. Medium voltage drive as capable of high power supply is being chosen over the low voltage drive. It has been in use for industrial applications of production and process industries [3]. The use of multilevel converter topologies have gained lot of interest recently, and was initiated by the introduction of Neutral point clamped inverter by Nabae et al [4]. The reason behind this is the existence of multiple advantages of multilevel converter over the conventional converters [1, 2, 5, 6, 7 and 8].

The major issue with any converter topology is to reduce the harmonic distortion, conduction and switching losses. Thus, it is desirable to have low switching frequency topology with reduced harmonics. Since, the harmonics normally tends to increase with the reduction in the switching frequency; it is a challenge to determine the methodology to reduce the harmonics while having low switching frequency by using the unconventional methods. On the other hand, programmed modulation technique has

been tested to be used for the selective harmonics elimination [9, 10, and 14] which gives low ripple output and [8 and 11] shows the implementation of programmed modulation to get low switching frequency with acceptable harmonics. In [12], it has shown that asynchronous PWM can be used with low switching frequency with the prediction of trajectory patterns of the stator current vector but is limited to error of the torque producing current components and exerts no constrains on harmonics of the field producing current. An adaptive duty cycle modulation algorithm for reduced switching frequency with acceptable harmonics is suggested in [13].

Eliminating several lower order harmonics by programmed PWM can generate low ripple, reduced torque pulsation with overall improved performance [14] however, total minimum loss will be obtained by reducing both switching frequency and ripple. This requires setting the target (the criterion function), for minimizing the harmonics using the advanced programmed technique and implementing it at low switching frequency. IGBT's are the obvious choice for medium voltage high power drives but the problem is with the switching loss. Switching loss can be lowered by reducing the switching frequency, however; it generally increases harmonics. A synchronous PWM technique is used for obtaining low switching frequency and acceptable harmonics for higher amplitude modulation and asynchronous modulation is used for low amplitude modulation due to the difficulty in computation and application of increased number of pulses per quarter cycle with low amplitude modulation.

Amplitude common mode voltage (CMV) and high frequency generated in PWM inverters has to be negotiated to avoid failure of drive due to the shaft voltage and bearing current on the rotor side. DC link voltage variation is a factor adding to the above problems in addition to increasing harmonics. There have been various researches to negotiate such problems [15and 16] however these methods have their own limitations and are limited to the Space Vector Modulation (SVM) and carrier based modulations. It is thus, essential to determine a suitable method for balancing the neutral point in the multi-level inverter.

In this report minimum loss programmed modulation technique will be developed and implementation results from the simulation and lab will be presented. Various issues such as, starting of the machine, neutral point balancing and switching from one modulation technique to another in real time will also be addressed. On the overall

system, 200 HZ of switching frequency for 50 Hz fundamental output frequency will be considered.

1.1 Background and Previous Works

This thesis work is the continuation of master's project [17], and basic theories including the simulation study was done during the earlier semester. The work during this thesis includes the improvement in the simulation and methodology on theory in addition to the implementation of the hybrid modulation technology (Space Vector PWM for lower amplitude for lower amplitude and Programmed Synchronous Modulation (PSM) for higher amplitude).

The similar approach as this thesis has been done by my seniors at Norwegian University of Science and Technology (NTNU) under the close supervision from Roy Nilsen of Wärtsilä Norway AS and Professor Tore Marvin Undeland of NTNU. Modelling and controlling high performance medium voltage drives was done by Roger Enes [18], where he simulated the patterns for programmed modulation in Simulink and implemented stator flux trajectory control stated in [11]. The criterion function written by Roy Nilsen in [19] is used for the minimization of weighted total harmonic distortion with respect to DC link (WTHD0). Details regarding this criterion function are given in Chapter 5. The approach is to identify the starting angles and find the different angles pattern to get the minimum of WTHD0 with the increase in amplitude. Set of starting angles are identified by trials as it has been found that different starting angles result in different minimum WTHD0. This is because the system gets stuck at the local minima instead of getting to the global minima. Further, the work was implemented in the lab by Mamata and Biruk [20, and 21] using the open loop configuration for three phase inverter and tested the system with six phase induction machine working on three phase supply. High speed processor called Field Programmable Gate Array (FPGA) was used for the purpose where all the angles as established in [18] were stored in the memory and fetched to perform the modulation. The above work doesn't however include the flux trajectory control.

The industrial need of low loss modulation technology has increased the research in the field of programmed modulation in order to get the low switching frequency while having the acceptable current ripple. Two approaches are common for the loss minimization, namely selective harmonic elimination (SHE) and minimum loss PWM. In 1973 Patel and Hoft presented an approach for harmonic elimination and voltage control [22]. They developed a generalized method to eliminate a fixed number of harmonics in half and full bridge inverter output and showed the solution for up to five harmonics level. Latter in 1977, Buja and Indri [23] developed optimal PWM by analytical calculation for proper choice of commutation angles to minimize the current ripple. Murphy, Howard and Hoft in 1979 realized the use of digital circuit with micro-processor for look up table based implementation [24]. In 1991, Holtz showed the use of low switching frequency for the field-oriented asynchronous PWM to implement with the high performance ac machine drives [14]. Latter in 1994, he employed the method called Synchronous Optimal Pulse Width Modulation (SOM) for medium voltage drives at low switching frequency [34]. The contribution of Holtz ever since with the SOM has been fascinating. In 2007, he along with Oikonomou implemented the SOM and stator flux trajectory control for medium voltage drives [11]. In more recent 2010, the similar hybrid modulation technology as in this thesis was implemented by him along with Rathore and Boller [8] except that it had different criterion function and was implemented for the five level inverter with neutral point balancing algorithm implemented for five or higher level inverters.

In this thesis, an approach of hybrid modulation technology to achieve the global minimum loss has been used. Different pulse number is used for the range of high amplitude modulation while space vector modulation is used for low amplitude modulation. Asynchronous modulation is also kept at very low switching frequency as is the synchronous programmed modulation. Special attention is given to neutral point balancing for the entire range of modulation. Implemented technology doesn't store the pre-calculated values but real time values are generated in the very fast operating processor called Field Programmable Gate Array (FPGA).

1.2 Motivation

Developments in special-purpose application specific power and control systems have been escalating at an unprecedented rate. We can see that these developments are well blended in the lifestyles of people as they are relying more and more on the automated stand-alone systems offered by the highly specialized embedded applications in many parts of the world. I have been fascinated by the world of possibilities offered by Microcontrollers and FPGAs.

Medium voltage drives with multilevel converter for multiphase machine has lots of complications in getting low loss system. Not only it has been great interest for the researchers, but it has been fierce competition to get such system for ship propulsion system from various companies such as Azipod® Propulsion from ABB [25], Wärtsilä Low Loss Propulsion system [26], eSiPOD (Siemens Solution for Podded Propulsion) [27] among the many. SOM facilitated by the use of highly specialized processer such as FPGA has been observed to be a great choice to obtain the minimum loss system.

Multilevel converter has facilitated the medium voltage drives with the ability to synchronize multiple IGBT's and perform the high power drives operation. The advancement in the power electronic semiconductor device has given us higher flexibility at economic operation.

As such, it is of great interest to be able to work with such a complicated system with continuous eye on the technological advancement in the field from many researchers and companies around the world.

1.3 Scope of the Work

The generation of low ripple output current with the switching frequency maintained at very low give Minimum loss synchronous PWM. The PWM is based on the calculation of switching patterns by advanced computer software such as Matlab that determines the solution for the minimum value of criterion function. However, programmed synchronous modulation is not a self-sufficient modulation technique as already

described. This inherent problem comes from the fact that the calculation and implementation complexity for the lower amplitude is very rigorous because of high number of switching pulses required at this level.

This thesis work describes the technique implemented for the Minimum loss synchronous PWM along with the simulations in Matlab for space vector modulation and calculation of optimal angles. Optimal technique is developed for the six-phase induction machine, but it is implemented on six-phase machine working with three phase supply. Since, the system is tested in the lab conditions, subsequent changes are required before it can be implemented for the industrial purposes.

1.4 Organization of the Report

In Chapter 1, introduction of the thesis work, including the background, motivation, objectives and scope of the work are discussed. Introduction to multiphase machine, multilevel converters, and existing modulation technologies are discussed in Chapters 2, 3 and 4 respectively. Theory used for this thesis, minimum loss synchronous PWM and neutral point balancing are discussed in Chapters 5 and 6. The part of thesis includes both simulation and implementation. Chapter 7 is used to describe the simulations used and their corresponding results. Field Programmable Gate Array (FPGA) is used for the implementation with the help of software packages from Xilinx Inc. Chapter 8 is used to discuss the tools used for the implementation of programmed modulation with space vector modulation in the lower amplitude range. Details regarding the output and validity of the output with the hardware implementation are presented in Chapter 9. Chapter 10 is used for conclusion and future works.

Chapter 2
Multiphase Machine

In this chapter description of multiphase machine (mainly six-phase machine) is presented. Various advantages and disadvantages are discussed over the conventional three phase machine along with its use in the industrial applications. Since, one of the uses of this project can be ship propulsion drive; it has been discussed in short here. Further, starting of the induction machine is described along with the harmonic loss.

2.1 Introduction

Three phase AC machine drives are widely used in the present day's industries replacing the traditional DC machines that gives more effective electromechanical energy conversion than AC machine drive consisting of IGBT's converter. However, it has been found that using the phase number higher than 3, the performance of AC drives with IGBT's converter can be improved [28]. This has led to the development of multiphase machine and advancement in power electronics switching devices is increasing the rise of interest. As compared to the conventional three phase drives, multi-phase machine has reduced amplitude and increased frequency of torque pulsation, reduced rotor harmonic currents, reduced current per phase and dc link current harmonics, high reliability and increased power [1]. Increased efficiency arises from the reduction in stator copper loss and a bit of reduction rotor copper loss while reduction in torque pulsations come from attenuation of the phase belt harmonics [28].

Multi-phase machines can be built with multiples of three phase windings, such as six-phase machine can be built with two three-phase windings and nine-phase machine can be built from three three-phase windings. For the rest of this chapter, six-phase induction machine is considered as the multiphase machine as it is used in the lab during the experiments. Although, the topic considers the multi-phase machine, the lab is performed on six-phase machine with three phase configuration. This thesis however, outlines the modulation technique that is designed for multi-phase machine and results can be implemented to any multi-phase induction machine.

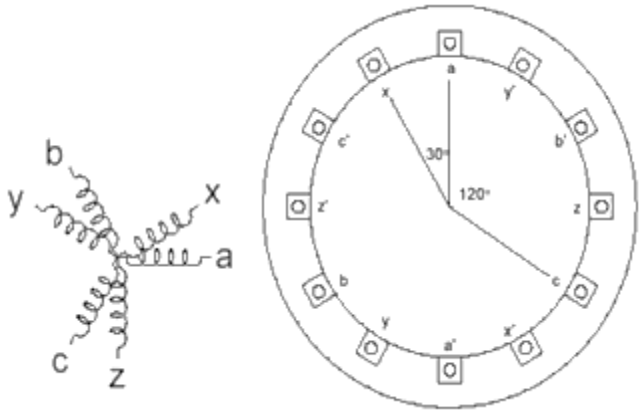


Figure 1: Six phase Induction machine (Left Electrical Diagram, right Basic Construction) [21]

Six-phase induction machine (SPIM) has gained interest in industrial applications. There are two types of SPIM, namely symmetrical SPIM and asymmetrical SPIM. The winding configuration makes the earlier symmetrical by making equally spaced windings while the later asymmetrical by unequal space distribution between the windings.

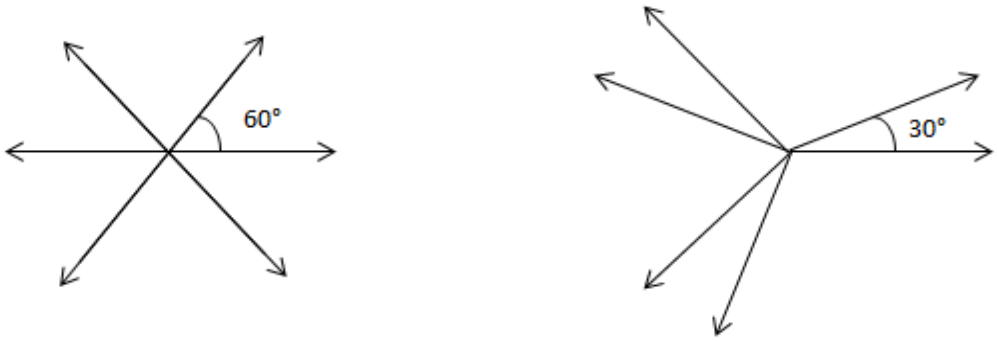


Figure 2: Six phase Induction machine Types (left Symmetrical SPIM, right Asymmetrical SPIM)

Symmetrical SPIM are more common in industries as they have less pulsating torque and fewer harmonics as compared to the asymmetrical SPIM [30]. Symmetrical SPIM however has a problem of magnetic circulating currents. Asymmetrical SPIM eliminates (6n + 1) order harmonics, where n = 1, 3, 5……. [31] and mainly used for the research purposes. Drives of multiphase machine (SPIM) can be done by using two three phase inverter which are synchronized with each other. For programmed modulation or hybrid modulation (the combination of space vector modulation and programmed modulation), the same processor can be used to generate the synchronized PWM for

both of the inverter at the same time. The block diagram below represents the drive of SPIM with the use of two inverter synchronized using the same processor.

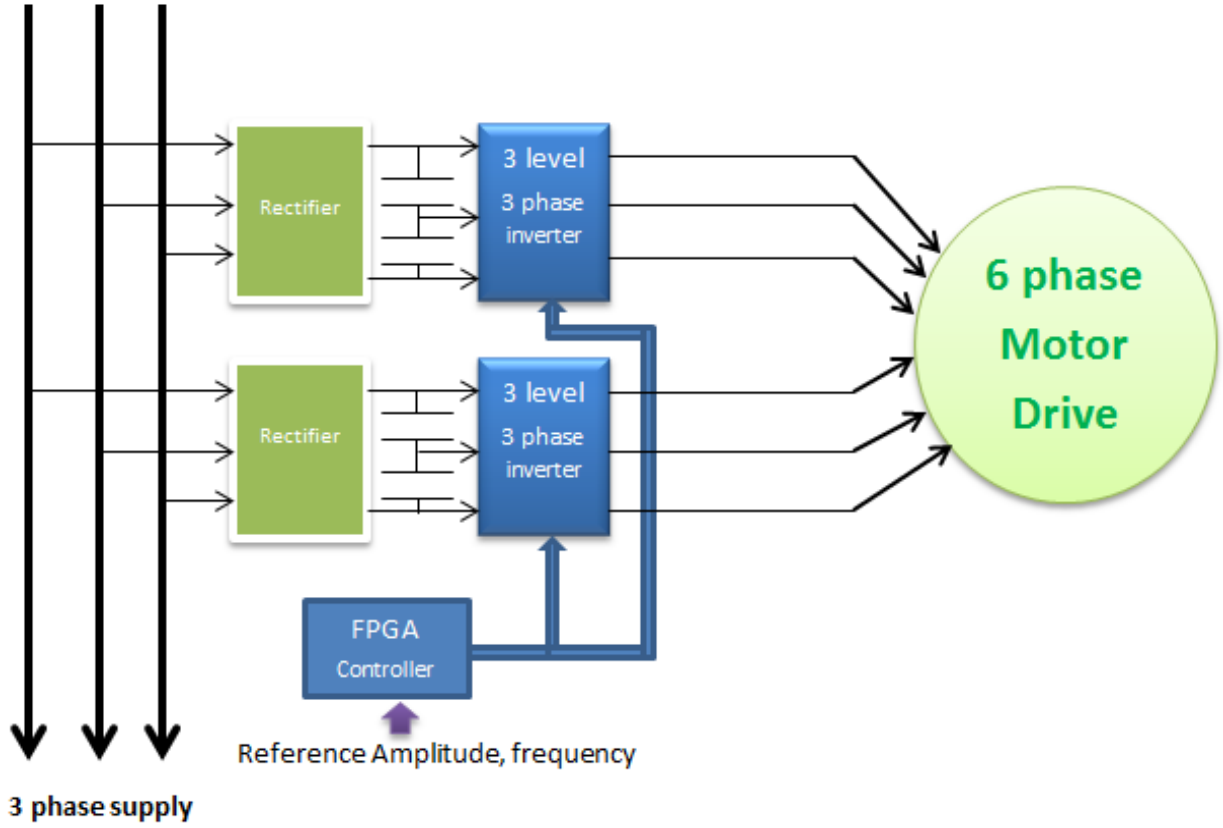


Figure 3: Six phase Induction machine Drive with two three level inverter synchronized

Adjustable Speed Drive (ASD) can be used with the multi-phase machine drive to achieve the performance of converter and motor combination.

2.2 Ship Propulsion Drive

By using the multi-phase machines, multiple phases of supply are required but limited rating power electronics converters can supply higher levels of power. Use of multi-phase machine in Electric ship propulsion systems has accelerated the development of high power multiphase machines as it can operate smoothly even during the loss of a phase.

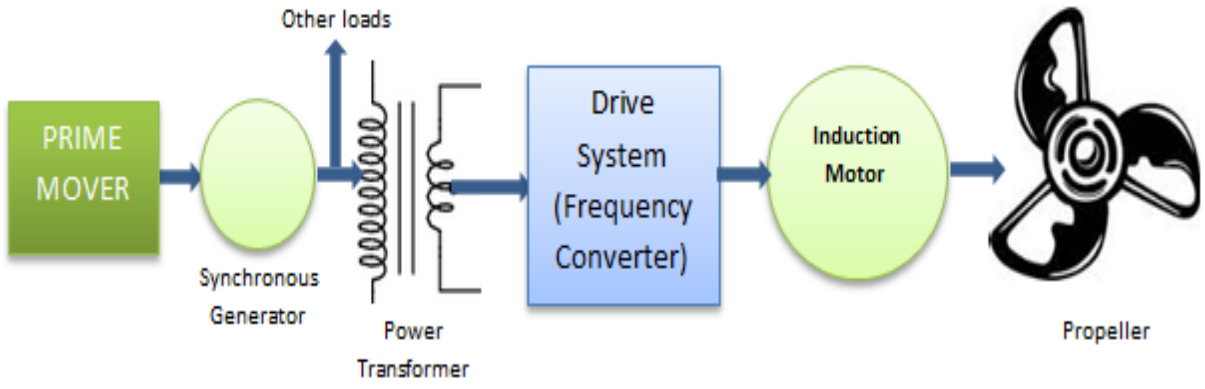


Figure 4: Block Diagram of Ship Propulsion system

Steam turbines were used in the past as the prime movers; however medium and high speed diesel engines are commonly used these days. The power for the propulsion system including all the loads and auxiliary loads come from the common energy source which is generator system. Electrical propulsion can reduce the noise and vibration while producing the smooth operation with high level of reliability. Further, it is also easier to maintain and repair as compared to steam turbines prime mover.

2.3 Starting of Induction Machine

There are multiple methods of starting the induction machine. Stator voltage control or rotor resistance control is a general approach for starting the three phase induction machine but they result in low efficiency. Voltage/Frequency (V/F) control is a more efficient method for speed control as well as starting of the machine.

If the induction machine is working below the rated speed then V/F is kept constant to make constant flux which keeps the torque capacity of the machine at the same speed. However with low frequency, the capacity of torque reduces and it has to be adjusted by increasing the voltage. As the frequency increases, the voltage/ frequency ratio reduces along with the torque.

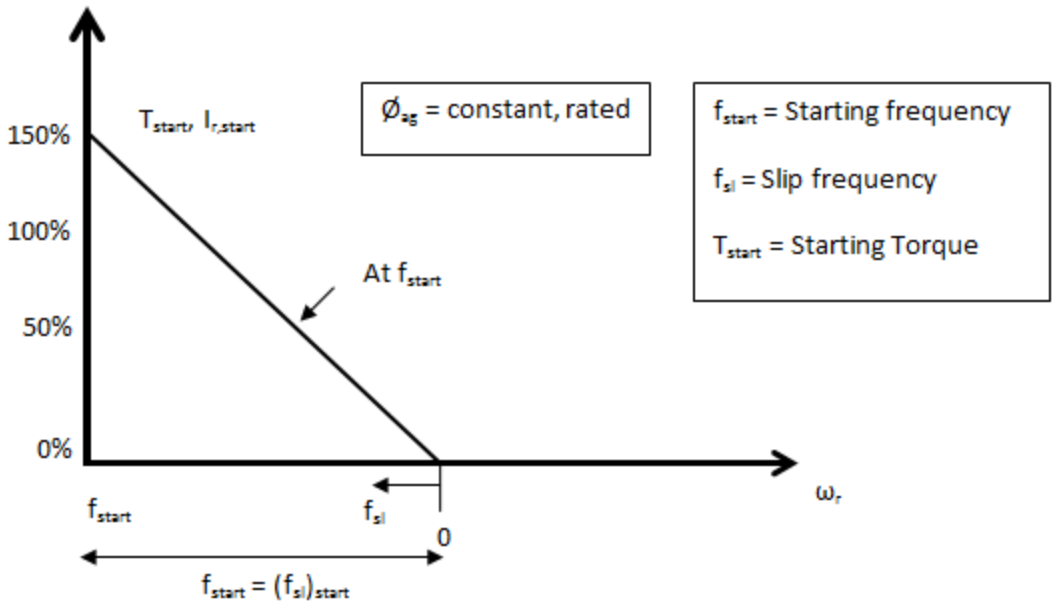


Figure 5: Start-up Frequency [32]

From the above figure, the starting frequency can be calculated as,

$$f_{start} = \frac{T_{start}}{T_{rated}} . (f_{sl})_{rated}$$

The stator frequency is increased to the rated speed without letting the stator current to exceed the rated values. When the base speed is achieved, the voltage is kept constant while frequency is increased. This varying frequency control method provides good operating and transient behaviors of the induction machine drive.

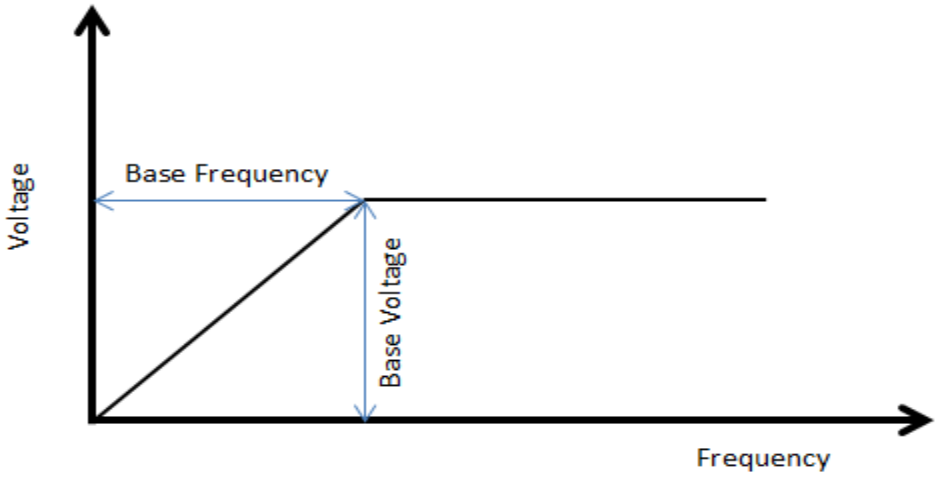


Figure 6: Ramping of frequency during the start-up

2.4 Harmonic loss in Induction Machine

Harmonics is the inherent problem with power electronics AC voltage generating circuits which actively participate in the power loss except the first harmonics which is also called fundamental harmonics. The loss in the motor is in the form of heating. Torque pulsations are an effect of harmonic current because of harmonic currents interacting with fundamental magnetic field. This has interested many researchers to reduce the harmonics in the power electronics applications with AC voltage output.

Fourier analysis acts as a tool to deal with the harmonics where it can separate the different combination sinusoidal waveform to generate the output waveform. In a balanced three phase system, 1^{st}, 4^{th}, 7^{th} and others act as the harmonics that rotate the motor in the positive direction while 2^{nd}, 5^{th}, 8^{th}, and others act as negative harmonics which tend to rotate the motor in the reverse direction. In the meanwhile, 3^{rd}, 6^{th}, 9^{th} and others are zero sequence and contribute only on heating the machine. The higher harmonics has low loss effect as compared to the lower harmonics, because of low impedance and thus high current for lower harmonics. Higher order harmonics can be removed by using the low pass filter at the inverter output.

Performance index is needed to determine the effect of harmonics in the system as the effect of high order and low order harmonics are different. Weighted Total Harmonic Distortion (WTHD) is one of such measure. This can be defined as the normalized Total Current Harmonic Distortion with maximum inrush current.

Chapter 3
Multilevel Converter

In this chapter different types of multilevel converters and their structures will be introduced along with the highlight on Neutral point clamped inverter, which is used during the lab and simulation. The advantages and disadvantages of multilevel converter will be discussed in details. Most of this chapter focuses on modern and industrial applications of multilevel converters.

In specialized applications such as ship propulsion drives, industries have begun to seek for high power in the rates of megawatt drives using the medium voltage supply. Due to the limitation of semiconductor device (specially the IGBT's) voltage handling capacity, it is not possible to connect single semiconductor device switch directly. The main aim of multilevel converter is to get higher power applications by using the series of semiconductor devices with lower dc switching voltages by synthesizing a staircase of voltage spectra. The semiconductor devices in multilevel converter use less voltage per device while having low switching frequency. The major advantages of multilevel converter is summarized as [1, 5, and 6]

- It can operate at lower switching frequency as a result the switching loss can be reduced highly as compared to the conventional converters.
- It draws input current with less distortion.
- Common mode voltage can be reduced by the use of multilevel converter, reducing the stress in the bearings of the motor.
- Low output voltage distortion with reduced dv/dt stress
- Electromagnetic compatibility (EMC) problems are reduced

These advantages has been the key for using multilevel inverter over the conventional one in the recent years for advanced drive system with high power requirement. However, there are disadvantages of multilevel converter as well, which includes

- It uses the greater number of semiconductor switching with increased cost and complexity
- With multilevel inverter there is an inherent problem of Neutral point voltage

3.1 Multilevel Converters Structures

Numerous multilevel technologies are proposed over the last few decades with the constant ability to use different modulation technology at the same time reduce the overall loss in the system. These topologies can be used to operate in both the modes inverter and rectifier. The different topologies can be distinguished into three types

3.1.1 Cascaded H-Bridges

Separate m number of DC source is fed to the number of cascaded H bridges to form m-level cascaded inverter. Each of this cascaded inverter is able to generate either $+V_{dc}$, 0 or $-V_{dc}$ by the combination of four switching semiconductor devices. The combination of four switching's are listed as below

Table 1: Switching states for Cascaded H-Bridge inverter

Switching Signals				Output
S_1	S_2	S_3	S_4	
ON	OFF	ON	OFF	$+V_{dc}$
OFF	ON	OFF	ON	$-V_{dc}$
ON	ON	ON	ON	0

The AC outputs of all the full bridges are connected in series to synthesize the output voltage waveform.

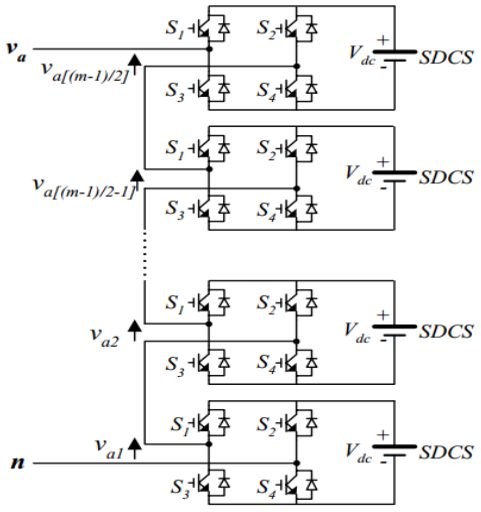


Figure 7: Single-phase structure of a multilevel cascaded H-bridges inverter [33]

For m-level cascaded inverter, there exists m = 2n+1 number of output phase levels where n is the number of independent dc sources.

3.1.2 Diode clamped multilevel Inverter

Diode clamped multilevel inverter consists of number of capacitor which subdivides the dc source voltage into n+1 levels, where n is the number of capacitor.

Table 2: Switching states for diode-clamped six level inverter voltages

Voltage	Switch State									
	S_{a5}	S_{a4}	S_{a3}	S_{a2}	S_{a1}	$S_{a'5}$	$S_{a'4}$	$S_{a'3}$	$S_{a'2}$	$S_{a'1}$
V_5=5 V_{dc}	1	1	1	1	1	0	0	0	0	0
V_4=4 V_{dc}	0	1	1	1	1	1	0	0	0	0
V_3=3 V_{dc}	0	0	1	1	1	1	1	0	0	0
V_2=2 V_{dc}	0	0	0	1	1	1	1	1	0	0
V_1=1 V_{dc}	0	0	0	0	1	1	1	1	1	0
V_0=0 V_{dc}	0	0	0	0	0	1	1	1	1	1

The above table shows the switching states of diode clamped six-level inverter. The voltage level across each semiconductor device is limited to the voltage across each of the capacitor. For ideal design, the size of all the capacitors should be equal so that voltage across each capacitor is limited to V_{dc}.

The m-level of diode clamped inverter consists of m-level of output phase voltage and 2m-1 level of output line voltage. The entire phases share a common dc bus as a result minimizes the capacitance size. The main disadvantage of this type of inverter is the overcharge or discharge of dc levels without precise monitoring and control making unwanted dc link voltage.

3.1.3 Flying capacitor multilevel inverter

Flying-capacitor based inverter was introduced in 1992, by Meynard and Foch [34]. It consists of ladder structure with different voltage level of dc side capacitors. In an m-

level flying capacitor multilevel inverter, it has m-levels of phase voltage, m-1 capacitors and 2m-1 levels of line voltage.

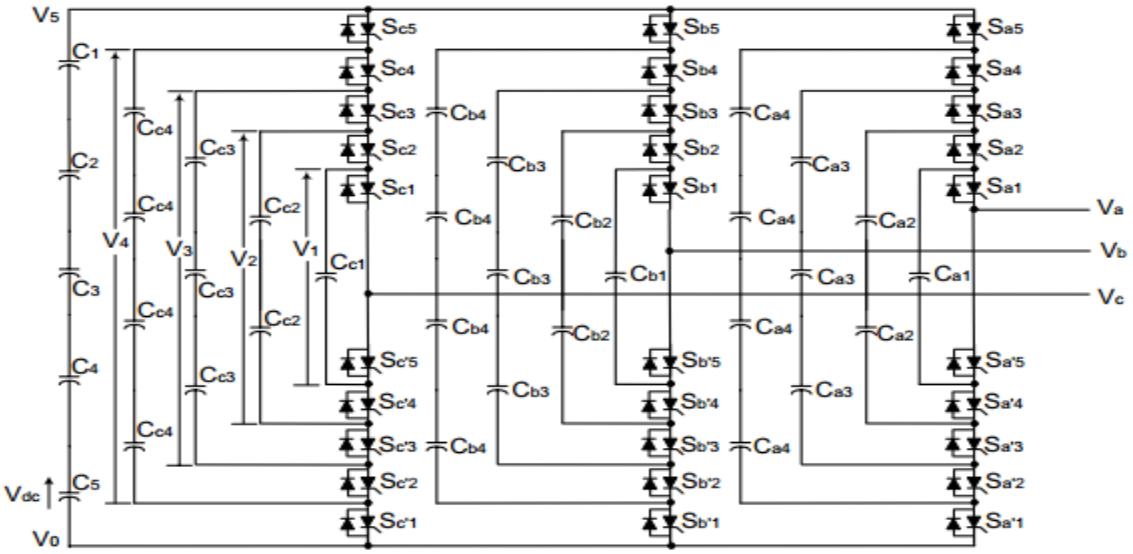


Figure 8: Six-level structure of flying capacitor inverter with three phases output [33]

3.2 Neutral Point Clamped Inverter

Neutral point clamed inverter is widely popular topology and was the first known multilevel inverter which was introduced in 1981 by Nabae et. al [4]. The switching characteristics of this inverter gives, either positive, negative or zero DC voltage level which is why it is often referred as three level diode clamped inverter as well.

Table 3: Switching patterns NPC inverter

IGBT1	IGBT2	IGBT3	IGBT4	Output
0	0	**1**	**1**	$-V_d/2$
0	**1**	**1**	0	0
1	**1**	0	0	$+V_d/2$

To control the three level inverters, only two switches are switched at a time out of the four switches on one leg. The switching of any one leg gives three different output with $+V_d/2$, 0 and $-V_d/2$.

The structure consists of four semiconductor device per phase which allows each device to operate by blocking only half of the DC link voltage i.e. lowers voltage ratings semiconductor devices can be added in series to operate for higher voltage.

Neutral point voltage is inherent problem with this type of inverter as compared to the conventional two level inverters where DC link is not present. Two independent sources can be used for avoiding the problem but the cost of the system rises and it is not advised to do so. As such, it has been a researcher's interest field to design a control method for the DC link capacitors voltage.

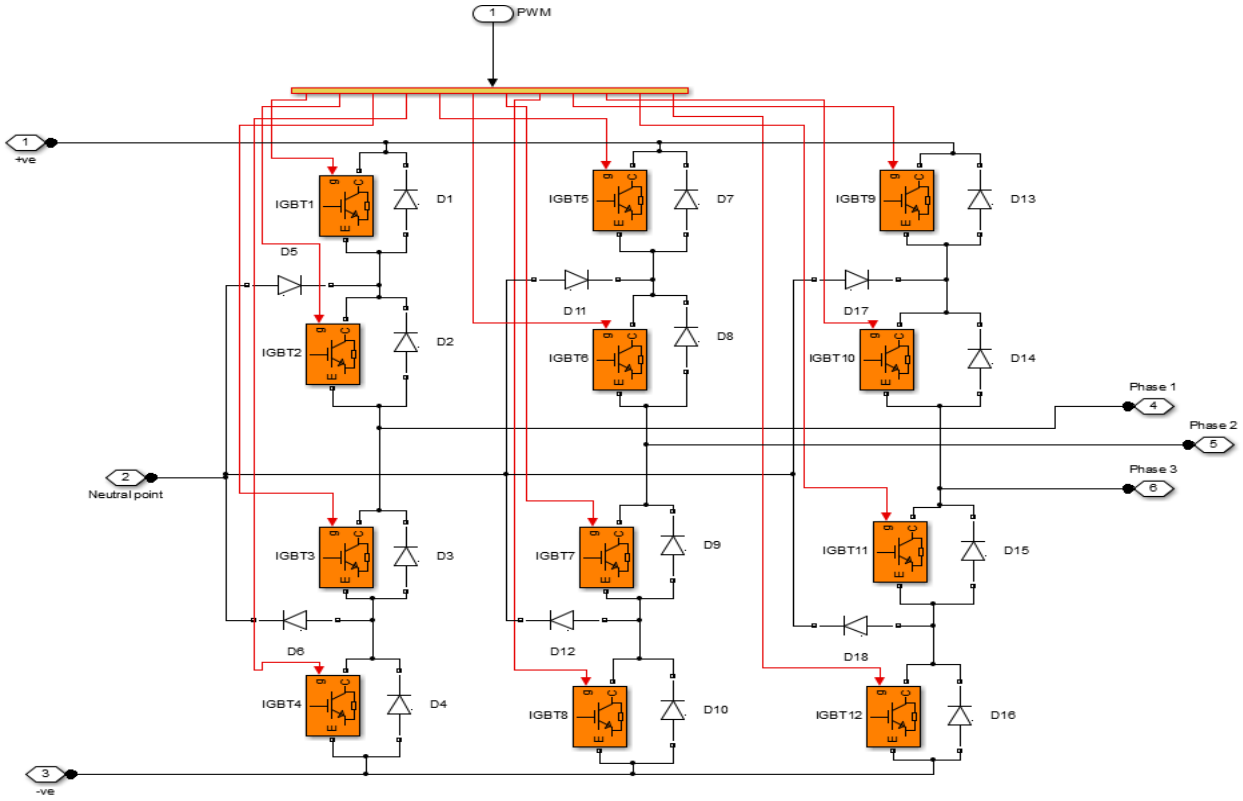


Figure 9: Neutral Point Clamped (NPC) Inverter with IGBT

Chapter 4

Modulation Strategies and Harmonics Loss

This chapter was first published in [17], which is the background of this thesis. Some of the contents are added and modified from the original text whenever required. Various approaches to the modulation technique are introduced with their advantages and disadvantages. Analytical calculation is carried for the modeling of the strategies and harmonics loss is discussed.

The concept of Neutral Point Clamped (NPC) inverter was introduced by Nabae et. al. [4] in 1981. Ever since, the research have been very interesting in the field of controlling the NPC inverter as it is the heart of any power electronic converter. Various approaches of modulation strategies to suit different converter topologies are suggested by different studies.

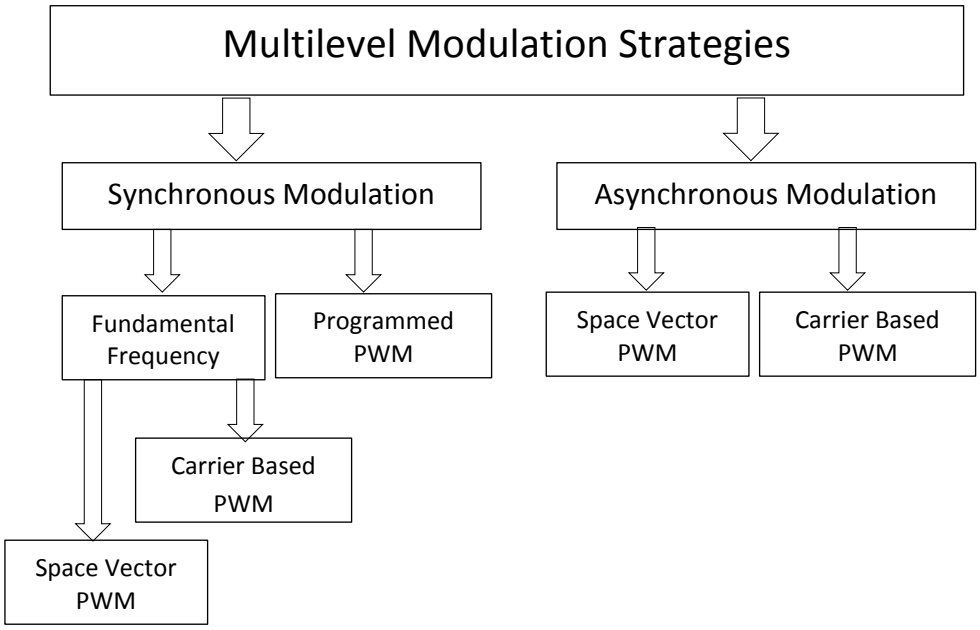


Figure 10: Modulation Strategies

These different approaches have been set according to the different priorities of optimization and requirements. The techniques for the Minimum loss PWM and

Selective Harmonics Elimination (SHE) for the multilevel modulation has been investigated in many different approaches. [3, 10, and 11] suggest various methods of categorizing PWM. Based on whether it is Synchronous or Asynchronous, these modulation techniques can be categorized as in Figure 10 for multilevel converter.

4.1 Asynchronous Modulation

If the zero crossing between carrier wave and reference wave is not synchronized, than it is called asynchronous modulation which results in unequal number of pulse numbers between positive and negative half cycle. If the number of pulse is high enough than such unequal pulse will not result in high sub-harmonic component however it is a problem with low number of pulses.

The two types of asynchronous modulation

i. Carrier Based PWM
ii. Space Vector Based PWM

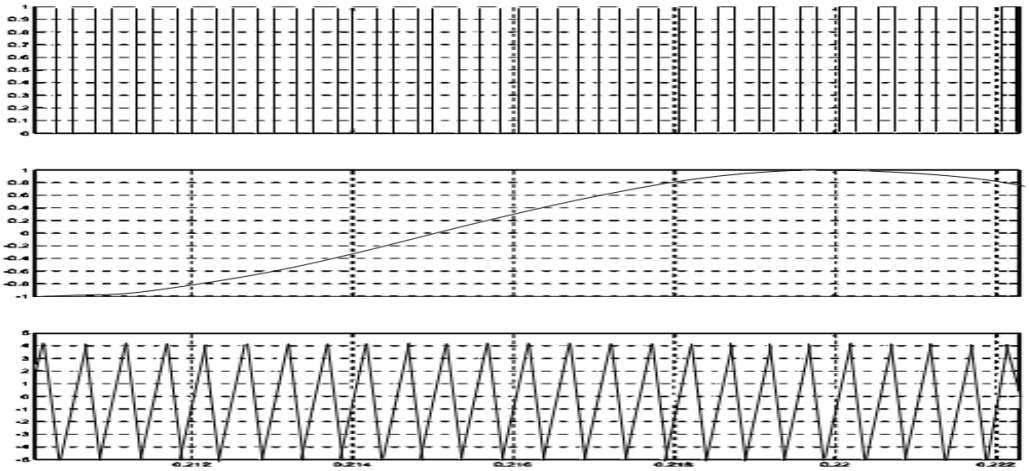


Figure 11: Carrier Wave PWM [3]

4.1.1 Carrier Based PWM

Carrier based PWM are the classical and most widely used method for Pulse width modulation. The reference wave is compared with the carrier wave to obtain the

switching pulse. The typical harmonic spectrum of carrier based PWM has prominent harmonic amplitudes around the carrier frequency and its harmonics [35].

i. Naturally Sampled PWM

This is one of the earliest and most easily realized modulation strategies which compare the low frequency target reference wave with the high frequency carrier wave. It is based on the comparator which compares the sinusoidal reference wave with very high frequency triangle wave.

Such a naturally sampled PWM have higher presence of harmonic content and generally not desired for the high power system where efficiency is of concern.

ii. Regularly Sampled PWM

In case of regularly sampled PWM, reference voltage is sampled at top or top and bottom of the triangular wave and then kept constant until next sample. To avoid multiple switching transitions, the reference voltage is sampled at the point where the peak of the carrier is reached. It can either be the case that the reference voltage is sampled at the top of the triangular wave and kept constant throughout the triangle period or the case of reference voltage is sampled at both top and bottom of the triangular wave while keeping constant in the half symmetry of the triangle. The earlier is termed as Symmetrical Regular sampled PWM and the latter is termed as Asymmetrical Regular Sampled PWM.

iii. Analytical Calculation of AC side Harmonic Losses for Carrier Wave Modulation

The weighted total harmonic distortion with reference to DC voltage of one capacitor in DC link is given by

$$WTHDO^2 = \frac{1}{U_{dc1}^2} \cdot \sum_{h=2}^{\infty} \left(\frac{U_h}{h}\right)^2$$

$$WTHDO^2 = \frac{1}{U_{dc1}^2} \cdot \sum_{h=2}^{\infty} \left(\frac{U_h}{h}\right)^2 \cdot \left(\frac{w_o L}{w_o L}\right)^2$$

$$= \frac{(w_o L)^2}{U_{dc1}^2} . \sum_{h=2}^{\infty} \left(\frac{U_h}{h w_o L}\right)^2$$

$$= \frac{(w_o L)^2}{U_{dc1}^2} . \Delta\, I_{N,rms}^2 . 2$$

$$Where, \quad \Delta\, I_{N,rms}^2 = \sum_{h=2}^{\infty} \left(\frac{U_h}{h w_o L}\right)^2 \; is\ the\ total\ harmonic\ current.$$

$$WTHDO^2 = \frac{(w_o L)^2}{U_{dc1}^2} . \Delta\, I_{N,rms}^2 . \frac{\Delta\, i_h^2}{\Delta\, i_h^2}$$

$$Where, \quad \Delta\, i_h = \frac{U_{dc} T_{sw}}{8L}$$

Now,

$$WTHDO^2 = \frac{(w_o L)^2}{U_{dc1}^2} . \Delta\, i_h^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \frac{(w_o L)^2}{U_{dc1}^2} . \left(\frac{U_{dc} T_{sw}}{8L}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \frac{(w_o L)^2}{U_{dc1}^2} . \left(\frac{2U_{dc1} T_{sw}}{8L}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \frac{(w_o L)^2}{U_{dc1}^2} . \left(\frac{2U_{dc1} T_{sw}}{8L}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \left(\frac{w_o T_{sw}}{4}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \left(\frac{2\pi f_o . \frac{1}{f_s}}{4}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$= \left(\frac{\pi f_o . \frac{1}{f_s}}{2}\right)^2 . \frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2}$$

$$WTHDO^2 = \left(\frac{\pi}{2}.\frac{f_o}{f_s}\right)^2.\frac{\Delta I_{N,rms}^2}{\Delta i_h^2} \ldots\ldots\ldots\ldots (1)$$

It has been proven from [7 and 12] that carrier wave with 3rd harmonic injection gives lower weighted total harmonics distortion compared to normal carrier wave. As such our analysis will be focused on carrier wave with 3rd harmonic injection.

A simplified expression for $\frac{\Delta I_{N,rms}^2}{\Delta i_h^2}$ is derived in [35] based on [36] for sinusoidal CBPWM with 1/6th 3rd harmonic injection of 3 level inverter,

$$\frac{\Delta I_{N,rms}^2}{\Delta i_h^2} = \begin{cases} \frac{1}{\pi}\left(-\frac{5}{14}M^3 + \frac{7}{16}M^4\pi - \frac{11}{24}\sqrt{3}M^2 + \frac{17}{36}M^2\pi - \frac{8}{9}\sqrt{3}M^3\right) & \boldsymbol{for\ 0 \le M \le 1/\sqrt{3}} \\ \frac{1}{\pi}\left(\frac{91\pi}{81}M^2 + \frac{4}{9}M^4\pi + \frac{7}{12}\sqrt{3}M^2 + \frac{4}{27}\pi - \frac{127}{315}M^3 - \frac{19}{9}\sqrt{3}M^3 - \frac{4}{3}\sqrt{3}M\right) & \boldsymbol{for\ 1/\sqrt{3} \le M \le 2/\sqrt{3}} \end{cases}$$

Based on the above expression, $\frac{\Delta I_{N,rms}^2}{\Delta i_h^2}$ can be plotted

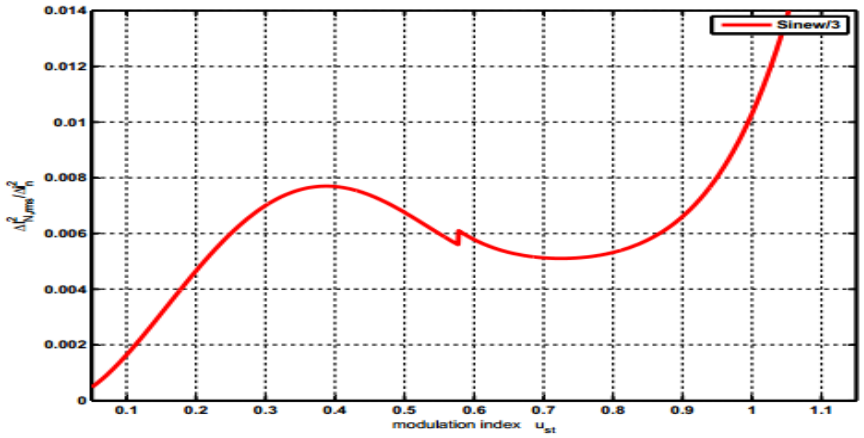


Figure 12: $\frac{\Delta I_{N,rms}^2}{\Delta i_h^2}$ for sinusoidal CBPWM with added 3rd harmonic injection

Simplified expression for $\frac{\Delta I_{N,rms}^2}{\Delta i_h^2}$ is enough for the analysis in our case because our analysis of this result is limited for the lower values of modulation index where it is found that full analytical expression and simplified analytical expression does fallow the same path. This can be verified with [35] page 111.

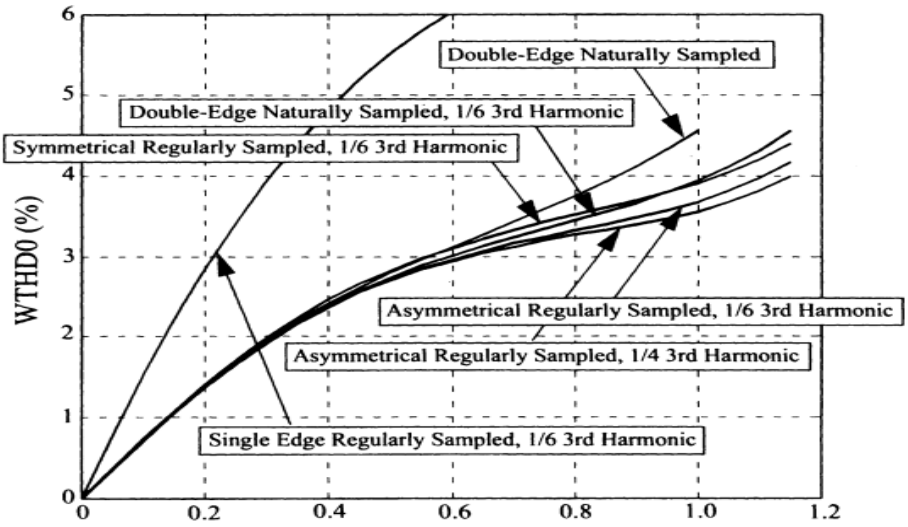


Figure 13: WTHD0 of l-l switched voltage for there-phase inverter vs modulation technique, fc/f0 = 11 [3]

The above plot shows the comparison of different sinusoidal PWM techniques which will later be shown to have very high weighted total harmonic distortion compared to the programmed synchronous modulation.

4.1.2 Space Vector Modulation

Space vector pulse width modulation (SVPWM) have now been used for quite a lot with the idea of vector representation of three-phase systems that is from the research contributions of Park [37] and Kron [38]. Kovacs and Racz [39] made the final step for systematically using the Space Vectors. They have discussed the mathematical treatment, physical description and drive transients.

Space vector modulation can be extended from two level to three level inverter, with slightly increase in computational overload. There are various papers regarding optimal space vector modulation utilizing the redundant vectors offered by multilevel converter. According to [36] sine wave with third harmonic injection PWM is essentially identical to space vector PWM except for the treatment of the zero space vector.

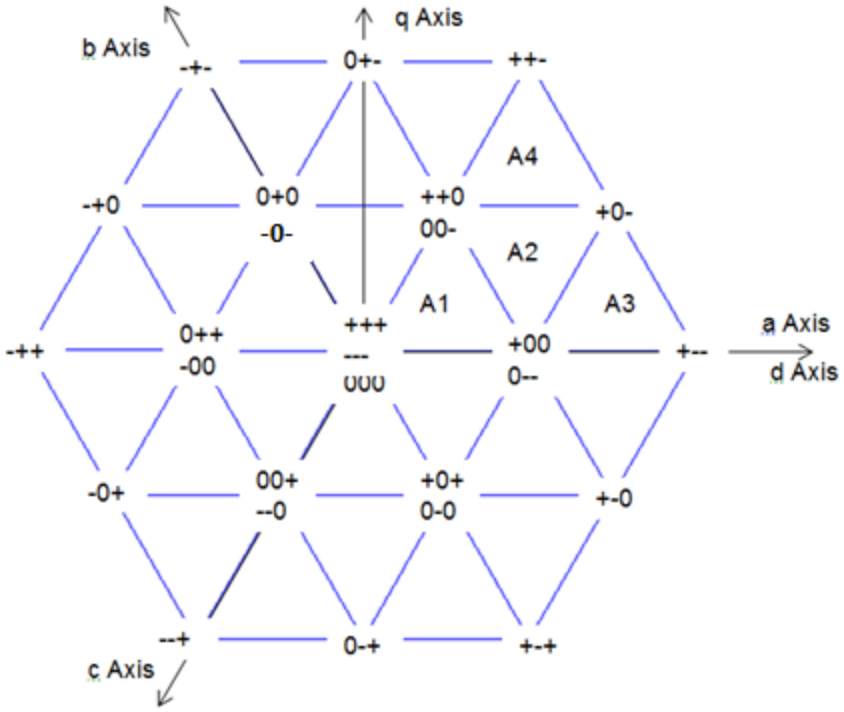


Figure 14: Space Vector Modulation for 3 level converters

For 2-level inverters 2^3=8 space vectors can be created, while two of them are the zero vectors. For 3-level inverters 3^3=27 space vectors can be created as shown in above figure, where three of them are equal zero and the small vectors are redundant. This means that the same space vector can be created by different combination of bridge leg voltage.

$$X_{x=+,-,0} = \begin{cases} +\frac{V_d}{2} \; for + state \\ 0 \; for \; 0 \; state \\ -\frac{V_d}{2} \; for - \; state \end{cases}$$

A popular modulation method is a Vector Modulation method called NTV; Nearest Triangular Vectors method. This means that when generating an average space vector within a switching period, only the corner vectors belonging to the triangle where the reference vector is located is used. This means that for triangle A2 in sector 1 of the above figure, only the vectors (+00), (0--), (00-), (++0) and (+0-) can be used.

In most methods the harmonics of the bridge-leg voltages are the one focused on. However, as shown focus is on α- and β- voltages (stator oriented dq-system), while

this is the voltages creating harmonic currents. As pointed out earlier these voltages can be expressed as:

$$U_{sd}^{s}(t)=\frac{1}{3}\cdot\left(2\cdot U_{a0}(t)-U_{b0}(t)-U_{c0}(t)\right)=\frac{1}{3}\cdot\left(U_{ab}(t)-U_{ca}(t)\right)$$
$$U_{sq}^{s}(t)=\frac{1}{\sqrt{3}}\cdot\left(U_{b}(t)-U_{c}(t)\right)=\frac{1}{\sqrt{3}}\cdot\left(U_{b0}(t)-U_{c0}(t)\right)=\frac{U_{bc}(t)}{\sqrt{3}}$$

For a symmetrical load as a three phase motor, the zero system voltage is zero, which means:

$$U_{sd}^{s}(t)=\frac{1}{3}\cdot\left(2\cdot U_{a0}(t)-U_{b0}(t)-U_{c0}(t)\right)=U_{a}(t)$$
$$U_{sq}^{s}(t)=\frac{1}{\sqrt{3}}\cdot\left(U_{b}(t)-U_{c}(t)\right)=\frac{1}{\sqrt{3}}\cdot\left(U_{b0}(t)-U_{c0}(t)\right)=\frac{U_{bc}(t)}{\sqrt{3}}$$

A very important point for motor drives is thus to look at the harmonics in the phase voltage of phase a, and line voltage between phase b and c. For Active Rectifiers line all three line voltages should be investigated, while the load is not necessarily symmetrical.

4.2 Synchronous Modulation

Synchronous modulation is the technology where the switching of the semiconductor device is synchronized with the reference voltage at zero crossing. The approach is based on the restriction that the ratio number of pulse (N), which is the ratio of switching frequency to fundamental frequency (f_s/f), is kept an integer number. If asynchronous modulation is implemented for the lower value of modulation index (u_{st}), it will result in output current with sub harmonic current components [19]. However, synchronous modulation can be used for the low switching frequency. Synchronous modulation technology can be applied with carrier based and space vector based technology as well but in this report priority is given to the Programmed synchronous modulation which is the foundation for the presented Minimum loss synchronous PWM.

4.2.1 Programmed Synchronous Modulation

With the advancement in digital control system and processors, the trend of programmed modulation technique is increasing. The general approach for this is to perform the optimization offline by the help of computer and use the resulted switching patterns by storing the PWM data into the memory of the controller. In order to get the low switching frequency and also to make the symmetry of the system, the optimized switching angles are made to operate with synchronized PWM. The synchronization of angles with the fundamental component avoids the sub-harmonic content. One of the draw backs of this type of modulation is the need of computational complexity of switching angles for lower modulation range. As a result, the programmed modulation technique is always implemented with asynchronous modulation which can be either natural or regular sample method for the lower amplitude modulation.

Programmed modulation is implemented in two common methods, Harmonic elimination PWM and minimization of the performance index often referred as optimal PWM.

i. Harmonic Elimination PWM

In a balanced three phase inverter with symmetrical PWM, the even and multiple of third harmonics are not present. Harmonic elimination PWM technique is used to eliminate the lower order harmonics effectively from the 5^{th} harmonics. The number of harmonics level that can be eliminated depends upon the degree of freedom i.e. number of switching events. If α_1, α_2, α_3, ….. α_n, are the n switching events in the quarter symmetry (90° out of 360°) of the PWM, than harmonics component corresponding to any n harmonics can be eliminated to get the low WTHD0. For example when N =4 , four lowest non multiple of three harmonics can be eliminated to get the low ripple. This is fifth, seventh, eleventh and thirteenth as even and multiple of third harmonics are essentially not present. Solution of harmonic elimination can be done using the advanced software tools such as Matlab.

ii. Optimal PWM

The concept of optimal PWM comes from the performance index of optimization. The criterion function of the optimization is determined and corresponding switching patterns are generated using the advanced computer calculation tools for the minimum of the criterion function. This task is often cited as more advanced than selective

harmonic elimination method as the solution gets stuck into the local minima making it difficult to find the global minima. It this thesis, WTHD0 is taken as the criterion function of minimization and used together with the low switching frequency in order to get the minimum loss synchronous PWM. Details regarding the minimum loss synchronous PWM is given in Chapter 5.

Chapter 5

Minimum Loss Synchronous PWM

In this chapter, the objective function of minimization to get the Minimum Loss Synchronous PWM will be investigated. Since the relevant work is already done by Roy Nilsen of Wärtsilä, this chapter is taken from [19]. The understanding of this chapter is very important to understand the entire project work.

One of the fundamental papers of programmed modulation strategies is by Patel and Hoft [22]. They investigated half-bridge and full-bridge inverters, but these patterns are valid for one bridge leg in a three-phase inverter 2-level and 3-level respectively. This means that for a 3-level inverter the pattern for a full-bridge one-phase inverter should be investigated.

$$U_{dc} = U_{dc1} + U_{dc2} \qquad U_{dc1} = U_{dc2} \Rightarrow U_{dc1} = U_{dc2} = \frac{U_{dc}}{2}$$

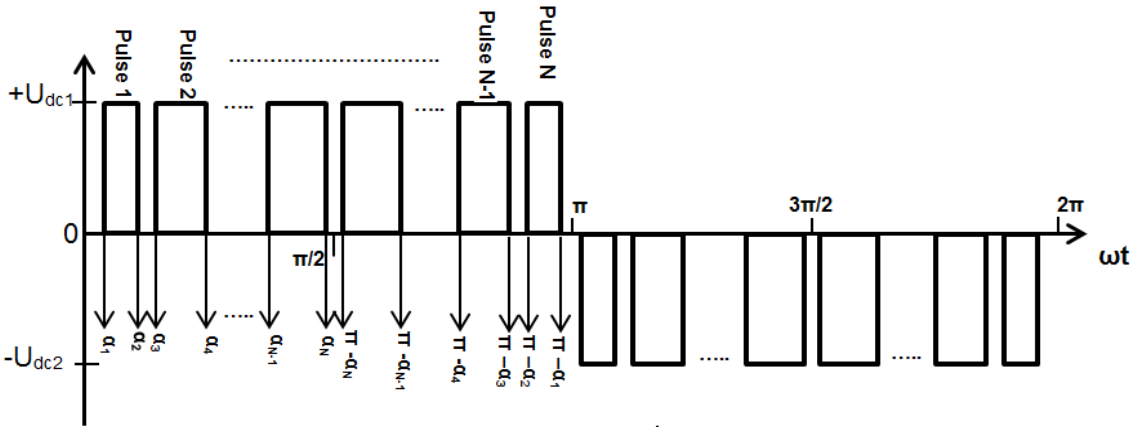


Figure 15: Synchronous PWM pattern (N = even number)

By inspection of Figure 15, it can be seen that the number of pulses per half period is N. In addition, the number of turn-on of upper switch is N and number of turn-off is N as well. The total number of commutations is thus 2*N per half period and 4*N per period.

In a complete three-phase inverter this becomes 12*N switching's. The average switching frequency over one period for the upper switch is thus:

$$f_{sw1,avg} = N \cdot f_s \qquad P_{sw1,avg} = N \cdot f_s \cdot \left(E_{on} + E_{off}\right)$$

When compared with a synchronized asymmetrical sampled carrier based modulation with the same number of pulses N per half period the equivalent ratio m_f becomes:

$$N = \begin{cases} 0.5 \cdot \left(m_f + 1\right) \\ 0.5 \cdot \left(m_f - 1\right) \end{cases} \qquad m_f = \frac{f_c}{f_s} = 3,9,15,21,27,.....$$

The Fourier series coefficients for the pattern will only have sine-term and odd number of harmonics due to Half Wave Symmetry (HWS) and the Quarter Wave Symmetry (QWS). The peak value of the h^{th} harmonic becomes [19]:

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

If N is odd then one will have the value $U_{dc1}/2$ at $\pi/2$. This means that voltage will not go to zero in the middle of a half period, but still be symmetrical. If there are N angles then there exists degrees of freedom to control the 1^{st} harmonic output and then to minimize the losses in the machine.

5.1 Criterion function

In drives applications the motor leakage inductance is limiting the harmonic currents. The weight total harmonic distortion factor for current is given in [40]:

$$THD_i = \frac{1}{\omega_1 L} \cdot \sqrt{\sum_{h=2}^{\infty} \left(\frac{\hat{U}_{a,h}}{h} \right)^2}$$

$$WTHD = \frac{THD_i}{\hat{I}_{a,1}} = \frac{\omega_1 L \cdot THD_i}{\hat{U}_{a0,1}} = \frac{1}{\hat{U}_{a0,1}} \cdot \sqrt{\sum_{h=2}^{\infty} \left(\frac{\hat{U}_{a0,h}}{h} \right)^2}$$

The inductance is the total leakage inductance in an induction motor, sub-transient inductance in a synchronous machine and stator inductance in a PM machine. When optimizing WTHD for an AC machine, equal harmonics as well as harmonic odd multiple of 3 can be skipped in the calculation. The current $I_{a,1}$ is the inrush current in the machine which is very large and $U_{a,1}$ is the harmonic voltage in the machine which includes the induced voltage in addition to the voltage across the inductances.

In general THD_i and WTHD is a measure of harmonic currents, and thus additional losses in the machine. The different factors could be weighted even more if taken into consideration frequency dependency of resistances and iron losses. This has been discussed in details in [8] and [40]. The conclusion in [8] is, however, that the optimal PWM pattern became almost the same as when not taken all this effects into consideration. This means that the expression above can be used for optimization.

Taken into considerations that no even harmonics and no harmonics multiple of three exist in the currents, the WTHD can be expressed as [40]:

$$WTHD = \frac{1}{\hat{U}_{a0,1}} \cdot \sqrt{\sum_{h=2}^{\infty} \left(\frac{\hat{U}_{a0,h}}{h} \right)^2} = \frac{1}{\hat{U}_{a0,1}} \cdot \sqrt{\sum_{i=1}^{\infty} \left(\left(\frac{\hat{U}_{a0,6i-1}}{6i-1} \right)^2 + \left(\frac{\hat{U}_{a0,6i+1}}{6i+1} \right)^2 \right)}$$

The optimization task is then, for each modulation index u_{st} , to find the set of N angles α_1 , …, α_N , which minimize the function above. The harmonic voltages can be calculated as:

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

$$h = 6 \cdot \mathrm{i} \pm 1, \qquad \mathrm{i} = 1,2,3,4....$$

To divide on the 1st harmonic component is a problem for low frequencies, while this voltage is proportional with frequency in a motor drive. It will thus at zero frequency be zero. It is no point to "scale" the WTHD with a frequency dependent 1st harmonic

component, while it is the relative values for different angles at this frequency which has to be compared. A scaled value with respect to dc-link voltage U_{dc1} is proposed in [3]:

$$WTHD0 = \frac{1}{U_{dc1}} \cdot \sqrt{\sum_{h=2}^{\infty}\left(\frac{U_{a0,h}}{h}\right)^2} = \sqrt{\sum_{h=2}^{\infty}\left(\frac{u_{a0,h}}{h}\right)^2} = \sqrt{\sum_{i=1}^{\infty}\left(\left(\frac{u_{a0,6i-1}}{6i-1}\right)^2 + \left(\frac{u_{a0,6i+1}}{6i+1}\right)^2\right)}$$

Where,

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

In this case the modulation index u_{st} is equal:

$$u_{st} = u_{a0,1} = \frac{4}{\pi} \cdot \sum_{k=1}^{N} (-1)^{k+1} \cdot \cos(\alpha_k) \qquad 0 < \alpha_1 < \alpha_2 ..... < \alpha_N < \frac{\pi}{2}$$

So far it has been discussed for bridge leg voltages. It will here be shown that this is sufficiently for minimization of harmonic h=6i±1, i=1,2,3,…etc. For a symmetrical three phase machine one obtains:

$$U_{sd}^{s}(t) = U_{sa} = \frac{1}{3} \cdot (2 \cdot U_{a0}(t) - U_{b0}(t) - U_{c0}(t)) = \frac{1}{3} \cdot (U_{ab}(t) - U_{ca}(t))$$
$$U_{sq}^{s}(t) = \frac{1}{\sqrt{3}} \cdot (U_b(t) - U_c(t)) = \frac{1}{\sqrt{3}} \cdot (U_{b0}(t) - U_{c0}(t)) = \frac{U_{bc}(t)}{\sqrt{3}}$$

This means that the phase voltage is equal the difference between two line voltages. The harmonics in the d- and q-voltages are:

$$U_{sa,h} = U_{sd,h}^{s}(t) = \frac{2 \cdot \hat{U}_{a0,h}}{3} \cdot \sin\left(\frac{h \cdot \pi}{3}\right) \cdot \left(2 \cdot \sin\left(h \cdot \left(\frac{\pi}{3} - \omega t\right)\right) + \sin(h \cdot (\pi - \omega t))\right)$$
$$U_{sq,h}^{s}(t) = \frac{2 \cdot \hat{U}_{a0,h}}{\sqrt{3}} \cdot \sin\left(\frac{h \cdot \pi}{3}\right) \cdot \sin(h \cdot (\pi - \omega t))$$

From the expressions above it can be seen that the amplitude of a component is modulated with a factor of $\sin(h\pi/3)$. This means that the same frequency components can be seen in the phase voltage as for bridge leg voltages, except for odd multiple of 3. The same dependency of modulation index is expected, except for the scaling with

sin(hπ/3) for the h[th] harmonic. Even though some addition "scaling" may occur due to the last two sine-terms in the expression for the phase voltage $U_{sa,h}$, the conclusion is that one should try to optimize the line voltage (~U_{sq}) of the machine. This seems to be in accordance with [22]. Taken into consideration that h=6*i±1 is the harmonics to be considered during optimization, one gets (h=5, 7, 11, 13, 17, 19, 23, 25, …):

$$\begin{aligned} U_{sa,6i\pm1} &= \pm\frac{\hat{U}_{a0,6i\pm1}}{\sqrt{3}}\cdot\left(2\cdot\sin\left(-(6i\pm1)\cdot\omega t - /+\frac{\pi}{3}\right)+\sin\left((6i\pm1)\cdot\omega t\right)\right) \\ &= \hat{U}_{a0,6i\pm1}\cdot\cos\left((6i\pm1)\cdot\omega t\right) \\ U^{s}_{sq,6i\pm1}(t) &= \pm\hat{U}_{a0,6i\pm1}\cdot\sin\left((6i\pm1)\cdot\omega t\right), \\ \underline{U}^{s}_{s,6i\pm1} &= \begin{cases}\hat{U}_{a0,6i-1}\cdot e^{-j(6i-1)\cdot\omega t} \\ \hat{U}_{a0,6i+1}\cdot e^{j(6i+1)\cdot\omega t}\end{cases} \qquad i=1,2,3,4,..... \end{aligned}$$

From equations above it is quite clear that both d,q- and phase components are proportional with the bridge leg voltage harmonics. Minimization of bridge-leg harmonics means minimization of the same harmonics in the dq-system. The expression WTHD0 above can thus be used for optimization.

The Matlab-routine fmincon which find the minimum of a constrained nonlinear multivariable function is used for optimization. In general several local optimal solutions exist; the challenge is to find the correct initial conditions. The criterion minimized is WTHD0 and constrains are:

$$ceq(\underline{\alpha}) = u_{st} - \frac{4}{\pi}\cdot\sum_{k=1}^{N}(-1)^{k+1}\cdot\cos(\alpha_k) \qquad 0<\alpha_1<\alpha_2.....<\alpha_N<\frac{\pi}{2}$$

These constrains can be split in two; one for the angles and one for the 1[st] harmonic amplitude:

$$\begin{aligned} &A\cdot\underline{x}\le\underline{b} \\ &\underline{x}^T=\left[\alpha_1 \quad \alpha_2 \quad .... \quad \alpha_N\right] \\ &\underline{b}^T=\left[-\Delta\alpha_{\min} \quad -\Delta\alpha_{\min} \quad ..... \quad -\frac{\pi}{2}-\Delta\alpha_{\min}\right] \\ &ceq(\underline{x})=0 \end{aligned}$$

The differences between angles are set to minimum angles due to minimum on/off-times. If the angles can be equal, some of the pulses can be removed by the optimization algorithm. The matrix A has the form:

$$A = \begin{bmatrix} -1 & 0 & 0 & ..... & 0 & 0 & 0 \\ 1 & -1 & 0 & ..... & 0 & 0 & 0 \\ 0 & 1 & -1 & ..... & 0 & 0 & 0 \\ 0 & 0 & 1 & ..... & 0 & 0 & 0 \\ .... & .... & ..... & ..... & .... & .... & .... \\ 0 & 0 & 0 & ..... & -1 & 0 & 0 \\ 0 & 0 & 0 & ..... & 1 & -1 & 0 \\ 0 & 0 & 0 & ..... & 0 & 0 & -1 \end{bmatrix}$$

5.2 Choice of N as the function of Modulation index

Minimization of the loss involves the minimization of WTHD0 which is again dependent upon the switching angles. As we increase the number of switching angles, the freedom of harmonic elimination increases. However, also the number of possible solution increases rapidly. The number of solution for the minimum-loss becomes even more complicated than the harmonic elimination.

Switching frequency (f_s) of high-power systems is limited by the loss in the IGBT's. This condition forces us to limit the number of switches as we have,

$$N = \frac{f_s}{f_1} \qquad (1)$$

Where, N is the number of switching angles per $\pi/2$ period as shown in figure 15.

But for an electrical machine with rotating magnetic field, modulation index (u_{st}) is proportional to fundamental frequency in order to maintain the stator flux at a constant level [41].

$$u_{st} = kf_1$$

$$= k.\frac{f_s}{N}$$

$$= k.\frac{f_{1max}}{N}\frac{f_s}{f_{1max}} \quad (2)$$

Fundamental frequency f_1 is maximum when modulation index is 1, hence

$$u_{st} = kf_1\Big|_{f_1=f_{1max}} = 1$$

$$\text{i.e } k = \frac{1}{f_{1max}} \quad (3)$$

Now, from equation (2) and (3)

$$N = \frac{f_{sw}}{f_{1max}}.\frac{1}{u_{st}} \quad (4)$$

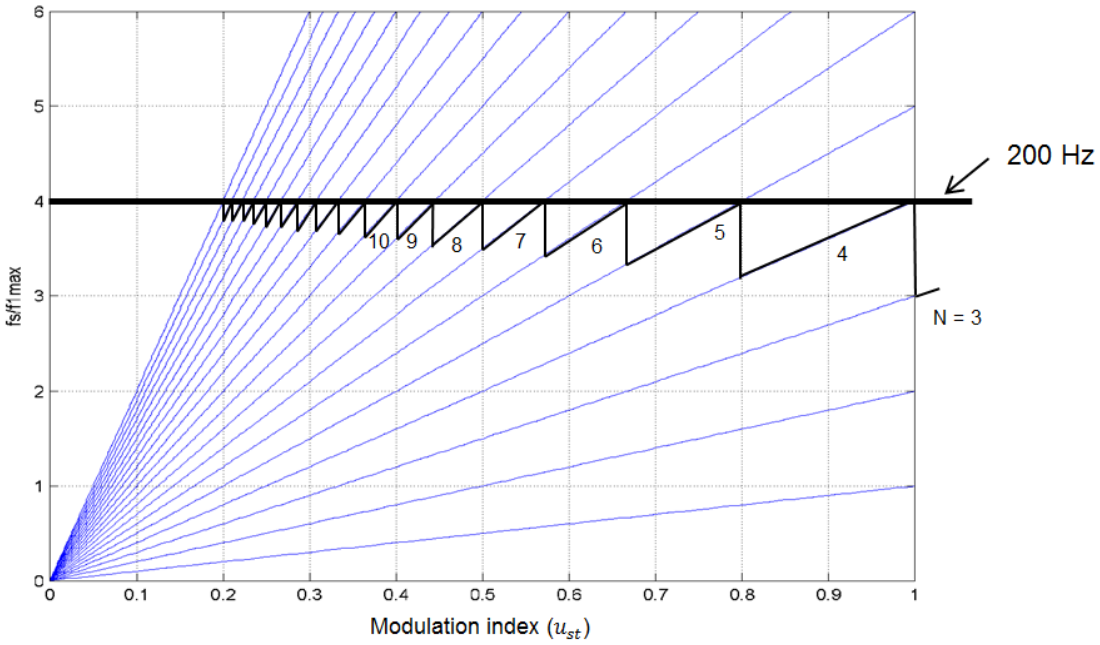


Figure 16: Acceptable number of switching angles versus modulation index (f=50 Hz)

Taking switching frequency as 200 Hz and fundamental frequency as 50 Hz the above characteristics is seen. It shows that, higher number of pulses at low modulation index is acceptable and number of pulses to be used reduces with the increase of modulation index. Also, the higher value of fundamental frequency reduces the number of pulses that can be utilized hence reducing the freedom of harmonic loss elimination.

When modulation index (u_{st}) reduces from the unity, number of switching angles (N) increases. When N is large enough, optimization procedure is not very effective as there is no significant achievement over space vector PWM.

As the fundamental frequency increases to 60 Hz, either switching frequency should be increased or the number of switching angles should be reduced as f_s/f_{1max} reduces at unity modulation index.

When the fundamental frequency is increased to 100 Hz at switching frequency of 200 Hz and modulation index of unity, only two switching angles can be applied.

For f_{sw} = 200 Hz and f_{1max} = 50 Hz

$$N = 4 . \frac{1}{u_{st}}$$

As such, programed PWM with switching angles varied over the variation of modulation index is preferred. This type of switching angle variation can be referred as feed forward PWM.

Chapter 6

Neutral Point Balancing

In this chapter, the problem of DC link balancing with the three level inverters will be introduced and the methods to negotiate it will be discussed. Methods of DC link balancing for three level inverters using carrier wave modulation, space vector modulation and programmed modulation will be introduced. This chapter is updated from [17].

Neutral-point (NP) voltage variation, due the voltage splitting between the capacitors, is a major problem with the NPC inverter. The system must overcome such NP voltage variation because they can cause start-up failure, CMV and over-voltage across the switching devices.

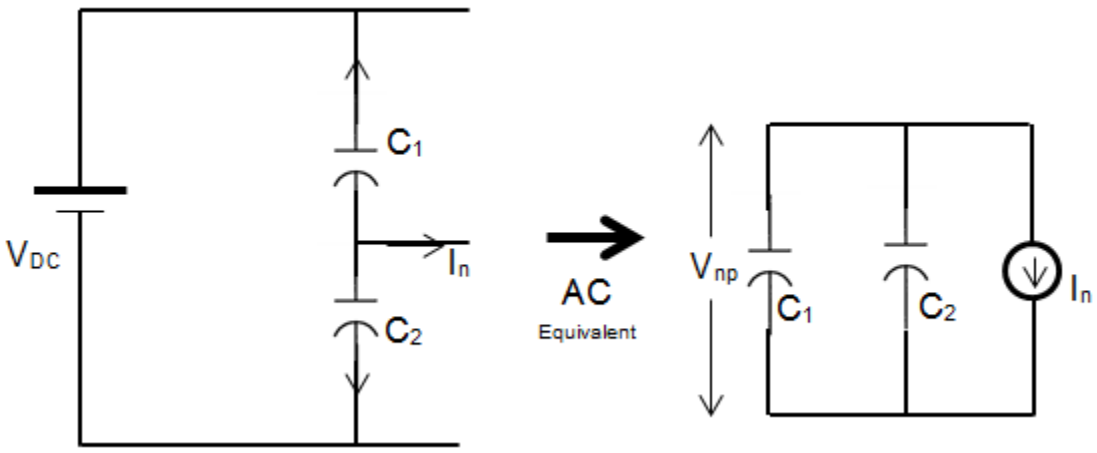


Figure 17: NP voltage in 3 levels Converter

The above figure on the right represents the equivalent circuit for the AC model (small signal model) of neutral point voltage V_{np} and neutral current I_n. The neutral point voltage is

$$V_{np} = \frac{1}{C_1 + C_2}\int I_n(t)dt$$

$$= \frac{1}{2C}\int I_n(t)dt \qquad \text{Since, } C_1 = C_2 = C$$

Since I_n is not zero for the system, V_{np} cannot be made zero without some techniques. Based on above equation we can either reduce I_n to zero or make C very high to minimize V_{np} close to zero. The first method uses additional topologies added to the system and modification on PWM while the second method can be used on higher frequency only because NP voltage variation increases widely for the low frequency making additional capacitance bank compulsory.

Although, various methods are purposed to maintain the Neutral-point voltage, they can be categorized into two types.

1. Carrier Based PWM [42]
2. Space Vector Modulation based PWM [16]

Carrier based PWM modulations methods require output zero sequence voltage in modulation schemes with closed loop control while Space vector modulations methods require some patterns of small vectors.

Both of the technologies are identified as not very satisfactory. With the carrier-based PWM, modulation distortion is a major problem. There will be additional problems if the system consists of another control loop as well. With SVPWM, NP voltage control is not available when Common-mode voltage cancellation method is used as small vectors that are used for the process is already removed. In [43] a method of selecting the appropriate redundant inverter sub-bridge for balancing the neutral-point is explained. The [44] suggests one of the possible ways to solve the problem in the programmed PWM, however the number of switching increases proportionally with the method and it is not recommended using such techniques for the concern where the loss is being optimized. One of the early papers describing the neutral point balancing on the programmed PWM method is [45]. It suggests the alteration of the PWM switching function. The paper claims to produce total immunity to the dc-link voltage ripple. However, it is found that at higher inverter output voltages a possibility of insufficient margin to incorporate higher instantaneous values of the modulating signal to be existent. Thus, the application of the technique can be limited. In [8], a method of neutral point balancing is given however it is found to be applicable for the multilevel inverter with 5 or more levels.

To be implemented in the minimum loss synchronous PWM, we need methods for the low amplitude modulation Space vector and another for the programmed modulation.

6.1 Neutral Point Balancing with Low Amplitude

For the low amplitude region, Space vector modulation is used. Not all vectors in the space vector affect the neutral point balance.

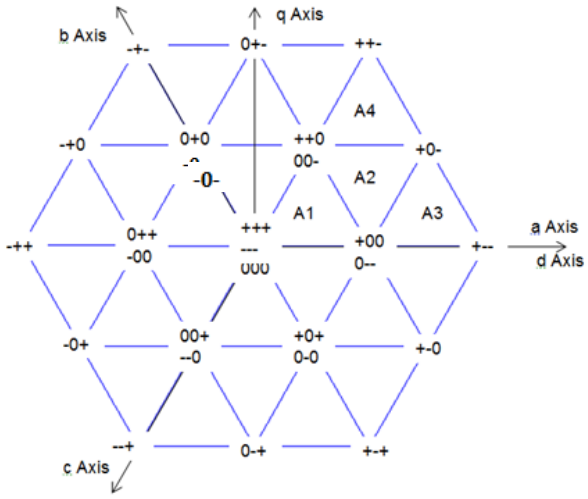


Figure 18: Space vector Modulation for 3 level converters

The space vectors can be divided into the category of zero vectors, small vectors, medium vectors and large vectors. Zero vectors (+++, 000, ---) and large vectors (+--, ++-, -+-, -++, --+, +-+) do not affect the NP balance. Positive small vectors (0--, ++0, -0-, 0++, --0, +0+), negative small vectors (+00, 00-, 0+0, -00, 00+, 0-0) and medium vectors (+0-, 0+-, -0+, 0-+, +-0) are the vectors that affect the NP balancing. However, for the low amplitude modulation only small vectors and zero vectors are used and we are interested in cancelation of the effects caused by the small vectors. The neutral point current I_n can be equal to twice the capacitor current.

$$i_{np} = 2\, i_{c1} = -2 i_{c2}$$

Where, i_{np} is NP current, i_{c1} and i_{c2} are the current through capacitor C_1 and C_2

The details regarding the NP balance using the small vectors is given in []. The redundancies of the small vectors are used to make the NP balance. This is done by controlling the neutral current with variation in the duration of positive and negative vectors. For the region of sector A1 in figure 6.1, the switching's can be made of the order as shown in figure 6.2, where the duty cycles for each state can be calculated as the function of control parameters f_1 and f_2 by varying which neutral point voltage/current is controller without changing the overall duty cycle and switching frequency.

The duty cycle of small vectors are broken down into two parts, which are

$$d_{s1} = d_{s1p} + d_{s1n} \; and \; d_{s2} = d_{s2p} + d_{s2n}$$

Where, p and n distinguishes the duty cycle positive and negative

Now,

$$d_{s1p} = f_1 . d_1$$

$$d_{s1n} = (1 - f_1) . d_1$$

$$d_{s2p} = f_2 . d_2$$

$$d_{s2n} = (1 - f_2) . d_2$$

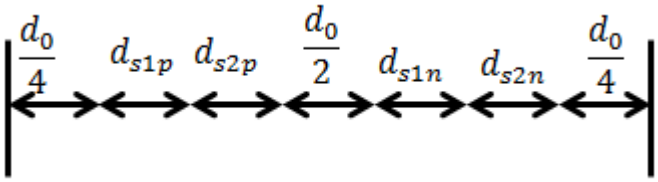


Figure 19: Arrangement of duty cycle with NP balance for Sector A1 in Figure 18

6.2 Neutral Point Balancing with PSM

Neutral point balancing in programmed modulation is different concept as compared to the ones for the space vector modulation. The flexibility of duty cycle selection is

removed with the optimal switching patterns generated. There is more advanced method that can be used for the purposes. One of such method is mentioned below but is not implemented in this thesis.

A method of neutral point balancing with a time-offset estimation scheme is discussed in [46]. Explanation regarding the method is described here as it can be one of the good alternatives to employ for the Neutral point Balancing in the Programmed Synchronous PWM method. The neutral point is balanced by adding time-offset by a time-offset estimation method which can be implemented without the use of additional hardware and complex analysis. The small vectors present in the NPC inverter vary the NP voltage. N-type small vector decreases the NP voltage while the P-type small vector increases. Hence, by balancing the N-type and P-type duration if NP unbalance occurred, we can balance the NP. This duration can be balanced by adding time-offset to the turn-on times of the switches.

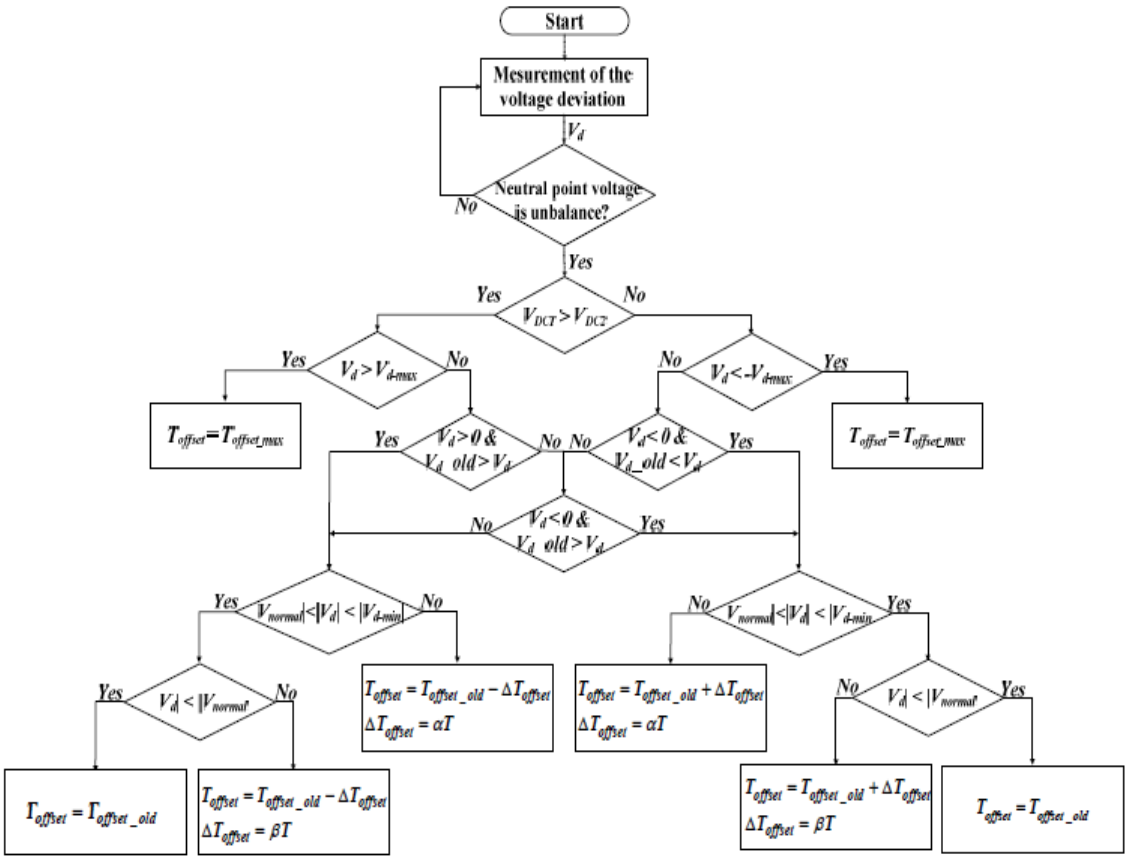


Figure 20: Block diagram for Neutral point balancing [46]

In [47] which is a similar technology as used in this system, Minimum dc-link ripple is suggested to be obtained by minimizing the integration of the modulation index as a function of sinusoidal waveform for the period of half wave symmetry. This solution can be directly implemented to the system designed by adding the additional function while generating the optimal pulse for the given range of pulse number. Hence for this thesis work the technology used in [47] is used to determine the DC link. The method of linearization is used where the changes from one amplitude level to another amplitude is made without much transient.

Chapter 7

Simulation and Results

In this chapter, the simulations that are carried out will be described along with the results. The simulation is carried out according the requirement that the lower modulation values require the asynchronous modulation while the upper modulation values require synchronous modulation.

It is observed in Chapter 5 that Asynchronous modulation is required for lower modulation index values and Programmed Synchronous Modulation can be carried only above certain value of modulation index. This is shown in the figure below and 0.25 is chosen as the value of transition from one method to another method making hybrid modulation.

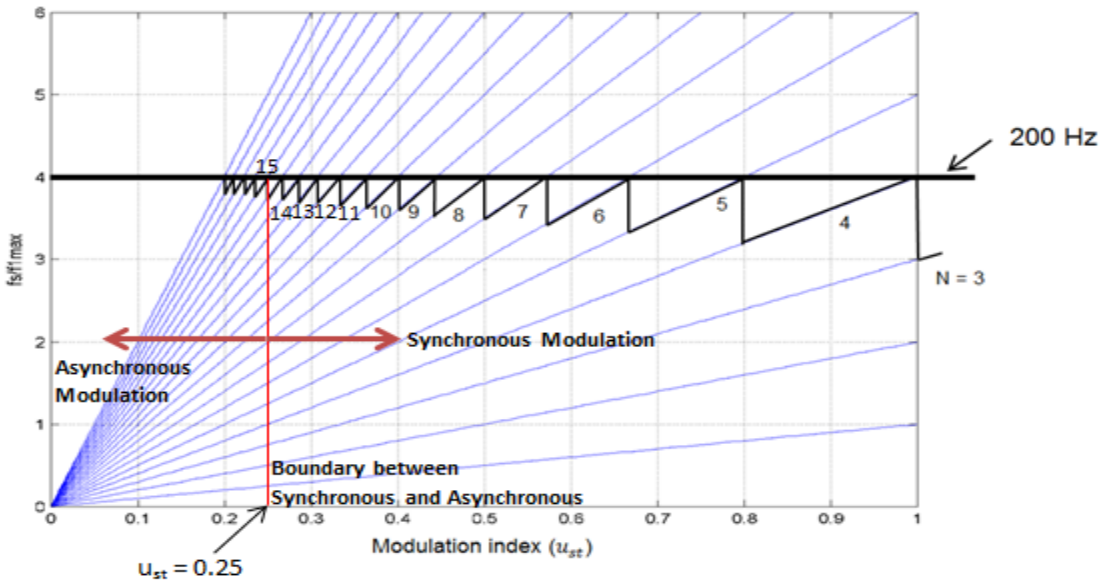


Figure 21: Modulation index vs. number of pulses

7.1 Space Vector Modulation Technology

Based on [48], Matlab function is developed and tested to produce the optimal values. The result shows that correct values are identified. To do this, the hexagon of space

vector is divided into sectors and blocks. There exist six sectors and four blocks in each of the sector.

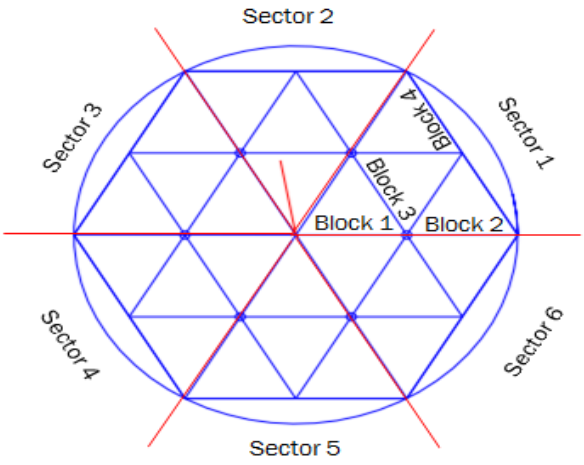


Figure 22: Division of Space vector into sectors and blocks

To identify the switching sequence and timing the algorithm presented in [48] is set to follow, which produces the sequence of the switching states and time associated to each state. Each sequence is characterized by four states and time for the four states is limited by the switching frequency. Out of these four states, one is a redundant state which is characterized by V_{01} and V_{02}, as shown in Figure 23.

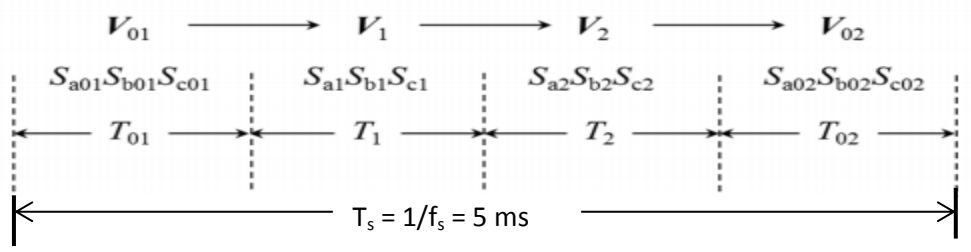


Figure 23: Switching Sequence and Duty Cycle

For example, during the simulation with modulation index 0.2, angle 170°, Currents (I_a,I_b, I_c) = (1, -0.5, -0.5), DC link voltage 3 KV, Mid-point Voltage 1 KV and Capacitance of 20 mH, the following results were obtained

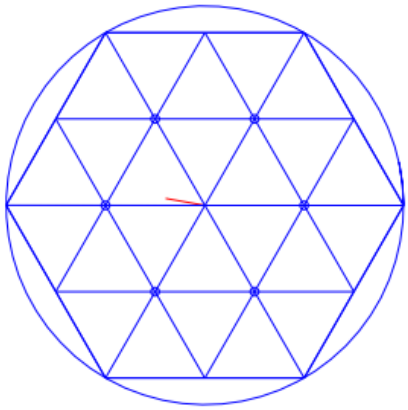


Figure 24: Identification of Sector and Block based on input

The program correctly identified the reference to be in sector 3 and block 1. Further, it identified the correct sequence and the time required for each state.

Further the Simulink Model is made to test the validity of the modulation and it is observed that the current harmonics to be directly related to the switching. With generally high frequency the solution had low ripple current while had high ripple with low switching frequency. A typical result with the modulation index of 0.2 and switching frequency of 5 KHz using the Simulink is shown below.

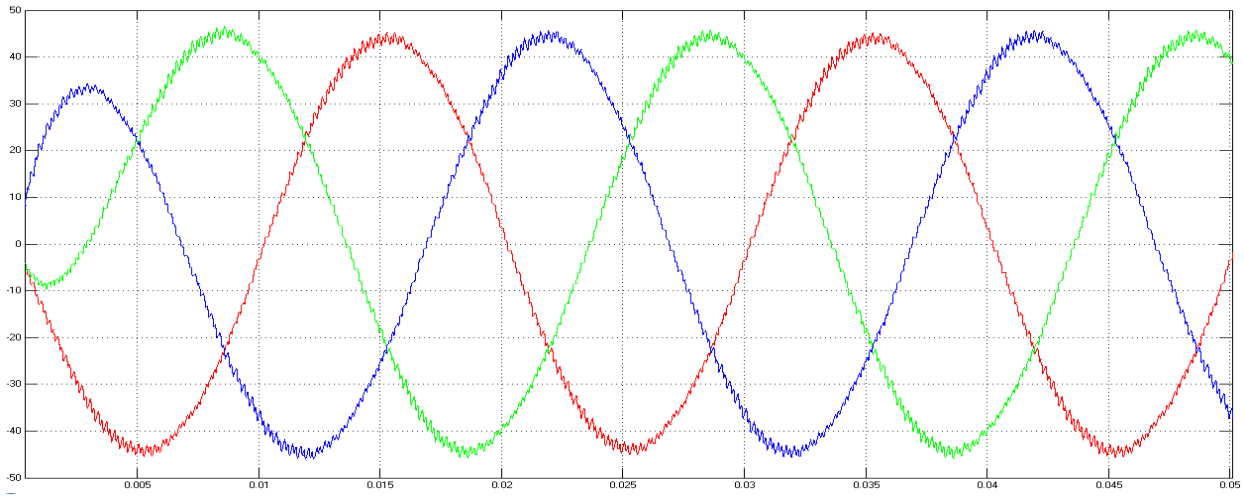


Figure 25: Low ripple output current with SVPWM (u_{st} = 0.2, f_{sw} = 5 KHz)

The Simulink model for the above task is given in Appendix 1. The three phase voltage waveform is achieved as expected.

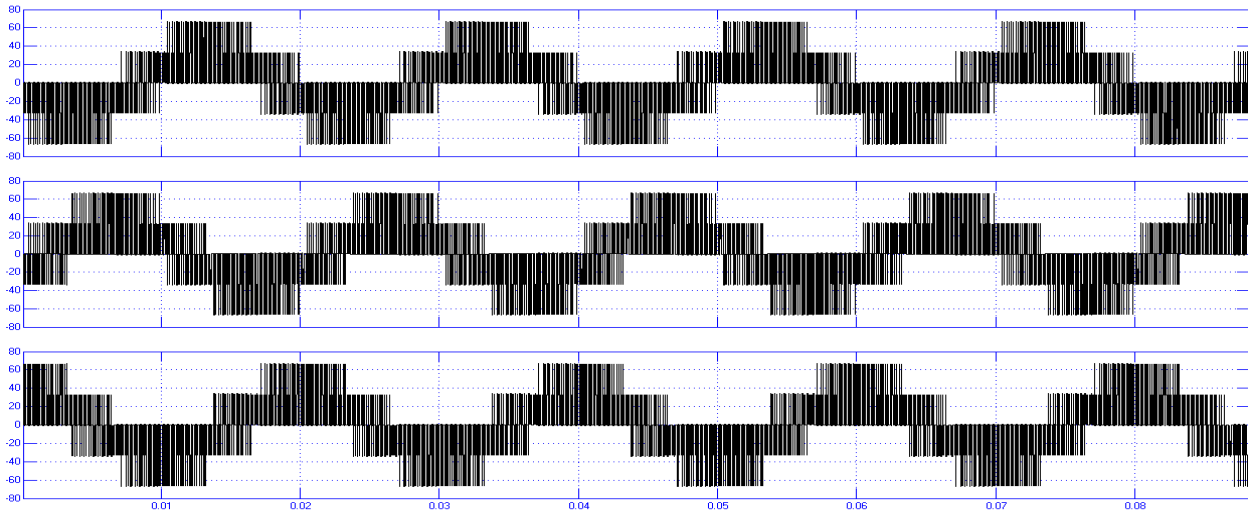


Figure 26: Three phase output Voltage with SVPWM (u_{st} = 0.2, f_{sw} = 5 KHz)

7.2 Programmed Synchronous Modulation

Programmed synchronous modulation is carried out based on the pulses classified as even number of pulses per half period and odd number of pulses per half period.

7.2.1 N equal even number

With equal number of N, optimization has been performed for even numbers of N pulses per half period. The modulation index u_{st} is varied from 0.05 to 1.23 and the optimization criterion is applied in order to minimize WTHD0. This numerical optimization algorithm performs global optimization of WTHD0, however; it consists of number of local optima where it is prone to get trapped. To overcome such effects several initial angles are tested simultaneously in order to get the minimization close to the lowest value. The algorithm is set to check the multiple angles and determine the lowest of the possible optima among the different set of angles generated with different initial angles. The solution space should be chosen carefully to expand the probability of including all the possible local optima. Throughout the calculation and description

the operation with switching frequency of 200 Hz and fundamental frequency of 50 Hz is assumed.

As stated earlier, in order to negotiate the local optima effects, several starting angles must be tested.

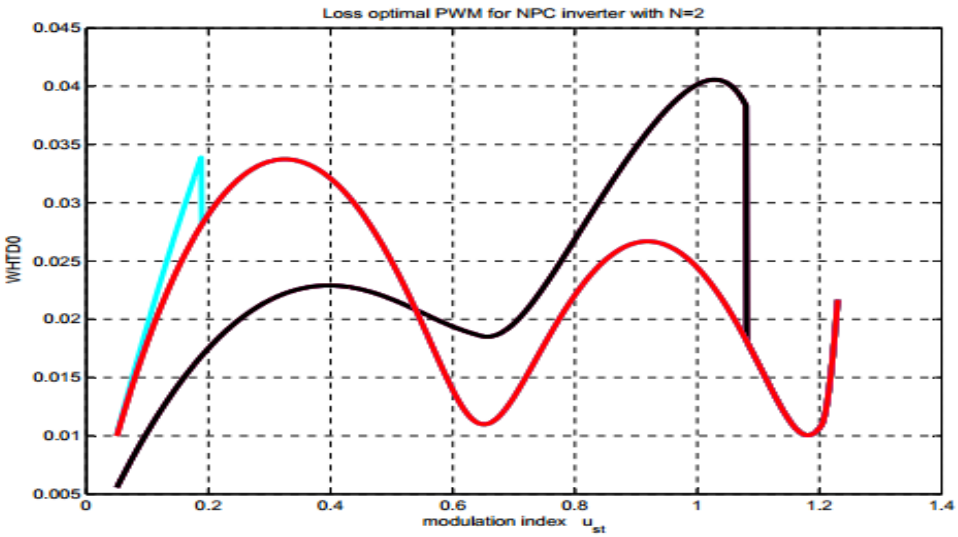


Figure 27: WTHD0 for N = 2 and different angles operated

Each starting angle follows with the sinusoidal WTHD0 output and switches from one wave to another wave to form the minimum loss PWM in the local region. However, it is unable to identify all the wave patterns and there is a chance of missing the optimum region. With the application of range of starting angles the optimum region is closely covered. When changing the angles at different regions modulation index, the observation is almost the same concluding it to be very much close to the minimum loss PWM. It is also concluded that, starting angles very close to each other will lead to the same optimization pattern. Figure 27 for N =2 shows that only 3 different optimization of WTHD0 pattern even though 7 different combination of starting angles used. This is due to the fact that optimization patterns in the local region soon get merged to form one particular pattern.

Generally for N = 2, WTHD0 is high compared to higher angles, however due to the limitation of freedom of number of pulses for higher modulation index, the overmodulation region with $u_{st} > 1$ can be the area of interest provided that it gives lower WTHD0 compared to N =3.

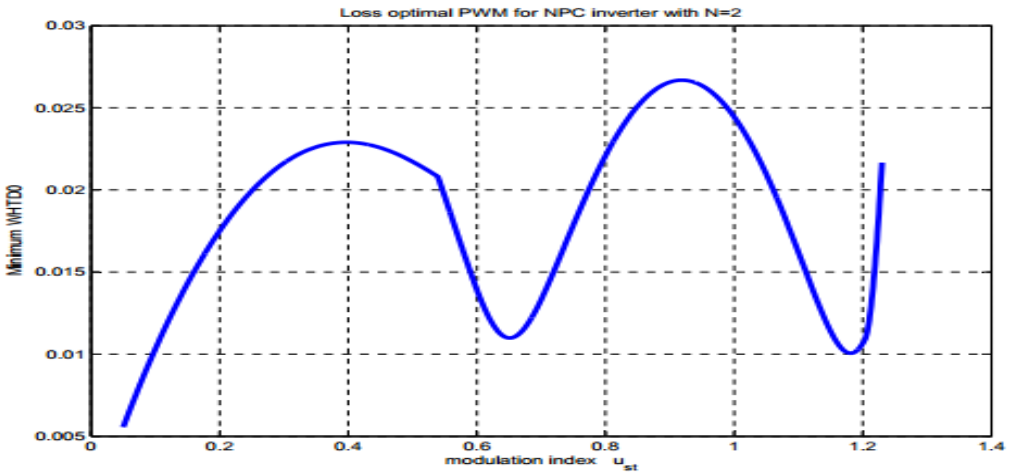


Figure 28: Selected minimum WTHD0 for N =2

Though multiple starting angles were used to figure out the patterns they give. Only the region with minimum WTHD0 for any particular value of modulation index is of interest. The algorithm is set to operate to produce the lowest of all the obtained patterns and this lowest WTHD0 pattern is used to identify the corresponding angles of minimum loss PWM.

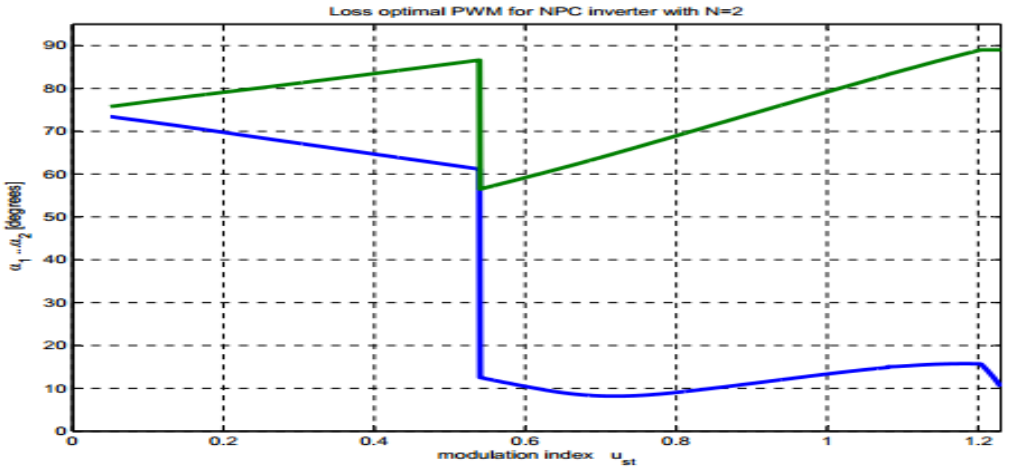


Figure 29: Optimal PWM angles for N = 2

For the most of the regions the angles are quite smooth however at about modulation index of 0.54 there is a drastic change is the angles. This is because the WTHD0 pattern at this instant is being shifted from one set of combination to the other. At overmodulation higher than 1.2, one of the angle is observed to saturate at 90º.

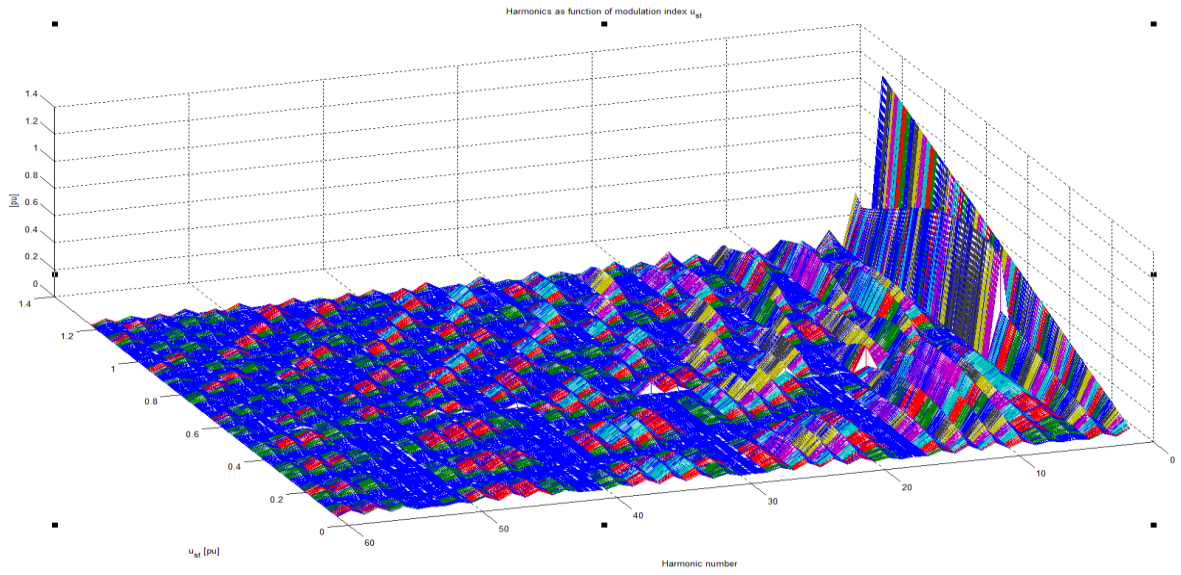


Figure 30: Harmonic spectrum for N =2

The presence of significant amount of multiple of 3rd order harmonics is observed which theoretically should be zero.

The similar method is used to obtain the solution of minimal WTHD0 for different value of even number of N. It is observed that the complexity of calculation increases with the increase in the number of angles. Refer appendix 3 for the patterns obtained with different value of even number of N.

After selecting the minimum WTHD0, reverse operation is applied for getting the angles of minimum loss PWM i.e. for each value of minimum WTHD0, corresponding angles are identified. If the angles are not smooth than it will lead to the DC link balancing problem hence techniques are required to overcome this effect.

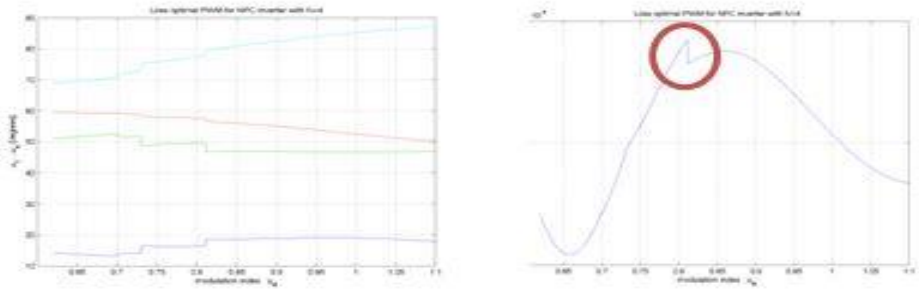


Figure 31: Angles with forced optimization changes (left), WTHD0 with forced optimization changes (right)

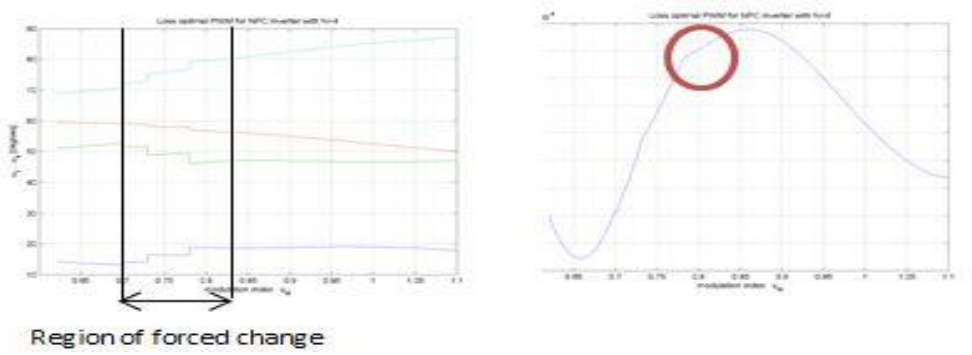


Figure 32: Angles with forced optimization changes (left), WTHD0 with forced optimization changes (right)

It is identified that the regions to which the angles should be smooth are dependent factors of interest. A closer look such forcing over the region of interest, the above result is seen while using N=4. The circled region shows that we can further enhance the region by forcing changes in the optimization manually.

7.2.2 N equal odd number

Similarly optimization is performed for odd numbers of N pulses per half period as well. The modulation index u_{st} is varied from 0.05 to 1.23 and the optimization criterion is applied in order to minimize WTHD0. Several initial angles are tested simultaneously in order to get the minimization close to the lowest value and avoid local optima. The algorithm is set to check the multiple angles and determine the lowest of the possible optima among the different set of angles generated with different initial angles.

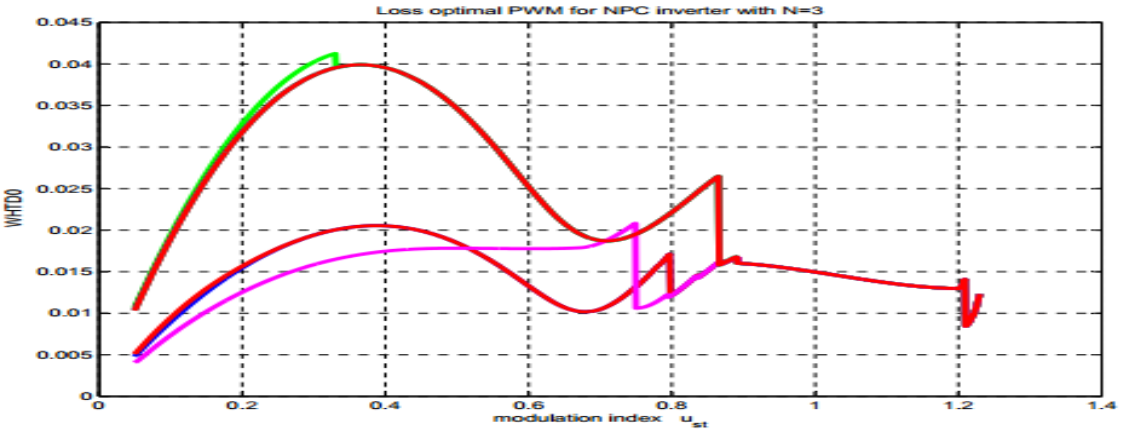


Figure 33: WTHD0 for N = 3 and different angles operated

The solution space should be chosen carefully to expand the probability of including all the possible local optima. With the application of range of starting angles the optimum region is closely covered as in case explained for even number of pulses. When changing the angles at different regions modulation index, the observation is almost the same concluding it to be close to the minimum loss PWM.

The optimization for odd number of angles, i.e. N =3 optimize WTHD0 to better as compared to even number N=2, however; is significantly worse than the case of 4 number of pulses. As the modulation index is higher than 1, the number of pulses allowed would be below four and we are interested in 3 angels rather than 2 angles as shown by the simulation that the WHTD0 with 3 number of pulses is better than 2 pulses. Also it would be obvious choice as we consider that we don't want the switching frequency well below 200 Hz.

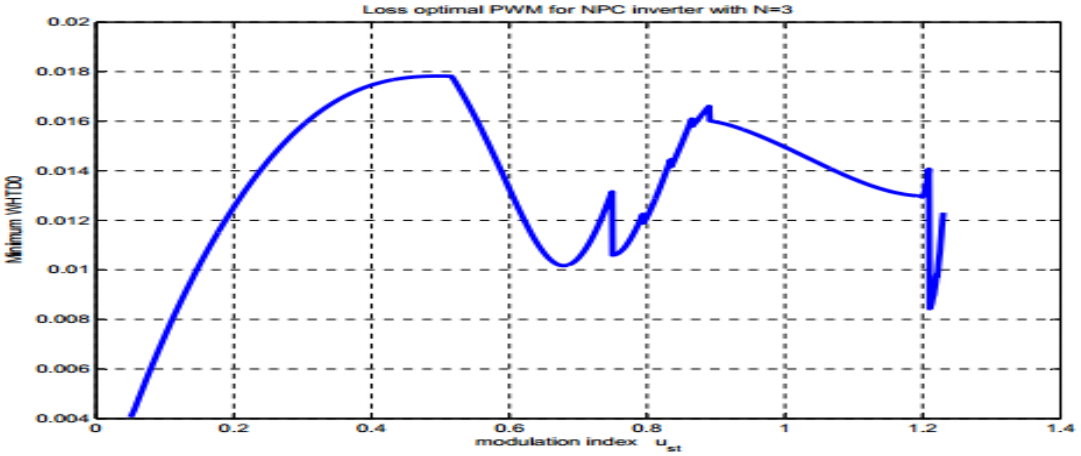


Figure 34: Selected minimum WTHD0 for N =3

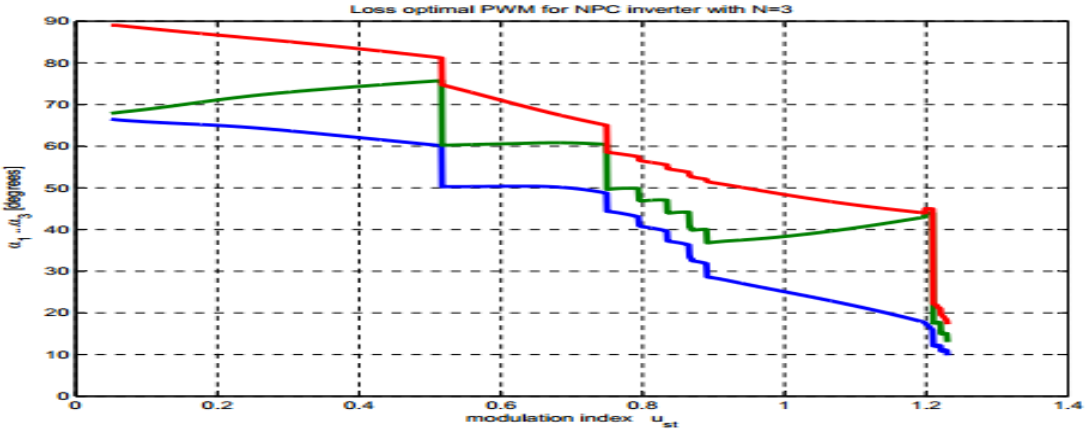


Figure 35: Optimal PWM angles for N = 3

Refer appendix 3 for patterns and WTHD0 level with different odd number of pulses applied into the system.

7.2.3 Comparison of Odd and Even number of pulses

It is observed that the even number of pulses give lower overall WTHD0 as compared to the odd number of pulses refer to the Appendix 3.

7.2.4 Implemented Programmed Modulation

According to the relation,

$$N = \frac{f_{sw}}{f_{1max}} . \frac{1}{u_{st}}$$

Large number of pulses can be implemented at the lower modulation index to obtain better WTHD0. However during the simulation it is obtained that, the optimization is not effective at low modulation index because of the following reasons.

1. For higher number of N, the computation of correct switching angles becomes very complicated that it becomes almost impossible to avoid local minima.

2. There is no significant achievement over the Space vector PWM [3].

The complexity of the system is largely increased with changing the number of switches in accordance to the modulation index. However, when the system is properly tuned with the switching angles, the system can be reduced to minimum WTHD0 and that is too only at 200 Hz switching frequency. But again, if we use only few pulses with lower modulation index than it will make very low switching frequency. This will produce problem with DC link balance and starting problem. As such, asynchronous space vector modulation is the preferred method used for lower modulation index until 0.25 and for higher modulation index programmed synchronous modulation is used.

It has been a common practice to make the system store the values for the programmed modulation in the memory and fetch the values for the modulation. But storing the data's in the memory and fetching it makes the post optimization is difficult, have chance of missing lots of values in between and lot of processing power of the chip is wasted in redundant cycles of reaching the correct values of angles for the given modulation index. It is also prone to lot of technical flaws and errors. As such, a method

of calculations in the processor itself is chosen to be the best alternative while the Natural point balancing with the linearization is implemented with the smooth transition from one modulation index value to another. All the sets of pulses according to modulation index is first calculated and then linearized for its working values to implement it in the system.

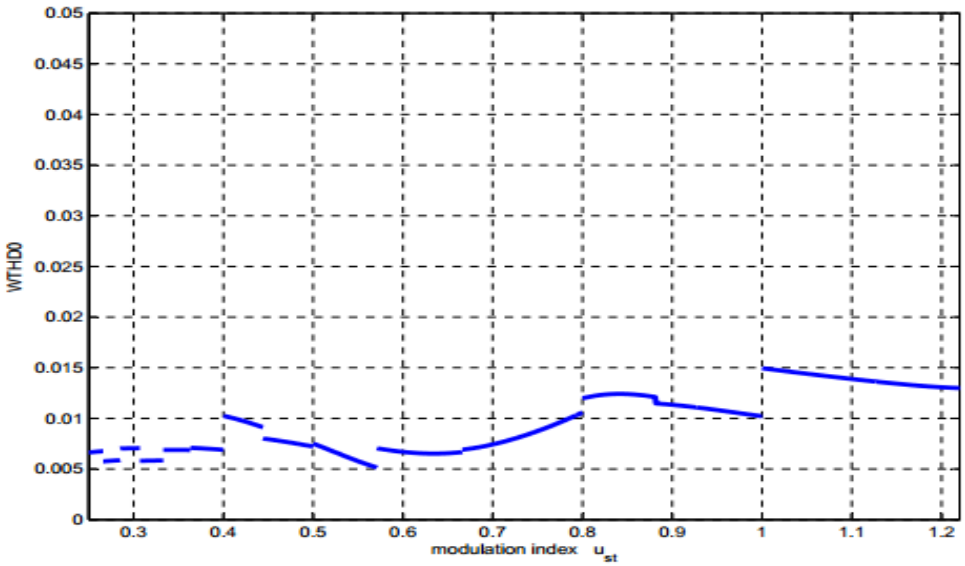


Figure 36: WTHD0 for Programmed Synchronous Modulation with DC balancing

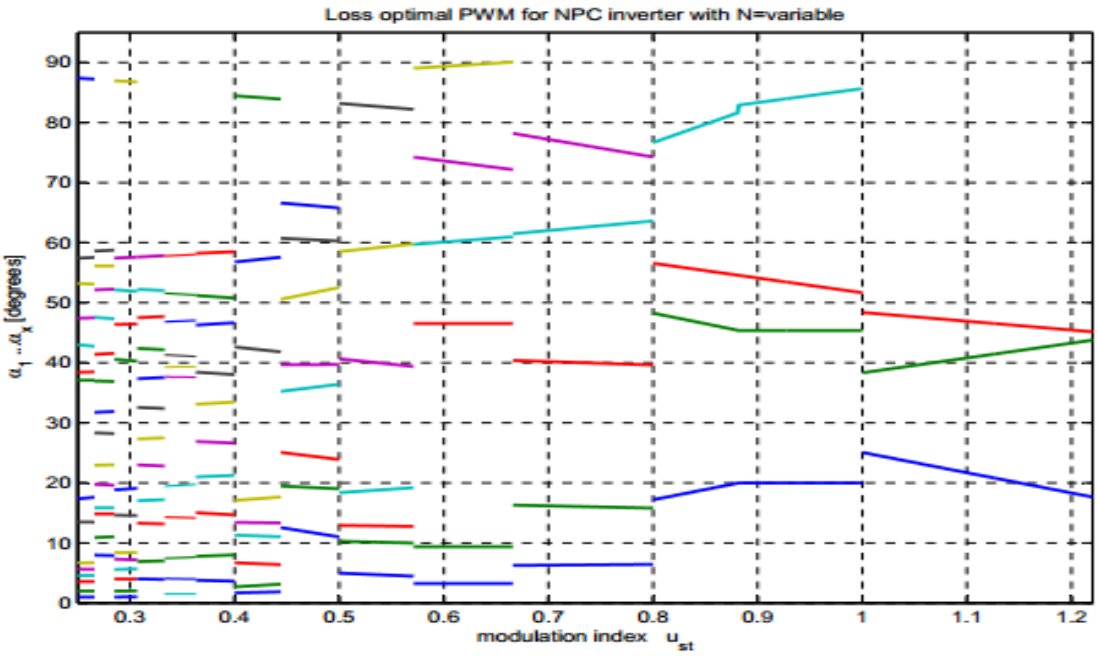


Figure 37: Pulses for Programmed Synchronous Modulation with DC balancing

7.2.5 Comparison of the result with existing results

In [35], R. Lund has compared the different existing techniques of modulation. But the factor of minimization he used for the comparison is $\frac{\Delta I_{N,rms}^2}{\Delta i_h^2}$ and hence it is required to convert our measurement into the same unit as he has used. By using the scaling, we get

$$\frac{\Delta\, I_{N,rms}^2}{\Delta\, i_h^2} = \left(\frac{\sqrt{2}}{\pi} . \frac{f_s}{f_o}\right)^2 . WTHDO^2$$

$$= \left(\frac{4.\sqrt{2}}{\pi}\right)^2 . WTHDO^2$$

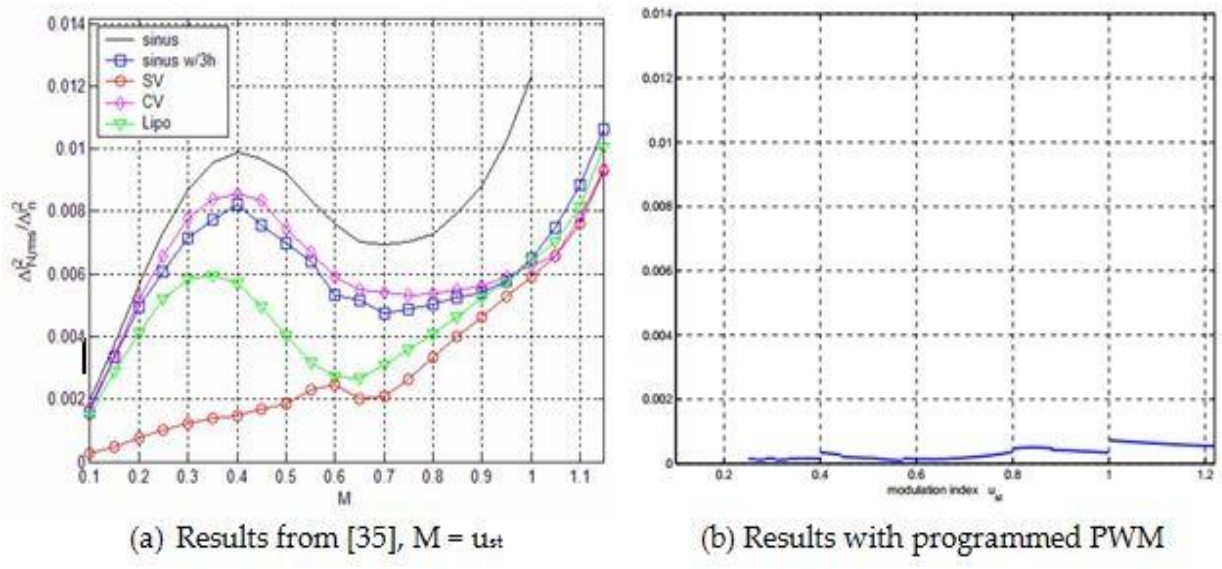


Figure 38: Comparison of different PWM techniques with Programmed PWM

This result confirms the superiority of the programmed PWM technique that is calculated above over the current existent techniques. It is to be noted that the solution of programmed synchronous modulation is stating in the above figure from 0.25 instead of 0.1 as in [35] because the technique will be used only for the modulation index above 0.25.

In [34] an advanced programmed synchronous modulation has been applied but the minimization factor used is distortion factor. Since, it is also implementing the Synchronous programmed modulation we can expect a bit similar result however not

exact. The other main difference between the two is the factor chosen for the minimization, which in [34] is distortion factor as compared to WTHD0. To make the comparison with [47] we must first convert the unit from WTHD0 to distortion factor, which is done as below

From [20] we have,

$$d = \frac{i_h}{i_{h,six\ step}} \dots\dots\dots\dots\dots\dots (1)$$

Where,

$$i_h = \frac{1}{2\pi f_1 . l_\sigma} . \sqrt{\sum_h \left(\frac{u_h}{h}\right)^2} \dots\dots\dots\dots\dots\dots\dots (2)\ and$$

$$i_{h,six\ step} = \frac{0.046\ (L-1)U_{dc}}{2\pi f_1 . l_\sigma * \pi} \dots\dots\dots\dots\dots (3)$$

From, (1), (2) and (3)

$$d = \frac{\frac{1}{2\pi f_1 . l_\sigma} . \sqrt{\sum_h \left(\frac{u_h}{h}\right)^2}}{\frac{0.046\ (L-1)U_{dc}}{2\pi f_1 . l_\sigma * \pi}}$$

$$= \frac{\sqrt{\sum_h \left(\frac{u_h}{h}\right)^2}}{\frac{0.046 * 2 * U_{dc}}{\pi}} \qquad [Since, L = 3, for\ 3\ level\ inverter]$$

$$= \frac{WTHD0}{\frac{0.046 * 2 * 2}{\pi}}$$

Where,

$$WTHD0 = \frac{1}{U_{dc1}} . \sqrt{\sum_h \left(\frac{u_h}{h}\right)^2}\ and\ U_{dc1} = \frac{U_{dc}}{2}$$

Hence,

$$d = WTHD0.\frac{\pi}{0.184}$$

Since the scaling of the modulation index is also different in [47], we need to scale it in accordance to [32].

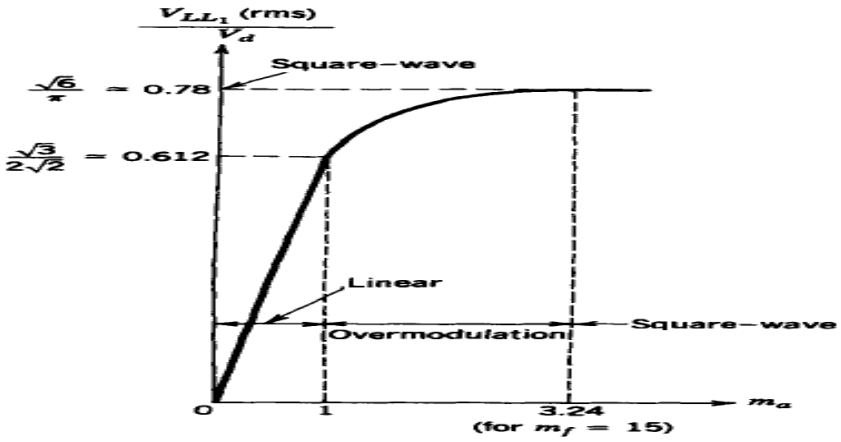
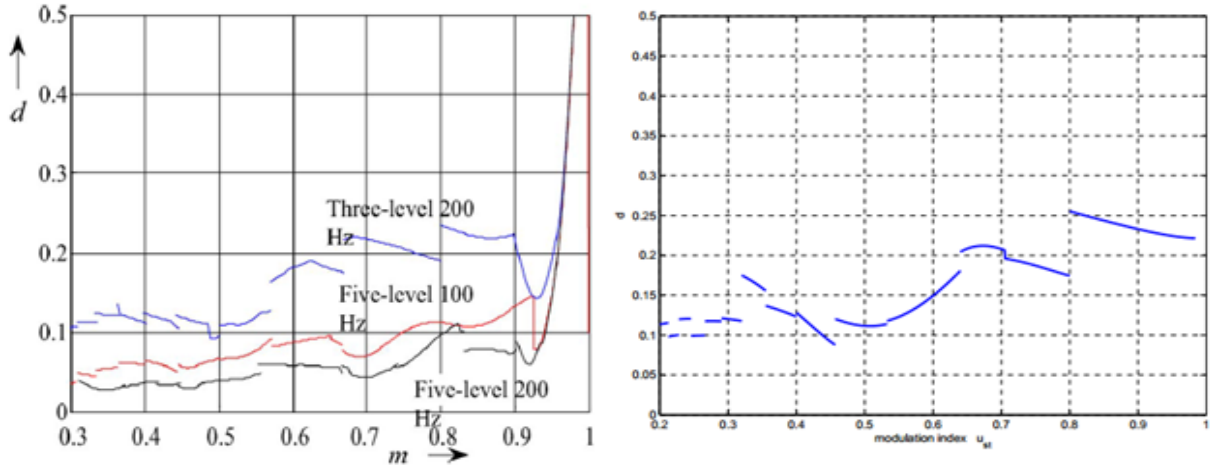


Figure 39: Operation of Inverter [32]



(a) Distortion factor for [47], m= u_{st} (b) Distortion factor with Programmed PWM

Figure 40: Comparison with [47] based on distortion factor

The result shows that, modulation method applied on [47] gives the result very closely resembles the result obtained as expected for the same kind of modulation technique. Since the number of angles and pattern for it is not mentioned in [47], it is not possible to know the number of pulses used at different levels. However, it can be concluded from the results

that programmed synchronous modulation technique produces very low THD as compared to the other modulation techniques.

Chapter 8

Introduction to Hardware Implementation Tools

Various tools including both software and hardware that is used as a part of the thesis are introduced in this chapter. This includes the fast processing field programmable array (FPGA), Xilinx Software solutions and Lab setup.

The use of digital signal processing units into the hardware has been immense recently as the offer high flexibility and better control than the analog counterpart. With the development within the field of embedded systems, control of power electronics converters have been more sophisticated with the software tools and mathematical modelling of the solution rather than hardware complication that existed with the analog approach. Use of Digital Signal Processers (DSPs) have been escalating as they can take the real world signals such as pressure, temperature, position among the few as the digital signal and perform the mathematical computation very quickly. The key module that converts this real world signal to digital signal for DSPs is Analog to Digital Converter (ADC).

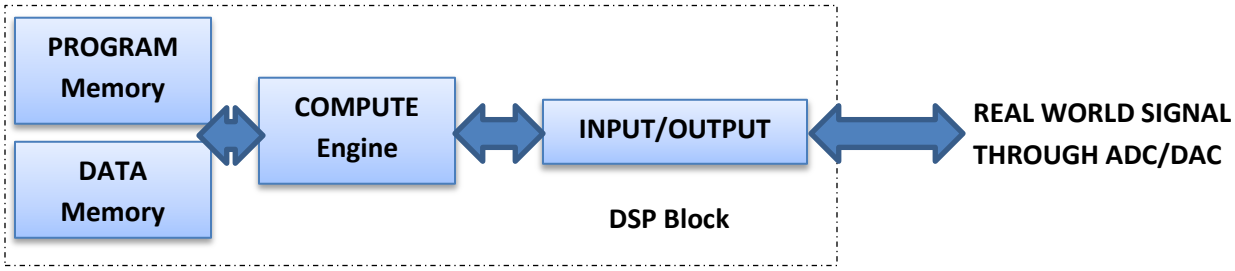


Figure 41: DSP Components

The DSPs after the successful completion of the designated task of mathematical and logical operation in real time gives signal back to the real world through the Digital to Analog Converter (DAC). The requirement of highly specialized purposes with high speed operation and reconfiguration capability requirement has given the superiority to the Field Programmable Gate Arrays (FPGAs) over DSPs. Increasing sophisticated tools

and requirements for power electronics converters is dragging embedded control system designers to move towards FPGA-based applications.

8.1 Field Programmable Gate Array (FPGA)

FPGAs are the matrix of reconfigurable gate array logic circuitry which can be configured and reconfigured into a system that creates hardware implementations with the software routine. They offers the flexibility of reconfigurable logic with hardware changes such as reconfigurable interconnect and have the potential to work as parallel processing. They can also provide multiple functions working at the same time because of their ability to work in parallel. The common units of FPGAs are RAM, clock management system, Digital Signal Processing, and input/output system.

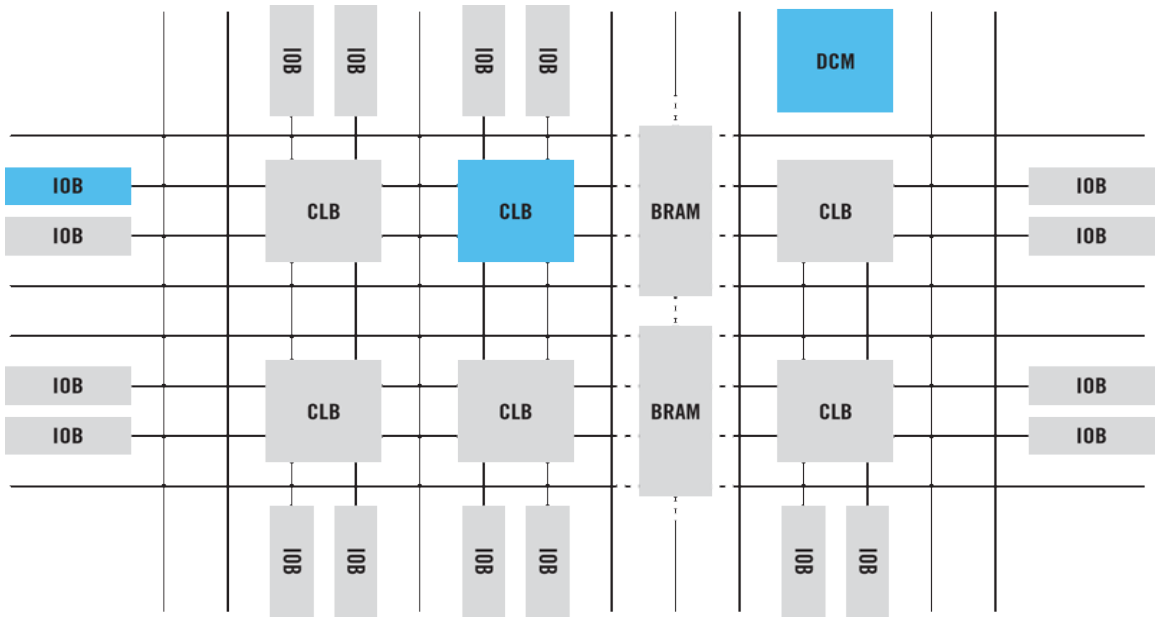


Figure 42: General Architecture of FPGA [49]

For every code that is written under the FPGA, the system converts into a physical digital circuit with facilities from RAM and interconnect system. The way the programmer works with the software affects the speed and performance of the system as glitches and redundancies often make the system operate slow. Intellectual Property (IP) is used to perform certain time critical tasks within the FPGAs.

8.1.1 Configurable Logic Blocks (CLB)

CLB is a logic unit in FPGA with configurable switch matrix, selection circuit and flip-flops. Hardware modules are built upon this matrix by programming. With the use of programming tools such as Xilinx ISE Project Navigator, we can see the RTL schematics that show the actual digital circuit connection made for the particular program.

8.1.2 Interconnect

Interconnect are the flexible connecting signal routes between the various components within the FPGA. These mainly include CLBs, Input /Outputs, S-RAM, flip-flops. The switches are programmed and re-programmed as desired for the possible connection of the programmer desire.

8.1.3 SelectIO (IOBs)

The input/output mechanism for the system is provided by the IOBs and it can exist in multiple numbers. These blocks give the programmer ways to communicate with the system. The feedback can be used by the programmer to enhance the system as the IOBs are reprogrammable.

8.1.4 Memory

The on-chip memory in the FPGAs is available as Block RAM (BRAM) which allows the designer to quickly communicate with the data's and storage. These memory interfaces are based on hardware-verified reference and software tools that enable quick generation of custom design.

8.1.5 Clock Management

The FPGAs can offer digital clock management and phase-looped locking. One of the major advantages is the ability to use the multiple clock generation from the single input clock as FPGAs can operate in parallel. With the digital clock management the clock frequency can be either reduced or increased from its reference value.

8.2 Implementation Board

System on Chip (SOC) integrates multiple features such as digital and analog part, hardware and software, equipped with communication infrastructure. The Xilinx Vertex FX30T Board [53] which is built by SINTEF is used for the project which provides all the necessary requirement of the control and conversions. It consists of Vertex-5 ppc440 FX30T with embedded Power processor core in it. The board can be used to design a full featured PWM generator with the hybrid of Space Vector Modulation and Programmed Synchronous Modulation. The important components of the board are

8.2.1 Memory

The board has the following memory units

1. DDR2 DRAM -Micron MT74H64M16HR-3E in 1GB, 16x64Mbit 333MHz
2. EROM-Renesas HN58V257A. 32k x 8 bits.
3. FLASH-Spansion S29GL512P is 512 Mbit flash chip, 32Mx16 64Mx8.

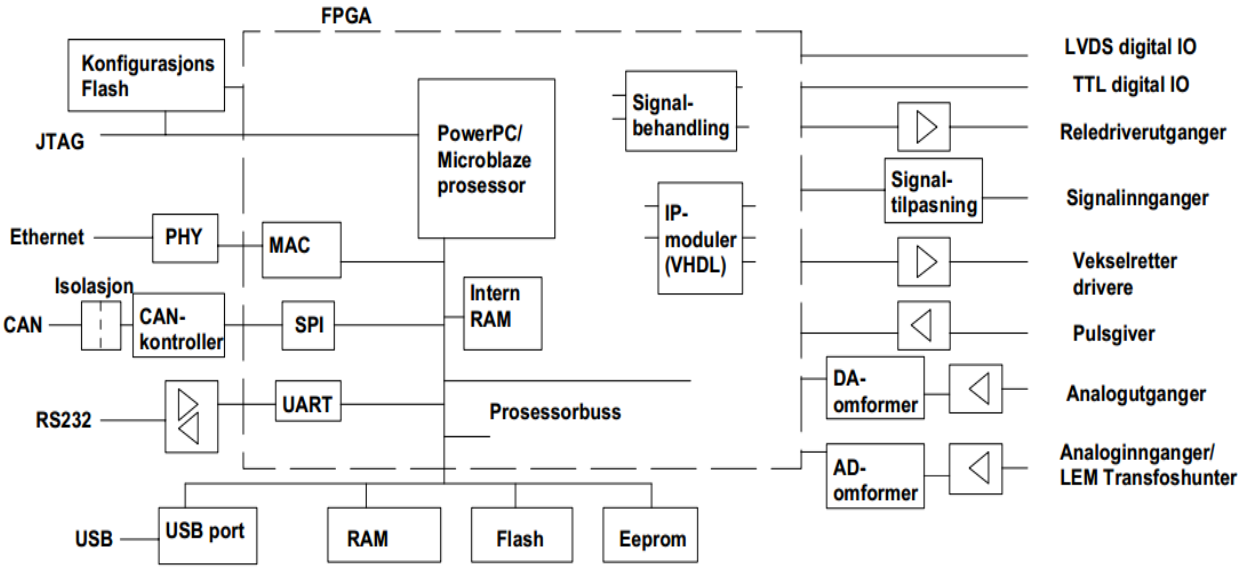


Figure 43: FPGA Board [53]

8.2.2 Clock

The board is supplied with the 40 MHz clock. Digital Clock Manager (DCM) can be used to generate the desired clock cycle rate either higher or lower than rated value.

8.2.3 Communication port

Following communication ports are present

1. RS232 serial port connected with a male 9 pol D-sub connector
2. Ethernet port for 10MB/sec and 100 MB/sec connection
3. High speed V.2 USB port with the separate controller NXP ISP1582 available on board outside FPGA

8.2.4 Analog to Digital Converter (ADC)

The board has inbuilt 12 bit ADC which is supplied with 40 MHz clock cycle.

8.2.5 Relay Drivers

Card is equipped with four relay driver. The output voltage is 5V.

8.2.6 LVDS

The board has high speed serial communication with LVDS signals with the Speed of 600 mbps.

8.2.7 Signal Inputs

The board has six general purpose signal inputs and can accept both analog and digital signals. The analog signals are read through the ADC while the digital signal is read with signal level being 5V CMOS.

8.2.8 Digital Input Output Port

There are three channels on card with 16 bit digital IO ports with signal level 0-3V as the FPGA blocks are fed with 3V.

8.3 Xilinx ISE Design Suite

The Xilinx ISE Design suite with embedded edition is a complete tool to work the Xilinx FPGA and is used for this thesis. It consists of Xilinx Platform Studio (XPS), Software Development Kit (SDK), ISE Project Navigator, and large number of repository. It provides with fundamental components to achieve the optimal design for the programmer. The complete list of available features with the different version of Xilinx ISE Design suite is listed in Figure 44.

Features	ISE WebPACK	Embedded Edition	System Edition
Device Support	Limited	All	All
ChipScope Pro and the ChipScope Pro Serial I/O Toolkit		✓	✓
CORE Generator	✓	✓	✓
Design Preservation	✓	✓	✓
Embedded IP Peripherals	Device locked to three smallest Zynq devices	✓	✓
ISE Simulator (ISim)	Limited	✓	✓
MicroBlaze Soft Processor	Device locked to three smallest Zynq devices	✓	✓
Partial Reconfiguration*	Option	Option	Option
PlanAhead™	✓		
Platform Studio	Device locked to three smallest Zynq devices	✓	✓
Power Optimization	✓	✓	✓
Project Navigator	✓	✓	✓
Software Development Kit (SDK)	✓	✓	✓
System Generator for DSP			✓
Timing Driven Place & Route, SmartGuide, and SmartXplorer	✓	✓	✓
XST Synthesis	✓	✓	✓

Figure 44: Features of Xilinx ISE Design Suite (with different edition) [50]

The Xilinx ISE Design Suite enables the designer to make complete embedded system processor to implement it with the FPGAs. XPS, SDK and Project Navigator were used for the purpose of this thesis.

8.3.1 Xilinx Platform Studio (XPS)

Xilinx Platform Studio (XPS) provides the path of communication between the embedded hardware system and software for the designer. Using this tool, designer can build, connect and configure multiple embedded systems design with processors such as state machines, microprocessor, FPGA among the few. It gives the graphical overview of any complex design and necessary assistance to configure the required system within few minutes.

The XPS has the ability to combine plug and play Intellectual Property (IP) cores available from Xilinx Embedded Intellectual Property catalog with custom design including the third party designs using Verilog and VHDL.

XPS automatically generates the critical system software like bare metal Board Support Package (BSP), boot loaders, and Linux BSPs. This removes the firmware development delay during the applications development with the embedded design.

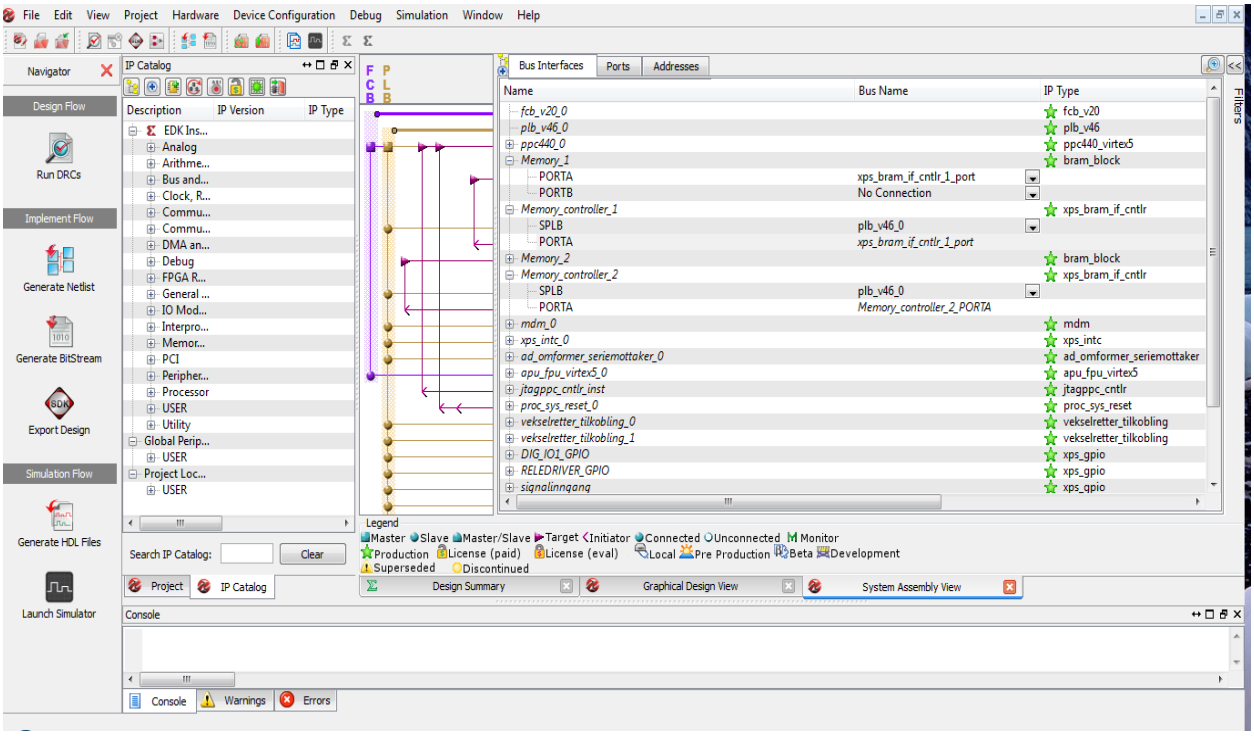


Figure 45: XPS Design Window

8.3.2 Xilinx Software Development Kit (SDK)

The SDK is the integrated design software solution provided by the Xilinx for creating embedded design and applications for all of the Xilinx microprocessors. It is the first application to enable true homogenous and heterogeneous multi-processor design []. It understands all the Xilinx-based design made using the XPS mentioned in earlier in 8.2.1.

This is easy to learn platform which give auto-configuration for many parameters such as memory mapping, tools and library paths, JTAG settings among the few. A designer can implement the design using the software with minimal effort and time to learn because of its auto pre-configuration ability.

SDK contains drivers that are user customizable. It has the ability to identify the bottle neck in the code and optimize it by converting the entire function into an optimized programmable logic or by splitting the function. SDK also supports trigger and debug for critical hardware/software design.

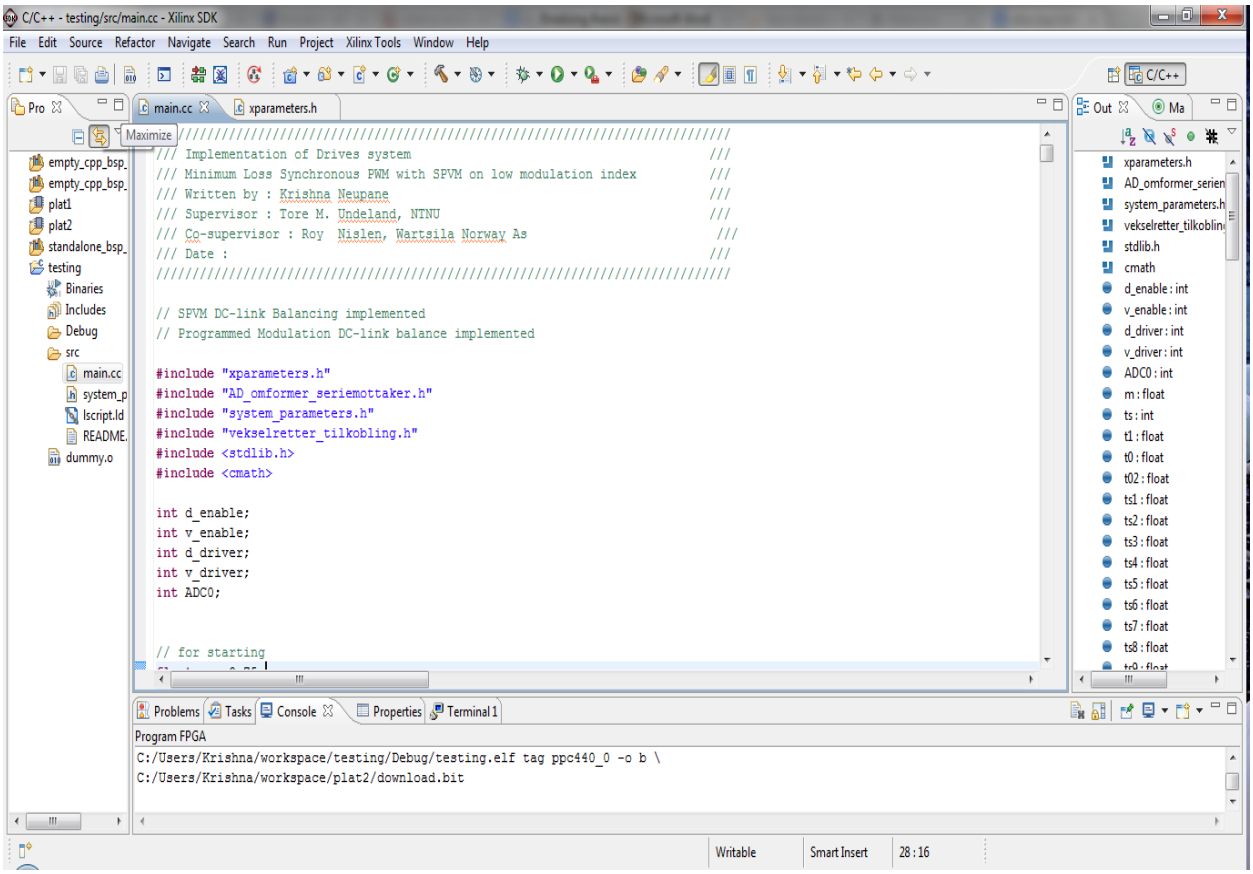


Figure 46: Xilinx SDK window

8.3.3 ISE Project Navigator

ISE project Navigator is the anther tool provided by Xilinx to support the embedded design with Xilinx processors which organizes the design and help implement the design by moving it to the targeted Xilinx based device. It is a high level design manager to work with FPGA and CPLD.

Project Navigator allows the designer to add and create source file to work with Verilog and VHDL. It is also integrated with ChipscopePro where designer can see the data configuration and make simulations with direct hardware interactions. Designer can modify source file, run process and view output from the process by using the Project Navigator. It also allows the designer to view the RTL schematics. The integrated PlanAhead with the Project Navigator offers RTL to bit stream flow with easy to use user interface.

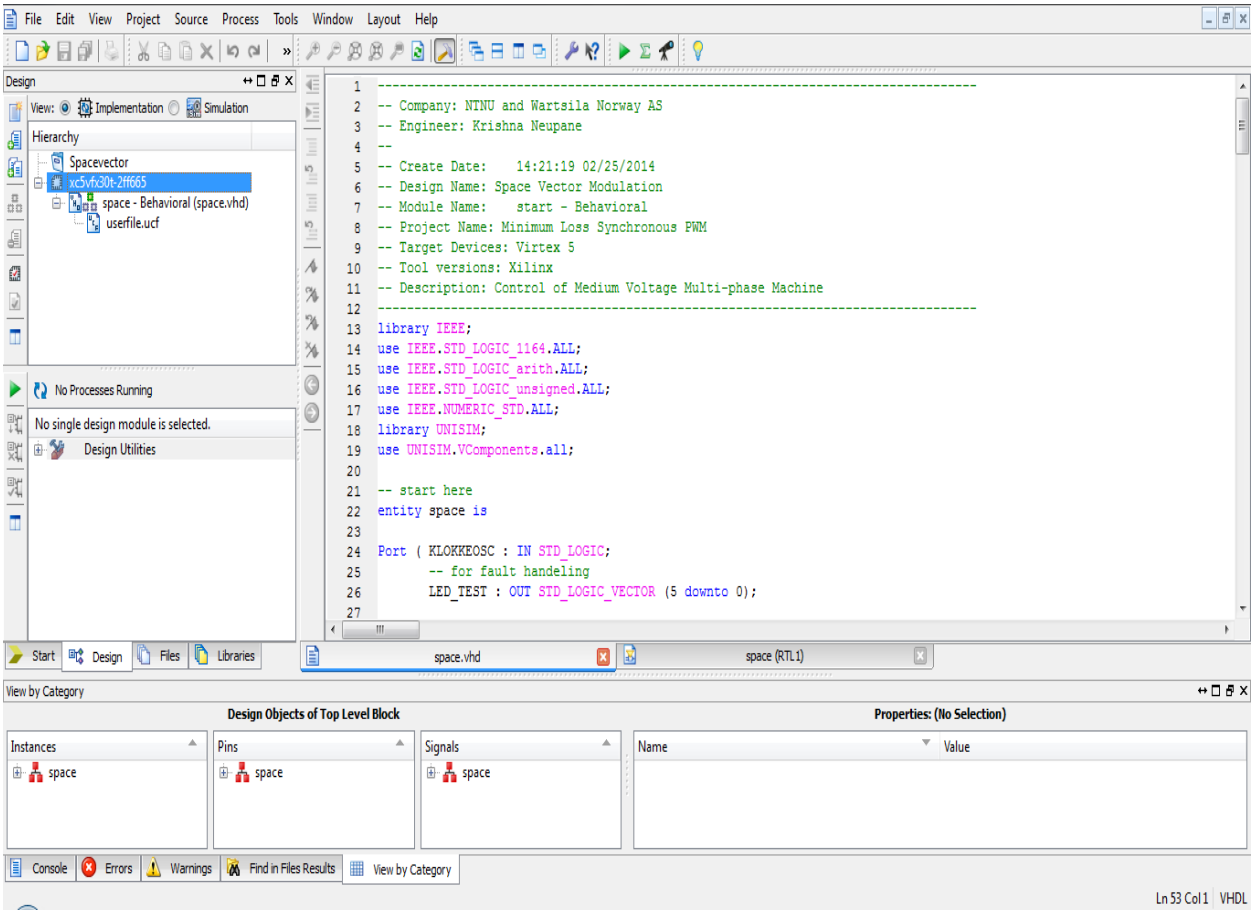


Figure 47: ISE Project Navigator Window

8.4 VHSIC Hardware Description Language (VHDL)

VHDL is the hardware description language for very high speed integrated circuit. This language provides a medium for direct communication with the hardware although it's quite complicated as compared to other high level languages such as C and Java. The main reason of the complication is that the entire language used for describing hardware system and the programmer has to think as the digital circuit.

VHDL is very useful tool for documentation, verification, and synthesis of complex digital circuit with language. This gives the flexibility as the VHDL code can theoretically achieve all of the above characteristics thus saving time and lot of effort. VHDL can also be used to take three different approaches for describing hardware

i. Structural method of hardware description
ii. Data flow method of hardware description
iii. Behavioural method of hardware description

A mixture of all these three methods is very common for the implementation. Execution of a VHDL program results is essential for many digital design as it is a simulation of the digital system allowing us to validate the design prior the fabrication. The book by Volnei A. Pedroni [52], helped the basics and advanced method of coding with VHDL.

8.5 Experimental Setup

The hardware components that are used in this thesis work are described in this section. The major components of the hardware are

1. The Six phase Induction Machine
2. Three phase Three Inverter
3. AC/DC Converter
4. FPGA Control Card
5. Measuring Devices
6. DC Machine with rectifier

8.5.1 The Setup

Even though six phase induction machine is used in the lab, only three phases were used for the experiment but this thesis can be extended to six phase drive.

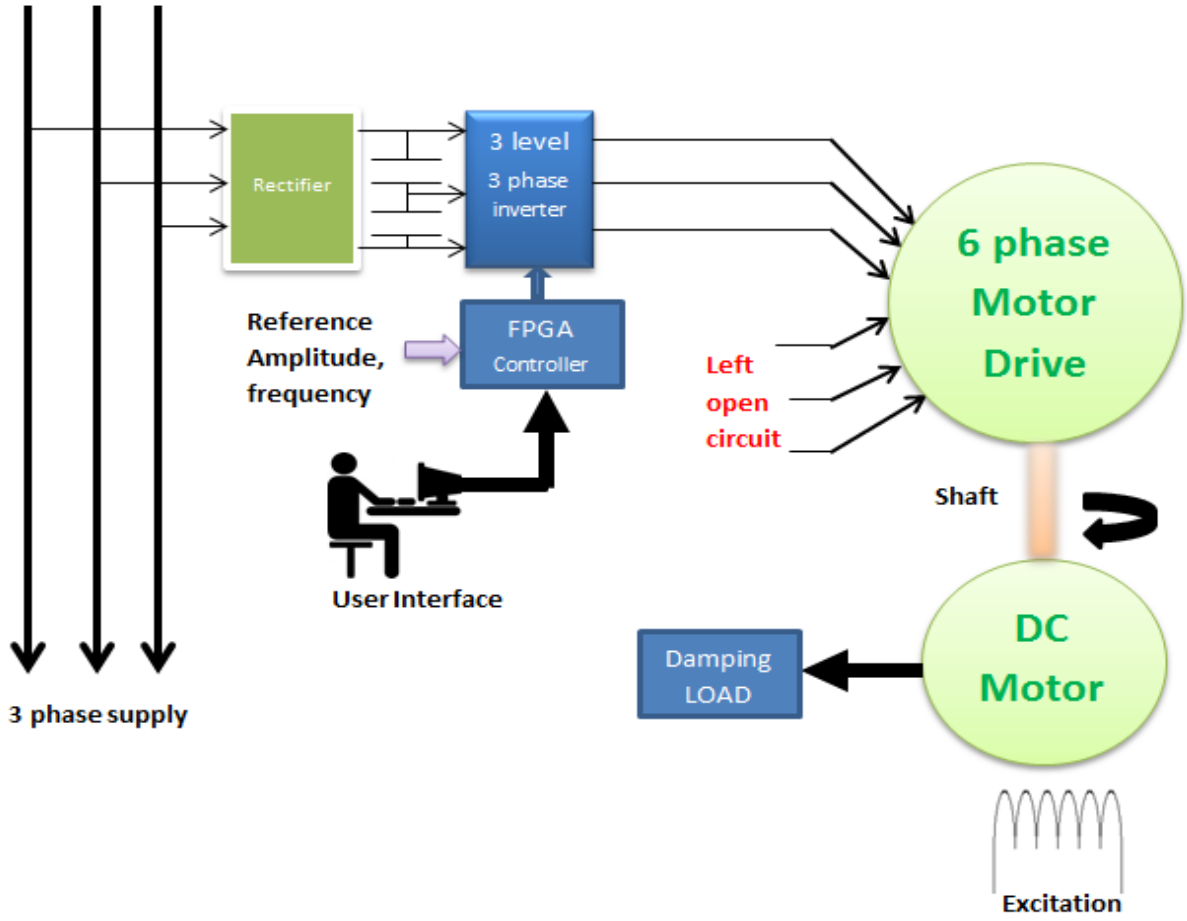


Figure 48: The lab setup (Block Diagram)

The three phase supply mains are taken into the system where, the rectifier is used to make the DC link. The embedded board is used to control the three phase inverter by controlling the amplitude modulation, frequency of the inverter output. The shaft of the machine is connected to the DC motor to regenerate the power and dump it to the resistive load. The FPGA board is controlled from the PC by the user. The provisions for monitoring the input and output data is done by using the oscilloscope, THD meter, Voltmeter among the few. The DC link voltage is varied by the use of auto-transformer that is connected between the rectifier and the three phase supply. This gives the freedom to choose the DC link voltage.

8.5.2 Six phase Multilevel Inverter

The six phase induction machine (SPIM) is used in the lab, which consists of three phase asymmetrical stator windings spaced at 30°, as stated in Chapter 2. The 11.7 kW rated machine has squirrel-cage rotor. Refer appendix 3 for the nameplate data.

8.5.3 Three phase Three Inverter

Three phase three level inverter (Neutral Point Clamped Inverter) is used to supply the three phases of the machine. It consists of IGBTs as the semiconductor switching device. It is rated as 400 kW, 1200 V and 200 A.

8.5.4 AC/DC Converter

Three phase diode rectifier is used to convert the three phase supply from the auto-transformer to the DC link voltage. The input and output ratings of the rectifier is 0-400 V RMS input and 0-540 V DC output while the current rating is 63 A.

8.5.5 FPGA Control Card

Xilinx Virtex FX30T is used in the lab. The Board is made by SINTEF Energy. The details regarding it can be found at Appendix 4.

8.5.6 Measuring Device

MSO 2024 is used as the oscilloscope for measuring the waveforms as well as reading the output values. THD meter is used to recognize the harmonics in both current and voltage in the lab.

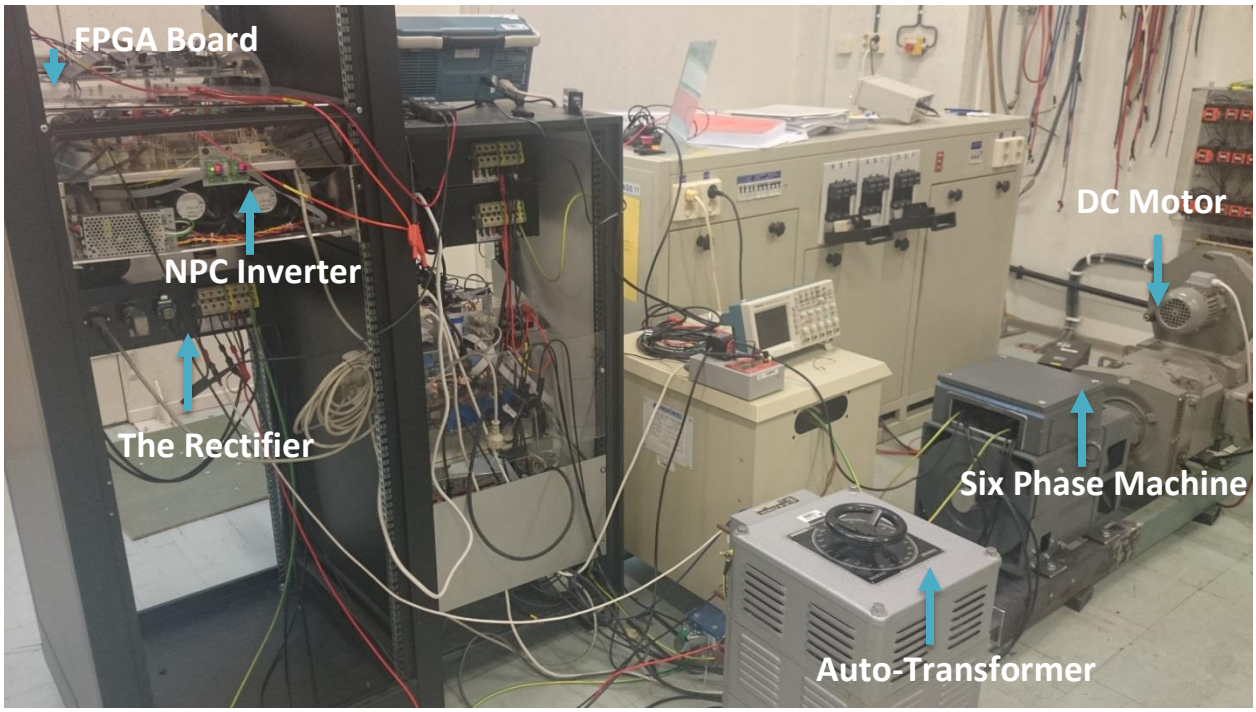


Figure 49: The lab setup

Chapter 9

Implementation and Results

In this chapter, details regarding how the system is applied are described with the aid of flowchart. The problem description and results of the experiments is discussed and analyzed. Further, THD (which is directly related to the criteria of minimization) with different values of amplitude modulation for both space vector modulation and programmed modulation, is explained.

As stated earlier, the implementation has three phases. The first is to start the machine, second is to make choice between asynchronous and synchronous modulation and finally the implementation of either of them. The FPGA board made by SINTEF is used for the purpose of implementation with Virtex 5 FX30T and power processor. It has been observed that the current THD for the programmed modulation is much lower as compared to the current THD for the space vector modulation.

9.1 Implementation Methodology

For the implementation of the system, the strategy described in Chapter 5 is used, where code for starting of the machine is made to run with every starting of the machine. Every time there is a change in the amplitude required during the operation of the machine, the interrupt detects it and shifts the system to run with the new modulation value. With programmed modulation the numbers of pulses are varied from 3 to 15 depending upon the amplitude of modulation.

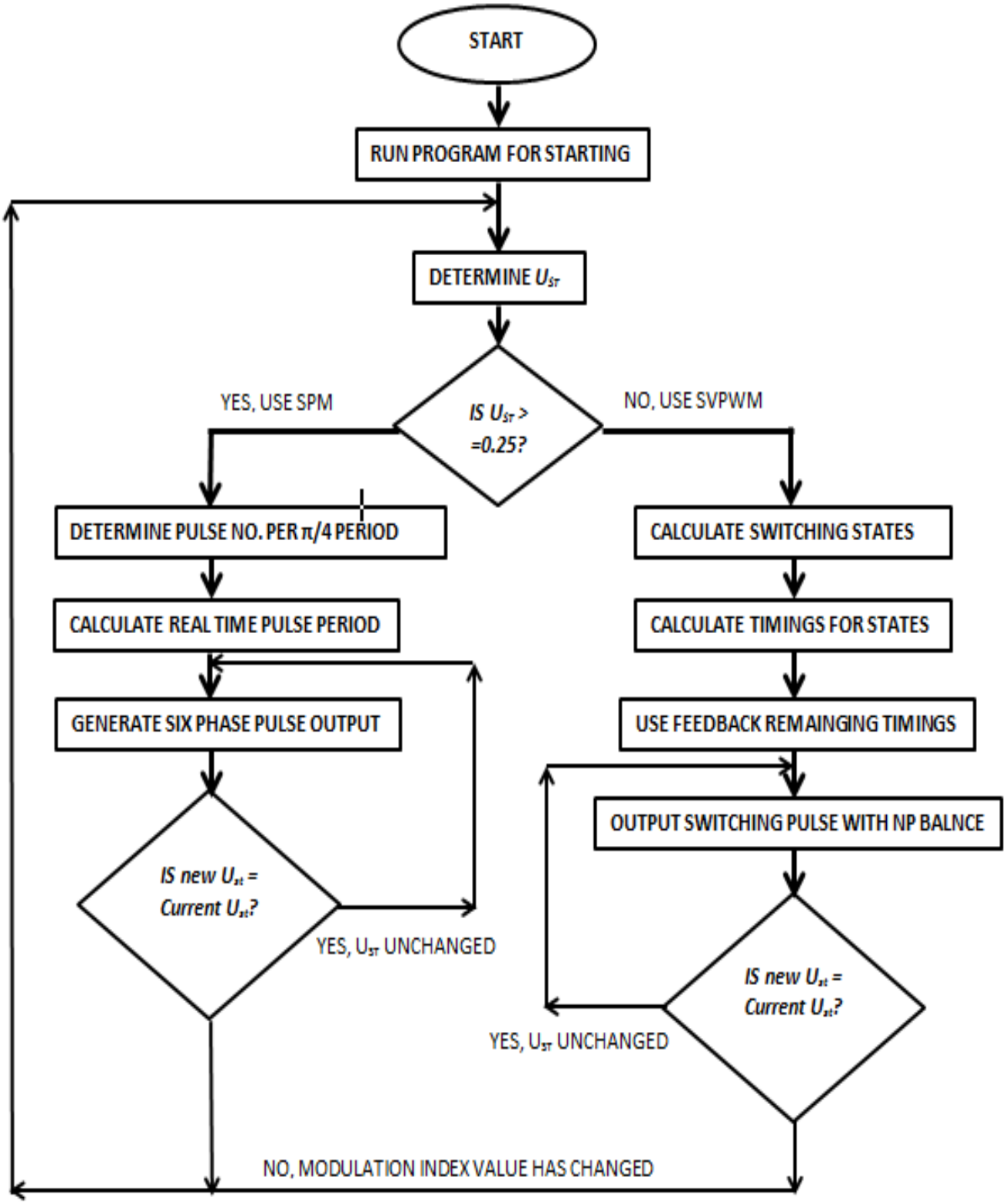


Figure 50: Implementation of the Overall Modulation Technique

Figure 50 shows the flow chart of the method implemented in the lab. Since online values generation of the modulation is used rather than storing them into the DRAM the system has a faster response to the change in the amplitude. During the space vector modulation for the lower amplitude, the calculation time for the duty cycle is highly reduced by using the constant values of sinusoidal function rather than performing the sinusoidal operations. While during the programmed modulation, the fast calculated programmed modulation values are sent to the function to generate the synchronous PWM until the new interrupt of changed amplitude requirement is sent to the function.

9.1.1 Implementation of Starting of Machine

The speed control of the three phase induction machine is achieved in the lab by using the control to vary the output frequency supplied to the induction machine. This resulted in the improvement of the stating of the machine. The fundamental problem is that it couldn't be stated with the low amplitude modulation and high modulation amplitude as 0.60 is used in the lab at the starting. The machine started with the frequency of 10 Hz supply and is gradually improved to 50 Hz supply. The starting of machine is not the main part of this work, as such the trail methods as described above is used in the lab to get the working condition of machine starting. It is therefore advised to use an improved method of starting the three phase induction machine which is not covered here.

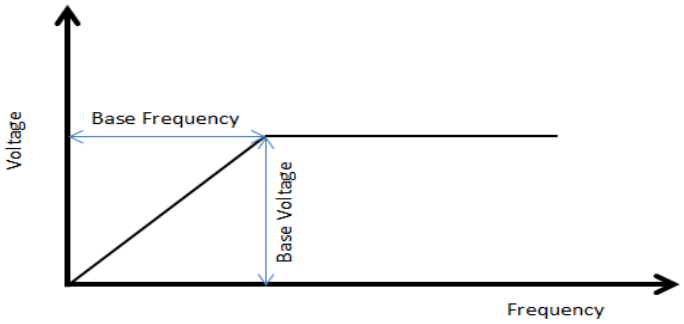


Figure 51: Starting of the Induction Machine

The voltage is kept constant when the base speed is achieved while the frequency is increased.

9.1.2 Implementation of Space Vector Modulation

The space vector modulation is achieved by using the categorization of sector and regions of the sectors. Particular sequence is defined for each regions and the sequence is carried in that manner.

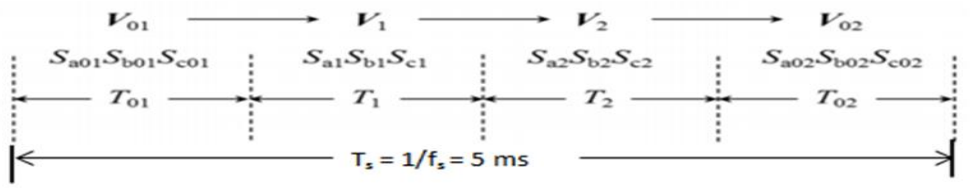


Figure 52: Switching Sequence for time period Ts

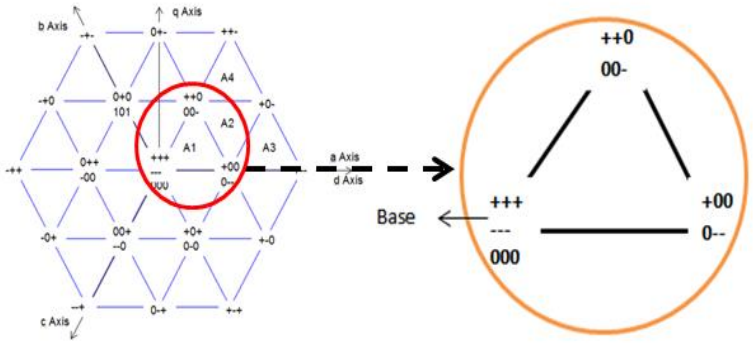


Figure 53: Determination of Switching Sequence in region A1

During the implementation, T_1 and T_2 as shown in Figure 52 is calculated normally while NP balancing algorithm is applied in order to determine the time period for T_{01} and T_{02}. The details regarding this can be found in [48]. For the purpose of simplification of the system from [48], assumptions such as stable operating conditions were made during the project. Another factor of importance for the implementation is the understanding of the correct driver signal that is associated with the different states. The table below shows the driver signals association with the states in the system for the first sector regions.

The implementation of space vector modulation for the six phase supply is not conducted in the lab rather it is applied for the three phase supply.

Table 4: Switching states and corresponding driver signals

Sector 1												
	A+		A-		B+		B-		C+		C-	
		AA+		AA-		BB+		BB-		CC+		CC-
Region 1 (1)												
+++	2	1	0	0	2	1	0	0	2	1	0	0
0	2	1	0	0	2	1	0	0	0	1	2	0
0	2	1	0	0	0	1	2	0	0	1	2	0
0	0	1	2	0	0	1	2	0	0	1	2	0
Region 2(2)												
0	2	1	0	0	0	1	2	0	0	1	2	0
+0-	2	1	0	0	0	1	2	0	0	0	2	1
+--	2	1	0	0	0	0	2	1	0	0	2	1
0--	0	1	2	0	0	0	2	1	0	0	2	1
Region 3(3)												
0--	0	1	2	0	0	0	2	1	0	0	2	1
00-	0	1	2	0	0	1	2	0	0	0	2	1
+0-	2	1	0	0	0	1	2	0	0	0	2	1
0	2	1	0	0	0	1	2	0	0	1	2	0
Region 4(4)												
0	2	1	0	0	2	1	0	0	0	1	2	0
++-	2	1	0	0	2	1	0	0	0	0	2	1
+0-	2	1	0	0	0	1	2	0	0	0	2	1
00-	0	1	2	0	0	1	2	0	0	0	2	1

9.1.3 Implementation of Programmed Modulation

The implementation of programmed modulation to get the synchronous modulation PWM signals fast as well as not to get the limited memory of the processor filled, multiple functions is made that acted independently. The use of FPGA processor here would have been more effective as it can work in parallel operation. However, the optimization of lots of steps made the synchronous PWM very effective even with the use of PowerPC that is inbuilt in the board along with the Virtex 5 processor. The main function is used to determine the value of amplitude modulation and it passes to the first function which determines the switching angles. These switching angles are than used by the second function to generate the real time six phase driver signals. In the lab it is tested with the three phase driver signal generated to supply a three level inverter. The system is made to operate under the second function until the interrupt signal is sent to make sure that there the change of amplitude modulation is required.

The major problem here is the use of Power PC processor instead of parallel working Virtex 5, as a little time is wasted each time an interrupt is detected even though it is determined that the time loss is not significant.

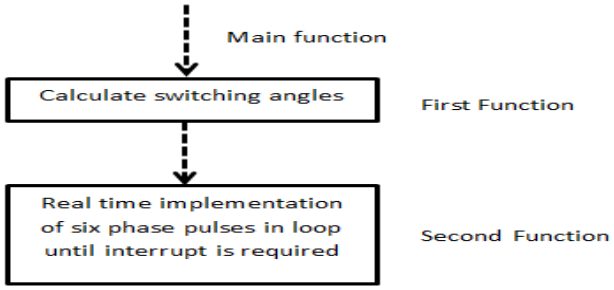


Figure 54: Implementation of programmed modulation

9.1.4Implementation of Feedback System

LV 25-600 is used as an actuator for the feedback of the system. The FPGA board is designed to receive the feedback signal from the current sensor LEM and process the results. This feedback signal is used to control the amplitude modulation level and hence the speed of the induction machine. The LEM can take up to 600 V/rms at 10 mA ratings and the result is passed through the ADC to get the digital signal of the inputs. For the purpose, the ADC has to be configured, refer Appendix 5 for various inputs and output configuration codes. Once the ADC has been configured, the values can be read directly by using a simple code from the SDK file.

```
ADC0 =      AD_SIGNAL(BASEADR_AD,0);
```

The ADC output value can be scaled as required and the value from ADC can be fetched whenever required. In this thesis work, the ADC value is fetched regularly to generate the interrupt signal for changed modulation condition and to calculate the new switching sequence for both SOM and SVPWM. The Board can take up to 8 ADC inputs; however, only one is used during this thesis.

9.1.5 Definition of User Constrain File

This file acts as the bridge between the software and the hardware by addressing the hardware components from the software. All the components used from the board should be addressed by this file. These include the Driver signals, ADC, Clock among the few. ISE Project Navigator is used to generate this file and imported to the XPS design. Refer Appendix 5 for examples on how to use user constrain file.

9.1.6 Real time implementation

The real time implementation of synchronous pulse width modulation is lot of work with being able get the optimal angles as fast as possible. These patterns are calculated and stored in temporary memory and multi-phase drive patterns are generated for the real time implementation. Refer Appendix 5 for the multi-phase drive patterns generation. It shows the three phase implementation by reading the temporary memory on real time. However, this can easily be extended to six phase or nine phase patterns.

9.1.7 Real time calculation of pattern

In figure 37, the optimal patterns for the minimum loss synchronous PWM is shown after the calculation. These patterns required for a cycle is quarter symmetrical (symmetrical about $\pi/2$), as such the rest of the patterns can be generated easily when the patterns for quarter symmetry is calculated. The patterns are linearized for the range of modulation amplitude and hence it is possible to directly jump into the level of interest. The linear calculation than makes it possible to calculate the patterns very fast. Further use of parallel processing can be utilized to calculate all the pulse required in short time. For example, while $U_{st} = 0.9$ it corresponds N =4 and four calculations can be operated in parallel to get the results in short time. Further, fraction numbers and division of numbers are removed from the calculations method to reduce the processing time. These patterns calculated are stored in temporary memory. For details regarding the optimal patterns refer Chapter 7.

9.2 Problems Encountered

Multiple problems were encountered during this project leading to the thesis which is in both Hardware and Software department. The list some of the major problems during the project are identified and presented below.

1. The limitation of the fast memory block (BRAM) within the board forced the limitation of code that can be used for the PowerPC processor. Multiple times fault is detected due to the limitation of such error.

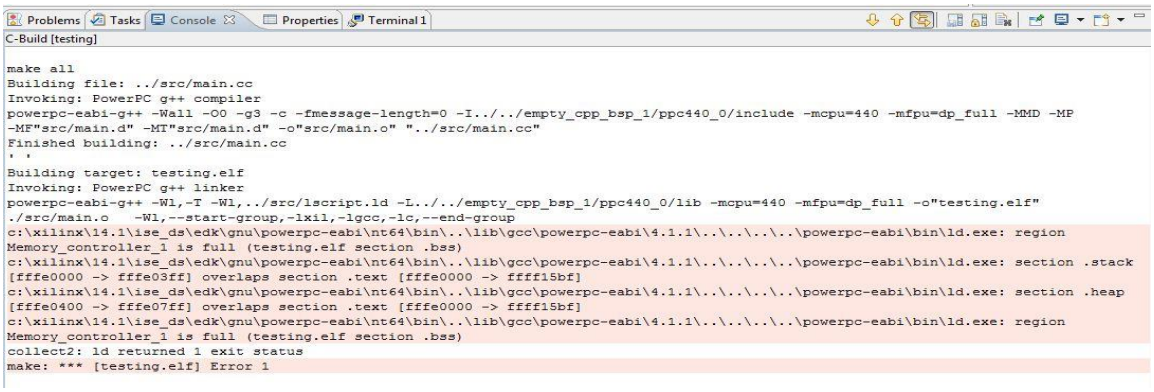


Figure 55: A typical fault with memory over loading

2. The loss of clock cycles each time the interrupt is request. Since the system has to regenerate switching sequence or pulse with the change in modulation amplitude, there is a small gap before the system can start regenerating the pulse in order. This can introduce unnecessary harmonics and transient's problem in the system.
3. The starting of the multiphase machine which is not covered well in this thesis lab work gave a problem and proper solution for the problem is yet to be implemented.
4. Identification of correct number of clock cycle utilization to define the fundamental frequency. This is done with many trials and 50 Hz fundamental frequency output as well as, 10 Hz, 20 Hz, 30 Hz and 40 Hz fundamental frequency is obtained in the lab.

9.3 Driver, Voltage and Current Signal

Driver, voltage and current signals are presented below to validate the result of the system.

9.3.1 Driver Signal

Driver signals are the major functions of the modulation, which determines the overall characteristics of the system including the harmonics and neutral point voltage/current. These signals are used with various functions under the different roles such as the

starter of the machine, space vector modulation with corresponding modulation amplitude, and finally programmed modulation.

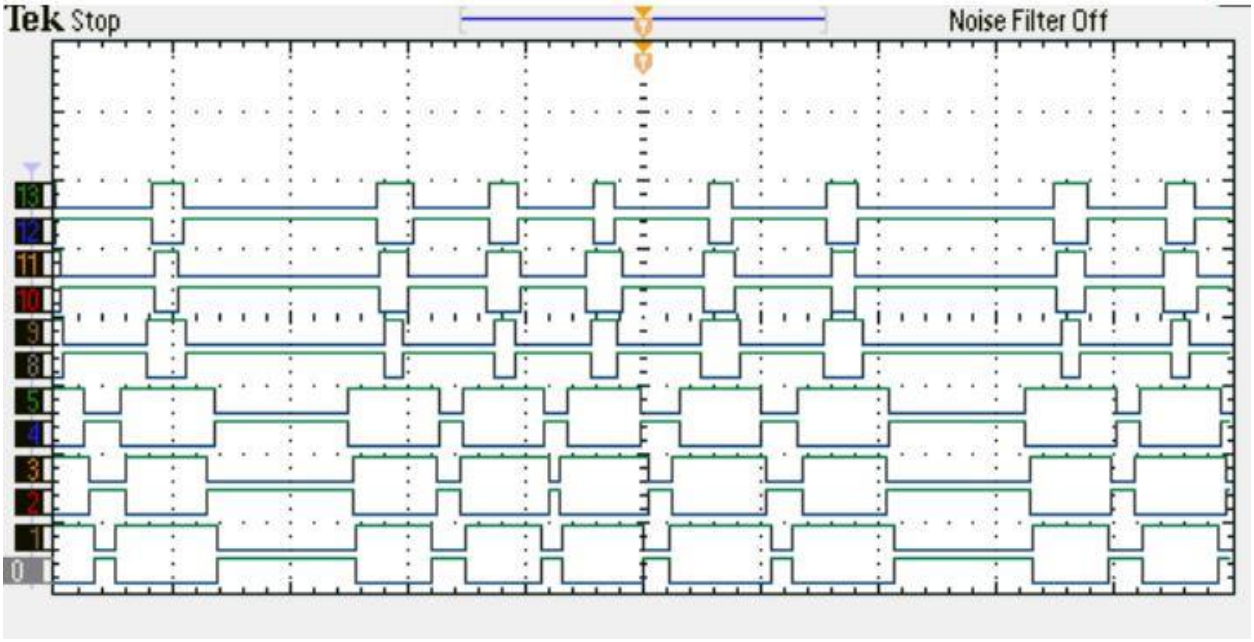


Figure 56: Driver signal for SVPWM with Ust = 0.2, f_{sw} = 200 Hz, f_s = 50 Hz

While the frequency of the modulation signals used varied during the start of the machine, the steady state operation had the constant frequency of 50 Hz for both SVPWM and PSM.

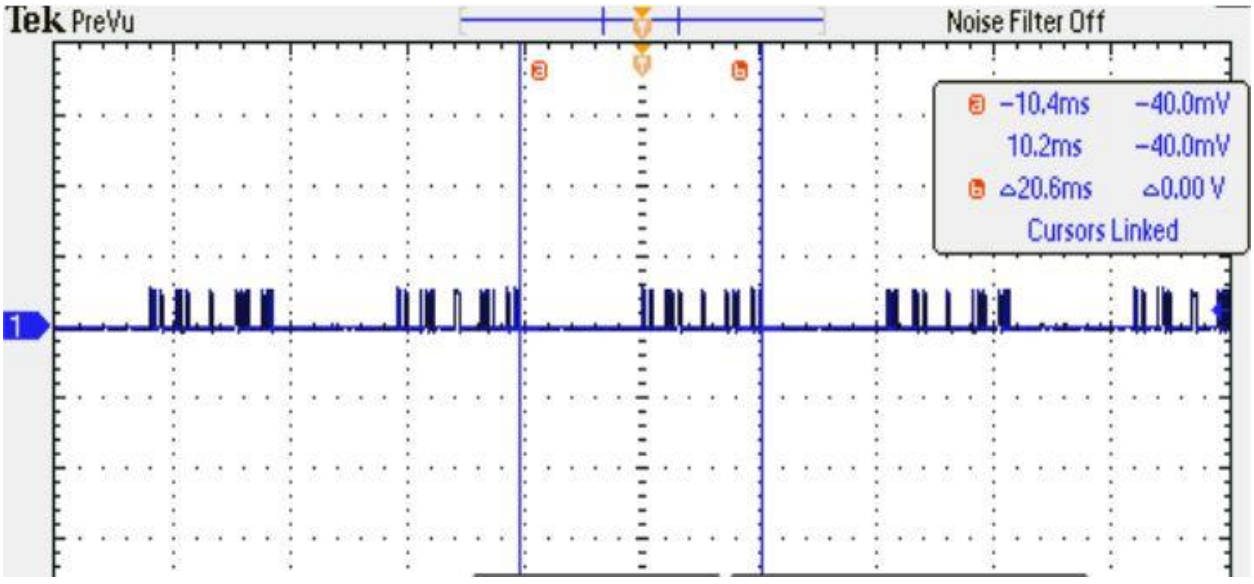


Figure 57: Driver signal for SOM with Ust = 0.256, N =15

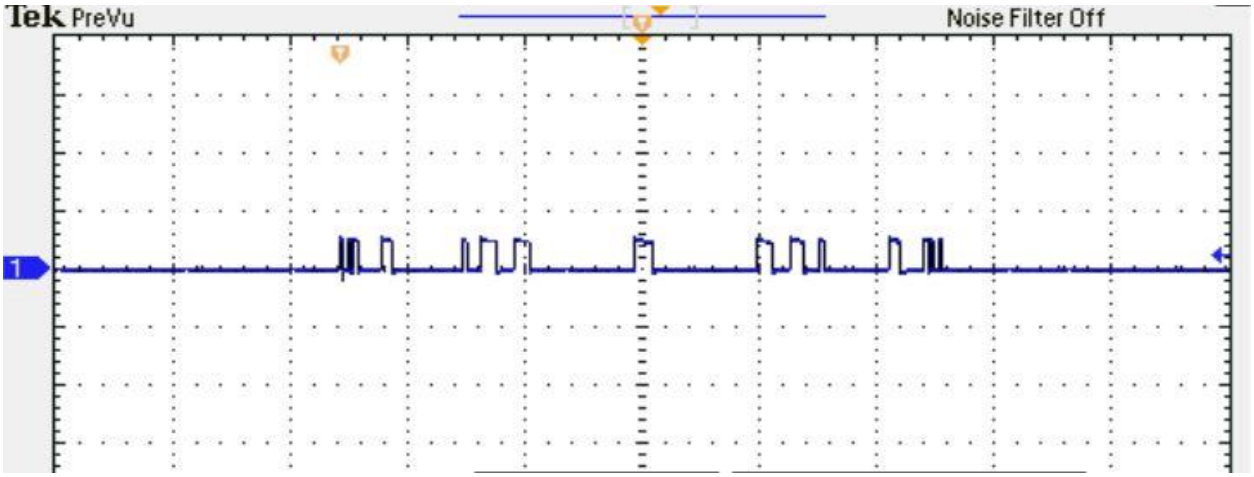


Figure 58: Checking Symmetry of Driver signal for SOM with Ust = 0.256, N =15

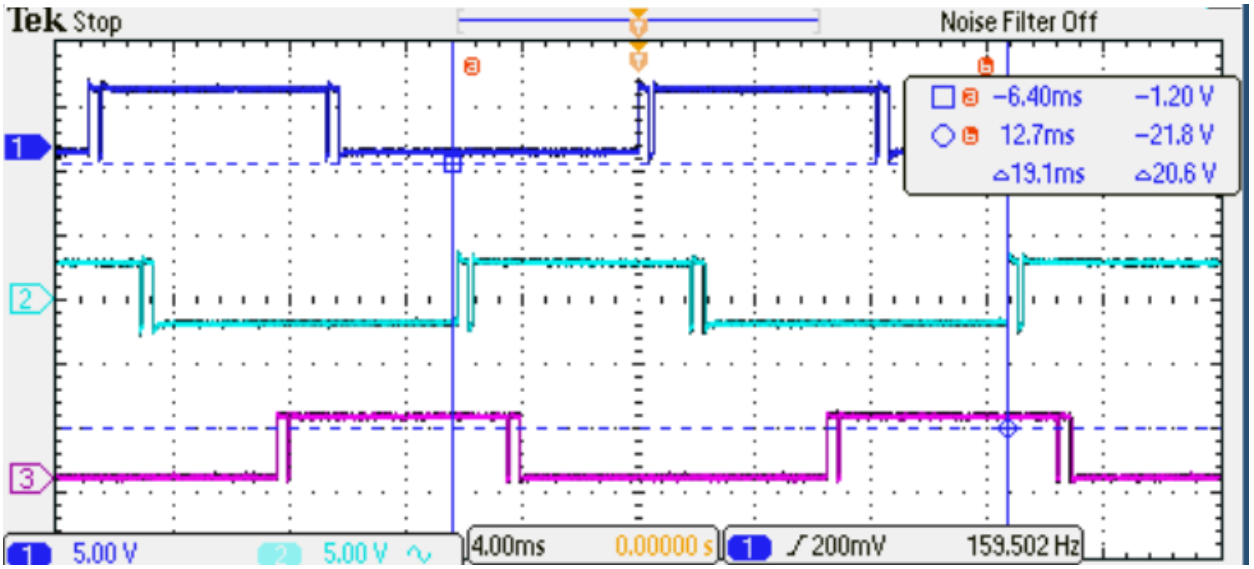


Figure 59: Three phase SOM with Ust = 1.1, N =3

The driver signal for the programmed modulation has the quarter wave symmetry as expected. This is the essence which is already described in the previous chapter. The frequency is found to increase to 52 Hz with modulation amplitude rise to 1.1. According to the Figure 21, the switching frequency should be below 200 Hz and the result matches with the theory. The signals for the three phase supply is found to be 120° phase shifted as expected. This provided the symmetrical three phase supply at the rated frequency of 50 Hz.

I. Variation of Pulse Number with Amplitude of Modulation

The number of pulse per quarter cycle is varied according the maximum number of pulses that could be applied for the system without going beyond the capacity of defined in Chapter 5. The changed number of pulses per period is observed in the lab as expected.

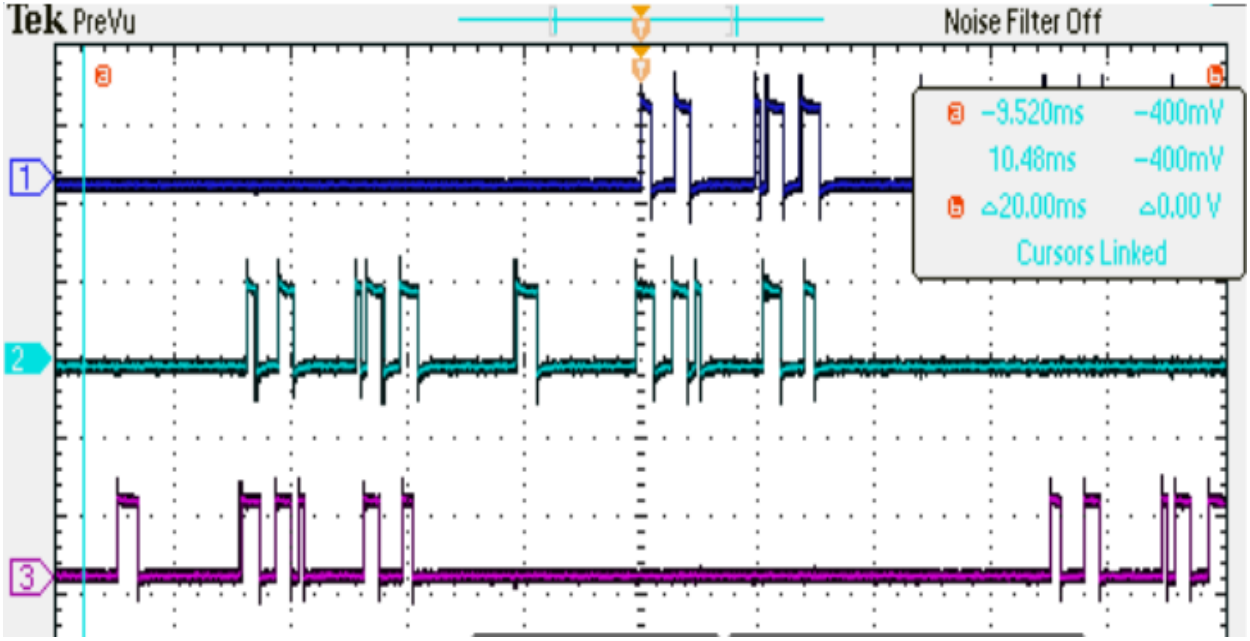


Figure 60: Driver signals for SOM with Ust = 0.34, N = 11, fs = 50 Hz

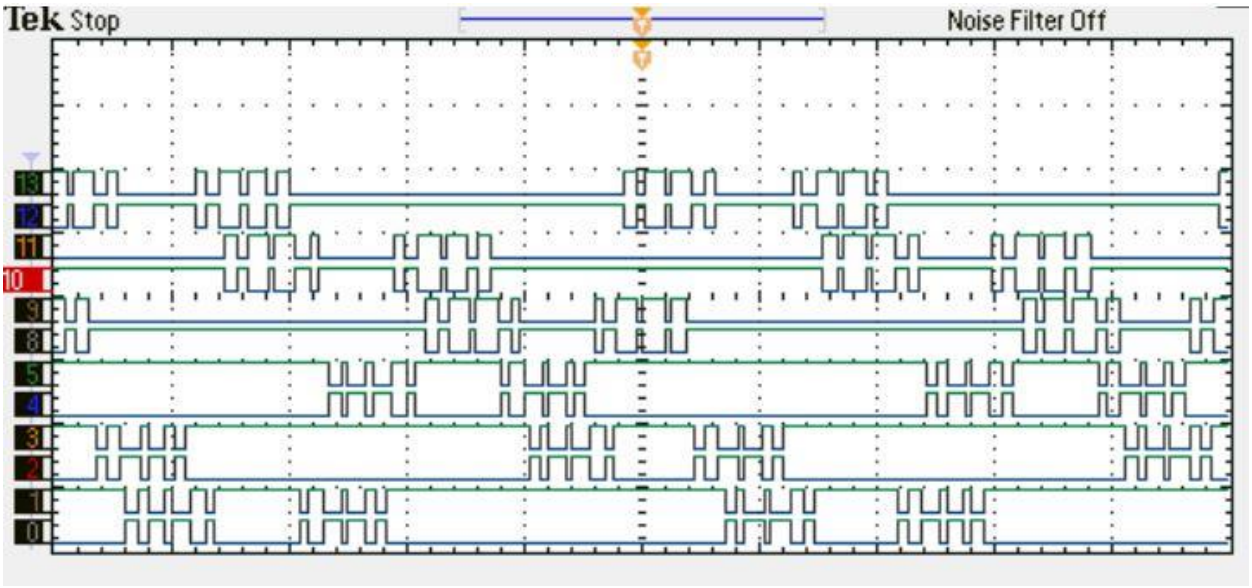


Figure 61: Driver signal for SOM with Ust = 0.48, N = 8, fs = 50 Hz

9.3.2 Voltage Signal

The driver signals for the three phase supply make the three phase voltage supply for the induction machine. This voltage waveform is found to have more harmonics as compared to the current waveform for this implementation.

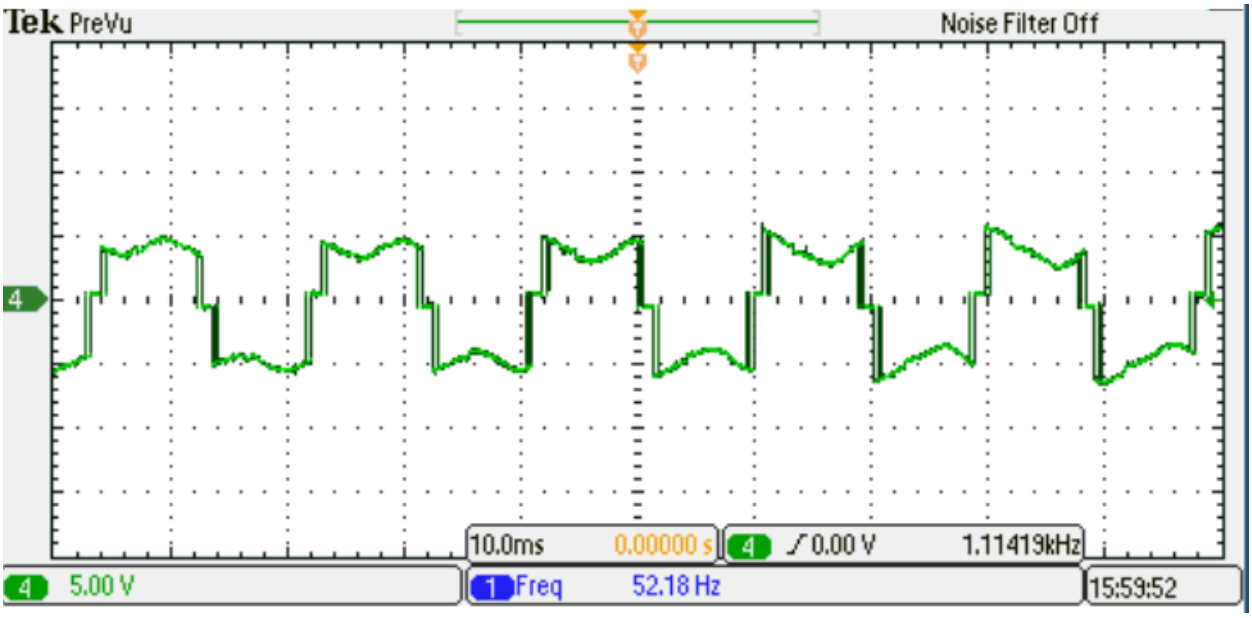


Figure 62: Phase Voltage for Ust = 1.1, SOM, fs = 52 Hz, N =3

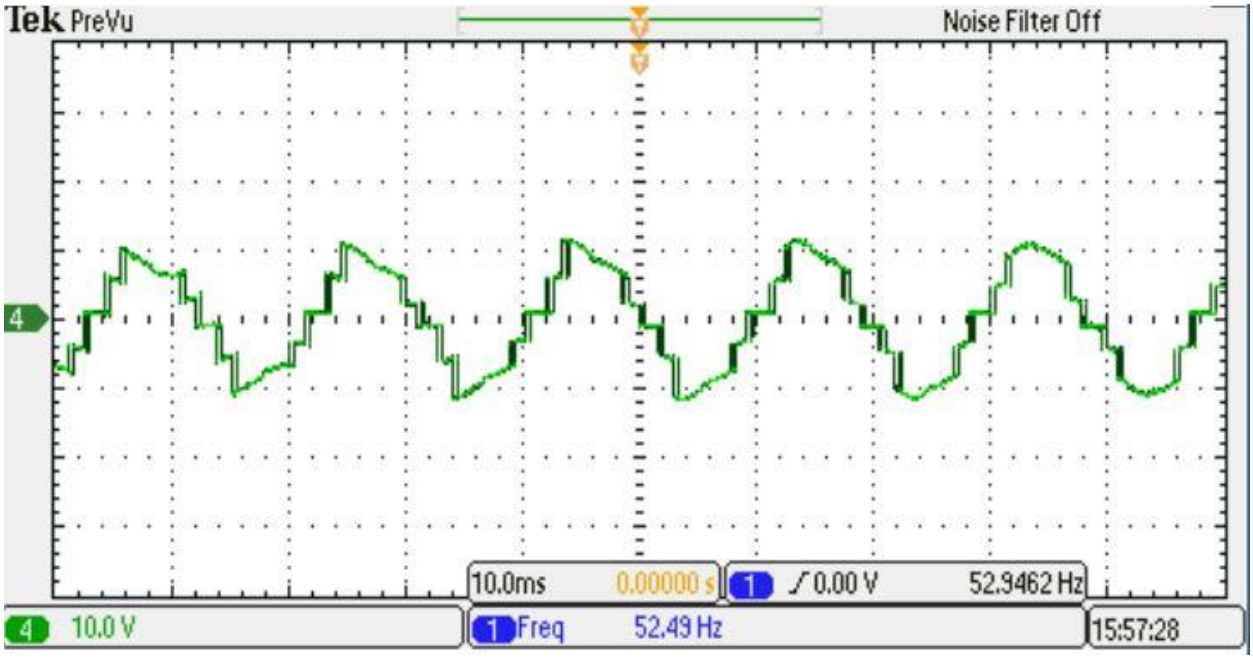


Figure 63: Line-Line Voltage for Ust = 1.1, SOM, fs = 52 Hz, N =3

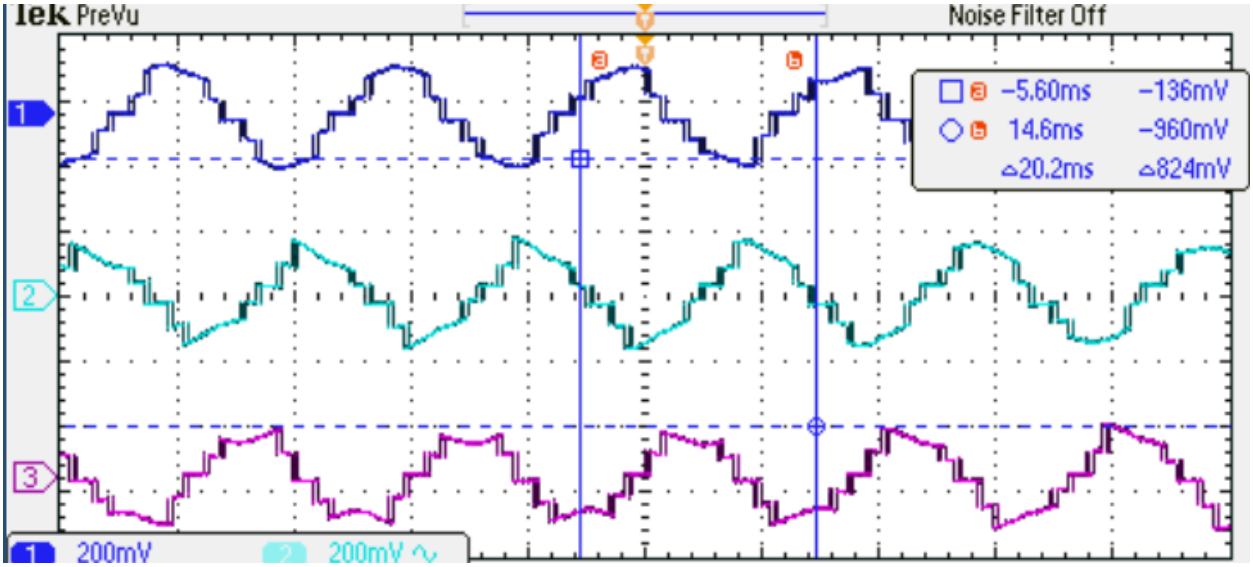


Figure 64: Line-Line Voltage for Ust = 1.1, SOM, fs = 50 Hz, N =3

This validates the fact of switching operation of the inverter performing as required except that the voltage ripple is quite high. The three R-Y-B signals with 120° phase shift is observed as it should be in the lab. The voltage results for all there instances (starting, space vector modulation and programmed modulation) were observe to be as expected except for a bit high ripple. The ripple during the space vector modulation is observed to be even higher while it remained almost constant with the programmed modulation. Since the timing of the clock cycle is managed well within the function of pulse generation and real time implementation, the three phase waveform are symmetrical. The implementation of programmed modulation in the over modulation range also showed the validity of symmetrical three phase output.

9.3.3 Current Signal

The voltage supplied to the induction machine, makes the flow of current into the induction machine. This current is again three phase current but with much lower harmonics on it as observed from the lab. The current waveform is found to be 120° phase shifted as it should be.

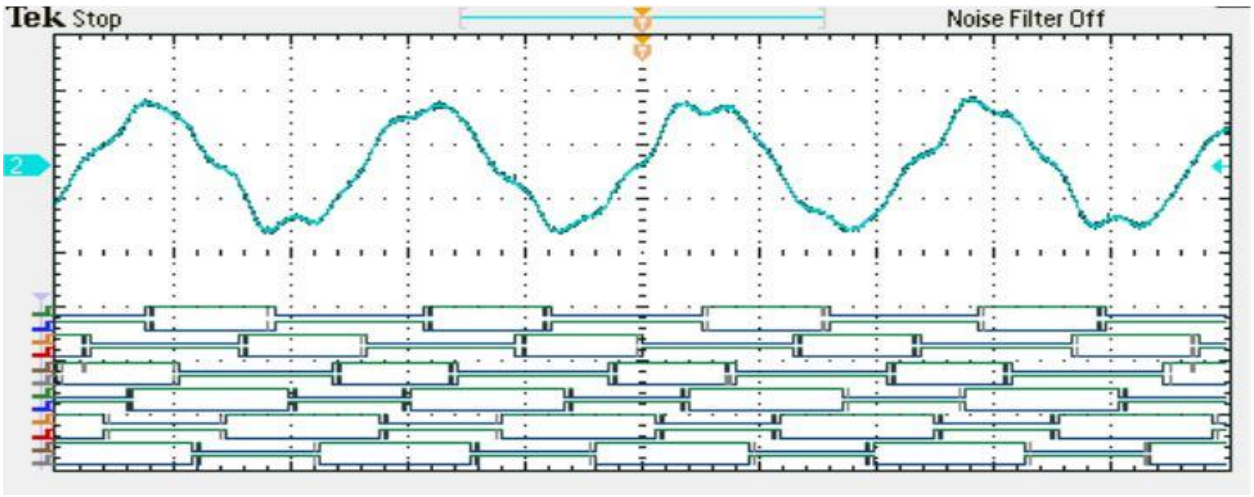


Figure 65: Phase current (up) and driver signal (down) with Ust = 1.1, SOM, fs= 50 Hz, N=3

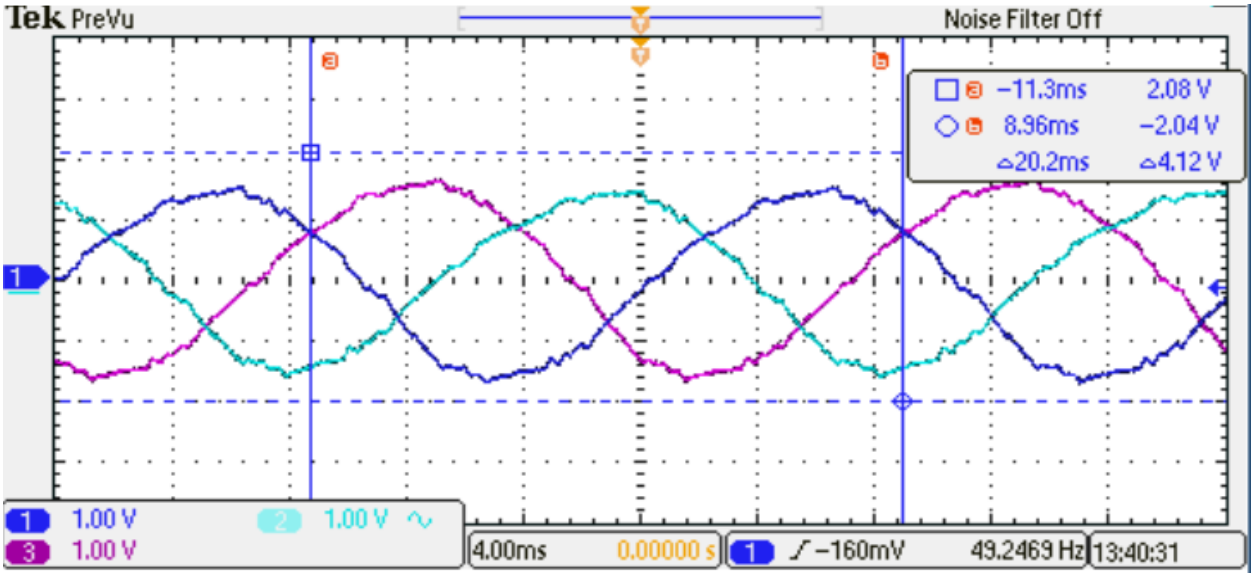


Figure 66: Three Phase currents with 120° phase shifted output, Ust = 0.65, SOM, N = 6

The current ripple during the space vector modulation is found to be quite high which is shown in Figure 67 below, but current ripple for programmed modulation is observed to be low as expected including in the overmodulation region where only three pulses per quarter cycle is used. In order to keep the switching frequency within 200 Hz, the number of switching pulses is limited per quarter cycle as already introduced in Chapter 5.

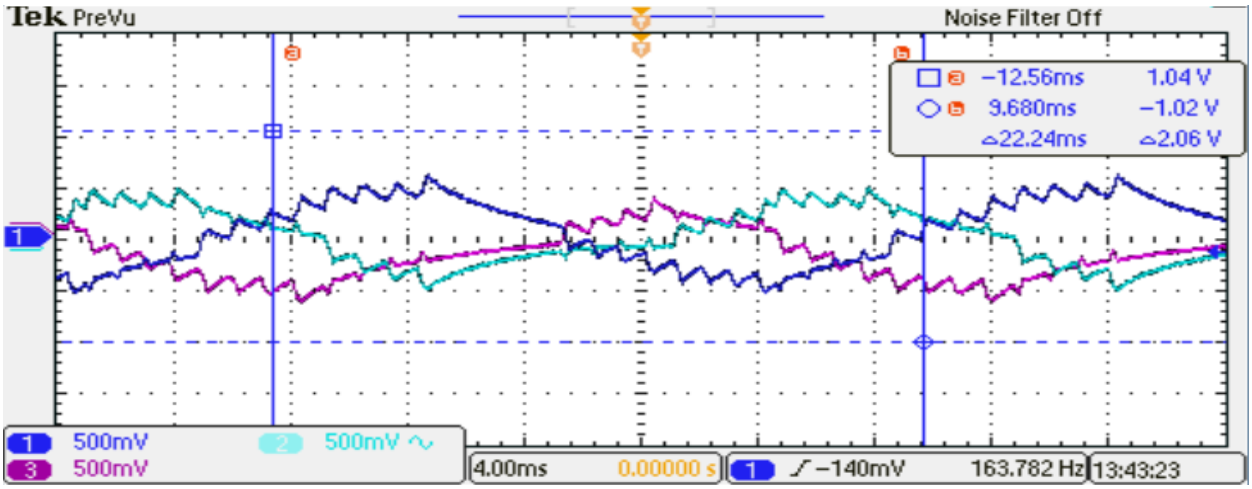


Figure 67: Three Phase currents with 120° phase shifted output, Ust = 0.22, SVPWM

9.4 Neutral Point Voltage

Neutral Point voltage is a major problem for the multiphase machine which increases harmonics, start-up failure, CMV and over-voltage across switching devices. The neutral point balancing approach is applied for both the SVPWM and SPM. The application on the SPM however is limited. The test in the lab is conducted for observing the Neutral Point Voltage and results below is observed.

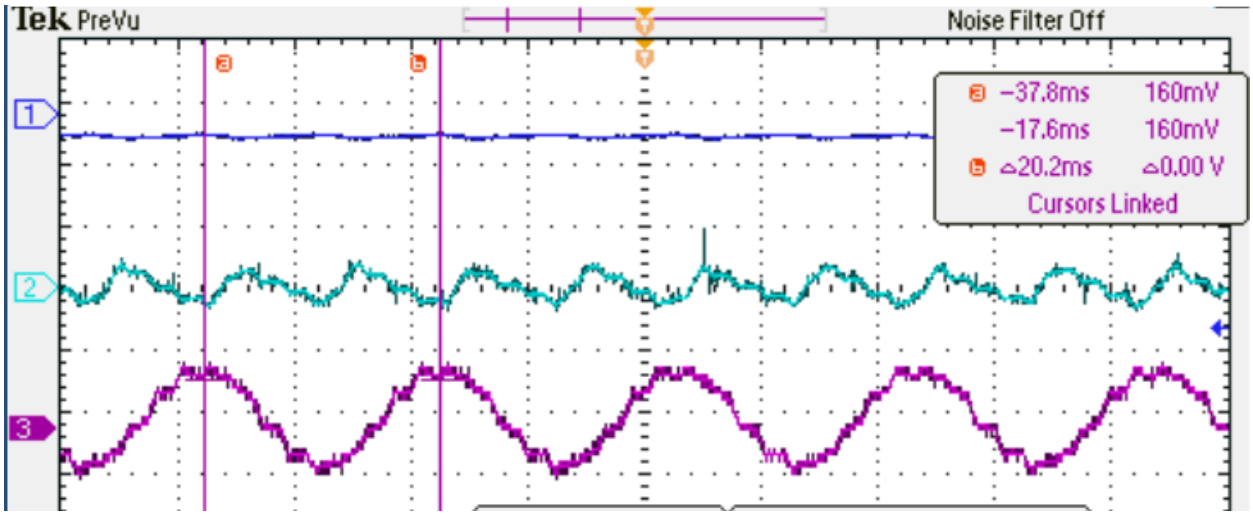


Figure 68: Current (down 3) and Capacitor voltages (1 and 2 up), Ust = 0.5, SOM

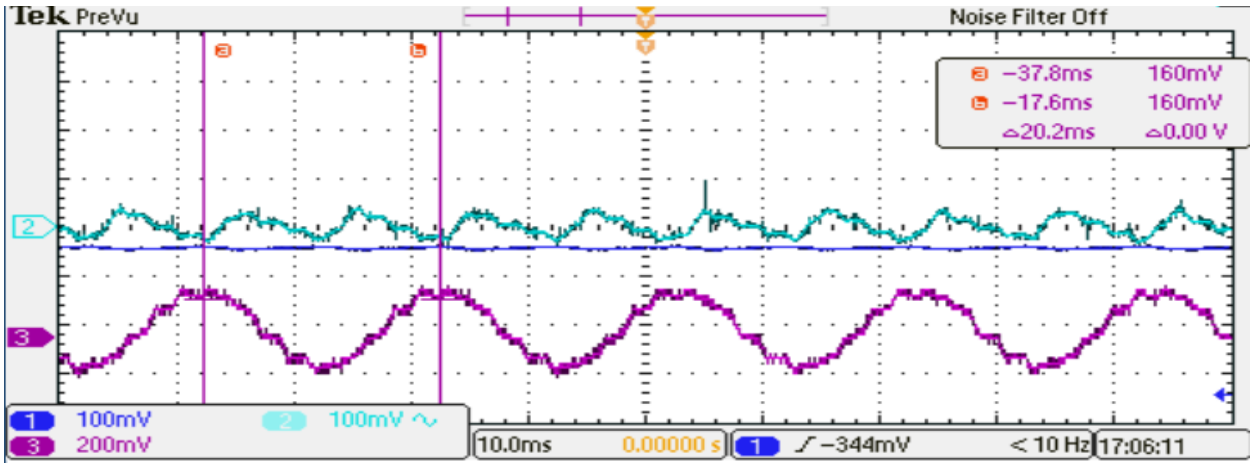


Figure 69: Current (down 3) and Capacitor voltages (1 and 2 together), Ust = 0.5, SOM

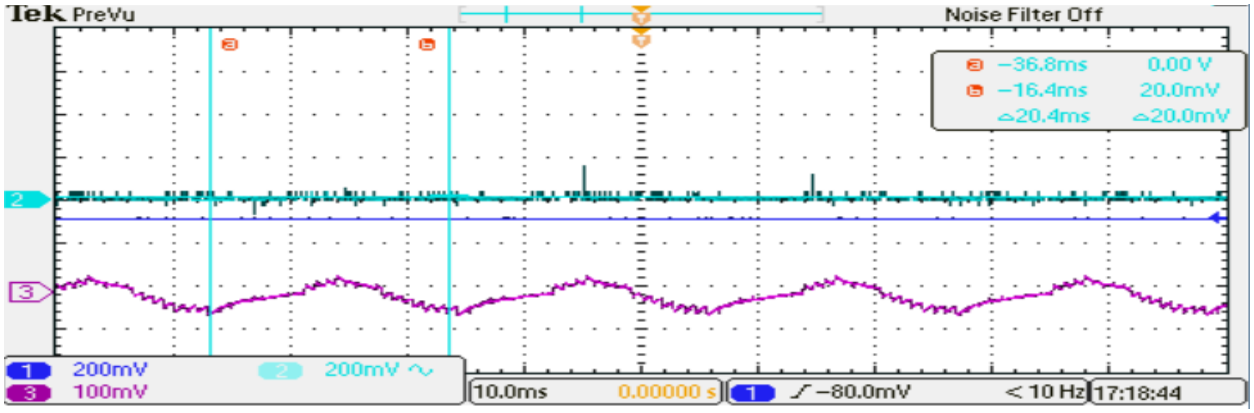


Figure 70: Current (down 3) and Capacitor voltages (1 and 2 together), Ust = 0.5, SVPWM

The ripple in DC link voltage is observed for programmed modulation and unbalanced DC link voltage is observed for space vector modulation. The technique used to make the DC link balancing is hence found to be inefficient. This is understandable with the fact that the harmonics for the space vector modulation is observed very high due to the low switching frequency. The problem with space vector modulation DC link balancing can be also because of the assumption that both the capacitors in the DC link being of same rating which in real can vary a bit. Also making the assumption in the feedback for calculation the timings of the pulses rather than actual feedback must have contributions in DC link problem. For smooth operation and dealing with DC link balance problem, it is thus advised to use the method [46] for programmed modulation which is already described in chapter 6 and proper feedback signals should be used for the space vector modulation.

9.5 Total Harmonic Distortion (THD)

The total harmonics distortion for both Voltage and Current are observed during the experiment. Both current and voltage harmonics for the space vector modulation with the low switching frequency of 200 Hz show higher total harmonic distortion as compared to the voltage and current for programmed modulation. 27% of Current harmonics and 67% of Voltage harmonics are observed for the space vector modulation as compared to 10% of current harmonics and 28% of voltage harmonics observed for programmed modulation with synchronous PWM. However, it is concluded that synchronous pulse width modulation with 200 Hz switching frequency is effective for low loss system as the harmonics is also reduced to the acceptable range. For space vector modulation it is essential to increase the switching frequency for the reduction of harmonics in the space vector region with current technique. This builds a question on a statement from [3] that states, harmonic losses for low modulation amplitude with space vector modulation becomes very close to its programmed counterparts. Due to the limitation that this thesis promises to build on switching frequency lower than 200 Hz, the consideration for higher switching frequency for reducing the harmonics in space vector modulation region is not considered.

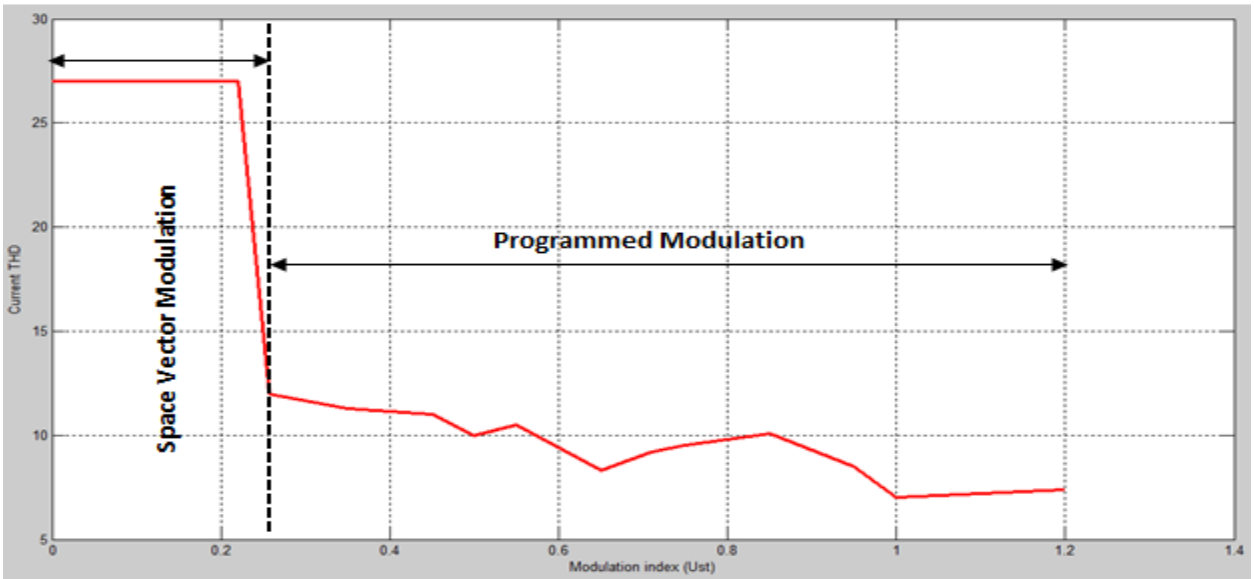


Figure 71: Harmonics in current vs the amplitude modulation level

Figure 72: Voltage harmonics with Ust = 0.65, fs= 50.2 Hz

It is noted that the odd harmonics is comparatively lowered as expected however the system considered the even harmonics not be existent and is not kept into the criterion of minimization. But, the result from the lab experiment showed that even harmonics such as 4^{th}, 6^{th}, and 8^{th} among the some are still dominant in the system. The need of dealing with the even harmonics is needed before the overall minimization can be obtained with the harmonics. Due to the scope and limitation of the project time, the even harmonics minimization is not considered in this thesis.

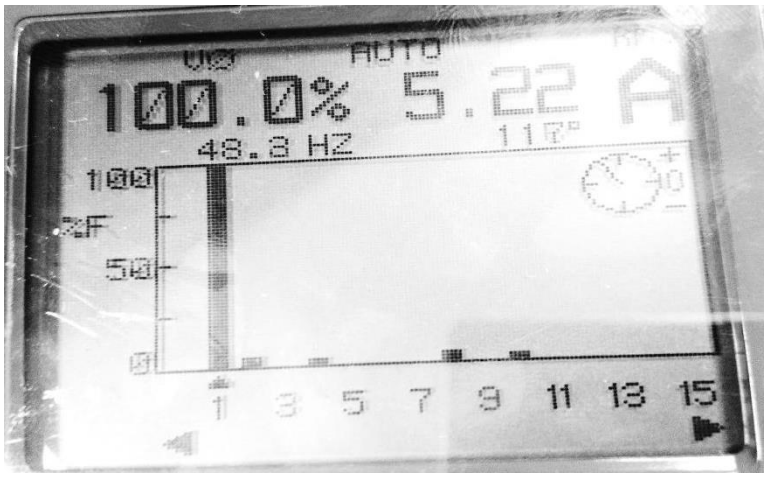


Figure 73: Voltage harmonics with Ust = 0.34, fs= 48.3 Hz

The even harmonics is also present in the current waveform even though it is in small amount. Since, Figure 66 and 67 shows that the current in three phase are symmetrical, only small amount of even and third order harmonics can be accepted. Simillarly,

Figure 64 shows the symmetry in three phase voltage. As such, the existance of even harmonics must be because of neutral point current and voltage build up.

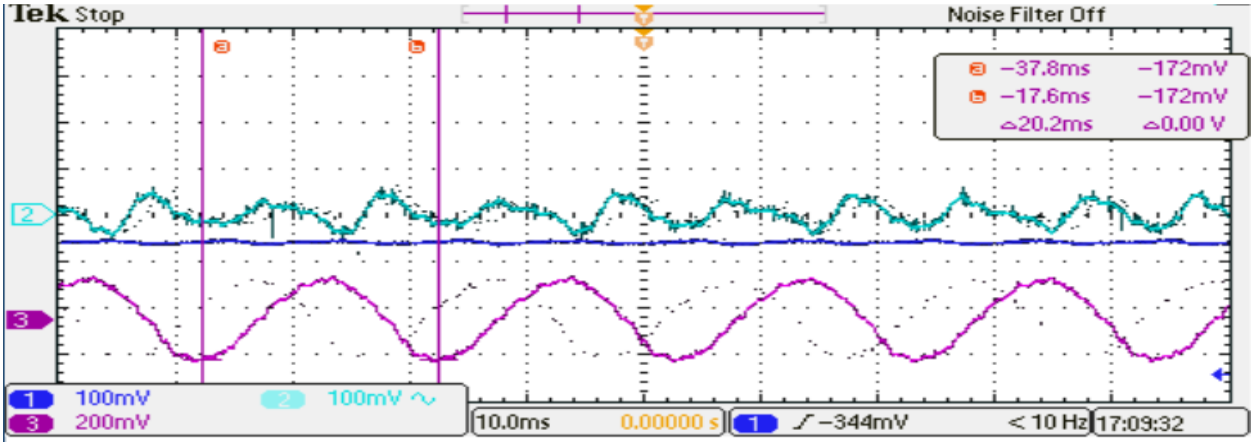


Figure 74: Current (down) and Capacitor voltages (1 and 2 up together), Ust = 0.6, SOM, 50 Hz

Figure 74 shows that there is imbalance of dc-link voltage which contirbuites to the even harmonics. Thus this even harmonics can be reduced by removing the DC-link imbalance. The approach of linieraization for smooth transisiton stated in Chapter 6 is not sufficient for the DC-link balancing and it is advised to use the method of time offset for future uses. Simillarly, the approach used for the space vector modulation as well needs upgrade to obtain low harmonics and DC link voltage.

Chapter 10
Conclusion and Future Works

In this chapter the summary of the approach to the method minimum loss synchronous PWM, its simulation and implementations and results is summarized. The regions for improvements and the techniques that is to be added to make the hybrid modulation work more effectively is detailed under the sub topic future works.

Medium voltage drives are replacing the Low voltage drives in the key areas where high power and low loss are needed. The introduction of multilevel inverter has facilitated the application of the medium voltage drives where the IGBT's are connected in series to build up voltage steps. Minimum loss synchronous refers to the system where the overall loss of the system is reduced. The key factors for these are the reduction in harmonics and switching frequency. Synchronous Optimal Modulation is the obvious choice as concluded from the results to get the switching loss down to 200 Hz while still having the harmonics in the order of acceptance. The optimal pulses patterns for the implementation of minimum loss synchronous PWM is generated implemented the technique of hybrid modulation with space vector modulation for low amplitude modulation and fundamental frequency of 50 Hz. For the entire propose the switching frequency is kept under 200 Hz.

The results from this master thesis prove the validity of synchronous optimal modulation technique as superior technique over conventional techniques such as space vector modulation and sinusoidal modulation techniques. The system is implemented for the three phase drive of six phase machine, but can be easily converted to make six phase implementation with addition of an additional three phase inverter to the system. In fact, once the synchronous PWM is generated for the single phase, it can be converted in real time to any number of phase machines such as six phase machine and nine phase machine. The total harmonic distortion with current waveform is observed to be very low however for the voltage the unacceptable amount of harmonics is found

to be introduced by even harmonics. The even harmonics is not considered during the criterion function. Also the low switching frequency contributed to the higher voltage harmonics for the space vector modulation. The DC link voltage is not balanced as expected from the theory and should contribute to the harmonics. For the same level of switching frequency the voltage harmonics in synchronous modulation is observed to much lower than compared to the space vector modulation.

The technique of Neutral point balancing is not well implemented in the system except the linearization of the switching patterns whereas for space vectors the assumption of parameters to neutralize the DC link voltage. The results as described in Chapter 9, however is found to be unacceptable for the programmed synchronous modulation with voltage ripple while the space vector modulation implementation showed constant DC link voltage difference. The more advanced technique of time offset method as described in Chapter 6 is thus advised for the implementation on overall system for future implementation.

10.1 Further Works

This project work leading to thesis has successfully implemented the hybrid modulation topology with space vector modulation and programmed synchronous modulation; however there are some key factors that could be improved to make this system more reliable and industrial friendly. The key areas of improvement that can fallow are as follows:

1. Use of parallel processing to generate the pulse sequence instead of serial.
2. Neutral point balancing has been addressed in this thesis to some extend but is not found to be reliable (refer Chapter 9) and switch to the approach is need. It is advised to implement the time-offset method as described in Chapter 6.
3. Starting of the machine has been addressed to some extend with trail methods. This however can be improved significantly with more scientific methods.
4. Low pass filter to eliminate the higher harmonics can be implemented in the FPGA processor to reduce the level of harmonics.

5. The space vector modulation for low amplitude is found to have higher harmonics; as such it can be improved with recent techniques or can be used with high switching frequency. The implementation of 5 KHz switching frequency is tested with the simulation (Refer chapter 7) and low harmonics content is achieved. This can be the alternative for implementation. Refer Appendix 1 for the Simulink model.

Bibliography/References

[1] Singh G.K., "Multi-phase induction machine drive research-a survey", *Electrical Power System Research*, vol. 61, pp. 139-147, March 28, 2002.

[2] CHORUS^{TM} MOTORS PLC, " http://www.chorusmotors.com/demo/index.demo.html," *online source*, unknown date.

[3] 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.

[4] A. Nabae, I. Takahashi, and H. Akagi, "A new neutral-point-clamped PWM inverter," *IEEE Trans. Industrial Applications*, Vol IA-17, No. 5, Sept/Oct. 1981.

[5] E. Levi; Bojoi I.R.; F. Profumo; H.A. Toliyat; S. Williamson, "Multiphase induction motor drives - a technology status review" *In: IET ELECTRIC POWER APPLICATIONS*, vol. 1 n. 4, pp. 489-516. - ISSN 1751-8660, 2007.

[6] Zhao, Y.; Lipo, T.A., "Modeling and control of a multi-phase induction machine with structural unbalance," *Energy Conversion, IEEE Transactions on* , vol.11, no.3, pp.570,577, Sep 1996 doi: 10.1109/60.537009.

[7] Holtz, J., "Pulsewidth modulation-a survey," *Power Electronics Specialists Conference*, 1992. PESC '92 Record., 23rd Annual IEEE , vol., no., pp.11,18 vol.1, 29 Jun-3 Jul 1992 doi: 10.1109/PESC.1992.254685.

[8] Akshay K. Rathore, J. Holtz and Till Boller, "Synchronous Optimal Pulsewidth Modulation for Low-Switching-Frequency Control of Medium-Voltage Multilevel Inverters", *IEEE Trans. Industrial Electronics*, Vol. 57, No. 7, July 2010.

[9] Li Li; Czarkowski, D.; Yaguang Liu; Pillay, P., "Multilevel selective harmonic elimination PWM technique in series-connected voltage inverters," *Industry Applications, IEEE Transactions on* , vol.36, no.1, pp.160,170, Jan/Feb 2000.

[10] Agelidis, V.G.; Balouktsis, A.I.; Dahidah, Mohamed S A, "A Five-Level Symmetrically Defined Selective Harmonic Elimination PWM Strategy: Analysis and Experimental Validation," *Power Electronics, IEEE Transactions on* , vol.23, no.1, pp.19,26, Jan. 2008.

[11] J. Holtz and N. Oikonomou, "Synchronous Optimal Pulsewidth Modulation and Stator Flux Trajectory Control for Medium-Voltage Drives", *IEEE Trans. Industrial Electronics*, Vol. 43, No. 2, March/April 2007.

[12] J. Holtz, "Field-Oriented Asynchronous Pulse-Width Modulation for High-Performance ac Machine Drives Operating at Low Switching Frequency," *IEEE Trans. Industrial Electronics*, Vol. 27, No. 3, May/June 1991.

[13] J. Rodr´ıguez, S. Kouro, J. Rebolledo and J. Pontt, "A Reduced Switching Frequency Modulation Algorithm for High Power Multilevel Inverters," *IEEE Trans. On I.E.*, Vol. 54, NO. 5, Oct. 2007.

[14] Enjeti, P.N.; Ziogas, P.D.; Lindsay, J.F., "Programmed PWM techniques to eliminate harmonics - A critical evaluation," *Industry Applications Society Annual Meeting*, 1988., Conference Record of the 1988 IEEE , vol., no., pp.418,430 vol.1, 2-7 Oct. 1988.

[15] Nikola C., Dushan B.," A Comprehensive Study of Neutral-Point Voltage Balancing Problem in Three-Level Neutral-Point-Clamped Voltage Source PWM Inverters", *IEEE Trans. Power Electronics*, Vol. 15, No. 2, MARCH 2000.

[16] Bo Gong, Shanmei C., Yi Qin, "Simple three-level neutral point voltage balance control strategy based on SVPWM*", Archives on Electrical Engineering*, Vol. 62(1), pp. 15-23 (2013).

[17] Krishna Neupane, "Control of Medium Voltage Multi-phase Machines: Minimum Loss Synchronous Optimal Modulation with Neutral Point Balancing", *Specialization project*, NTNU, December 2013.

[18] Roger Enes, “Modelling and Control of High Performance Medium Voltage Drives - Simulation and analysis of the Programmed Modulation strategy”, *Master Thesis*, NTNU, June 2012.
[19] Roy Nilsen “Modulation Methods for three levels Inverter,” *Medium Voltage Drives*, Wärtsilä Norway AS, Confidential, Feb., 2010.
[20] Mamata Maharjan, “Synchronous Optimal Modulation for medium voltage multi-phase machines – implementation of three-level modulators in Field Programmable Gate Arrays*”, Mater Thesis*, NTNU, June 2013.
[21] Biruk Bekele Yenore, “Low switching frequency modulation scheme for high power three level converter – FPGA based implementation”, *Master Thesis*, NTNU, June 2013.
[22] H. S. Patel and R. G. Hoft, “Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part I--Harmonic Elimination,” *Industry Applications*, IEEE Transactions on, vol. IA-9, pp. 310-317, 1973.
[23] G. S. Buja and G. B. Indri, "Optimal Pulsewidth Modulation for Feeding AC Motors," *Industry Applications, IEEE Transactions on*, vol. IA-13, pp. 38-44, 1977.
[24] J. M. D. Murphy, L. S. Howard, and R. G. Hoft, “Microprocessor control of a PWM inverter induction motor drive,” *Institute of Electrical and Electronics Engineers*, Inc, pp. 344-348, 00/1979.
[25] ABB Marine Systems and Solution “http://www.abb.com/industries/db0003db002805/c12571f4002ab83dc1256fdf003b2929.aspx,” *Azipod® Propulsion*, © Copyright 2014 ABB.
[26] Wärtsilä, “http://www.wartsila.com/en/power-electric-systems/electric-propulsion-packages/electric-propulsion,” *Wärtsilä Low Loss Propulsion system*, © 2014 Wärtsilä.
[27] Siemens Solution for Podded Propulsion, “http://www.industry.siemens.com/verticals/global/en/marine/cargo_ships/propulsion/siship_ssp/Pages/Default.aspx,” *eSiPOD*, © Siemens AG 1996-2014
[28] Williamson, S.; Smith, S., "Pulsating torque and losses in multiphase induction machines," *Industry Applications Conference*, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE , vol.2, no., pp.1155,1162 vol.2, Sept. 30 2001-Oct. 4 2001.
[29] Archana Nanoty and A. R. Chudasama, “Control of Designed Developed Six Phase Induction Motor,” *International Journal of Electromagnetics and Applications*, p-ISSN: 2168-5037, e-ISSN: 2168-5045, 2012.
[30] R. H. Nelson and P. C. Krause, “Induction Machine Analysis for Arbitrary Displacement Between Multiple Winding Sets,” *Power Apparatus and Systems*, IEEE Transactions on, vol. PAS-93, pp.841-848, 1974.
[31] Emil Levi, “Recent Developments in High Performance Variable-Speed Multiphase Induction Motor Drives” *Sixth International Symposium Nikola Tesla*, Belgrade, Serbia. 18th – 20th October, 2006.
[32] Ned Mohan, Tore M. Undeland, William P. Robbins, “Power Electronics: Converters, Applications, and Design, 3rd Edition”, *John Wiley & Sons, Inc*., 2003, ISBN: 978-0-471-22693-2.
[33] Surin Khomfoi and Leon M. Tolbert, “Multilevel Power Converter,” *The University of Tennessee*, Tennessee, USA.
[34] T.A. Meynard, H. Foch, “Multi-level conversion: High voltage choppers and Voltage-Source Inverters,” *IEEE Power Electronics Specialists Conference*, 1992.
[35] Richard Lund, “Multilevel Power Electronic Converters for Electrical Motor Drives”, *PHD Thesis*, NTNU, Trondheim, April 2005.
[36] Johann W. Kolar, Has Ertl, and Franz C. Zach, “Influence of the Modulation Method on the Conduction and Switching Losses of a PWM Converter System”, *Trans. On Industrial Applications*, Vol. 27, No. 6, Nov/Dec 1991.
[37] Park, R.H., “Two-Reaction Theory of Synchronous Machines”, *AIEE Trans*. No.48, 1929, pp.716-730 and no. 52, 1933, pp.352-355.
[38] Kron, G., “The Application of Tensors to the Analysis of Rotating Electrical Machinery”, Schenectady, NY, USA, General Electric Review, 1942
[39] Kovacs, K.P., Racz, I., “Transiente Vorgange in Wechselstrommachinen”, Budapest, Hungary, Akad.Kiado, 1959.
[40] Kolar J.W., Ert H., Zach F.C. “Calculation of the passive and active component stress of three phase PWM converter system with high pulse rate.” *EPE Aachen* 1989.
[41] Leonhand W., “Control of Electrical Drives”, 3rd Edition, Springer-Verlag, New York.

[42] Javier Chivite-Zabalza, Pedro Izurza-Moreno, Danel Madariaga, Gorka Calvo,and Miguel Angel Rodr´ıguez, "Voltage Balancing control in 3-Level Neutral-Point Clamped Inverters Using Triangular Carrier PWM Modulation for FACTS Applications", *IEEE Trans. On Power Electronics*, Vol. 28, No. 10, Oct 2013.

[43] Joachim Holtz and Nikolaos Oikonomou, "Neutral Point Potential Balancing Algorithm at Low Modulation Index for Three-Level Inverter Medium-Voltage Drives", *IEEE Tran. of Industrial Applications*, Vol. 43, No. 3, May/June 2007.

[44] Lazhar Ben-Brahim, "A Discontinuous PWM Method for Balancing the Neutral Point Voltage in Three-Level Inverter-Fed Variable Frequency Drives", *IEEE Trans. On Energy Conservation*, Vol. 23, No. 4, December 2008.

[45] Prasad N. Enjeti and Wajiha Shireen, "A New Technique to Reject DC-Link Voltage Ripple for Inverters Operating on Programmed Waveforms*", IEEE Trans. On P.E*., Vol. 7, No. 1, Jan 1992.

[46] Ui-Min Choi and Kyo-Beum Lee, "Neutral Point Voltage Balancing Method for Three-Level Inverter Systems with a Time-Offset Estimation Scheme", *Journal of Power Electronics*, Vol. 13, No. 2, March 2013

[47] Akshay K. Rathore, Joachim Holtz and Till Boller, "Generalized Optimal Pulsewidth Modulation of Multilevel Inverters for Low-Switching-Frequency Control of Medium-Voltage High-Power Industrial AC Drives", *IEEE Trans on Industrial Electronics*, Vol. 60, No. 10, Oct 2013.

[48] Deng Y., Teo K.H. and Harley R.G. "Generalized DC-link Voltage Balancing Control Method for Multilevel Inverters", *Mitsubishi Electric Research Laboratories*, TR2013-005, March 2013.

[49] Xilinx Inc., "http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm," *Field Programmable Gate Array*, © Copyright 2014 Xilinx Inc.

[50] Xilinx Inc., "http://www.xilinx.com/products/design-tools/ise-design-suite," *ISE Design Suite*, © Copyright 2014 Xilinx Inc.

[51] K. Ljøkelsøy, "FPGA based processor board for control of power electronics converters," *SINTEF*, Norway, Technical report, December 12, 2008.

[52] Volnei A. Pedroni, "Circuit Design and Simulation with VHDL," The Mit Press, Cambridge, Massachusetts, Second Edition, 2004.

[53] K. Ljøkelsøy, "Prosessorkort basert på Xilinx Virtex5 FPGA. V1. 1. Beskrivelse," *SINTEF*, Norway, Technical report, February 12, 2009.

Appendix

Appendix 1

SIX PHASE INDUCTION MOTOR SPECIFICATION

Table 5: Specification of Six phase Induction Motor

Parameter	Explanation	Value
U_N	Nominal line to line voltage [V_{rms}]	**400 V**
I_N	Nominal line current [A_{rms}]	**11.8 A**
f_N	Nominal frequency [Hz]	**75 Hz**
p	Number of pole pairs	**2**
N	Nominal speed [mechanical rpm]	**2235 rpm**
M_N	Nominal output torque [Nm]	**50 Nm**
P_N	Nominal power output [kW]	**11.7 kW**
cosϕ	Nominal power factor	**0.77**
n_{MAX}	Maximum speed [mechanical rpm]	**5000 rpm**

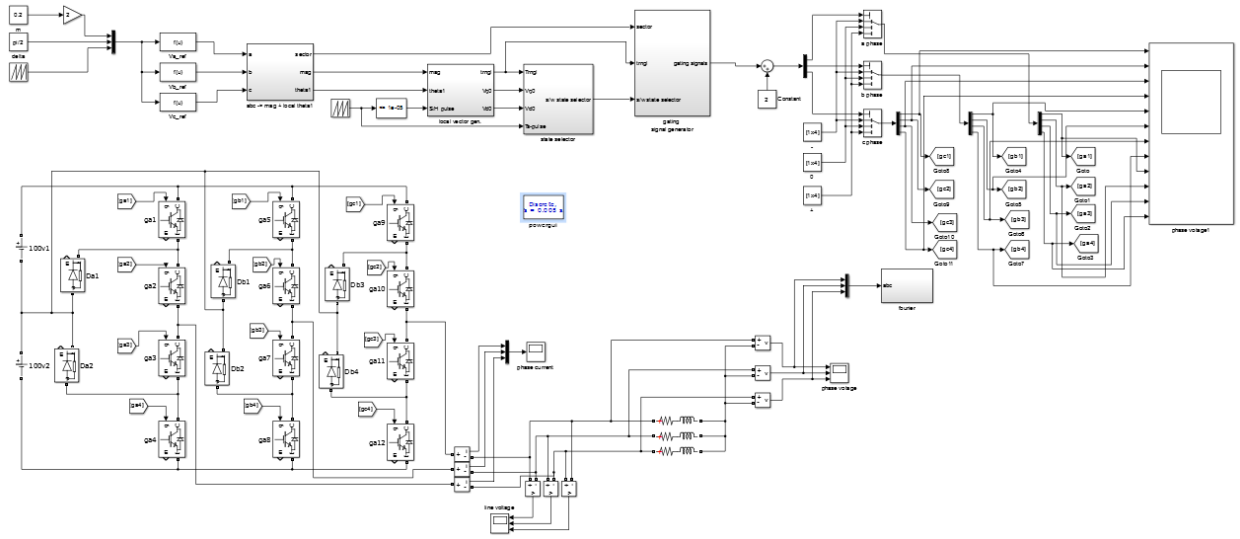


Figure 75: Simulink Model for Space Vector Modulation

Appendix 2

MATLAB CODE FOR PM, LINEARIZED PATTERNS AND HARMONICS CHECK

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PULSE PATTERN GENERATION, PROGRAMMED MODUALTION                              %
% Based on pre-calculated values for differnt number of pulses                 %
% Written by Krishna Neupane                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
clc;
close all;

% '''''''''''''''''''' N = 15
ustmax=0.266;
l=0.250:0.001:ustmax;
n=length(l);

for i=1:n
    f(1,i)=0.0;
    % N = 15
    x0(1,i) = 1.0000*pi/180;
    x0(2,i) = 2.0000*pi/180;
    x0(3,i) = (3.5588+((3.5939-3.5588)/(0.266-0.250))*(l(i)-0.250))*pi/180;
    x0(4,i) = (4.5587+((4.5939-4.5587)/(0.266-0.250))*(l(i)-0.250))*pi/180;
    x0(5,i) = (5.6432+((5.6094-5.6432)/(0.266-0.250))*(l(i)-0.250))*pi/180;
    x0(6,i) = (6.6700+((6.7329-6.6700)/(0.266-0.250))*(l(i)-0.250))*pi/180;
    x0(7,i) = (13.4969+((13.4674-13.4969)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(8,i) = (17.3504+((17.6012-17.3504)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(9,i) = (37.0737+((37.0993-37.0737)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(10,i) = (38.4213+((38.4815-38.4213)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(11,i) = (42.9992+((42.7454-42.9992)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(12,i) = (47.4189+((47.4929-47.4189)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(13,i) = (53.2151+((53.0890-53.2151)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(14,i) = (57.4092+((57.5475-57.4092)/(0.266-0.250))*(l(i)-
0.250))*pi/180;
    x0(15,i) = (87.3895+((87.2295-87.3895)/(0.266-0.250))*(l(i)-
0.250))*pi/180;

        for c=1:600
              u_h1 = 0.0;
              u_h2 = 0.0;
                   for j=1:15
```

```
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
            end

               f(1,i)=sqrt(f(1,i));

end
    x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;
    x0(11,:) = x0(11,:)*180/pi;
    x0(12,:) = x0(12,:)*180/pi;
    x0(13,:) = x0(13,:)*180/pi;
    x0(14,:) = x0(14,:)*180/pi;
    x0(15,:) = x0(15,:)*180/pi;

figure;
                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                              l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),...
                              l,x0(11,:),l,x0(12,:),l,x0(13,:),l,x0(14,:),...
                              l,x0(15,:),'Linewidth',2);
hold on;
                            grid;
                            axis([0.25 1.22 0.0 95]);
                            xlabel('modulation index   u_s_t');
                            ylabel(' \alpha_1 ...\alpha_x [degrees]');
                            title('Loss optimal PWM for NPC inverter with
N=variable');


%
ustmax=0.285;
l=0.266:0.001:ustmax;
n=length(l);


for i=1:n
    f(1,i)=0.0;
    % N = 14
    x0(1,i) = (7.9717+((7.8903-7.9717)/(0.285-0.266))*(l(i)-0.266))*pi/180;
    x0(2,i) = (10.8882+((11.0487-10.8882)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
```

```
    x0(3,i) = (14.8439+((14.8419-14.8439)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(4,i) = (15.8439+((15.8419-15.8439)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(5,i) = (19.7772+((19.6013-19.7772)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(6,i) = (22.9432+((23.0198-22.9432)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(7,i) = (28.3363+((28.2111-28.3363)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(8,i) = (31.7472+((31.8903-31.7472)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(9,i) = (36.9809+((36.8878-36.9809)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(10,i) = (41.3838+((41.5845-41.3838)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(11,i) = (47.5576+((47.3395-47.5576)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(12,i) = (52.1327+((52.2644-52.1327)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(13,i) = (56.1183+((56.1210-56.1183)/(0.285-0.266))*(l(i)-
0.266))*pi/180;
    x0(14,i) = (58.6010+((58.7625-58.6010)/(0.285-0.266))*(l(i)-
0.266))*pi/180;

        for c=1:600
              u_h1 = 0.0;
              u_h2 = 0.0;
                   for j=1:14
                       u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                       u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                   end
                   u_h1=u_h1*4/(pi*(6*c-1));
                   u_h2=u_h2*4/(pi*(6*c+1));
                   f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
        end

           f(1,i)=sqrt(f(1,i));

end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;
    x0(11,:) = x0(11,:)*180/pi;
    x0(12,:) = x0(12,:)*180/pi;
    x0(13,:) = x0(13,:)*180/pi;
    x0(14,:) = x0(14,:)*180/pi;
%     x0(15,:) = x0(15,:)*180/pi;
```

```
                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                              l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),...

l,x0(11,:),l,x0(12,:),l,x0(13,:),l,x0(14,:),'Linewidth',2);
hold on;


ustmax=0.307;
l=0.285:0.001:ustmax;
n=length(l);


for i=1:n
    f(1,i)=0.0;
    % N = 13
    x0(1,i) = (1.0010+((1.0104-1.0010)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(2,i) = (2.0010+((2.0104-2.0010)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(3,i) = (4.0111+((4.0150-4.0111)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(4,i) = (5.5834+((5.6761-5.5834)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(5,i) = (7.3036+((7.1876-7.3036)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(6,i) = (8.3998+((8.4190-8.3998)/(0.307-0.285))*(l(i)-0.285))*pi/180;
    x0(7,i) = (14.6247+((14.5661-14.6247)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(8,i) = (18.8326+((19.0975-18.8326)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(9,i) = (40.5808+((40.2610-40.5808)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(10,i) = (46.3642+((46.4652-46.3642)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(11,i) = (52.1637+((51.8750-52.1637)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(12,i) = (57.4294+((57.5962-57.4294)/(0.307-0.285))*(l(i)-
0.285))*pi/180;
    x0(13,i) = (86.9704+((86.7562-86.9704)/(0.307-0.285))*(l(i)-
0.285))*pi/180;


        for c=1:600
              u_h1 = 0.0;
              u_h2 = 0.0;
                  for j=1:13
                      u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                      u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                  end
                  u_h1=u_h1*4/(pi*(6*c-1));
                  u_h2=u_h2*4/(pi*(6*c+1));
                  f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
        end
           f(1,i)=sqrt(f(1,i));
end

  x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
```

```
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;
    x0(11,:) = x0(11,:)*180/pi;
    x0(12,:) = x0(12,:)*180/pi;
    x0(13,:) = x0(13,:)*180/pi;
    x0(14,:) = x0(14,:)*180/pi;
%     x0(15,:) = x0(15,:)*180/pi;


                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                              l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),...

l,x0(11,:),l,x0(12,:),l,x0(13,:),'Linewidth',2);
hold on;


ustmax=0.333;
l=0.307:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 13
    x0(1,i) = (4.0228+((3.9199-4.0228)/(0.333-0.307))*(l(i)-0.307))*pi/180;
    x0(2,i) = (6.9026+((7.0268-6.9026)/(0.333-0.307))*(l(i)-0.307))*pi/180;
    x0(3,i) = (13.2923+((13.1671-13.2923)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(4,i) = (17.0234+((17.2145-17.0234)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(5,i) = (22.9983+((22.8058-22.9983)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(6,i) = (27.3200+((27.4938-27.3200)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(7,i) = (32.5711+((32.3901-32.5711)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(8,i) = (37.3407+((37.5673-37.3407)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(9,i) = (42.3986+((42.1776-42.3986)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(10,i) = (47.5070+((47.7192-47.5070)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(11,i) = (52.2340+((52.0348-52.2340)/(0.333-0.307))*(l(i)-
0.307))*pi/180;
    x0(12,i) = (57.5990+((57.8478-57.5990)/(0.333-0.307))*(l(i)-
0.307))*pi/180;

         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
```

```
                    for j=1:12
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
        end

           f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;
    x0(11,:) = x0(11,:)*180/pi;
    x0(12,:) = x0(12,:)*180/pi;


                          plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                            l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                            l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),...
                            l,x0(11,:),l,x0(12,:),'Linewidth',2);
hold on;


ustmax=0.363;
l=0.333:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 13
    x0(1,i) = (4.0628+((3.9805-4.0628)/(0.363-0.333))*(l(i)-0.333))*pi/180;
    x0(2,i) = (7.4340+((7.6371-7.4340)/(0.363-0.333))*(l(i)-0.333))*pi/180;
    x0(3,i) = (14.3073+((14.1606-14.3073)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(4,i) = (19.4421+((19.7536-19.4421)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(5,i) = (37.6693+((37.6061-37.6693)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(6,i) = (39.2992+((39.3298-39.2992)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(7,i) = (41.3243+((41.0500-41.3243)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(8,i) = (46.8267+((47.0537-46.8267)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
```

```
    x0(9,i) = (51.5779+((51.2829-51.5779)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(10,i) = (57.7612+((58.0387-57.7612)/(0.363-0.333))*(l(i)-
0.333))*pi/180;
    x0(11,i) = (86.7233+((86.4157-86.7233)/(0.363-0.333))*(l(i)-
0.333))*pi/180;

         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:11
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;


                           plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                             l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                             l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),...
                             l,x0(11,:),'Linewidth',1.2);
hold on;

%
ustmax=0.400;
l=0.363:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 13
     x0(1,i) = (5.132+((3.635-5.132)/0.35)*(l(i)-0.05))*pi/180;
     x0(2,i) = (5.843+((8.009-5.843)/0.35)*(l(i)-0.05))*pi/180;
     x0(3,i) = (18.160+((14.666-18.160)/0.35)*(l(i)-0.05))*pi/180;
     x0(4,i) = (18.829+((21.256-18.829)/0.35)*(l(i)-0.05))*pi/180;
     x0(5,i) = (29.090+((26.630-29.090)/0.35)*(l(i)-0.05))*pi/180;
     x0(6,i) = (29.937+((33.464-29.937)/0.35)*(l(i)-0.05))*pi/180;
     x0(7,i) = (41.962+((38.028-41.962)/0.35)*(l(i)-0.05))*pi/180;
```

```
     x0(8,i) = (43.030+((46.665-43.030)/0.35)*(l(i)-0.05))*pi/180;
     x0(9,i) = (54.983+((50.770-54.983)/0.35)*(l(i)-0.05))*pi/180;
     x0(10,i) = (55.992+((58.489-55.992)/0.35)*(l(i)-0.05))*pi/180;

         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:10
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
    x0(10,:) = x0(10,:)*180/pi;


                           plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                             l,x0(4,:),l,x0(5,:),l,x0(6,:),...

l,x0(7,:),l,x0(8,:),l,x0(9,:),l,x0(10,:),'Linewidth',2);
hold on;


ustmax=0.444;
l=0.400:0.001:ustmax;
n=length(l);


 for i=1:n
    f(1,i)=0.0;
    % N = 13
     x0(1,i) = (1.6851+((1.8617-1.6851)/(0.444-0.400))*(l(i)-0.400))*pi/180;
     x0(2,i) = (2.6859+((3.1364-2.6859)/(0.444-0.400))*(l(i)-0.400))*pi/180;
     x0(3,i) = (6.6999+((6.3959-6.6999)/(0.444-0.400))*(l(i)-0.400))*pi/180;
     x0(4,i) = (11.3021+((11.0411-11.3021)/(0.444-0.400))*(l(i)-
0.400))*pi/180;
     x0(5,i) = (13.4353+((13.3375-13.4353)/(0.444-0.400))*(l(i)-
0.400))*pi/180;
     x0(6,i) = (17.0799+((17.6476-17.0799)/(0.444-0.400))*(l(i)-
0.400))*pi/180;
```

```
     x0(7,i) = (42.6305+((41.8187-42.6305)/(0.444-0.400))*(l(i)-
0.400))*pi/180;
     x0(8,i) = (56.7750+((57.5809-56.7750)/(0.444-0.400))*(l(i)-
0.400))*pi/180;
     x0(9,i) = (84.4690+((83.9025-84.4690)/(0.444-0.400))*(l(i)-
0.400))*pi/180;

         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:9
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;
    x0(9,:) = x0(9,:)*180/pi;
%     x0(10,:) = x0(10,:)*180/pi;


                           plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                             l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                             l,x0(7,:),l,x0(8,:),l,x0(9,:),'Linewidth',2);
hold on;

ustmax=0.500;
l=0.444:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 8
     x0(1,i) = (12.5672+((10.9988-12.5672)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(2,i) = (19.4729+((19.0186-19.4729)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(3,i) = (25.0861+((23.8928-25.0861)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(4,i) = (35.2381+((36.4225-35.2381)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
```

```
     x0(5,i) = (39.6658+((39.7053-39.6658)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(6,i) = (50.5692+((52.5459-50.5692)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(7,i) = (60.7336+((60.2714-60.7336)/(0.500-0.444))*(l(i)-
0.444))*pi/180;
     x0(8,i) = (66.6129+((65.7782-66.6129)/(0.500-0.444))*(l(i)-
0.444))*pi/180;


         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:8
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;
    x0(8,:) = x0(8,:)*180/pi;


                           plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                             l,x0(4,:),l,x0(5,:),l,x0(6,:),...
                             l,x0(7,:),l,x0(8,:),'Linewidth',2);
hold on;

ustmax=0.571;
l=0.500:0.001:ustmax;
n=length(l);


 for i=1:n
    f(1,i)=0.0;
    % N = 8
     x0(1,i) = (4.9795+((4.4639-4.9795)/(0.571-0.500))*(l(i)-0.500))*pi/180;
     x0(2,i) = (10.2818+((9.9832-10.2818)/(0.571-0.500))*(l(i)-
0.500))*pi/180;
     x0(3,i) = (12.9257+((12.7575-12.9257)/(0.571-0.500))*(l(i)-
0.500))*pi/180;
     x0(4,i) = (18.3740+((19.1955-18.3740)/(0.571-0.500))*(l(i)-
0.500))*pi/180;
```

```
     x0(5,i) = (40.6587+((39.3779-40.6587)/(0.571-0.500))*(l(i)-
0.500))*pi/180;
     x0(6,i) = (58.5092+((59.7847-58.5092)/(0.571-0.500))*(l(i)-
0.500))*pi/180;
     x0(7,i) = (83.1927+((82.2000-83.1927)/(0.571-0.500))*(l(i)-
0.500))*pi/180;


         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:7
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;
    x0(7,:) = x0(7,:)*180/pi;


                           plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...

l,x0(4,:),l,x0(5,:),l,x0(6,:),l,x0(7,:),'Linewidth',2);
hold on;
%
ustmax=0.666;
l=0.571:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 6
            x0(1,i) = 3.267*pi/180;
            x0(2,i) = 9.36*pi/180;
            x0(3,i) = 46.529*pi/180;
            x0(5,i) = (74.364+((72.292-74.364)/0.094)*(l(i)-0.566))*pi/180;
         x0(4,i) = (57.423+((60.9-57.423)/0.26)*(l(i)-0.4))*pi/180;
        x0(6,i) = (87.179+((90-87.179)/0.26)*(l(i)-0.4))*pi/180;


         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
```

```
                    for j=1:6
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
          end

             f(1,i)=sqrt(f(1,i));

 end
   x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;
    x0(6,:) = x0(6,:)*180/pi;

                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),l,x0(5,:),l,x0(6,:),'Linewidth',2);
hold on;


ustmax=0.800;
l=0.666:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 5
     x0(1,i) = (6.2499+((6.4376-6.2499)/(0.800-0.666))*(l(i)-0.666))*pi/180;
     x0(2,i) = (16.3088+((15.7985-16.3088)/(0.800-0.666))*(l(i)-
0.666))*pi/180;
     x0(3,i) = (40.3917+((39.6314-40.3917)/(0.800-0.666))*(l(i)-
0.666))*pi/180;
     x0(4,i) = (61.4821+((63.6099-61.4821)/(0.800-0.666))*(l(i)-
0.666))*pi/180;
     x0(5,i) = (78.1950+((74.2663-78.1950)/(0.800-0.666))*(l(i)-
0.666))*pi/180;


          for c=1:600
                u_h1 = 0.0;
                u_h2 = 0.0;
                     for j=1:5
                         u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                         u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                     end
                     u_h1=u_h1*4/(pi*(6*c-1));
                     u_h2=u_h2*4/(pi*(6*c+1));
                     f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
          end

             f(1,i)=sqrt(f(1,i));
```

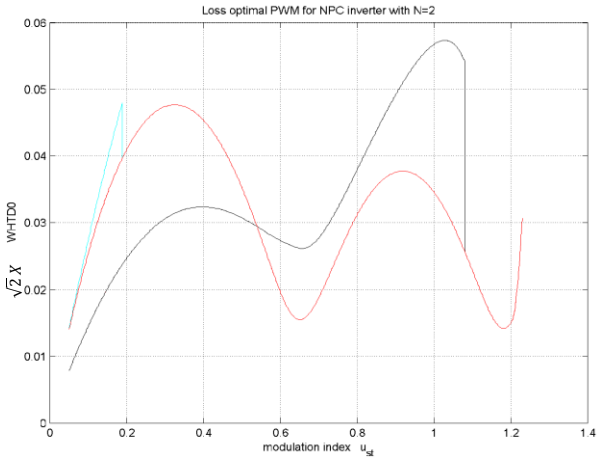
```
 end
 x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;
    x0(5,:) = x0(5,:)*180/pi;


                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),l,x0(5,:),'Linewidth',2);
hold on;
ustmax=1.00;
l=0.800:0.001:ustmax;
n=length(l);

 for i=1:n
    f(1,i)=0.0;
    % N = 4
        if l(i)<0.882
        x0(1,i) = (13.09+((19.99-13.09)/0.202)*(l(i)-0.68))*pi/180;
        x0(2,i) = (52.485+((45.38-52.485)/0.202)*(l(i)-0.68))*pi/180;
         else
         x0(1,i) = 19.99*pi/180;
         x0(2,i) = 45.38*pi/180;
        end
        x0(3,i) = (59.515+((51.663-59.515)/0.32)*(l(i)-0.68))*pi/180;
         if l(i)<0.882
        x0(4,i)= (69.3+((82.905-69.3)/0.222)*(l(i)-0.68))*pi/180;
         else
          x0(4,i)= (82.905+((85.655-82.905)/0.118)*(l(i)-0.882))*pi/180;
       end


         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:4
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));
 end
 x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;
    x0(4,:) = x0(4,:)*180/pi;

                            plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),...
                              l,x0(4,:),'Linewidth',2);
hold on;
```

```
ustmax=1.23;
l=1.00:0.001:ustmax;
n=length(l);


 for i=1:n
    f(1,i)=0.0;
    % N = 3
     x0(1,i) = (25.0727+((17.3091-25.0727)/0.23)*(l(i)-1))*pi/180;
     x0(2,i) = (38.3261+((44.0137-38.3261)/0.23)*(l(i)-1))*pi/180;
     x0(3,i) = (48.3850+((45.0137-48.3850)/0.23)*(l(i)-1))*pi/180;

     if l > 1.20
     x0(1,i) = (10.0+((9.8089-10.0)/0.03)*(l(i)-1.2))*pi/180;
     x0(2,i) = (14.0+((13.0004-14.0)/0.03)*(l(i)-1.2))*pi/180;
     x0(3,i) = (17.5+((17.2487-17.5)/0.03)*(l(i)-1.2))*pi/180;
     end


         for c=1:600
               u_h1 = 0.0;
               u_h2 = 0.0;
                    for j=1:3
                        u_h1=u_h1+(-1.0)^(j+1)*cos((6*c-1)*x0(j,i));
                        u_h2=u_h2+(-1.0)^(j+1)*cos((6*c+1)*x0(j,i));
                    end
                    u_h1=u_h1*4/(pi*(6*c-1));
                    u_h2=u_h2*4/(pi*(6*c+1));
                    f(1,i)=f(1,i)+(u_h1/(6*c-1))^2+(u_h2/(6*c+1))^2;
         end

            f(1,i)=sqrt(f(1,i));
 end
 x0(1,:) = x0(1,:)*180/pi;
    x0(2,:) = x0(2,:)*180/pi;
    x0(3,:) = x0(3,:)*180/pi;

plot(l,x0(1,:),l,x0(2,:),l,x0(3,:),'Linewidth',2);
hold off;
```

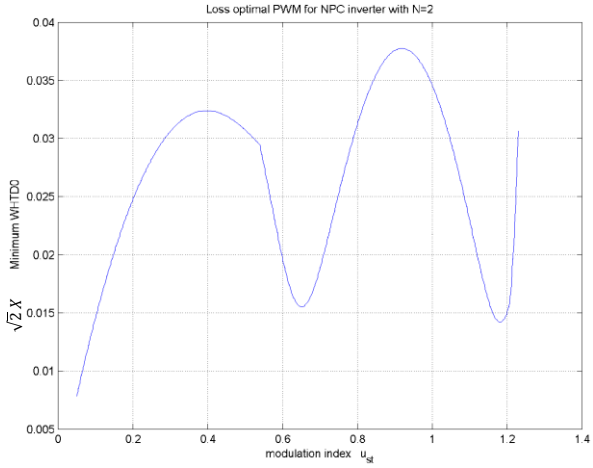
Appendix 3

WTHDO WITH EVEN NUMBER OF PULSES

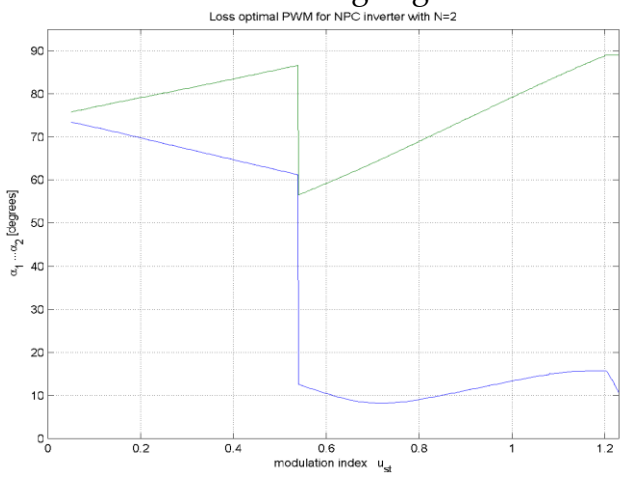
WITH N = 2



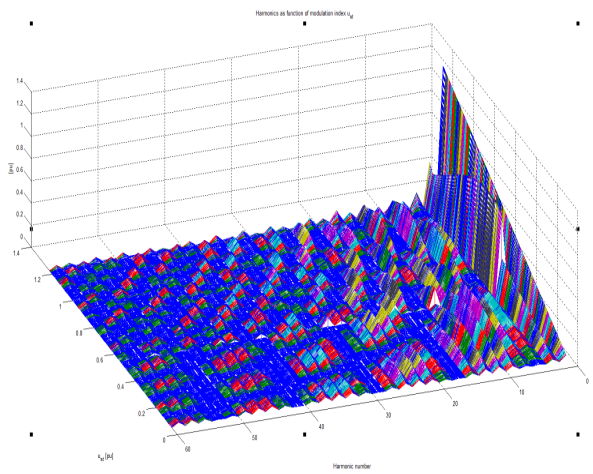
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =2



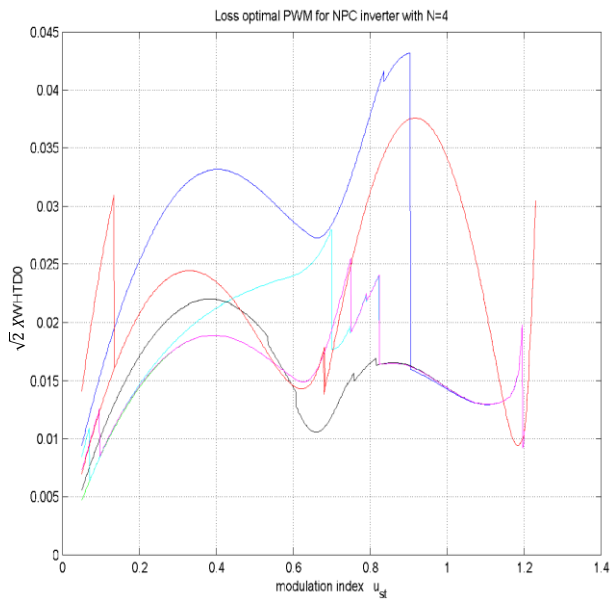
c. Angles pattern for minimum WTHD0, N=2



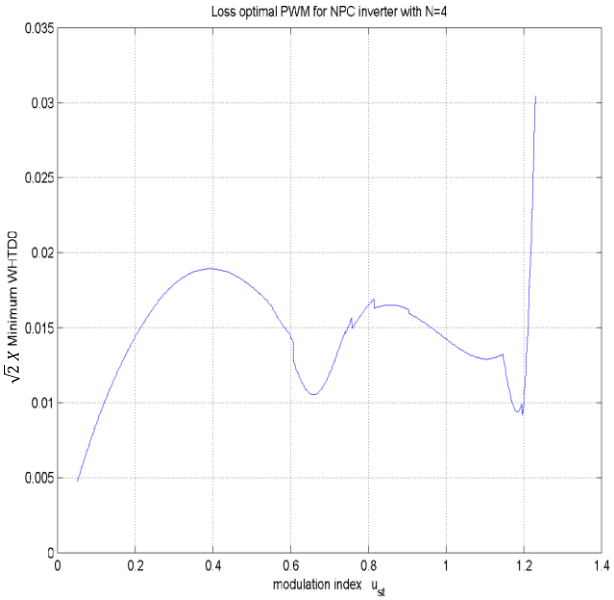
d. Harmonics for minimum WTHD0, N=2

Figure 76: Minimization of WTHD0 with N =2

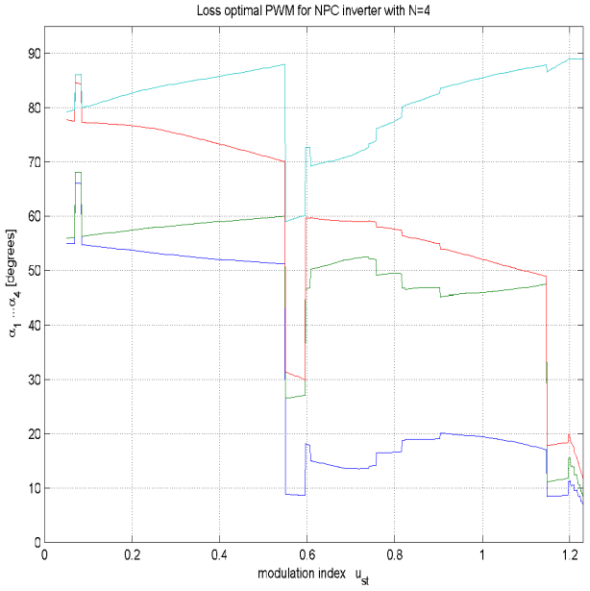
WITH N = 4



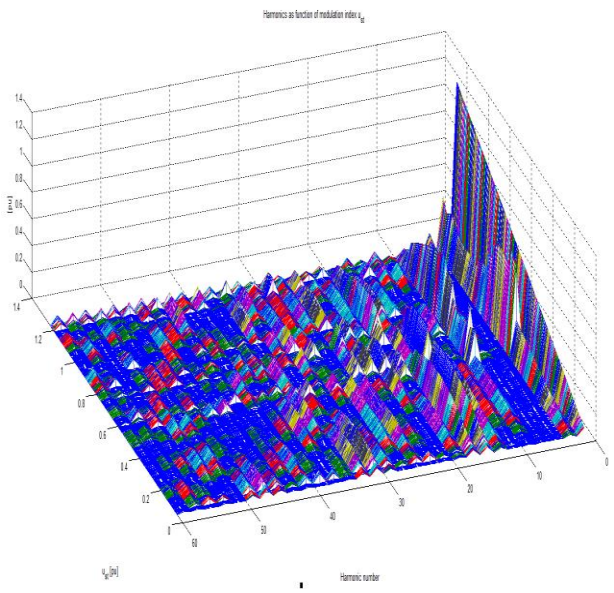
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =4



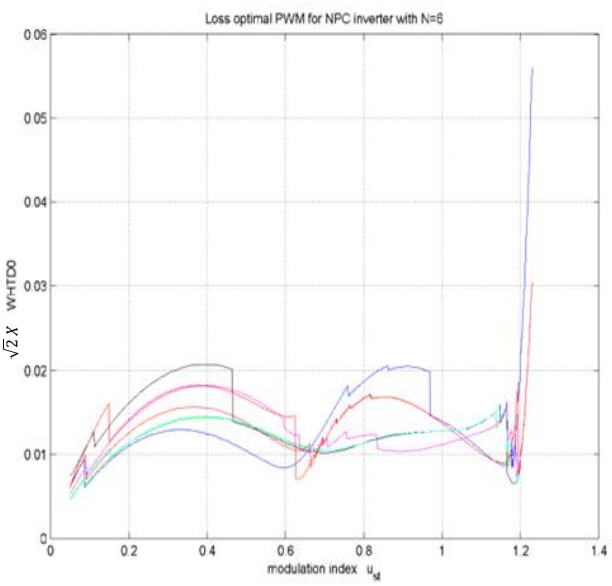
c. Angles pattern for minimum WTHD0, N=4



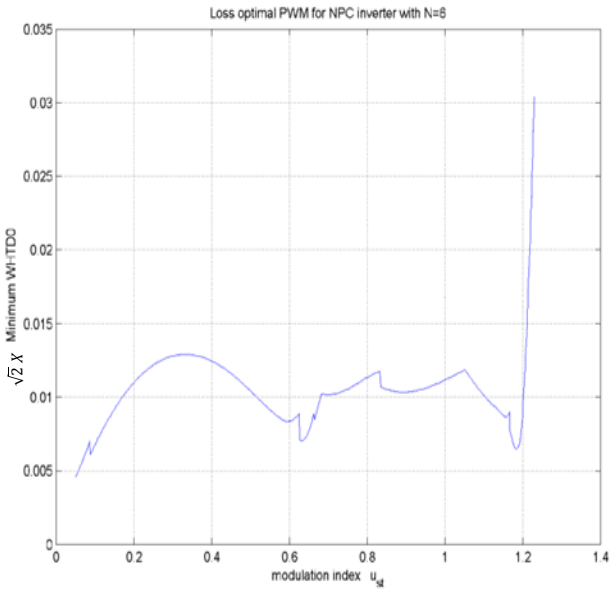
d. Harmonics for minimum WTHD0, N=4

Figure 77: Minimization of WTHD0 with N =4

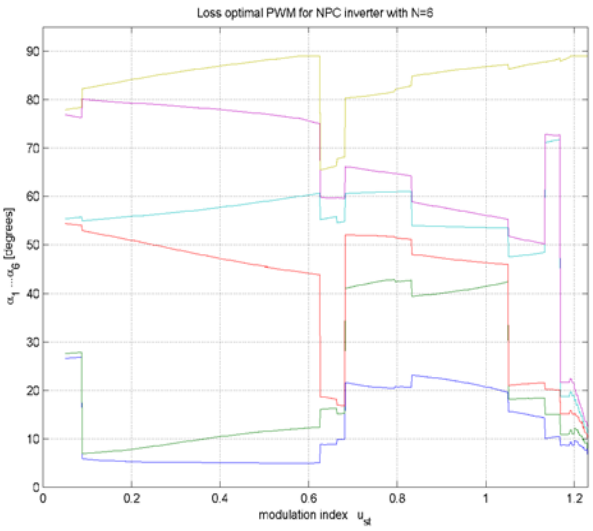
WITH N = 6



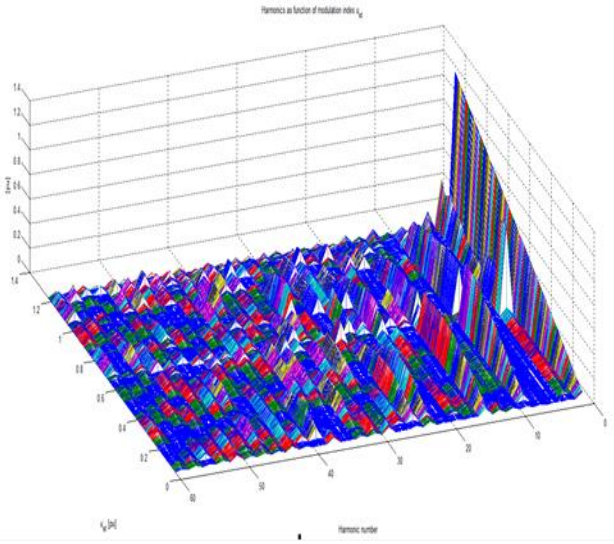
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =6



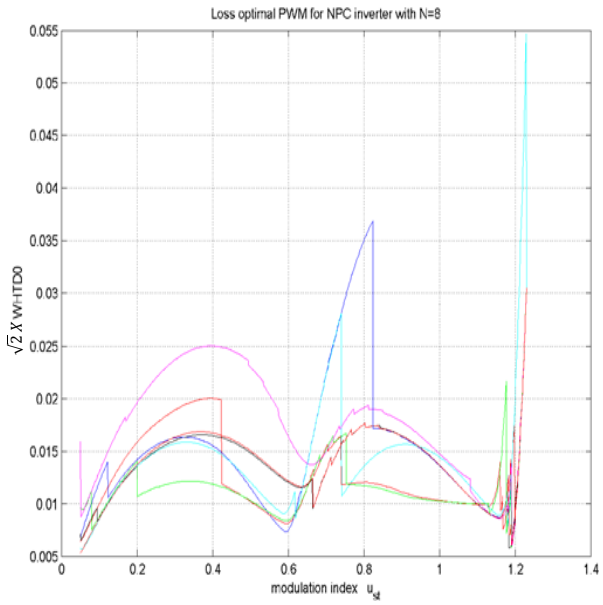
c. Angles pattern for minimum WTHD0, N=6



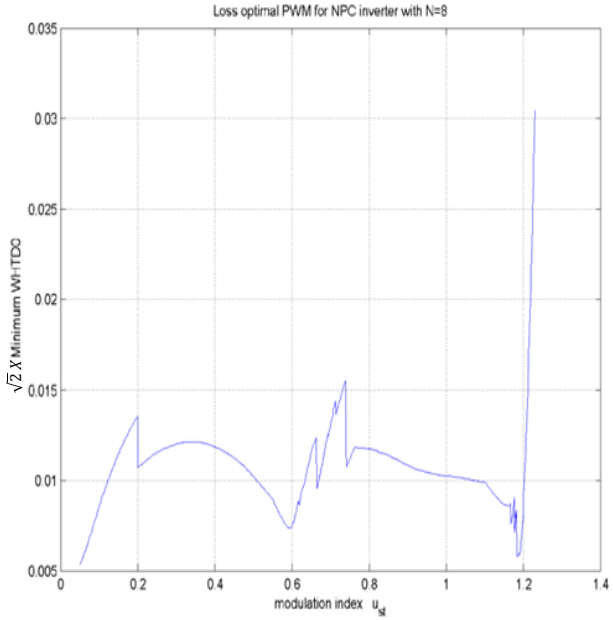
d. Harmonics for minimum WTHD0, N=6

Figure 78: Minimization of WTHD0 with N =6

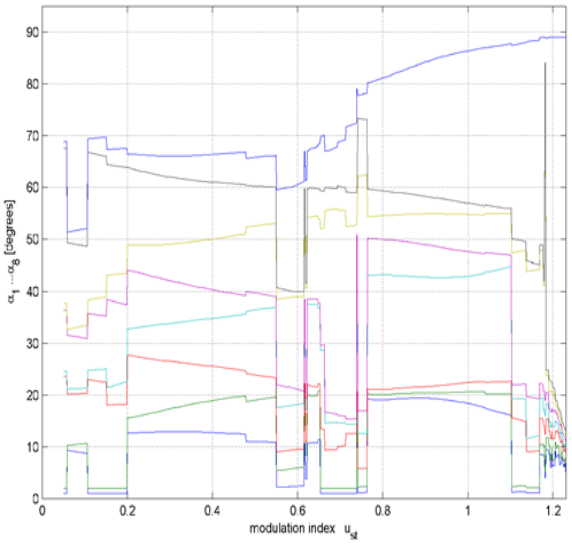
WITH N = 8



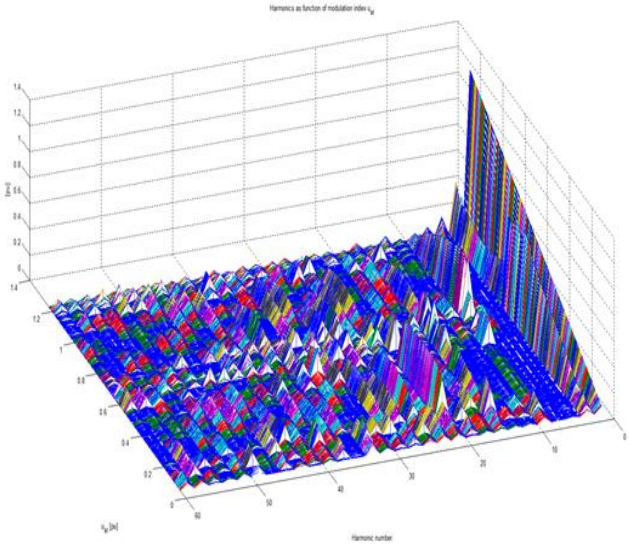
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =8



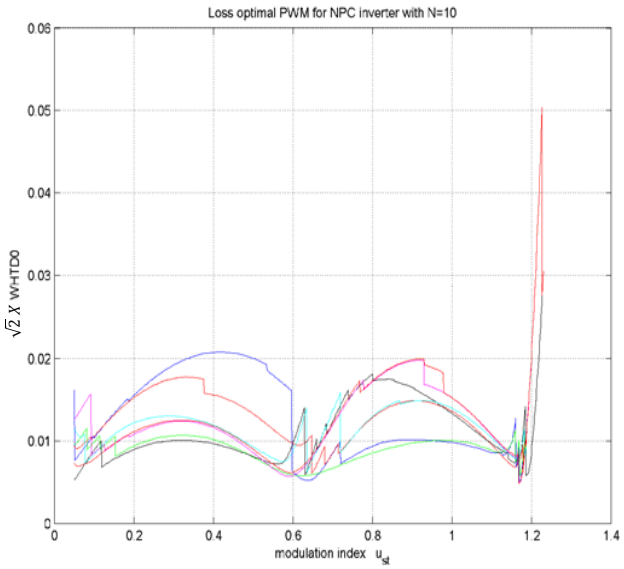
c. Angles pattern for minimum WTHD0, N=8



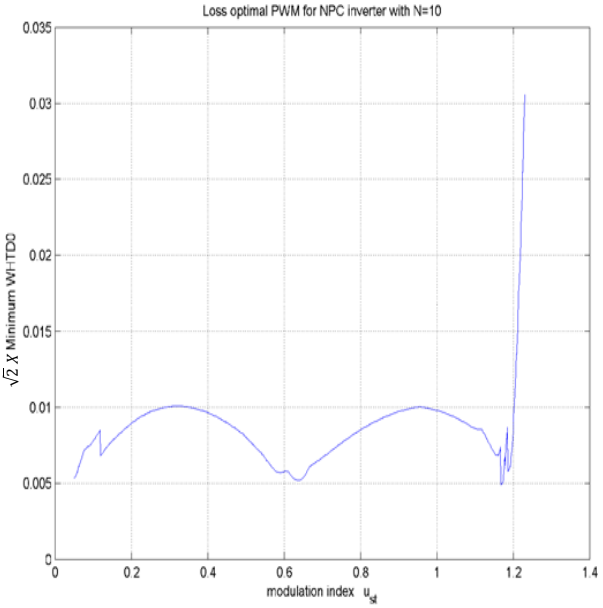
d. Harmonics for minimum WTHD0, N=8

Figure 79: Minimization of WTHD0 with N =8

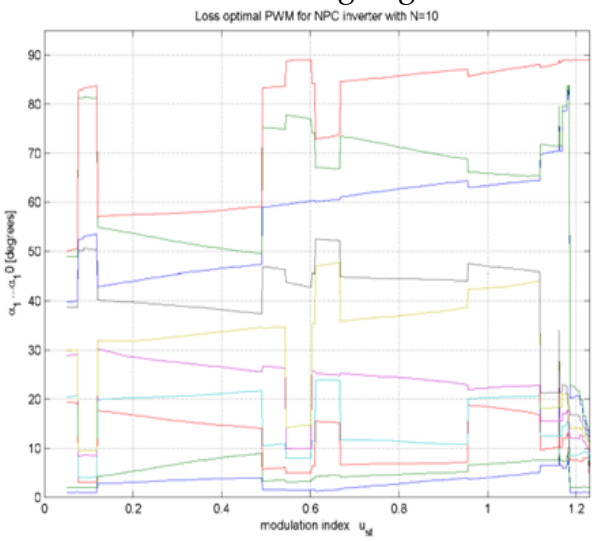
WITH N = 10



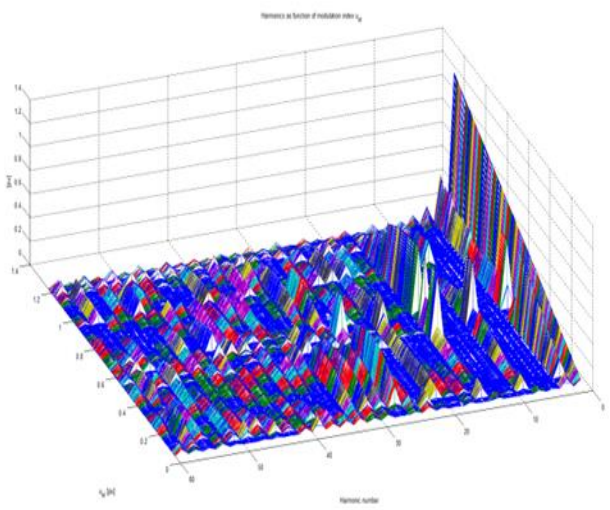
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =10



c. Angles pattern for minimum WTHD0, N=10

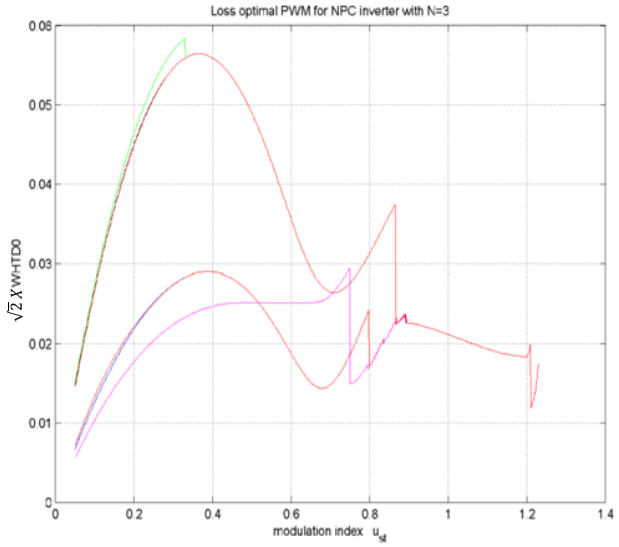


d. Harmonics for minimum WTHD0, N=10

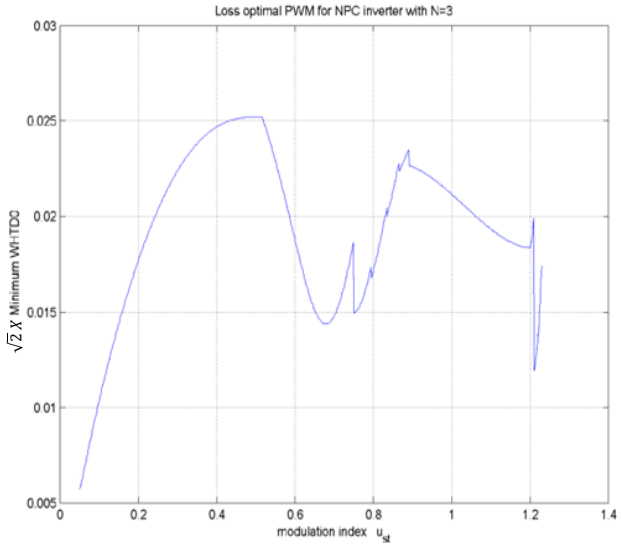
Figure 80: Minimization of WTHD0 with N =10

WTHDO WITH ODD NUMBER OF PULSES

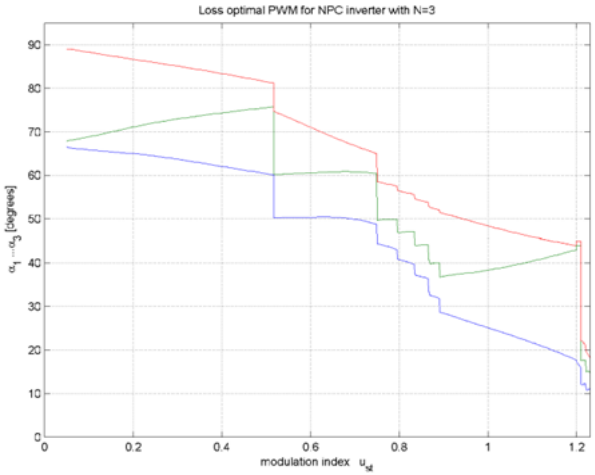
WITH N = 3



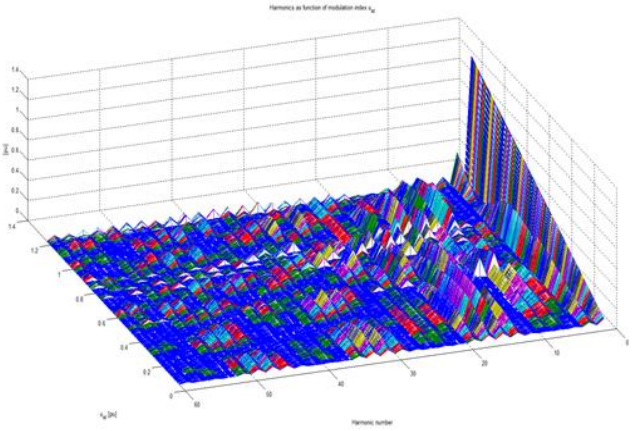
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =3



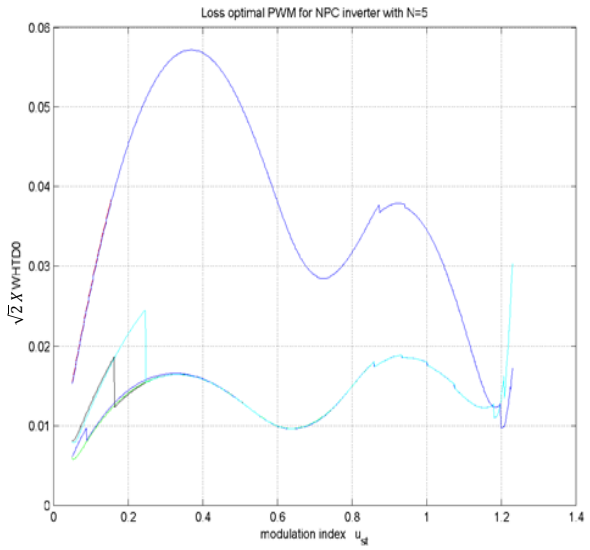
c. Angles pattern for minimum WTHD0, N=3



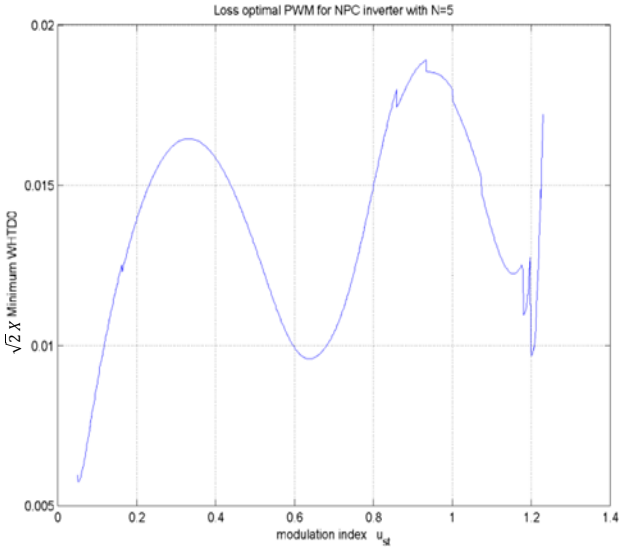
d. Harmonics for minimum WTHD0, N=3

Figure 81: Minimization of WTHD0 with N =3

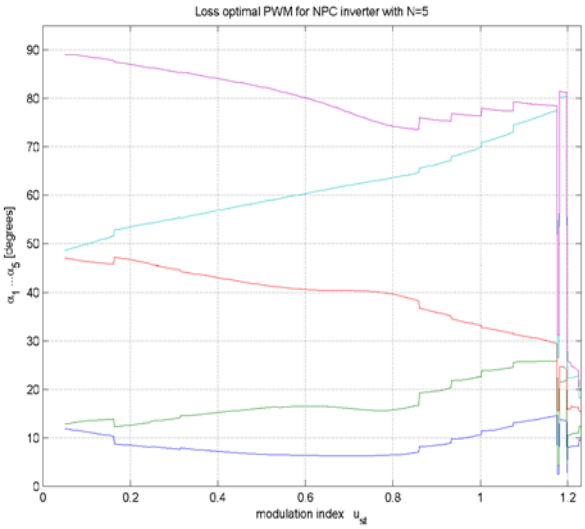
WITH N = 5



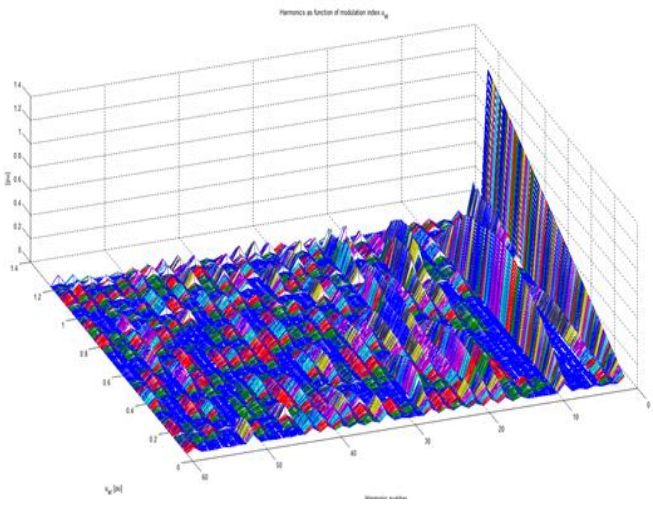
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =5



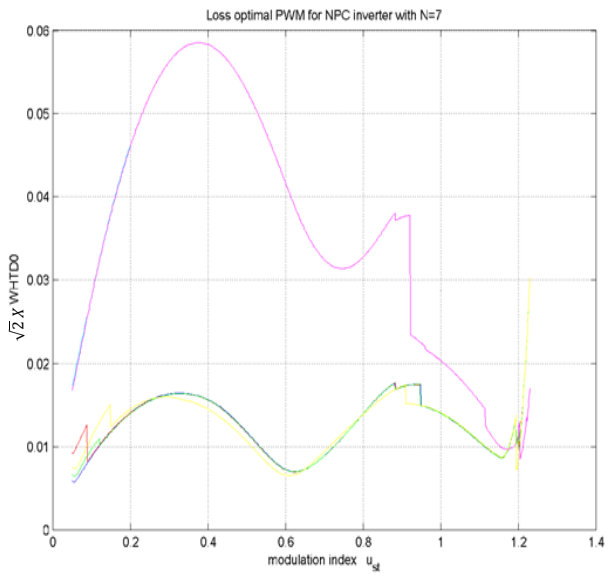
c. Angles pattern for minimum WTHD0, N=5



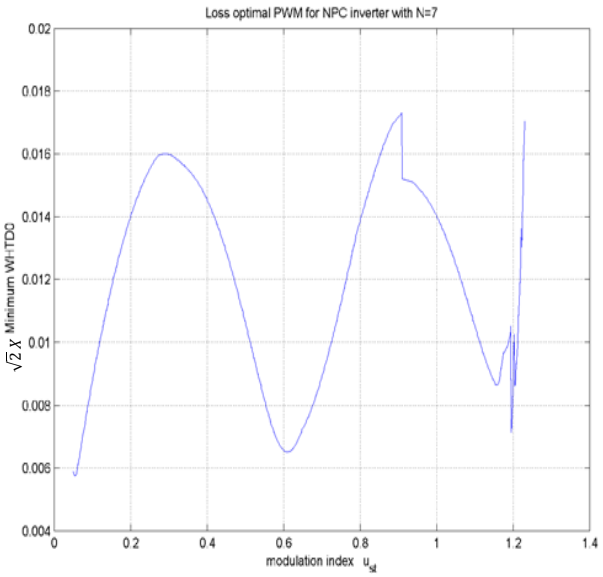
d. Harmonics for minimum WTHD0, N=5

Figure 82: Minimization of WTHD0 with N =5

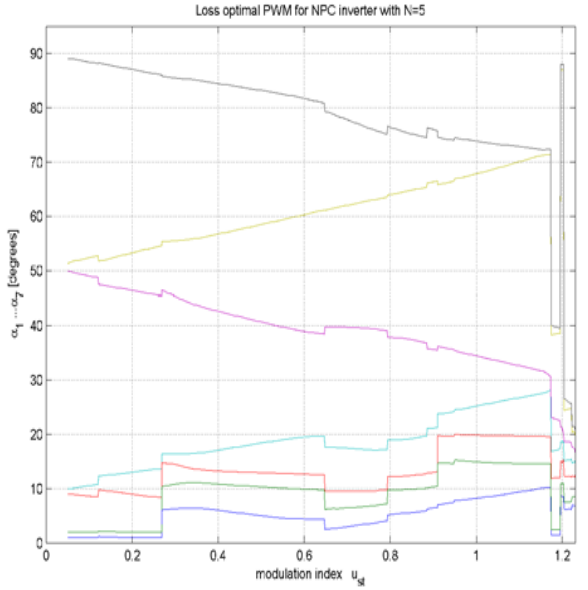
WITH N = 7



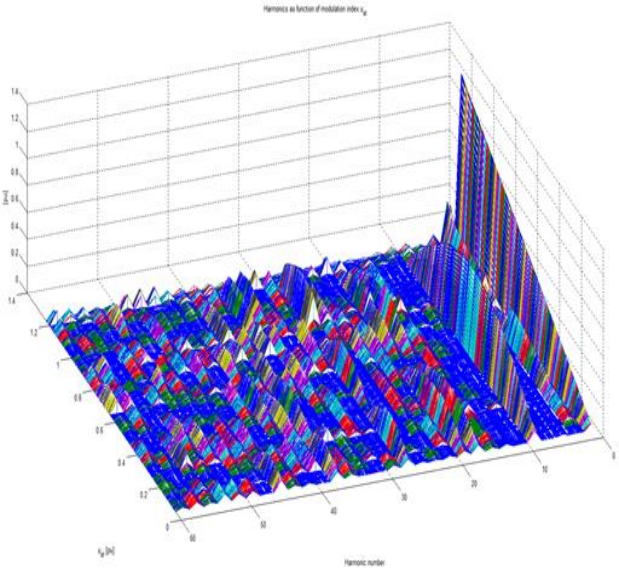
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =7



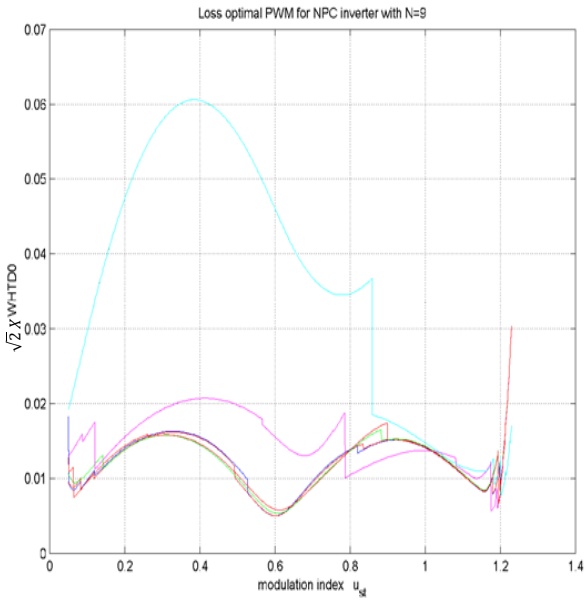
c. Angles pattern for minimum WTHD0, N=7



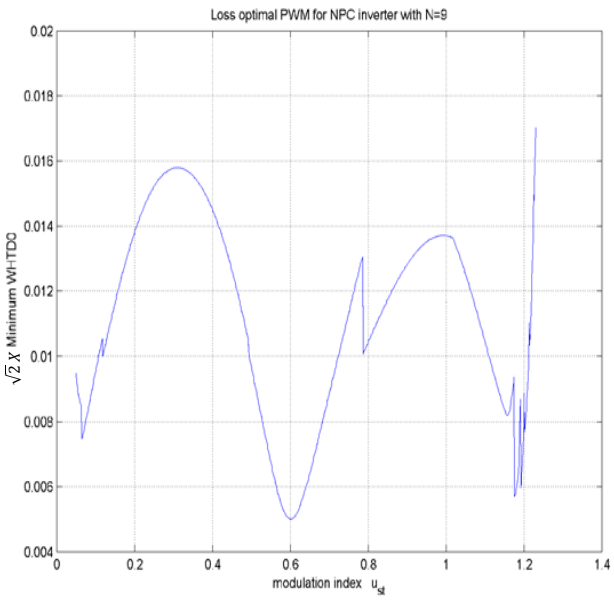
d. Harmonics for minimum WTHD0, N=7

Figure 83: Minimization of WTHD0 with N =7

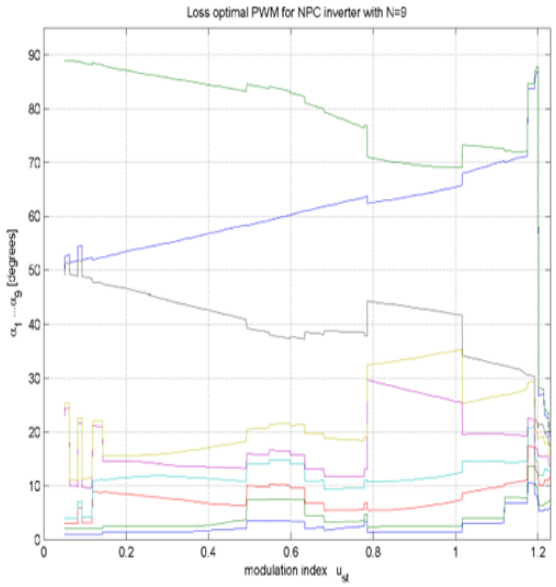
WITH N = 9



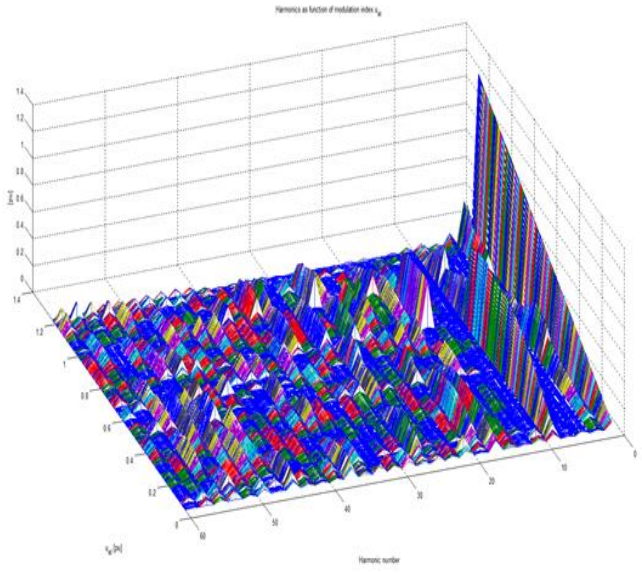
a. Minimum WTHD0 with multiple starting angles



b. Minimum WTHD0 with N =2



c. Angles pattern for minimum WTHD0, N=9

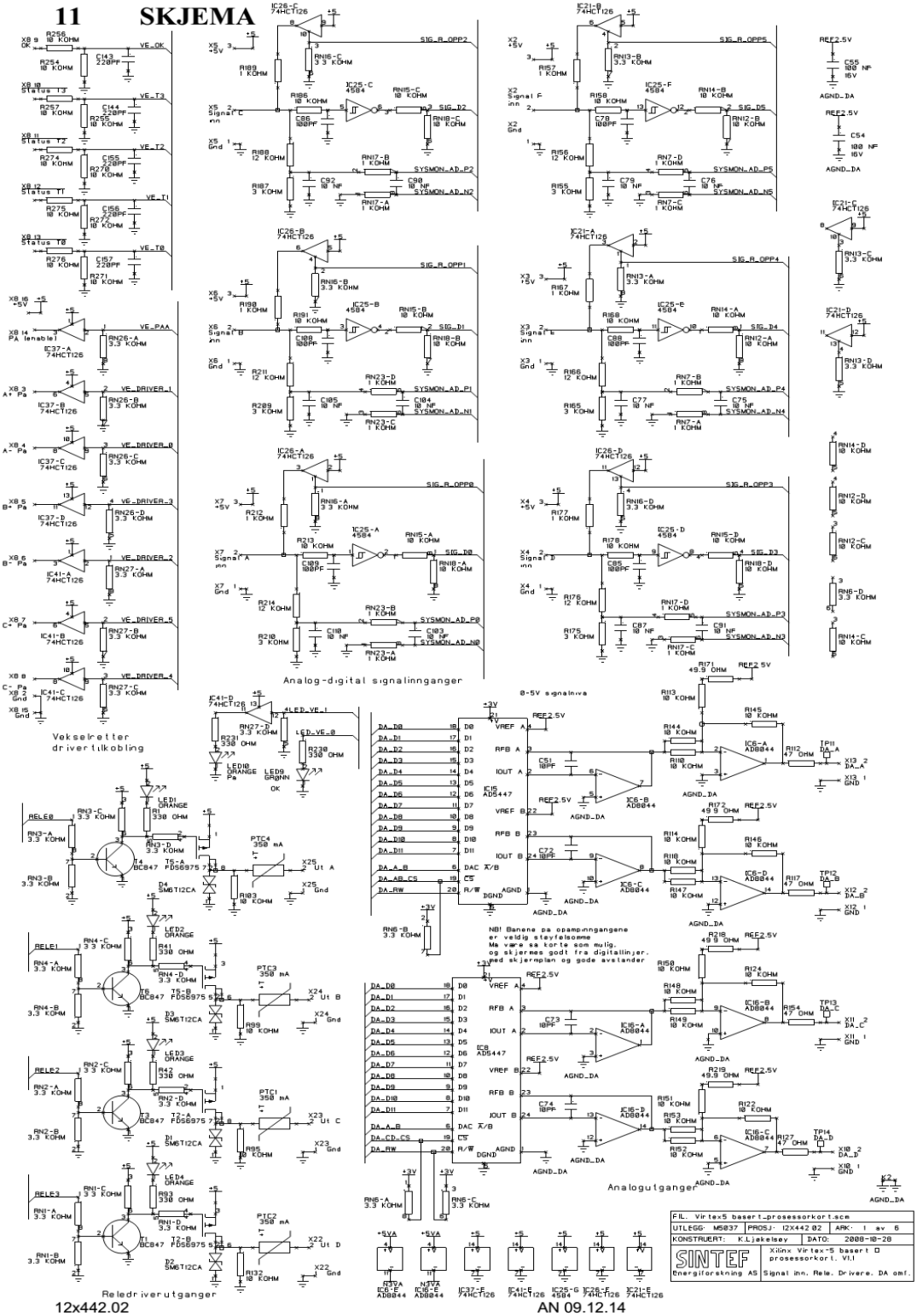


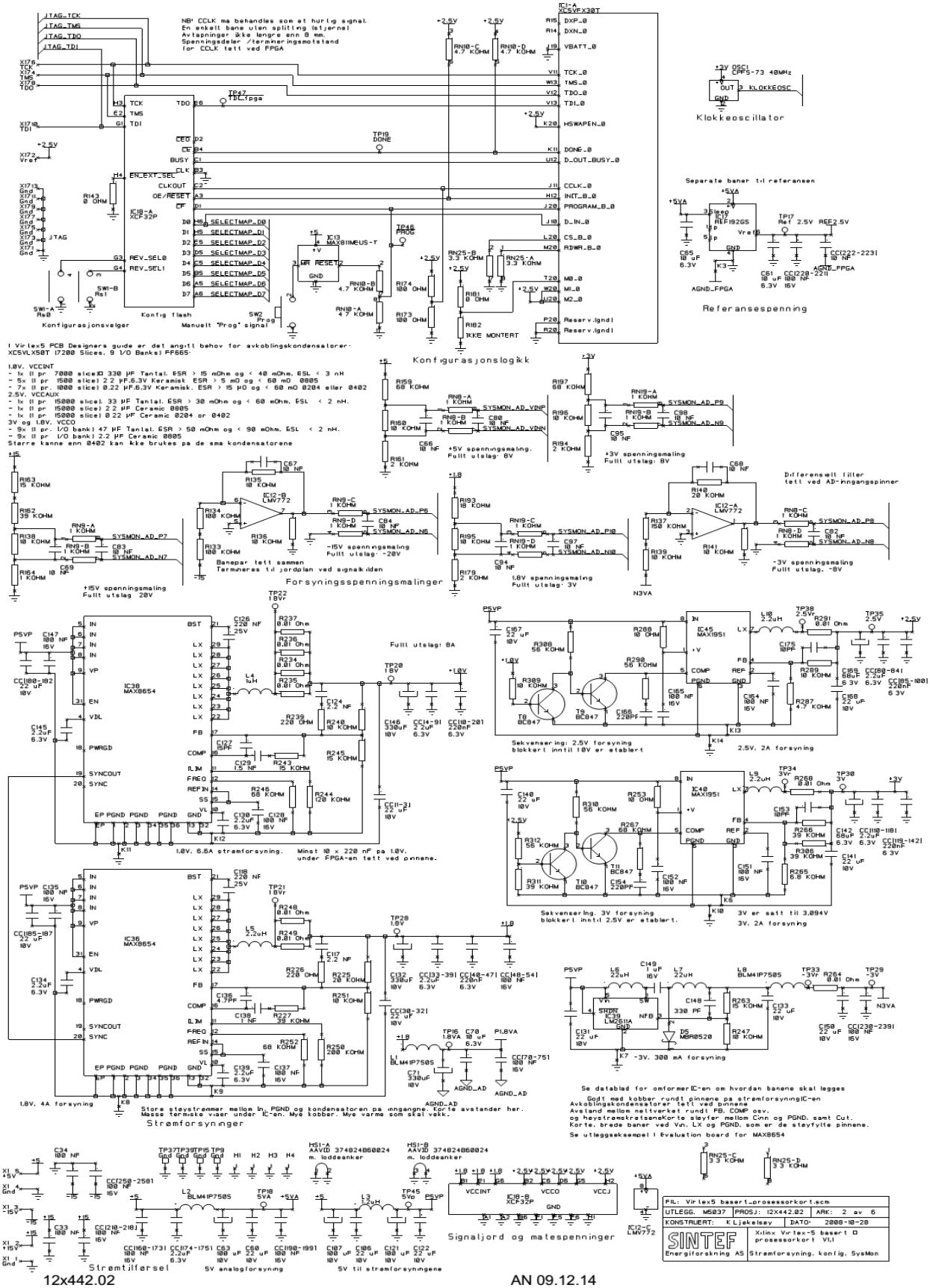
d. Harmonics for minimum WTHD0, N=9

Figure 84: Minimization of WTHD0 with N = 9

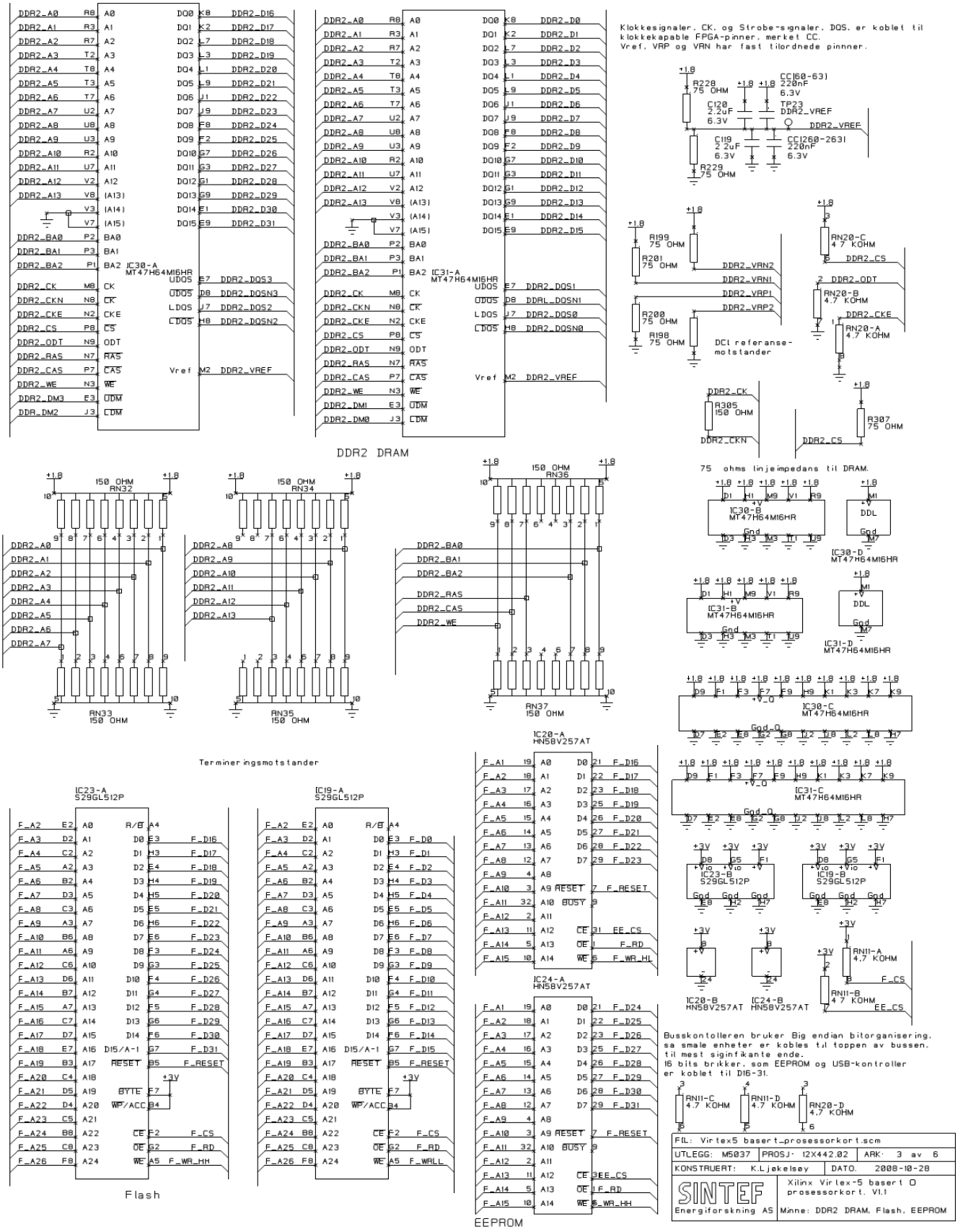
Appendix 4

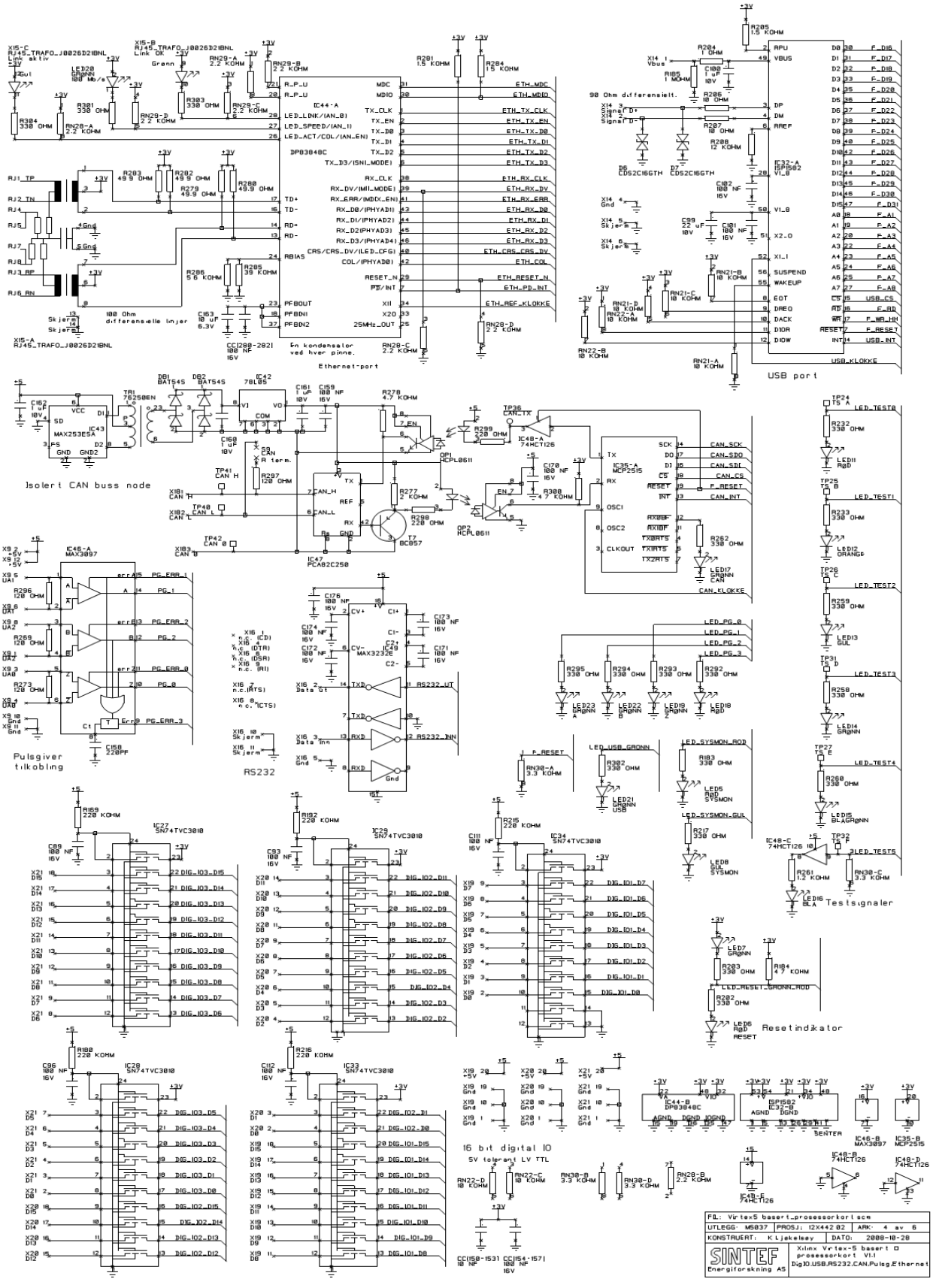
MANUAL OF THE FPGA BOARD [53]

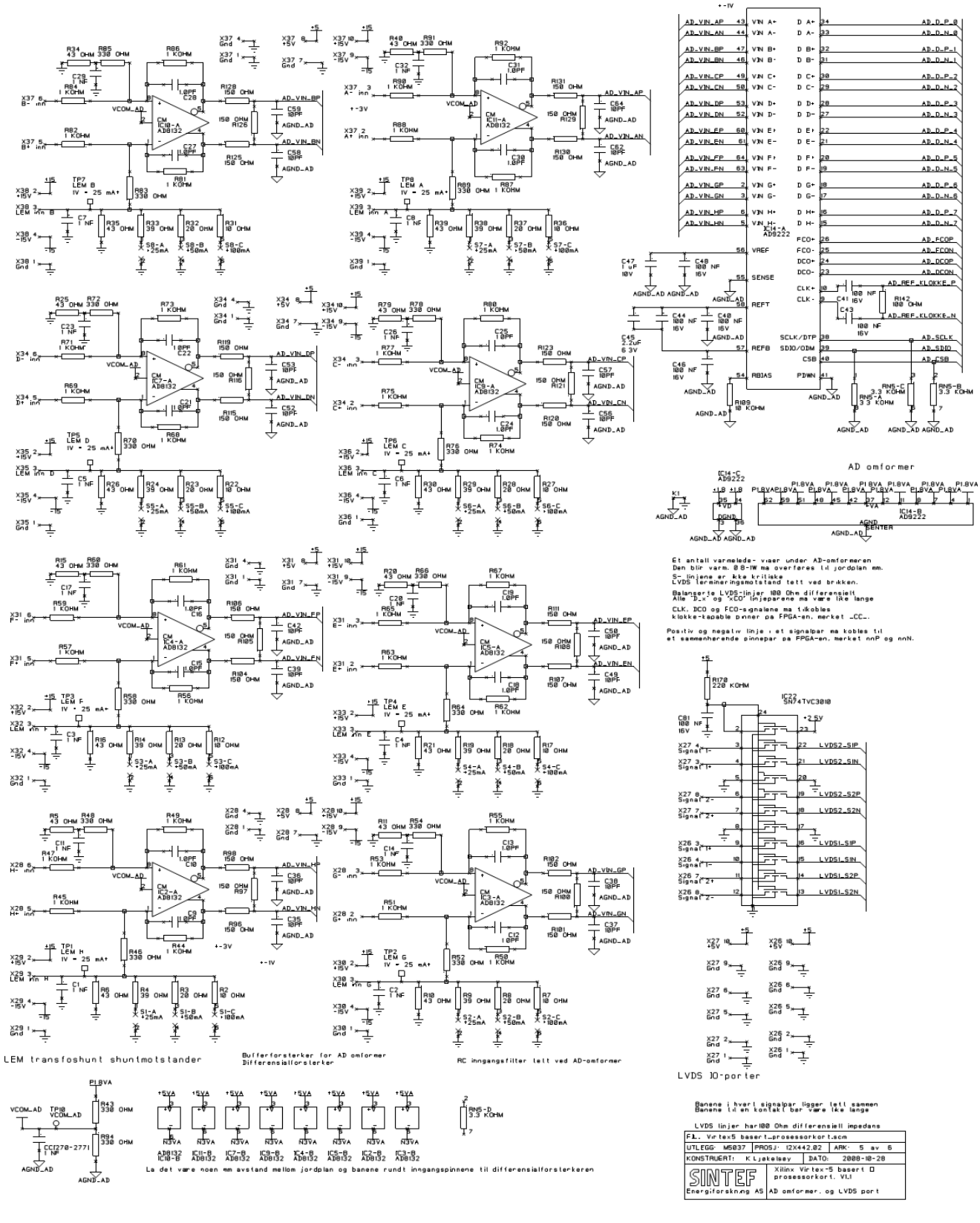


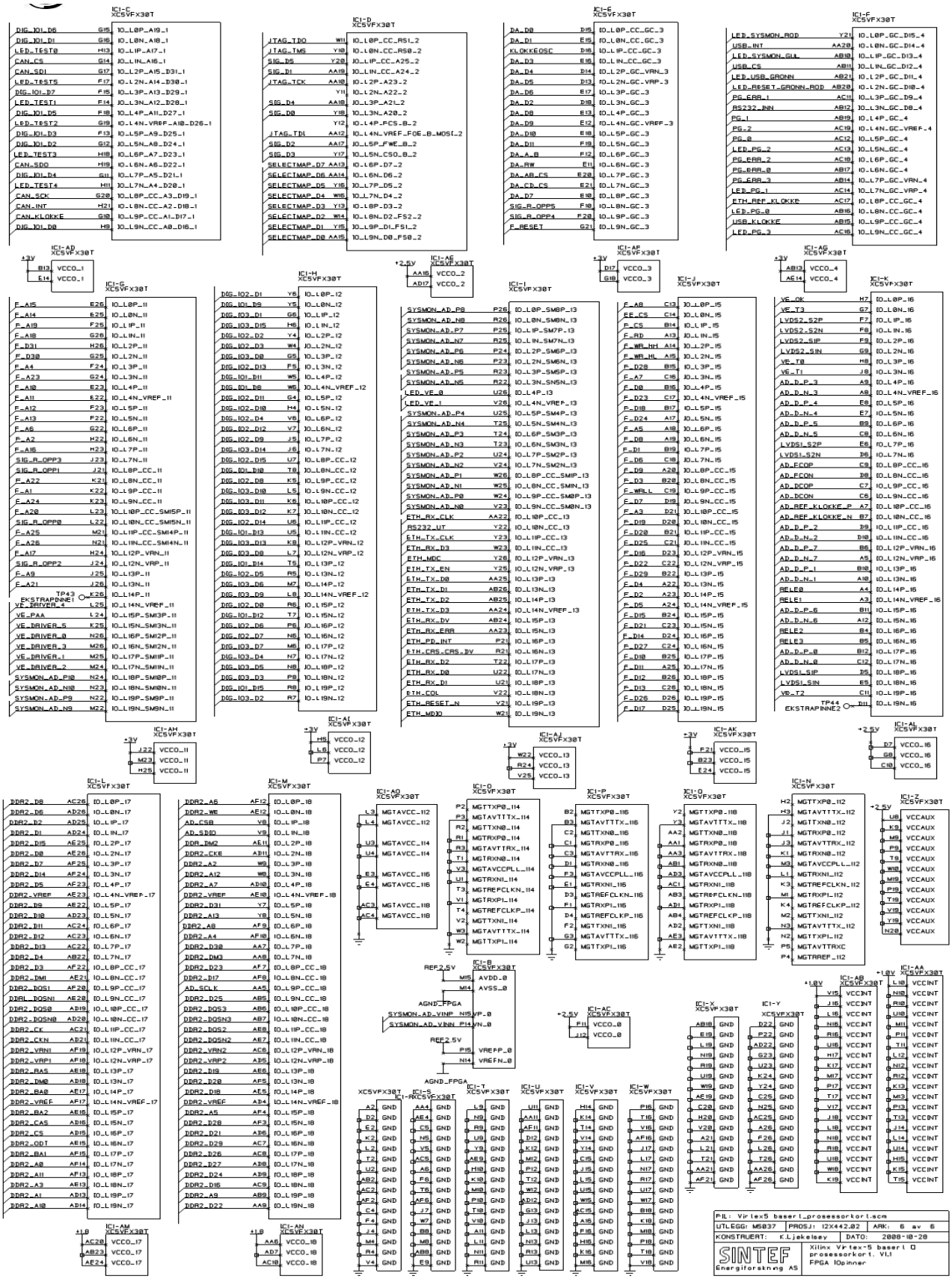


12x442.02 AN 09.12.14

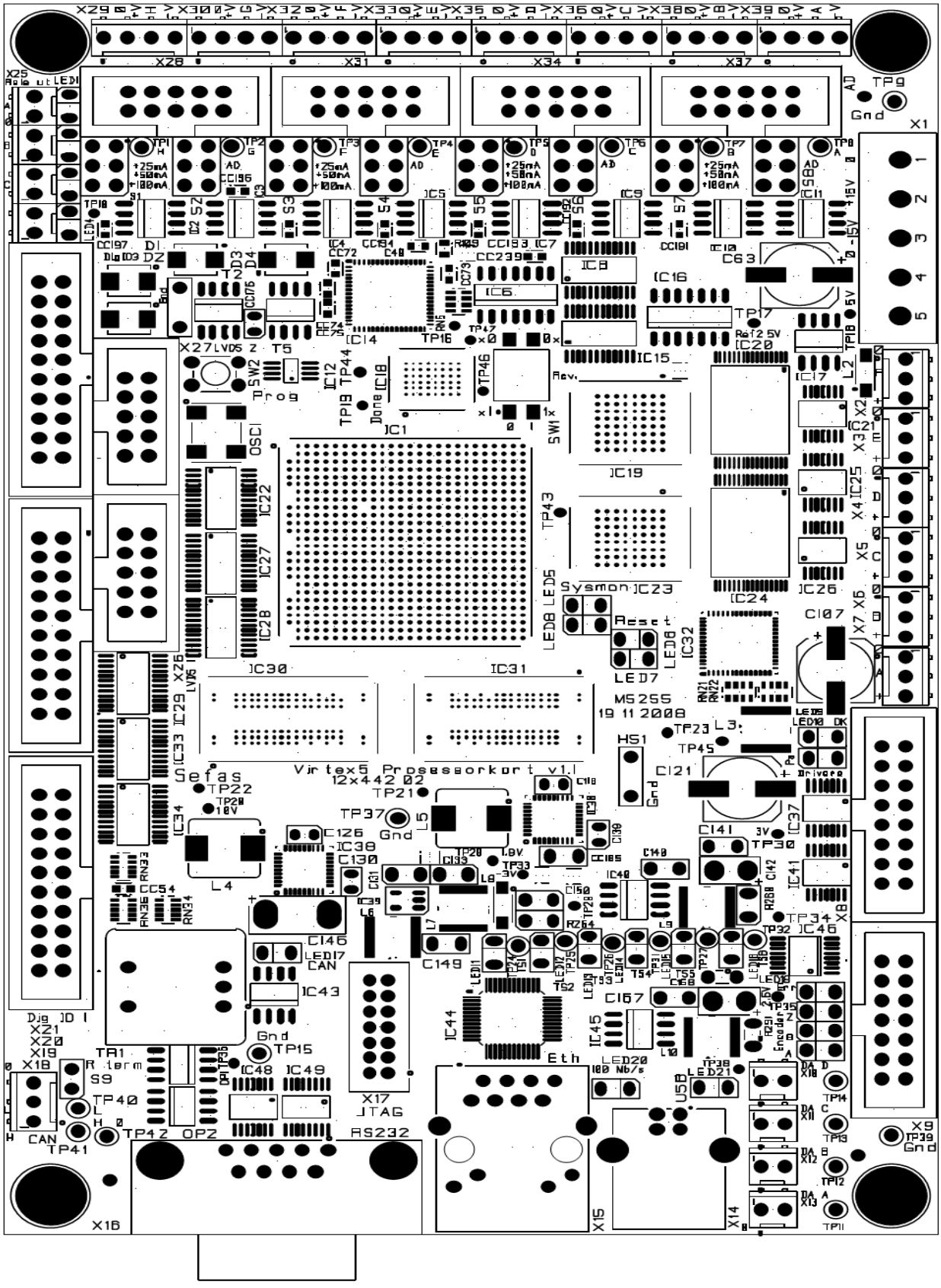

DDR2 DRAM
Klokkesignaler, CK, og Strobe-signaler, DQS, er koblet til klokkekapable FPGA-pinner, merket CC.
Vref, VRP og VRN har fast tilordnede pinnner.
75 ohms linjeimpedans til DRAM.
Termineringsmotstander
Flash
EEPROM
Busskontolleren bruker Big endian bitorganisering, sa smale enheter er kobles til toppen av bussen, til mest siginfikante ende.
16 bits brikker, som EEPROM og USB-kontroller er koblet til D16-31.
FIL: Virtex5 basert_prosessorkort.scm
UTLEGG: M5037
PROSJ: 12X442.02
ARK: 3 av 6
KONSTRUERT: K.Ljøkelsøy
DATO. 2008-10-28
SINTEF Energiforskning AS
Xilinx Virtex-5 basert D prosessorkort. V1.1
Minne: DDR2 DRAM, Flash, EEPROM


Ethernet-port
USB port
Isolert CAN buss node
Pulsgiver tilkobling
RS232
Testsignaler
Resetindikator
16 bit digital IO
5V tolerant LV TTL
Xilinx Virtex-5 basert prosessorkort V1.1
Dig10.USB.RS232.CAN.Pulsg.Ethernet
SINTEF Energiforskning AS


AD omformer
LEM transfoshunt shuntmotstander
Bufferforsterker for AD omformer
Differensialforsterker
RC inngangsfilter lett ved AD-omformer
LVDS IO-porter
Et antall varmelede- viaer under AD-omformeren
Den blir varm. 0.8-1W ma overføres til jordplan em.
S- linjene er ikke kritiske
LVDS termineringsmotstand tett ved brikken.
Balanserte LVDS-linjer 100 Ohm differensielt
Alle "_x" og "xCO" linjeparene ma være like lange
CLK, DCO og FCO-signalene ma tilkobles
klokke-kapable pinner på FPGA-en, merket _CC_.
Positiv og negativ linje i et signalpar ma kobles til
et sammenhørende pinnepar på FPGA-en, merket nnP og nnN.
Banene i hvert signalpar ligger tett sammen
Banene til en kontakt bar være like lange
LVDS linjer har 100 Ohm differensiell impedans
La det være noen mm avstand mellom jordplan og banene rundt inngangspinnene til differensialforsterkeren
Fil. Virtex5 basert_prosessorkort.sch
UTLEGG: M5037
PROSJ: 12X442.02
ARK: 5 av 6
KONSTRUERT: K Ljøkelsøy
DATO: 2008-10-28
SINTEF Energiforskning AS
Xilinx Virtex-5 basert FPGA prosessorkort. V1.1
AD omformer og LVDS port


FIL: Virtex5 baserl_prosessorkort.scm
UTLEGG: M5037
PROSJ: 12X442.02
ARK: 6 av 6
KONSTRUERT: K.Ljekelsey
DATO: 2008-10-20
SINTEF Energiforskning AS
Xilinx Virtex-5 baserl O prosessorkort. VLI
FPGA IOpinner

12 KOMPONENTPLASSERING



Appendix 5

CONFIGURATION OF INPUT OUTPUT DRIVERS

SYSTEM DEFINATION

It is done by using the XPS tool.

ppc440_0's Address Map

Memory_controller_2	C_BASEADDR	0x00000000	0x0001FFFF	128K	SPLB	plb_v46_0
ad_omformer_seriemottak...	C_BASEADDR	0x60000000	0x6000FFFF	64K	SPLB	plb_v46_0
signalinngang	C_BASEADDR	0x81400000	0x8140FFFF	64K	SPLB	plb_v46_0
RELEDRIVER_GPIO	C_BASEADDR	0x81420000	0x8142FFFF	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
mdm_0	C_BASEADDR	0x84400000	0x8440FFFF	64K	SPLB	plb_v46_0
DIG_IO1_GPIO	C_BASEADDR	0x88000000	0x8800FFFF	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
Memory_controller_1	C_BASEADDR	0xFFFE0000	0xFFFFFFFF	128K	SPLB	plb_v46_0

Figure 85: Configuration of Components in XPS

ADC CONFIGURATION

```
#define AD_SIGNAL_A          *(volatile int*)((AD_S_BASEADR) + 0x0)
#define AD_SIGNAL_B          *(volatile int*)((AD_S_BASEADR) + 0x4)
#define AD_SIGNAL_C          *(volatile int*)((AD_S_BASEADR) + 0x8)
#define AD_SIGNAL_D          *(volatile int*)((AD_S_BASEADR) + 0xC)
#define AD_SIGNAL_E          *(volatile int*)((AD_S_BASEADR) + 0x10)
#define AD_SIGNAL_F          *(volatile int*)((AD_S_BASEADR) + 0x14)
#define AD_SIGNAL_G          *(volatile int*)((AD_S_BASEADR) + 0x18)
#define AD_SIGNAL_H          *(volatile int*)((AD_S_BASEADR) + 0x1C)
```

DEFINATION OF USER CONSTRAIN FILE, example

```
NET "KLOKKEOSC" TNM_NET = "klokke_40MHz";
#TIMESPEC "TSklokke40MHz" = PERIOD "klokke_40MHz" 25 ns HIGH 50%;
#NET "AD_DCOP" TNM_NET = "ad_data_klokke";
#TIMESPEC "TSad_data_klokke" = PERIOD "ad_data_klokke" 4 ns HIGH 50%;
#INST "*/clk_bus*" TNM = "TNM_CLK_BUS_SYNK";
```

```
#INST "*/clk_ad*" TNM = "TNM_CLK_AD_SYNK";
##TIMESPEC "TS_AD_SERIEMOTTAKER_SYNK" = FROM "TNM_CLK_AD_SYNK" TO
"TNM_CLK_BUS_SYNK" 4 ns;


##  IC1 D
##NET   JTAG_TDO                                    LOC = "W11";
#CONFIG PROHIBIT  = "W11";
##NET   JTAG_TMS                                    LOC = "Y10";
#CONFIG PROHIBIT  = "Y10";
#NET   SIG_D<5>                          LOC = "Y20";
#NET   SIG_D<1>                          LOC = "AA19";
##NET   JTAG_TCK                         LOC = "AA10";
#CONFIG PROHIBIT  = "AA10";
#CONFIG PROHIBIT   = "Y11";
#NET   SIG_D<4>                          LOC = "AA18";
#NET   SIG_D0                            LOC = "Y18";
#CONFIG PROHIBIT  = "Y12";
##NET   JTAG_TDI                         LOC = "AA12";
#CONFIG PROHIBIT  = "AA12";
#NET   SIG_D<2>                          LOC = "AA17";
#NET   SIG_D<3>                          LOC = "Y17";
#CONFIG PROHIBIT = "AA13";
#CONFIG PROHIBIT = "AA14";
#CONFIG PROHIBIT = "Y16";
#CONFIG PROHIBIT = "W16";
#CONFIG PROHIBIT = "Y13";
#CONFIG PROHIBIT = "W14";
#CONFIG PROHIBIT = "Y15";
#CONFIG PROHIBIT = "AA15";
```

THREE PHASE IMPLEMENATION ON REAL TIME, THREE LEVEL

```
// ******************************************//
// Programmed module                        //
// ******************************************//
// Note that temporary patterns are stored in int variable phase1
void programmedModulation(int phase1[5373]){
      count =0;
      count1=1791;
      count2=3582;
                              while (count<=1791){
                                    count = count+1;
                                    if (phase1[count]==1){
                                          phaseA1=0x000020;
                                          phaseA2=0x000020;
                                    }
                                    else if (phase1[count]==2){
                                          phaseA1=0x000002;
                                          phaseA2=0x000002;
```

```
                        }
                        else {
                              phaseA1=0x000020;
                              phaseA2=0x000002;
                        }
                  // for B phase
                        count1=count1+1;
                        if (phase1[count1]==1){
                              phaseB1=0x002000;
                              phaseB2=0x002000;
                        }
                        else if (phase1[count1]==2){
                              phaseB1=0x000200;
                              phaseB2=0x000200;
                        }
                        else {
                              phaseB1=0x002000;
                              phaseB2=0x000200;
                        }
                  // for C phase
                              count2=count2+1;
                              if (phase1[count2]==1){
                                    phaseC1=0x200000;
                                    phaseC2=0x200000;
                              }
                              else if (phase1[count2]==2){
                                    phaseC1=0x020000;
                                    phaseC2=0x020000;
                              }
                              else {
                                    phaseC1=0x200000;
                                    phaseC2=0x020000;
                              }

d_driver=phaseA1+phaseB1+phaseC1;
v_driver=phaseA2+phaseB2+phaseC2;
DRIVER_KONFIGREG(BASEADR_INVERTER0)=v_driver;
DRIVER_KONFIGREG(BASEADR_INVERTER1)=d_driver;
                  }

// second last angle range 120-240
// ***********************************************
count2=0;
while (count<=3582){
      count = count+1;
      if (phase1[count]==1){
            phaseA1=0x000020;
            phaseA2=0x000020;
      }
      else if (phase1[count]==2){
            phaseA1=0x000002;
            phaseA2=0x000002;
      }
      else {
            phaseA1=0x000020;
            phaseA2=0x000002;
      }
```

```
// for B phase
      count1=count1+1;
      if (phase1[count1]==1){
            phaseB1=0x002000;
            phaseB2=0x002000;
      }
      else if (phase1[count1]==2){
            phaseB1=0x000200;
            phaseB2=0x000200;
      }
      else {
            phaseB1=0x002000;
            phaseB2=0x000200;
      }
// for C phase
            count2=count2+1;
            if (phase1[count2]==1){
                  phaseC1=0x200000;
                  phaseC2=0x200000;
            }
            else if (phase1[count2]==2){
                  phaseC1=0x020000;
                  phaseC2=0x020000;
            }
            else {
                  phaseC1=0x200000;
                  phaseC2=0x020000;
            }
d_driver=phaseA1+phaseB1+phaseC1;
v_driver=phaseA2+phaseB2+phaseC2;
DRIVER_KONFIGREG(BASEADR_INVERTER0)=v_driver;
DRIVER_KONFIGREG(BASEADR_INVERTER1)=d_driver;
}

// last angle range 240-360
// ************************************************
count1=0;
while (count<=5373){
      count = count+1;
      if (phase1[count]==1){
            phaseA1=0x000020;
            phaseA2=0x000020;
      }
      else if (phase1[count]==2){
            phaseA1=0x000002;
            phaseA2=0x000002;
      }
      else {
            phaseA1=0x000020;
            phaseA2=0x000002;
      }
// for B phase
      count1=count1+1;
      if (phase1[count1]==1){
            phaseB1=0x002000;
            phaseB2=0x002000;
      }
```

```
            else if (phase1[count1]==2){
                  phaseB1=0x000200;
                  phaseB2=0x000200;
            }
            else {
                  phaseB1=0x002000;
                  phaseB2=0x000200;
            }
      // for C phase
                  count2=count2+1;
                  if (phase1[count2]==1){
                        phaseC1=0x200000;
                        phaseC2=0x200000;
                  }
                  else if (phase1[count2]==2){
                        phaseC1=0x020000;
                        phaseC2=0x020000;
                  }
                  else {
                        phaseC1=0x200000;
                        phaseC2=0x020000;
                  }
      d_driver=phaseA1+phaseB1+phaseC1;
      v_driver=phaseA2+phaseB2+phaseC2;
      DRIVER_KONFIGREG(BASEADR_INVERTER0)=v_driver;
      DRIVER_KONFIGREG(BASEADR_INVERTER1)=d_driver;
      }

}
```

CODE FOR STORING PATTERN IN TEMPORARY MEMORY

```
// ********************************************************************//
// CODE FOR GENERATING PROGRAMMED SINGAL                               //
// FOR THE ENTIRE RANGE DEPENDING UPON THE N                           //
//******************************************************************** //

void getPattern(int K1, int K2, float tp[60]){
            int i=0;
            int count=0;
            while (i<K1){
                        i=i+1;
                  while (count <= tp[i]){
                        count = count +1;
                        phase1[count]=0;
                  }
                        i=i+1;
                  while (count <= tp[i]){
                  count = count+1;
                  phase1[count]=1;
                  }
            }

// reverse direction flow
```

```
            while (i<K2){
                              i=i+1;
                        while (count <= tp[i]){
                              count = count +1;
                              phase1[count]=0;
                        }
                              i=i+1;
                        while (count <= tp[i]){
                        count = count+1;
                        phase1[count]=2;
                        }
            }
// the gap
            while (count <= 5373){
                  count=count+1;
                  phase1[count]=0;
            }


}
```

OVERVIEW OF THE MAIN FUNCTION CODE

```
////////////////////////////////////////////////////////////////////////
/// Implementation of Drives system                                  ///
/// Minimum Loss Synchronous PWM with SPVM on low modulation index   ///
/// Written by : Krishna Neupane                                     ///
/// Supervisor : Tore M. Undeland, NTNU                              ///
/// Co-supervisor : Roy  Nislen, Wartsila Norway As                  ///
/// Date :                                                           ///
////////////////////////////////////////////////////////////////////////

// SPVM DC-link Balancing implemented
// Programmed Modulation DC-link balance implemented

#include "xparameters.h"
#include "AD_omformer_seriemottaker.h"
#include "system_parameters.h"
#include "vekselretter_tilkobling.h"
#include <stdlib.h>
#include <cmath>

int d_enable;
int v_enable;
int d_driver;
int v_driver;
int ADC0;

float m;
int ts; // setting time for space vector modulation
float t1, t0,t01, t02; // switching angles for space vector
float t[72]; // calculation for space vector overall regions
float new_m;

float tp[60]; // patterns timing, maximum 60 for N = 15, transition points
for full cycle 15 x 4 = 60
```

```
// for starting
void starting();

// for space vector
void calcspace();
void spacemodulation();

// for programmed
void getPattern(int K1, int K2);
void programmedModulation(int phase1[5373]);
void calcprogrammed();

// for programmed modulation up to N =15 (Ust = 0.25) maximum
float x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14;
int K1, K2; // for identification between negative and positive pulse
int phase1[5373]; // with standard 50 Hz output
int phaseA1,phaseB1,phaseC1;
int phaseA2,phaseB2, phaseC2;
int count =0;int count1=0;int count2=0; // for three phase


int main()
      {
      // initialization
                  int d_enable = 1;
                  int v_enable =1;
                  RELEDRIVER = 0xF;

                  VE_PAA_SIGNAL_REG(BASEADR_INVERTER0)=d_enable;
                  VE_PAA_SIGNAL_REG(BASEADR_INVERTER1)=v_enable;

x0=0;x1=0;x2=0;x3=0;x4=0;x5=0;x6=0;x7=0;x8=0;x9=0;x10=0;x11=0;x12=0;x13=0;x14
=0;

// infinite loop
while(1)
            {
            ADC0 =      AD_SIGNAL(BASEADR_AD,0);
            count = 0;
            m= ADC0;
            new_m = m;


// Space Vector Modulation if modulation index is less than equal to .25
                              if (new_m<=0.25) {
                                    calcspace();
                                                            while (new_m==m){

      spacemodulation();
                                                                  m =
AD_SIGNAL(BASEADR_AD,0);
                                                            }
                              }

// Programmed Modulation for Modulation index higher than 0.25
```

```
                              else if (new_m<= 1.2){
                                    calcprogrammed();
                                    getPattern(K1, K2);
                                                            while (new_m==m){

      programmedModulation(phase1);
                                                                  m =
AD_SIGNAL(BASEADR_AD,0);
                                                            }
                              }
// Determine the error in the value of modulation index
                              else {
                                    RELEDRIVER = 0x0; // indicates the error
                              }
            }
}
```

ADC IMPLEMENTATION WITH VHDL

```
library ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity missionADC is
port(
-- ADC ports configuration
AD_D_P : in std_logic_vector (7 downto 0);
AD_D_N : in std_logic_vector (7 downto 0);
AD_FCOP : in std_logic;
AD_FCON : in std_logic;
AD_DCOP : in std_logic;
AD_DCON : in std_logic;
AD_REF_KLOKKE_P : out std_logic;
AD_REF_KLOKKE_N : out std_logic;

-- bus protocal ports
Bus2IP_Clk : in std_logic;
Bus2IP_Reset : in std_logic;
Bus2IP_Data : in std_logic_vector (0 to 31);
Bus2IP_BE : in std_logic_vector (0 to 8);
Bus2IP_RdCE : in std_logic_vector (0 to 3);
Bus2IP_WrCE : in std_logic_vector (0 to 3);
IP2Bus_Data : out std_logic_vector (0 to 31);
IP2Bus_RdAck : out std_logic;
IP2Bus_WrAck : out std_logic;
IP2Bus_Error : out std_logic;
```

```
-- output terminals
AD_SIGNAL_A : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_B : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_C : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_D : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_E : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_F : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_G : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_H : OUT STD_LOGIC_VECTOR (11 downto 0);
AD_SIGNAL_NY : OUT STD_LOGIC
);
end missionADC;

architecture Behavioral of missionADC is

-- signals used
signal Clock : std_logic := '0';
signal reset_h : std_logic := '1';
signal AD_200MHZ : std_logic := '0';
signal resetpin : std_logic := '0';
signal addressteller : std_logic_vector (0 to 31) := (others => '0');
signal next_address : std_logic_vector (0 to 31) := (others => '0');
signal address : std_logic_vector (0 to 31) := (others => '0');
signal writebus : std_logic_vector (0 to 31) := (others => '0');
signal readbus : std_logic_vector (0 to 31) := (others => '0');
signal bytenable : std_logic_vector ( 0 to 4) := (others => '0');
signal readvalue : std_logic_vector (0 to 31) := (others => '0');

signal ad_klokke_d_p : std_logic := '0';
signal ad_klokke_d_n : std_logic := '0';
signal ad_klokke_f_p : std_logic := '0';
signal ad_klokke_f_n : std_logic := '0';

signal AD_SIGNAL_A : std_logic_vector (11 downto 0);
signal AD_SIGNAL_B : std_logic_vector (11 downto 0);
signal AD_SIGNAL_C : std_logic_vector (11 downto 0);
signal AD_SIGNAL_D : std_logic_vector (11 downto 0);
signal AD_SIGNAL_E : std_logic_vector (11 downto 0);
signal AD_SIGNAL_F : std_logic_vector (11 downto 0);
signal AD_SIGNAL_G : std_logic_vector (11 downto 0);
signal AD_SIGNAL_H : std_logic_vector (11 downto 0);

signal ad_d_ut_p : std_logic_vector (7 downto 0) :="00000000";
signal ad_d_ut_n : std_logic_vector (7 downto 0) :="00000000";

signal user_Bus2IP_Rd
```

```
begin

AD_D_P <= ad_d_ut_p,
AD_D_N <= ad_d_ut_n,
AD_FCOP <= ad_klokke_f_p,
AD_FCON <= ad_klokke_f_n,
AD_DCOP <= ad_klokke_d_p,
AD_DCON <= ad_klokke_d_n,

AD_REF_KLOKKE_P <= open,
AD_REF_KLOKKE_N <= open,
AD_REF_KLOKKE_INN <= ref_klokke_t,
AD_DATA_KLOKKE_UT <= open,

--AD_SIGNAL_NY_BUSCLK => open,
--AD_SIGNAL_NY_ADCLK => open,
AD_SIGNAL_UT <= AD_SIGNAL_UT,
AD_SIGNAL_A <= AD_SIGNAL_A,
AD_SIGNAL_B <= AD_SIGNAL_B,
AD_SIGNAL_C <= AD_SIGNAL_C,
AD_SIGNAL_D <= AD_SIGNAL_D,
AD_SIGNAL_E <= AD_SIGNAL_E,
AD_SIGNAL_F => AD_SIGNAL_F,
AD_SIGNAL_G => AD_SIGNAL_G,
AD_SIGNAL_H => AD_SIGNAL_H,

Bus2IP_Clk => klokke,
Bus2IP_Reset => reset_h,
Bus2IP_Data => skrivebuss,
Bus2IP_BE => "1111",
Bus2IP_RdCE => user_Bus2IP_RdCE,
Bus2IP_WrCE => user_Bus2IP_WrCE,
IP2Bus_Data => readbus,
--IP2Bus_RdAck => open,
--IP2Bus_WrAck => open,
--IP2Bus_Error => open



end Behavioral;
```

Appendix 6

MORE LAB RESULTS

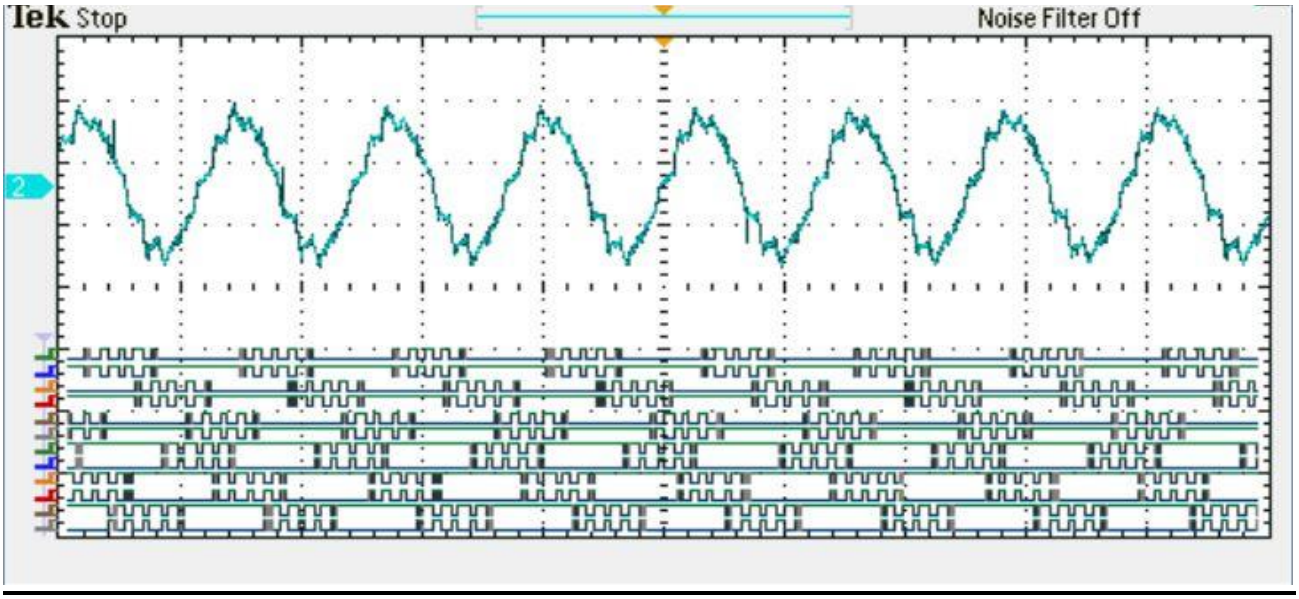


Figure 86: Current Signal along with Driver signals, U_{st} = 0.5, f_{sw} = 200 Hz, f_s = 50 Hz

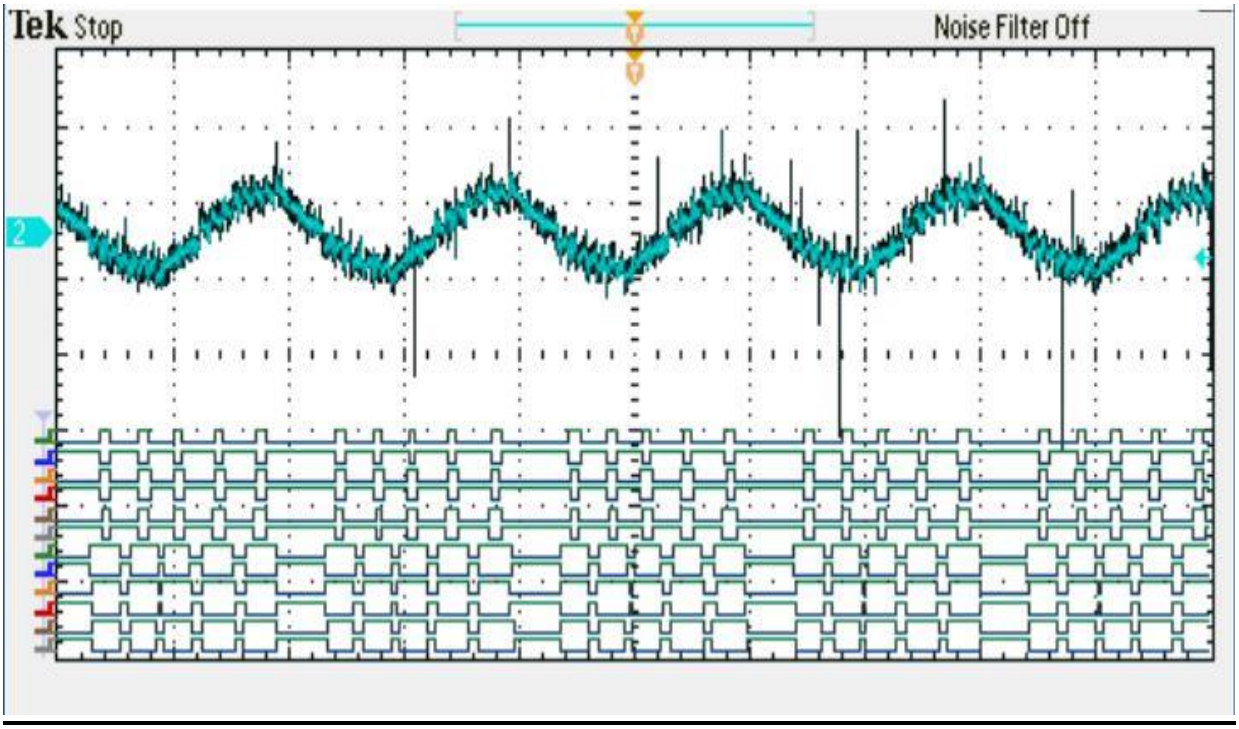


Figure 87: Current Signal with Driver Signals, SVPWM, f_{sw} = 200 Hz, f_s = 50 Hz, U_{st} = 0.12

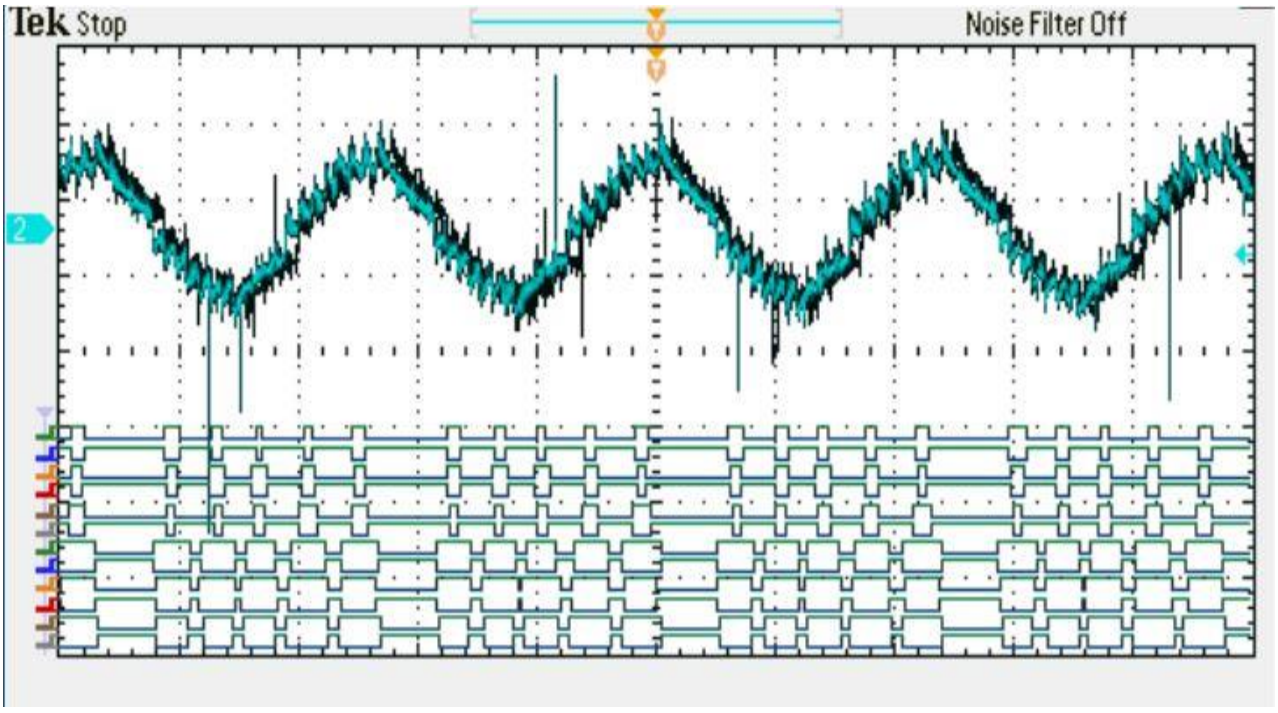


Figure 88: Current Signal with Driver Signals, SVPWM, f_{sw} = 200 Hz, f_s = 50 Hz, U_{st} = 0.2

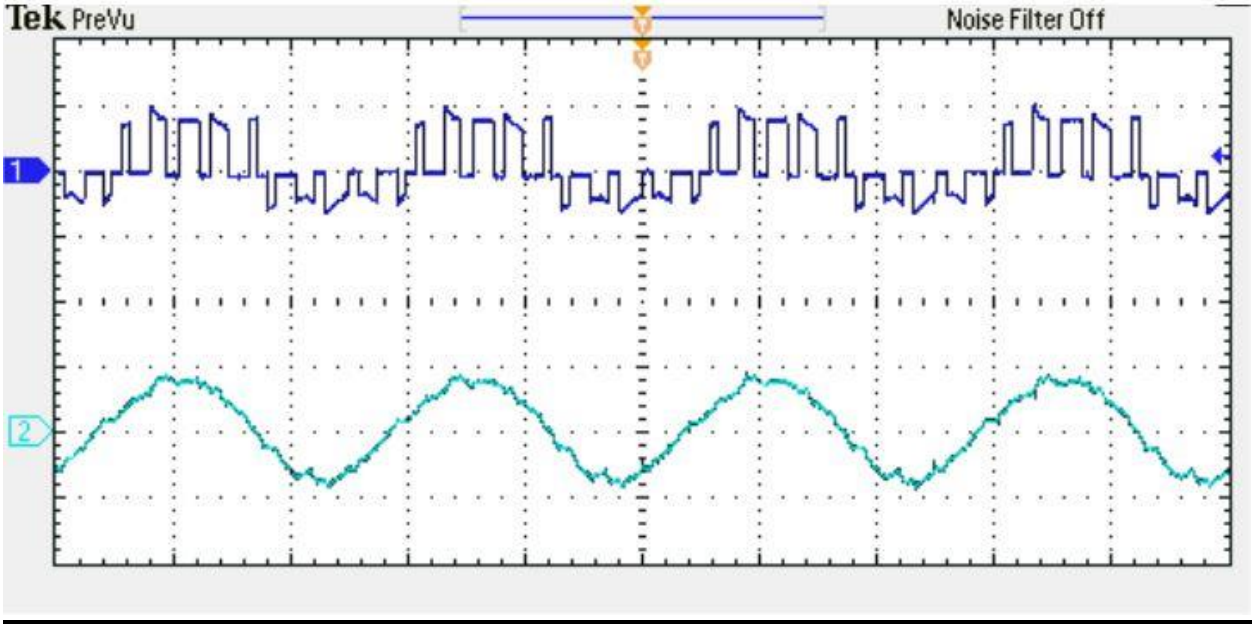


Figure 89: Phase Voltage Signal (Up) with Current Signal (Down), SOM, fsw = 200 Hz, fs = 50 Hz, Ust = 0.71

Table 6: Modulation Value vs Current Harmonics Level, from Lab

Modulation	Current THD %
0.12	27
0.22	27
0.258	12
0.35	11.3
0.45	11
0.5	10
0.55	10.5
0.65	8.3
0.71	9.2
0.75	9.5
0.85	10.1
0.95	8.5
1	7
1.1	7.2
1.2	7.4

Appendix 7

HARDWARE COMPONENTS

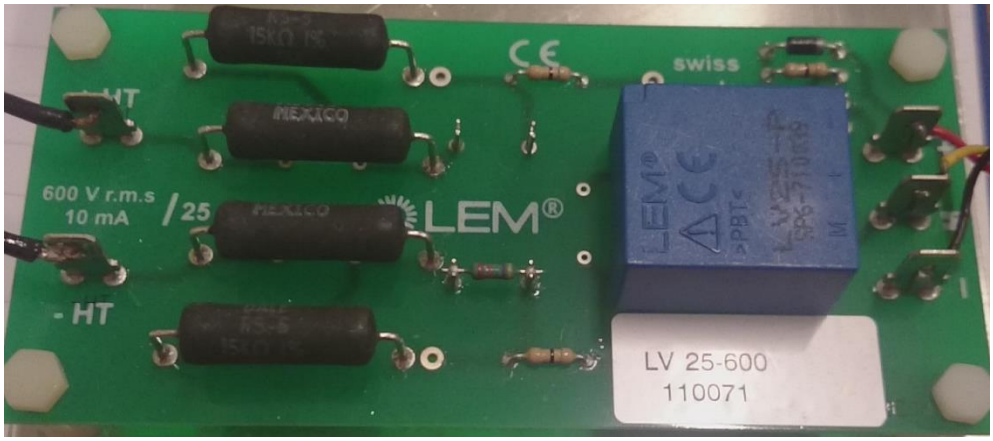


Figure 90: LV 25-600 for Feedback signal feeding to FPGA Board

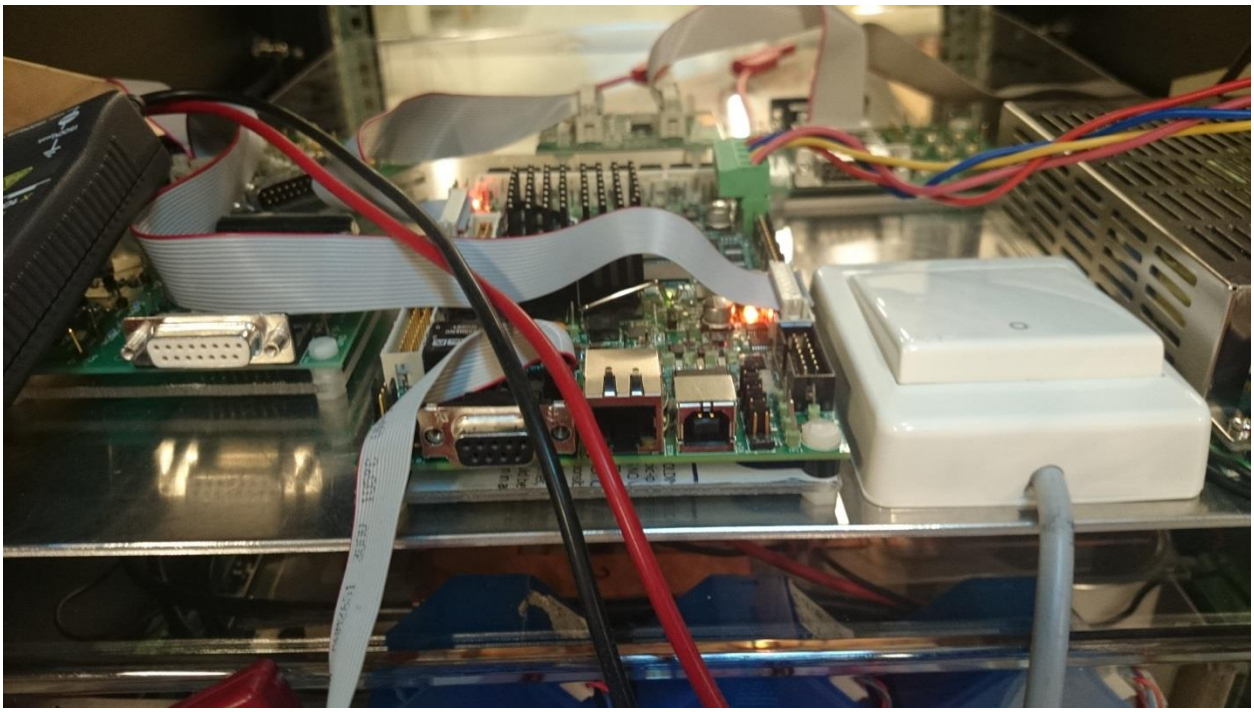


Figure 91: FPGA Board with Driver circuit

Figure 92: Three phase auto-transformer

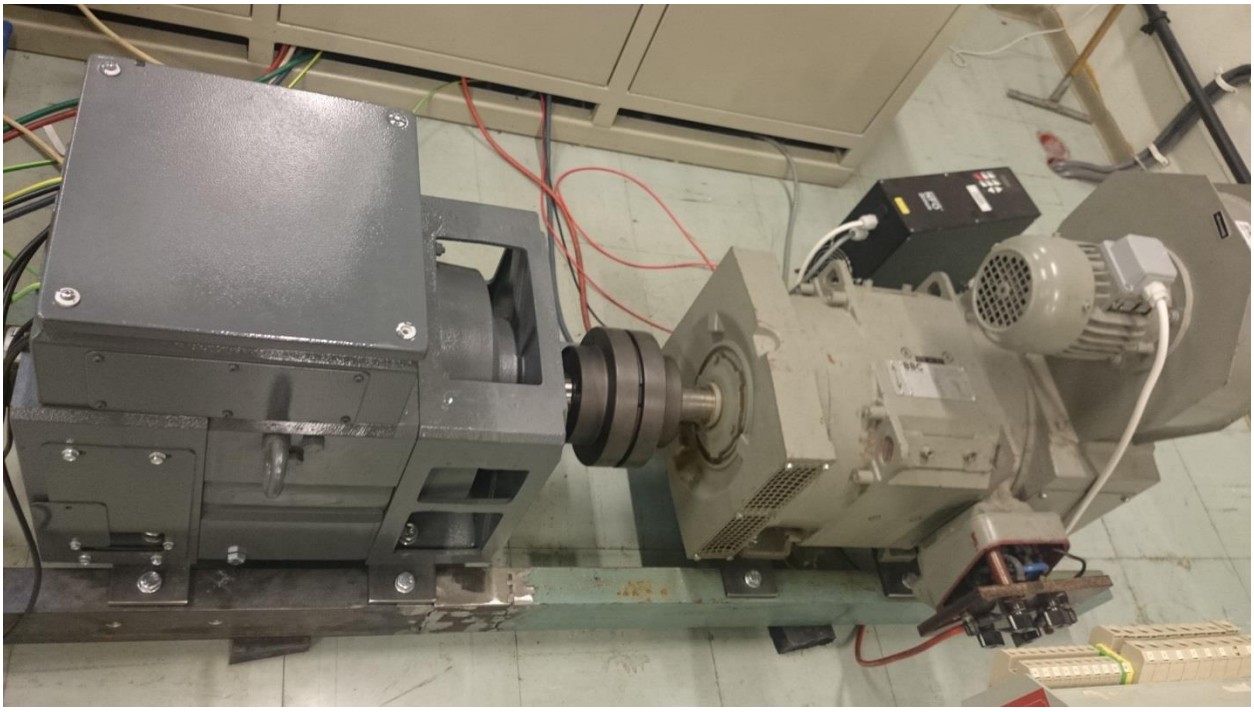


Figure 93: Induction machine connected to DC Motor

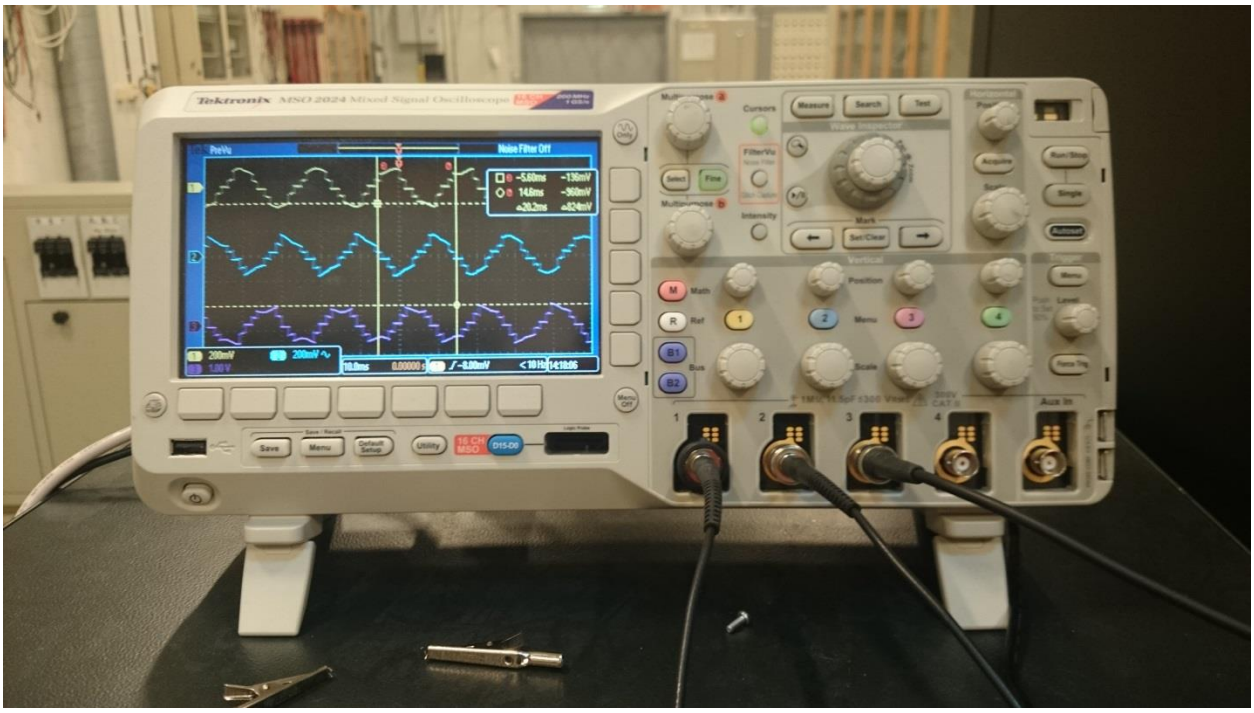


Figure 94: Oscilloscope MSO2024