CAPACITOR-LESS VAR COMPENSATOR

BASED ON A MATRIX CONVERTER

A Thesis

by

DIVYA RATHNA BALAKRISHNAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2010

Major Subject: Electrical Engineering

Capacitor-Less VAR Compensator

Based on a Matrix Converter

CAPACITOR-LESS VAR COMPENSATOR

BASED ON A MATRIX CONVERTER

A Thesis

by

DIVYA RATHNA BALAKRISHNAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Robert S. Balog
Committee Members,	Hamid Toliyat
	Shankar Bhattacharyya
	Won-jong Kim
Head of Department,	Costas Georghiades

December 2010

Major Subject: Electrical Engineering

ABSTRACT

Capacitor-Less VAR Compensator

Based on a Matrix Converter. (December 2010)

Divya Rathna Balakrishnan, B.Tech., National Institute of Technology, Karnataka, India

Chair of Advisory Committee: Dr. Robert S. Balog

Reactive power, denoted as volt-ampere reactive (VARs), is fundamental to ac power systems and is due to the complex impedance of the loads and transmission lines. It has several undesirable consequences which include increased transmission loss, reduction of power transfer capability, and the potential for the onset of system-wide voltage instability, if not properly compensated and controlled. Reactive power compensation is a technique used to manage and control reactive power in the ac network by supplying or consuming VARs from points near the loads or along the transmission lines. Load compensation is aimed at applying power factor correction techniques directly at the loads by locally supplying VARs. Typical loads such as motors and other inductive devices operate with lagging power factor and consume VARs; compensation techniques have traditionally employed capacitor banks to supply the required VARs. However, capacitors are known to have reliability problems with both catastrophic failure modes and wear-out mechanisms. Thus, they require constant monitoring and periodic replacement, which greatly increases the cost of traditional load compensation techniques.

This thesis proposes a reactive power load compensator that uses inductors (chokes) instead of capacitors to supply reactive power to support the load. Chokes are regarded as robust and rugged elements; but, they operate with lagging power factor and thus consume VARs instead of generating VARs like capacitors. A matrix converter interfaces the chokes to the ac network. The matrix converter is controlled using the Venturini modulation method which can enable the converter to exhibit a current phase reversal property. So, although the inductors draw lagging currents from the output of the converter, the converter actually draws leading currents from the ac network. Thus, with the proposed compensation technique, lagging power factor loads can be compensated without using capacitor banks.

The detailed operation of the matrix converter and the Venturini modulation method are examined in the thesis. The application of the converter to the proposed load compensation technique is analyzed. Simulations of the system in the MATLAB and PSIM environments are presented that support the analysis. A digital implementation of control signals for the converter is developed which demonstrates the practical feasibility of the proposed technique. The simulation and hardware results have shown the proposed compensator to be a promising and effective solution to the reliability issues of capacitor-based load-side VAR compensation techniques.

To my family

ACKNOWLEDGEMENTS

I would like to express my deep sense of gratitude and regard to my advisor and mentor, Dr. Robert Balog. His invaluable guidance and unstinted support have made this work possible. I would like to thank him for being so encouraging, understanding and patient at all times. I am happy to have had the opportunity to pursue research under his guidance and am grateful to him for the many things I have learnt during the course of my work.

My sincere thanks are due to Dr. Hamid Toliyat, Dr. Shankar Bhattacharyya and Dr. Won-jong Kim for sparing their valuable time to serve on my defense committee. I would also like to thank Dr. Prasad Enjeti for his initial guidance on my research.

I am grateful to the entire staff of the Electrical Engineering Department for supporting me during my academic program at Texas A&M University. I am thankful to my research group for making my laboratory experience so wonderful. I would like to specially thank Mehran, Daniel and Somu for their assistance with my research work.

I am deeply indebted to my parents, sister and brother-in-law whose unconditional love, support and encouragement have greatly contributed to the completion of this work. I owe my gratitude to them for always believing in me and helping me through my academic pursuit.

NOMENCLATURE

VAR	Volt-ampere reactive
MC	3-phase ac-to-3-phase ac matrix converter
$v_{1,LN}$	Line-neutral ac network voltage (phase 1)
$v_{2,LN}$	Line-neutral ac network voltage (phase 2)
$v_{3,LN}$	Line-neutral ac network voltage (phase 3)
$v_{i1,LN}$	Line-neutral matrix converter input voltage (phase 1)
$v_{i2,LN}$	Line-neutral matrix converter input voltage (phase 2)
$v_{i3,LN}$	Line-neutral matrix converter input voltage (phase 3)
$v_{o1,LN}$	Line-neutral matrix converter output voltage (phase 1)
$v_{o2,LN}$	Line-neutral matrix converter output voltage (phase 2)
$v_{o3,LN}$	Line-neutral matrix converter output voltage (phase 3)
i_1	Ac network line current (phase 1)
i_2	Ac network line current (phase 2)
i_3	Ac network line current (phase 3)
i_{Load1}	Load current drawn from ac network (phase 1)
i_{Load2}	Load current drawn from ac network (phase 2)
i_{Load3}	Load current drawn from ac network (phase 3)
i_{i1}	Matrix converter input current (phase 1)
i_{i2}	Matrix converter input current (phase 2)
i_{i3}	Matrix converter input current (phase 3)

i_{o1}	Matrix converter output current (phase 1)
i_{o2}	Matrix converter output current (phase 2)
i_{o3}	Matrix converter output current (phase 3)
Φ_i	Initial phase of matrix converter input voltage (phase 1)
Φ_o	Initial phase of matrix converter output voltage (phase 1)
ω_i	Angular frequency of matrix converter input voltages
ω_o	Angular frequency of matrix converter output voltages
ω	Angular frequency of ac network voltages
$V_{LN,rms}$	RMS value of ac network voltages
L_{MC}	3-phase choke at output of matrix converter
$\mathbf{H}$	Modulation matrix of matrix converter
$\mathbf{S}$	Switching matrix of matrix converter
$H_1\,H_2\,H_3$	Modulation functions of the Venturini method
$S_1\,S_2\,S_3$	Switching functions of the Venturini method
q	Modulation index of the matrix converter
f_{sw}	Switching frequency of the matrix converter

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Reactive power, known as volt-ampere reactive (VARs), is attributed to the complex impedance of typical ac power system loads and transmission lines. Even though its presence in the power system network is fundamental, it has several undesirable consequences including reduced stability limits (steady-state, dynamic and transient) and lowered power transfer capability. It also leads to increased transmission losses, inefficient performance of power system equipment and the potential for the onset of system-wide voltage instability (caused by variation in the reactive power demands of the loads), if not properly compensated and controlled [1]. *Reactive power compensation* is defined as the management and control of reactive power in the ac network, achieved by supplying or absorbing VARs from the system [2].

A. Classification of Compensation Techniques

VAR compensation may be achieved by absorbing or injecting reactive power at either the transmission level, known as *transmission compensation*, or near the load, known as *load compensation* [1]. Transmission compensation is used to maintain the specified voltage at different buses by injecting variable amounts of reactive power into the transmission lines. As a result, the stability limits and the power transfer capability are improved and tighter system voltage control is achieved. Transmission compensation

This thesis follows the style of *IEEE Transactions on Power Electronics*.

is performed at the ac network level in the system and does not address the load-side source of VARs in the system - the reactive load. Load compensation, on the other hand, is aimed at applying power factor correction techniques directly at the load to locally supply VARs required by the loads. Consequently, the effective system load – the original intended load together with the compensator, appear as unity power factor loads that do not draw VARs from the source. So, transmission losses and required current carrying capacity of lines are minimized. Further, the power transfer capability and stability limits of the system are not deteriorated by the load and voltage instability is simplified. Large scale customers such as industries are often penalized for drawing excessive reactive power from the ac network; so they are motivated to employ load compensation techniques to locally supply VARs to their loads.

B. Previous Work in Reactive Power Compensation

Several reactive power compensation solutions that are applicable to load compensation have been proposed in the past and are summarized below [2-5]. Load compensation employs shunt-connected techniques and so, series compensation methods have been omitted from the discussion.

1. Mechanically switched capacitors and reactors

Initially, mechanically switched capacitor and reactor banks were used to provide lagging and leading reactive power to support the power system. Depending on the VAR requirement, the banks are switched in and out of the system through mechanical relays and circuit breakers. This approach is fundamentally discrete and the compensation solution does not offer continuously variable reactive power support. It also suffers from other drawbacks including slow speed of response, lifetime wear-out of the switch elements, switching transients and lack of flexible, continuously variable VAR compensation.

2. Synchronous condensers

Synchronous condensers are essentially synchronous motors operated at no-load. By varying the excitation field, the rotating loads can be made to supply VARs (under-excited) or consume VARs (over-excited) from the ac system. The advantages of synchronous condensers include continuously variable reactive power support, high short-term overload capability and harmonic-free operation. However, they suffer from major disadvantages including high installation time and costs, maintenance costs, mechanical losses and slow response time. Further, they contribute to high system fault current and cannot be easily relocated due to their large size.

3. Static VAR Compensator (SVC)

With advances in semiconductor technology, the solutions listed above were subsequently replaced by the static VAR compensator (SVC) which consists of banks of capacitors and reactors that are switched on/off or phase-controlled using thyristors. The number of banks switched in at a time depends on the VAR requirement.

a. Thyristor-Switched Capacitor (TSC)

A thyristor-switched capacitor (TSC) consists of banks of capacitors switched in and out of the ac power system using a pair of anti-parallel thyristors as shown in Fig. 1. The switching in and switching out actions are carried out at instants when the capacitor voltage equals the positive or negative peak of the ac line voltage, to prevent switching transients. This introduces a maximum response delay of 1 cycle when switching in and ½-cycle when switching out the capacitor bank [2, 5]. A current-limiting series reactor is used to prevent any likelihood of switching transients. The reactor and capacitor form a notch filter to prevent resonance with the power system current harmonics. TSC banks may be connected in wye or delta configurations, though the latter is preferred when unbalanced VAR consumption is expected [5].

A TSC has good response time and low maintenance and installation costs as it has no moving parts. It also has negligible harmonics when switched appropriately. However, a TSC cannot provide continuous VAR control as it is essentially just a solid-

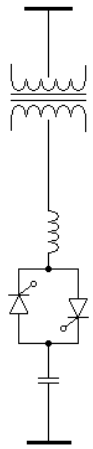


Fig. 1. Thyristor-switched capacitor

state version of the mechanically switched capacitor bank. As each TSC bank requires a pair of thyristors, it is uneconomical especially at high voltage levels due to the cost of high voltage thyristors and their gate drive circuits. The peak inverse voltage of each thyristor is twice the ac network voltage peak. Protective equipment must be installed to prevent thyristor failure due to line voltage transients and fault currents. Due to these disadvantages, a TSC is practically not an attractive compensation solution.

b. Thyristor-Controlled Reactor (TCR)

A thyristor-controlled reactor (TCR) consists of shunt capacitors in parallel with reactors connected to the power network through a pair of anti-parallel thyristors as shown in Fig. 2. The effective reactance of the TCR and so, the reactive power drawn by it, are controlled by varying the firing angles of the thyristors. As the firing angle increases, the effective inductance of the TCR increases. A filter is used in parallel with the TCR as the gating action generates low-order odd harmonics. In three-phase

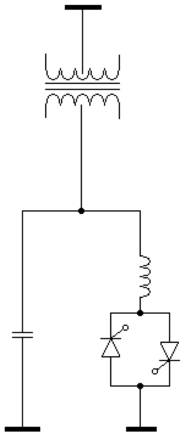


Fig. 2. Thyristor-controlled reactor

arrangements, the reactors of the TCR are connected in delta to remove unbalance, while the capacitors may be connected in delta or wye.

With fast response times, acceptable cost and the ability to balance loads, the TCR is considered to be a good compensation technique. However, though a continuous VAR control range can be achieved, it is discontinuous in time as firing angle adjustments can be made only once per ½-cycle. The other disadvantage of the TCR is that the reactor must be of a rating comparable to that of the capacitor to be able to provide leading VARs as well.

4. Static Synchronous Compensator (STATCOM)

A static synchronous compensator (STATCOM) is based on power electronic converters that behave as ideal ac sources such as the voltage-source and current-source inverters shown in Fig. 3 and Fig. 4, respectively. The converters are connected to the ac power system network through reactors as shown in Fig. 5. Depending on the relative

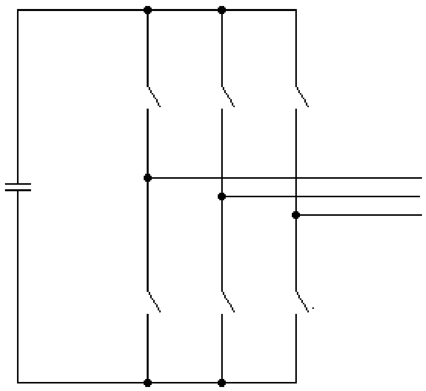


Fig. 3. VSI-based STATCOM

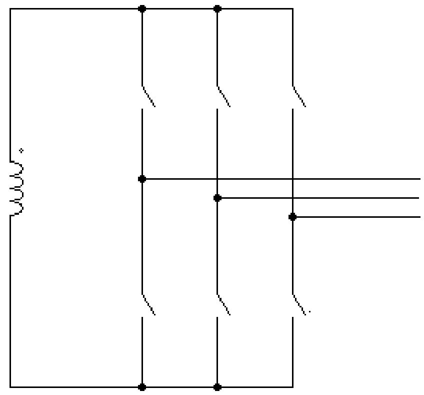


Fig. 4. CSI-based STATCOM

magnitudes of the ac network voltage E and the converter voltage V, variable amounts of reactive power are injected into the network. When |E| > |V|, the converter injects lagging VARs into the ac network and vice versa [2, 5].

The power electronic converters are made to operate at high switching frequencies using various pulse-width modulation (PWM) techniques. So, the STATCOM can provide smoothly variable reactive power as a continuous function of time. The high frequency switching harmonics generated can be easily filtered. More importantly, the STATCOM does not require large number of reactive elements, thus reducing the cost ad size of compensation techniques. Since the STATCOM is a force-commutated converter operating at high frequencies, it generally employs switches such as insulated-gate bipolar transistors (IGBTs) and integrated gate-commutated thyristors (IGCTs). However, these switches are not yet developed for high voltage ratings. To overcome this issue, multi-level converters are used [2]. The STATCOM is able to

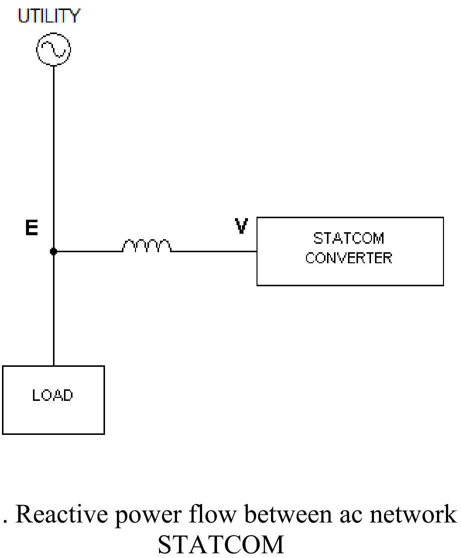


Fig. 5. Reactive power flow between ac network and STATCOM

provide compensation characteristics almost identical to the synchronous condensers. However, it does not have the high overload capability of the rotating synchronous machine.

5. Compensation using Thyristor-Based Cycloconverters

Direct ac-ac cycloconverters, naturally-commutated and force-commutated, can be used for reactive power compensation [5] in a way similar to inverters, as shown in Fig. 6. As the converter is made to draw only reactive or harmonic power, it is sufficient to use passive tank circuits at the input of the converter. While naturally commutated cycloconverters can inject only leading VARs into the system, force-commutated

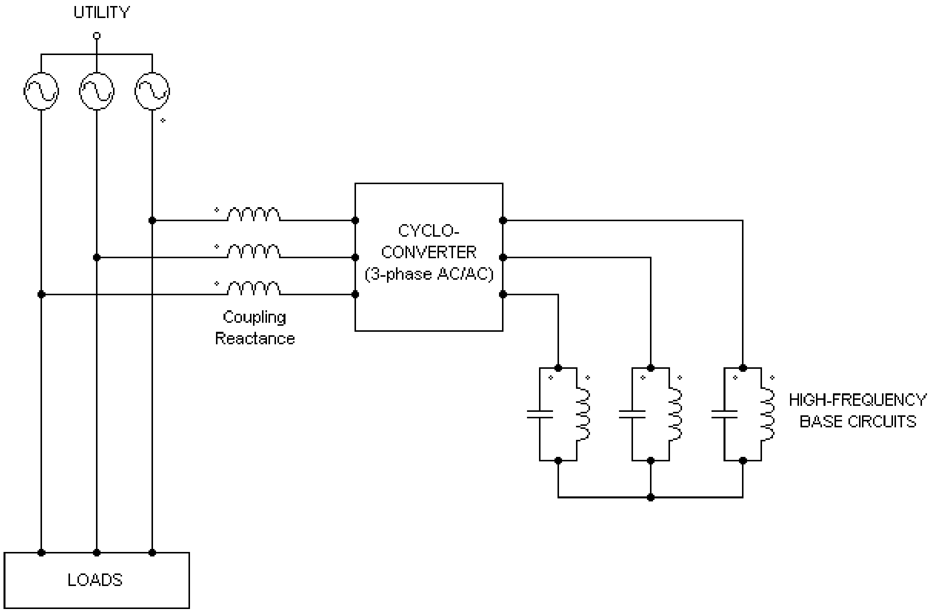


Fig. 6. STATCOM based on a cycloconverter

cycloconverters can be controlled to inject lagging VARs as well [5]. Another STATCOM based on the concept of "*power doubling*" was proposed by Gyugyi [5] and implemented in [6], wherein reactive power could be injected by the cyclo-converter through both its input and output terminals.

Though the cycloconverters seemed to be compact solutions to reactive power compensation, they did not gain much popularity as they required a very large number of thyristors (36 switches for three-phase converter) which needed to be switched at very high frequencies (approximately ten times the line frequency), which was not possible at that time.

C. Capacitors in Reactive Power Compensation

Many typical loads such as motors and inductive loads operate with lagging power factor, which means that they consume VARs. So, load compensation techniques employ capacitor banks to locally supply the VARs needed by the load. However, capacitors are known for being highly unreliable components with both catastrophic and wear-out failure mechanisms. With the extensive use of power electronics in VAR compensation (as in the voltage source inverter-based STATCOM), aluminum electrolytic dc capacitors are widely used due to their advantages of high energy density, reasonable voltage ratings and low cost per unit energy. But, 60% of power electronic failures are attributed to them [7]. They have a short life-span (generally, less than 10,000 hours at rated conditions) and must be frequently replaced. The reliability issues of aluminum electrolytic capacitors are discussed below.

1. Construction of the Electrolytic Capacitor

A cross-section of the electrolytic capacitor is shown in Fig. 7. The anode of the capacitor is an aluminum foil that is coated with aluminum oxide, the dielectric which is formed through chemical reactions. The cathode consists of a paper strip impregnated with an electrolyte, which is in contact with another aluminum foil. Several layers of cathode and anode are wound in an alternating manner [8].

2. Failure Types and Mechanisms of the Electrolytic Capacitor

Capacitor failure types can be listed as follows [9]:

i. Early failures: They occur during the first year of energizing the capacitors. They are attributed to defects in the manufacturing, testing and installation procedures.

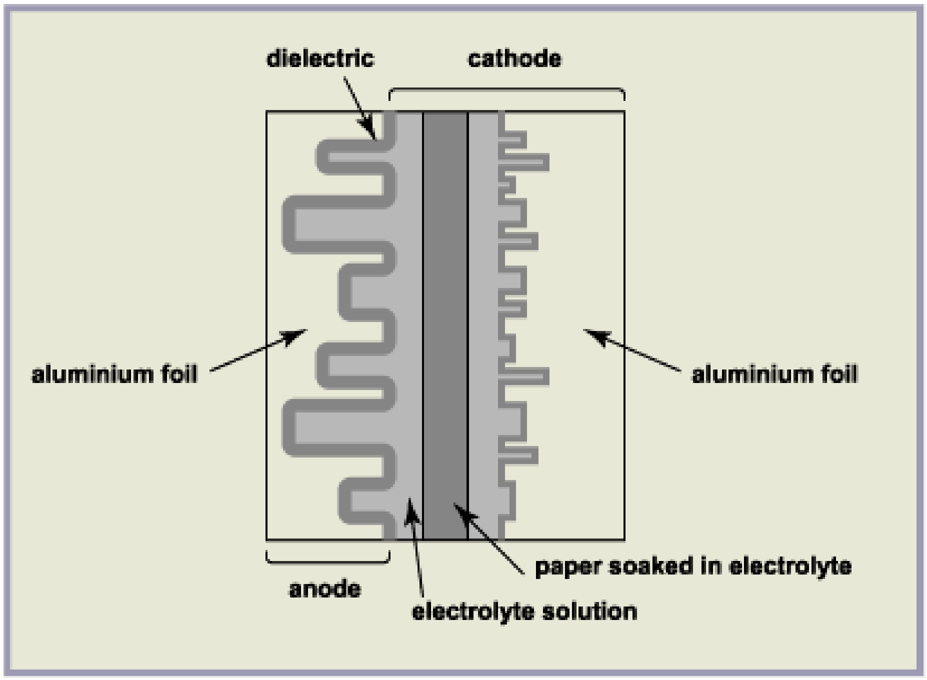


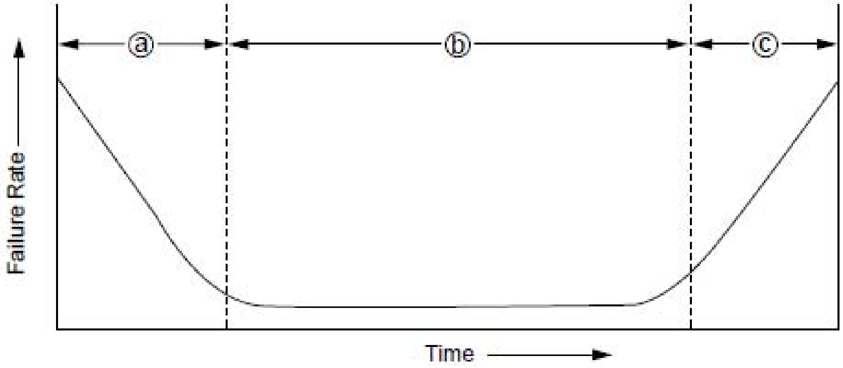
Fig. 7. Cross-section of an electrolytic capacitor – taken from [8]

ii. Random failures: These less probable failures occur due to operating conditions such as lightning surges.

iii. Wear-out failures: These are attributed to wear-out and aging of the capacitor dielectric. They form a large percentage of capacitor failures and will be discussed in detail.

These failures are shown as a function of time in the bathtub curve of Fig. 8.

Capacitor failures occur through the several mechanisms/modes listed in Fig. 9. Early and random failures are attributed to catastrophic mechanisms including open-circuit, short-circuit, open vent, increased leakage current and electrolytic leakage [10]. Wear-out failures in electrolytic capacitors are primarily driven by a mechanism called '*partial discharge*' which can be defined as the incomplete charge transfer (discharge) occurring across the space (such as gaps) between the electrode and the electrolyte [11]. Partial discharge leads to the decomposition of electrolyte, producing hydrogen gas



a – Early failure period b – Random failure period c – Wear-out failure period

Fig. 8. Bathtub curve of capacitor failures – taken from [10]

which decreases the corona inception voltage (breakdown voltage) below the voltage rating of the capacitor. Operation of such a capacitor results in high internal temperatures, arcing, bulging and its consequent failure. The factors contributing to the occurrence of partial discharge include higher operating temperatures, manufacturing defects and conditions such as over-voltages.

3. Failure Detection Mechanisms

The equivalent model of a capacitor, shown in Fig. 10, includes an effective series resistance (ESR) and an effective series inductance (ESL) along with the capacitor [12]. Failure of capacitors is accompanied by a decrease in its capacitance and, more

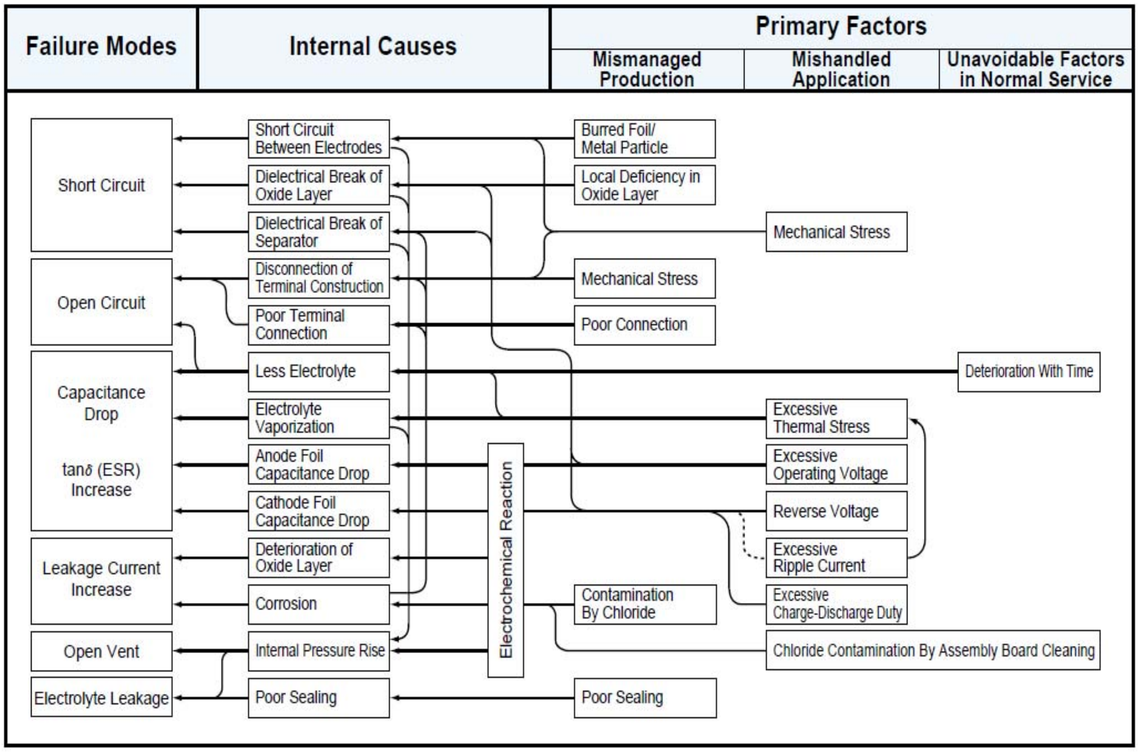


Fig. 9. Different failure mechanisms of capacitor failures – taken from [10]

importantly, by an increase in the ESR. Several techniques have been developed to use the ESR as a means to monitor the health of electrolytic capacitors and accurately detect faults [7, 12, 13].

From the above discussion, it is clear that aluminum electrolytic dc capacitors, with their catastrophic failure modes and wear-out mechanisms, require constant monitoring and periodic replacement. Ac capacitors also suffer from similar reliability problems. Thus, capacitors greatly increase the maintenance and operational costs of the traditional load compensation techniques and are undesirable components.

D. Proposed Compensation Technique

This thesis proposes a reactive power load compensator that uses inductors (chokes) instead of capacitors to supply reactive power to support the load. Inductors are regarded as robust and rugged elements and are not subject to service life-limiting failure mechanisms. However, it is also well known that they have lagging power factor and consume VARs, which is the opposite reactive power behavior of the capacitor. The proposed load compensator interfaces the inductive choke element to the ac network with a 3-phase ac-to-3-phase ac direct matrix converter (MC). The MC is controlled by the Venturini modulation technique which has the advantageous property that it can

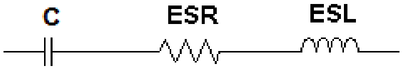


Fig. 10. Equivalent circuit of a capacitor

enable the MC to invert the phase of the current from the input to the output [14]. So, although the inductive choke draws lagging currents from the output of the converter, through this current phase reversal property of the modulation technique, the converter draws leading currents from the ac network. Thus, with the proposed compensation technique, lagging power factor loads can be compensated without using capacitor banks. This thesis will examine the fundamental relationships of the matrix converter controlled by the Venturini modulation technique and apply them to a shunt-connected load-side VAR compensator.

E. Overview of Thesis

The matrix converter and its historical background are introduced in Chapter II. Detailed operation of the converter using the Venturini modulation method is presented, and the derivation of the current phase reversal property is examined. Simulation results of the converter in MATLAB and PSIM environments are produced and discussed in detail.

The schematic of the proposed VAR load compensator is presented in Chapter III. The application of the current reversal property to the compensator is justified. The capacity of the compensator is given mathematically. MATLAB and PSIM simulations examining the performance of the compensator are also presented.

The digital implementation of the compensator in hardware is presented in Chapter IV. A possible solution to the hardware set-up of the entire system is explored.

Control signals of the system, generated using a DSP and a CPLD, are presented. The practical feasibility of the proposed compensator is thus shown.

Conclusion drawn from this thesis are presented in Chapter V. Future work is also discussed.

Portions of Chapters II, III and IV have been previously published in [15] at the IEEE 42nd North American Power Symposium (NAPS) 2010.

CHAPTER II

MATRIX CONVERTER AND ITS OPERATION

A. Introduction to the Matrix Converter

A *matrix converter* is a direct ac-to-ac force-commutated converter which has the ability to transform the magnitude, phase angle and frequency of the input voltage without using intermediate energy storage elements. An *m/n* matrix converter results in energy conversion between '*m*' output phases and '*n*' input phases. As 3-phase power is most widely used, the 3/3 matrix converter (MC) will be considered hereafter. The basic layout of the MC, shown in Fig. 11, illustrates that each of the output phases is connected to every input phase by a matrix of switches. There are no intermediate

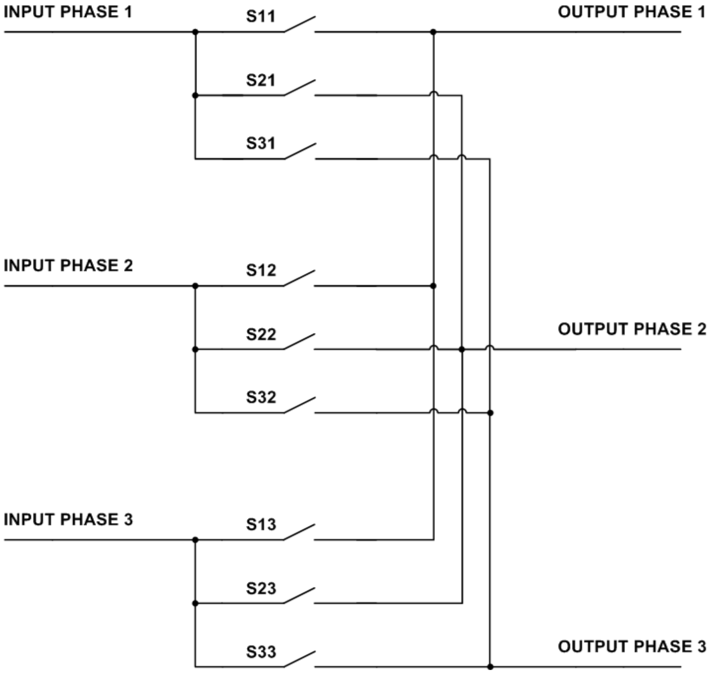


Fig. 11. Basic layout of a 3/3 MC

energy storage elements and the switches are *bi-directional* (*bi-lateral*) – capable of handling positive as well as negative voltages and currents (four-quadrant operation).

The concept of direct ac-to-ac converters was introduced by Hazeltine in 1923 [16], but a detailed treatment was first carried out by Gyugyi and Pelly in [16], where they are referred to as *frequency changers* or *cycloconverters*. These converters were based on the semiconductor device *thyristor* and were broadly classified as naturally commutated (NCCs) and force-commutated (FCCs). Though cycloconverters offered a wide range of operational flexibility, their widespread usage was hindered by the large requirement of thyristors (36 switches for 3-phase applications) and the difficulty of switching these early technology thyristors at relatively high frequencies (approximately ten times the line frequency). With developments in power transistor technology, interest in cycloconverters was reestablished [17]. However, the earnest development of direct ac-ac converters was initiated by the Venturini and Alesina [14, 18] in which they introduced the concept of the matrix converter. They developed and analyzed the principle of the '*low-frequency modulation matrix*', which views the converter as a single mathematical transfer function (direct modulation method). Following the work of Venturini and Alesina, several modulation methods were proposed as summarized in [17].

The Venturini modulation method is characterized by sinusoidal input and output waveforms. It has a maximum attainable voltage gain of 0.5, which could be increased to 0.866 by the addition of common-mode voltages to the output voltages [19]. But, this increase is obtained at the cost of greater switching frequencies and implementation complexities. The indirect modulation methods, which model the MC as a two-stage transfer function, were analyzed in [20]. These methods were able to increase the voltage gain to 1.05 at the expense of low-frequency distortion in the input/output waveforms or both [17]. Space vector modulation methods were developed in [21-24] with the aim of achieving the superior performance of space vector based inverters in MCs as well. But, even these methods have a limited voltage gain of 0.866 and offer similar performance as the Venturini modulation method.

This thesis implements the Venturini modulation method because it can be operated such that it has the advantageous current phase reversal property, which is fundamental to the operation of the proposed reactive power load compensator. The maximum voltage gain of 0.5 is not viewed as a limitation in this application of a matrix converter as the output terminals of the MC are internal nodes of the VAR compensator and do not connect to any load such as a motor, where the output voltage is important. Thus, other modulation methods need not be explored with the aim of improving the voltage gain.

B. Application of the Venturini Modulation Method to the Matrix Converter

A block diagram of the MC indicating the input and output currents and voltages is shown in Fig. 12. The line-neutral input voltages of the MC are given by $v_{i1,LN}$, $v_{i2,LN}$ and $v_{i3,LN}$, while the line-neutral output voltages are denoted by $v_{o1,LN}$, $v_{o2,LN}$ and $v_{o3,LN}$. The input and output line currents of the MC are given by i_{i1}, i_{i2}, i_{i3} and i_{o1}, i_{o2}, i_{o3} respectively. While the amplitudes of the input and output voltages are denoted by V_i

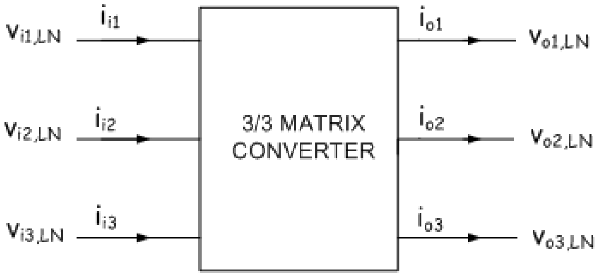


Fig. 12. Block diagram of MC showing voltages and currents

and V_o respectively, the phase angles are given by Φ_i and Φ_o respectively. The input and output side frequencies of the converter are denoted by ω_i and ω_o.

The MC can be modeled as a mathematical transfer function, the modulation matrix **H** which relates the input and output voltages.

$$\begin{bmatrix} v_{o1,LN} \\ v_{o2,LN} \\ v_{o3,LN} \end{bmatrix} = \mathbf{H} \times \begin{bmatrix} v_{i1,LN} \\ v_{i2,LN} \\ v_{i3,LN} \end{bmatrix} \tag{1}$$

where

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \tag{2}$$

The modulation matrix element H_{xy} is the transfer function between the x^{th} output phase and the y^{th} input phase. Since there are no energy storage elements in the MC and assuming that there is no power loss in the MC, the instantaneous three-phase powers are equal on the input and output sides. Consequently, the relationship between the input and output currents of the MC is given by (3).

$$\begin{bmatrix} i_{i1} \\ i_{i2} \\ i_{i3} \end{bmatrix} = \mathbf{H}^T \times \begin{bmatrix} i_{o1} \\ i_{o2} \\ i_{o3} \end{bmatrix} \tag{3}$$

where $\mathbf{H}^T$ is the transpose of the modulation matrix **H**.

According to the Venturini modulation method, the proposed generalized modulation matrix for the MC [14] is given by (4-6), the detailed mathematical treatment of which is presented in [18]. Depending on our requirements of output frequency, phase angle and amplitude in relation with the input voltage parameters, the corresponding **H** matrix is derived.

$$\begin{aligned} \mathbf{H} \ = \ & \frac{1}{3}\alpha_1 \begin{bmatrix} 1+2q\,CS(0) & 1+2q\,CS(-2\pi/3) & 1+2q\,CS(2\pi/3) \\ 1+2q\,CS(2\pi/3) & 1+2q\,CS(0) & 1+2q\,CS(-2\pi/3) \\ 1+2q\,CS(-2\pi/3) & 1+2q\,CS(2\pi/3) & 1+2q\,CS(0) \end{bmatrix} + \\ & \frac{1}{3}\alpha_2 \begin{bmatrix} 1+2q\,CA(0) & 1+2q\,CA(-2\pi/3) & 1+2q\,CA(2\pi/3) \\ 1+2q\,CA(-2\pi/3) & 1+2q\,CA(2\pi/3) & 1+2q\,CA(0) \\ 1+2q\,CA(2\pi/3) & 1+2q\,CA(0) & 1+2q\,CA(-2\pi/3) \end{bmatrix} \end{aligned} \tag{4}$$

where

$$\begin{aligned} & CS(x) = \cos(\omega_m t + x) \\ & CA(x) = \cos[(-\omega_m + 2\omega_i)t + x] \\ & \omega_m = \omega_o - \omega_i \\ & \alpha_1 = \frac{1}{2}[1 + \tan(\phi_i)\cdot\cot(\phi_o)] \\ & \alpha_2 = 1 - \alpha_1 \\ & q = \frac{V_o}{V_i} \end{aligned} \tag{5}$$

The above equations are subjected to the constraints given by (6) to ensure that the matrix **H** is finite, positive, semi-definite and therefore, invertible.

$$\begin{aligned} &\alpha_1 \geq 0 \\ &\alpha_2 \geq 0 \\ &0 \leq q \leq \frac{1}{2} \end{aligned} \tag{6}$$

C. Derivation of the Current Phase Reversal Property

Consider the following input voltages for the MC of Fig. 12 such that the input frequency ω_i is set to ω and the input phase angle Φ_i is set to 0.

$$\begin{aligned} v_{i1,LN} &= V_i \cos(\omega t) \\ v_{i2,LN} &= V_i \cos(\omega t - 2\pi/3) \\ v_{i3,LN} &= V_i \cos(\omega t + 2\pi/3) \end{aligned} \tag{7}$$

Further, let the desired output voltages be in phase with the input voltages and have the same frequency, such that

$$\begin{aligned} \omega_o &= \omega_i = \omega \\ \phi_o &= \phi_i = 0^\circ \end{aligned} \tag{8}$$

Then, from (5), we have

$$\begin{aligned} CS(x) &= \cos(x) \\ CA(x) &= \cos(2\omega t + x) \\ \omega_m &= 0 \\ \alpha_1 &= 0 \\ \alpha_2 &= 1 \end{aligned} \tag{9}$$

Substituting (9) into (4), we obtain the following modulation matrix

$$\mathbf{H} = \begin{bmatrix} H_1 & H_2 & H_3 \\ H_2 & H_3 & H_1 \\ H_3 & H_1 & H_2 \end{bmatrix} = \mathbf{H}^T \tag{10}$$

where

$$\begin{aligned} H_1 &= \frac{1}{3}\left(1 + 2q\cos(2\omega t)\right) \\ H_2 &= \frac{1}{3}\left(1 + 2q\cos(2\omega t - 2\pi/3)\right) \\ H_3 &= \frac{1}{3}\left(1 + 2q\cos(2\omega t + 2\pi/3)\right) \end{aligned} \tag{11}$$

It can be seen that the modulation functions *H_1, H_2* and *H_3* have twice the frequency '*2ω*' as that of the input and output voltages. The term '*q*', known as the modulation index, is the ratio between the output and input voltage amplitudes and is a control parameter of the modulation functions (5). The coefficient '1/3' is used to limit the duty cycle of the MC switches to 2/3, this maximum value being reached when one of the other modulation functions becomes zero. The coefficient '2' is used to ensure that the output phases are never left unconnected even when one of the modulation functions becomes zero. The 'unity' term asserts the same property when the required output voltages are zero and consequently, 'q' is made to be zero.

The modulation matrix of (10) and (11) must result in the MC operation as described in (7) and (8). As verification, (10)-(11) and (7) are substituted into (1). The resulting output voltages are obtained as in (12) and they are found to be in conformity with the desired output voltage parameters.

$$\begin{aligned} v_{o1,LN} &= qV_i\cos(\omega t) \\ v_{o2,LN} &= qV_i\cos(\omega t - 2\pi/3) \\ v_{o3,LN} &= qV_i\cos(\omega t + 2\pi/3) \end{aligned} \tag{12}$$

Let it now be assumed that the output currents have a phase angle *Φ* with respect to the output voltages and are given by

$$\begin{aligned} i_{o1} &= I_o \cos(\omega t + \phi) \\ i_{o2} &= I_o \cos(\omega t + \phi - 2\pi/3) \\ i_{o3} &= I_o \cos(\omega t + \phi + 2\pi/3) \end{aligned} \tag{13}$$

Substituting (10) and (13) into (3), the following input currents are obtained

$$\begin{aligned} i_{i1} &= qI_o \cos(\omega t - \phi) \\ i_{i2} &= qI_o \cos(\omega t - \phi - 2\pi/3) \\ i_{i3} &= qI_o \cos(\omega t - \phi + 2\pi/3) \end{aligned} \tag{14}$$

The output current i_{o1} can be expressed as a phasor as follows

$$\begin{aligned} & i_{o1} = \mathrm{Re}\left\{I_o \times e^{j(\omega t + \phi)}\right\} = \mathrm{Re}\left\{I_o e^{j\omega t} \times e^{j\phi}\right\} \\ \therefore \quad & \widetilde{I}_{o1} = I_o \angle \phi \end{aligned} \tag{15}$$

Similarly, the input current i_{o1} can be expressed as a phasor as

$$\begin{aligned} & i_{i1} = \mathrm{Re}\left\{qI_o \times e^{j(\omega t - \phi)}\right\} = \mathrm{Re}\left\{qI_o e^{j\omega t} \times e^{-j\phi}\right\} \\ \therefore \quad & \widetilde{I}_{i1} = qI_o \angle -\phi \end{aligned} \tag{16}$$

Comparing (15) and (16), it is clearly seen that there is a reversal in the sign of the phase angles of the output and input currents – the *current phase reversal* property. Similarly, this property can be observed in all the phase currents by expressing them in the phasor notation. Comparing the voltage and current expressions, it can be observed that while the output currents lead the corresponding output voltages by *Φ*, the input currents lag the corresponding input voltages by *Φ*. This implies that the current phase reversal property evaluates to a power factor reversal between the input and output terminals as both the input and output voltages have phase angles of 0 degrees. Thus, the modulation matrix **H** which operates the MC with the special current phase reversal property has been established.

D. Switch Realization of the Modulation Functions

The mathematical model of the MC presented in the previous section will now be implemented in the switch matrix of Fig. 11. Switching functions which define the on/off states of the switches as a continuous function of time are to be derived from the modulation functions. The relationships between the input and output voltages and currents of the MC can be expressed in terms of these switching functions as

$$\begin{bmatrix} v_{o1,LN} \\ v_{o2,LN} \\ v_{o3,LN} \end{bmatrix} = \mathbf{S} \times \begin{bmatrix} v_{i1,LN} \\ v_{i2,LN} \\ v_{i3,LN} \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} i_{i1} \\ i_{i2} \\ i_{i3} \end{bmatrix} = \mathbf{S}^T \times \begin{bmatrix} i_{o1} \\ i_{o2} \\ i_{o3} \end{bmatrix} \tag{18}$$

with

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \tag{19}$$

where the matrix element S_{xy} represents the switching function governing the switch connecting the x^{th} output phase and the y^{th} input phase.

Comparing (2) and (19), the modulation function associated with each switching function is easily inferred. As there are three basic functions H_1, H_2 and H_3 in the chosen

Venturini modulation matrix (10)-(11), the **S** matrix is constituted by three corresponding switching functions

$$\mathbf{S} = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_1 \\ S_3 & S_1 & S_2 \end{bmatrix} \tag{20}$$

It is assumed that the switches are operated at a frequency much higher than the input and output frequencies such that their average behavior, given by their duty cycles over each switching period, equals the modulation function values at each instant of time. Thus, while the **H** matrix is a 'low-frequency' description of the MC, the **S** matrix additionally includes a range of switching harmonics. To obtain the on-off times from the duty cycle values (modulation functions), a suitable switching period T_{disc} is obtained by dividing the time period of the modulation functions into '*N*' time intervals given by

$$T_{disc} = \frac{2\pi/2\omega}{N} = \frac{\pi}{N\omega} \tag{21}$$

Discrete-time versions of the modulation functions (switch-averaged functions), developed by applying a zero-order hold of time period T_{disc} as shown in Fig. 13, determine the switch duty ratios over each of the '*N*' intervals. The '*ON*' times of the three switching functions are then derived as

$$\begin{aligned} t_1[k] &= \frac{T_{disc}}{3} \times \left(1 + 2q\cos\left(2\omega T_{disc}k\right)\right) \\ t_2[k] &= \frac{T_{disc}}{3} \times \left(1 + 2q\cos\left(2\omega T_{disc}k - 2\pi/3\right)\right) \\ t_3[k] &= \frac{T_{disc}}{3} \times \left(1 + 2q\cos\left(2\omega T_{disc}k + 2\pi/3\right)\right) \end{aligned} \tag{22}$$

where $0 \le k \le N - 1$

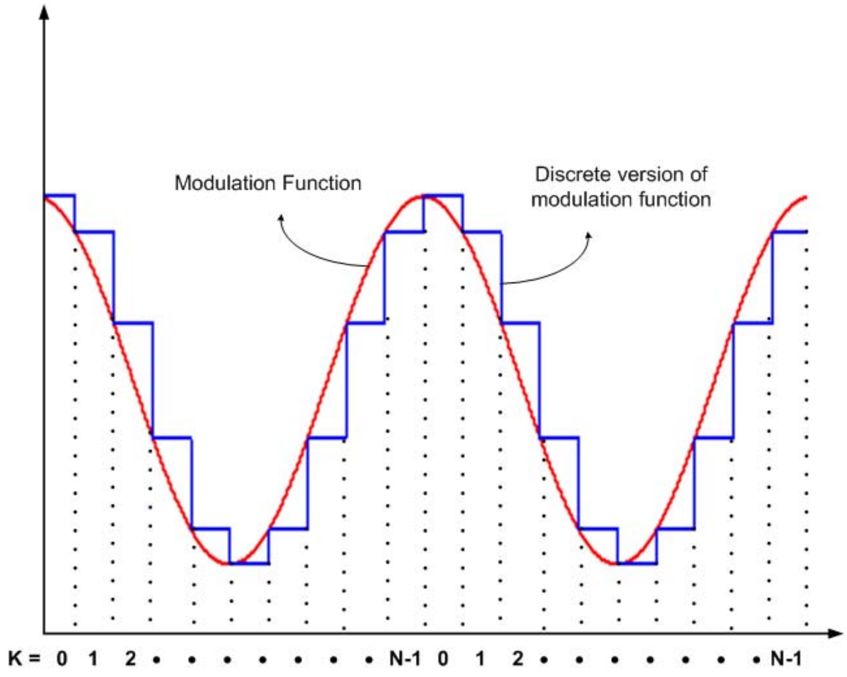


Fig. 13. Modulation function and its discrete version

Within each k^{th} interval, the switches governed by S_1 are turned on for time t_1, followed by the switches governed by S_2 and S_3 which are turned on for times t_2 and t_3, respectively, as shown in Fig. 14. Based on this pattern, the on-off states of the switches can be obtained as a function of time, thus, completely defining the switching functions.

The effective switching frequency is

$$f_{sw} = \frac{1}{T_{disc}} \tag{23}$$

Fig. 13 implies that as the value of N is increased, the switching functions more accurately model the modulation functions.

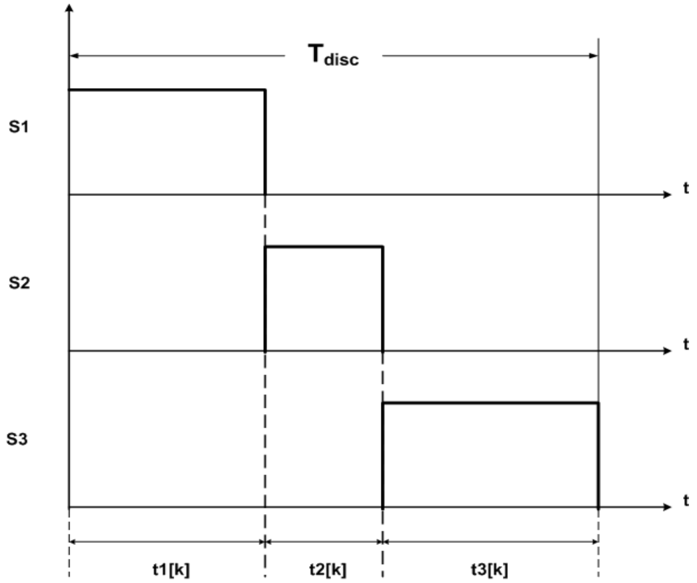


Fig. 14. Switching function for a discrete interval *k*, 0≤ *k* ≤ N-1

As mentioned before, the MC requires bi-directional switches; but, semiconductor switches with inherent bi-directional capability are not yet commercially available. So, bilateral functionality is constructed from one or more unidirectional

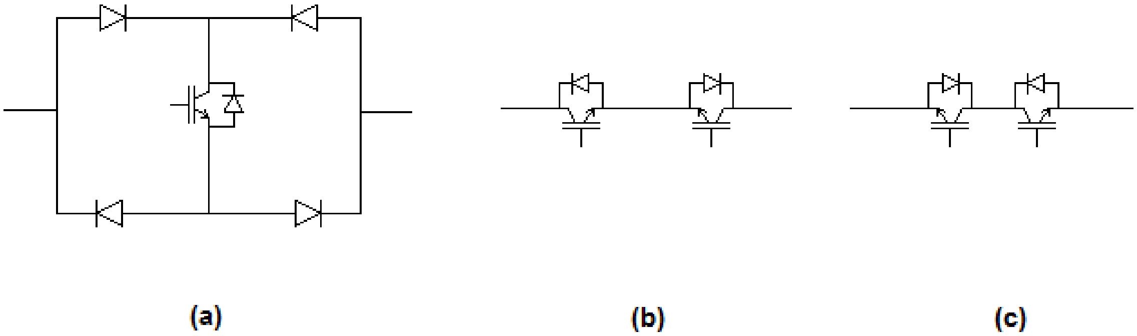


Fig. 15. Different bi-directional switch realizations based on discrete switches

switch elements. Three possible realizations are illustrated in Fig. 15 [17, 25]. The diode bridge type realization (Fig. 15(a)) requires only a single controlled switch but is generally not practical due to the diode losses. The common-emitter (Fig. 15(b)) and common-collector (Fig. 15(c)) configurations are more efficient realizations, but they require two gate-drive circuits and careful commutation techniques to prevent converter failures. The common-collector configuration is the more preferred switch arrangement as it requires lesser number of isolated gate-drive power supplies than the common-emitter configuration. For example, in the case of the MC, while the common-emitter configuration requires 9 isolated power supplies, the common-collector configuration requires only 6 supplies. This reduced number of supplies is also an advantage in view of the isolation distance between independent voltage potentials within the MC.

E. Simulation Results

The system shown in Fig. 16 has been considered for simulation study. The MC is connected to a 3-phase wye-connected source of 480V (rms) line-line voltage and 60

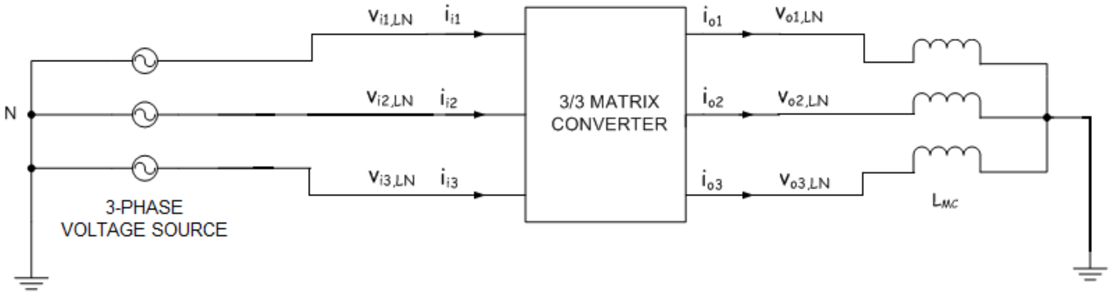


Fig. 16. System simulated to study MC operation

TABLE 1. System parameters used for simulation

Parameter	**Value**
Line-to-line input voltage of the MC $V_{LL,rms}$	480 V
Three-phase load at the MC output	20mH
Angular frequency of power grid voltage ω	2π*60 rad/s
Discrete intervals per period of modulation function *N*	100

Hz frequency. A 3-phase inductive load of 20 mH is considered at the output of the MC. The operating parameters of the MC are summarized in TABLE 1. From (21) and (23), it is seen that the resulting switching frequency is 12 kHz. Line filters must be used between the MC and the source to eliminate the high order harmonics in the input currents. They have been excluded from the initial study to obtain a better understanding of the MC operation. The effect of filters on the system will be considered later in this chapter.

1. Simulation in MATLAB

The '*ON*' times of the switches are obtained using (22). The switching functions, which are vectors of 1's and 0's, are derived by expressing these '*ON*' times in terms of a suitable time step. The simulated modulation and switching functions are shown in Fig. 17. The switch-averaged function, obtained by averaging the switching functions over every switching period T_{disc}, closely follows the corresponding modulation function. This indicates that the switching function models the modulation function as expected.

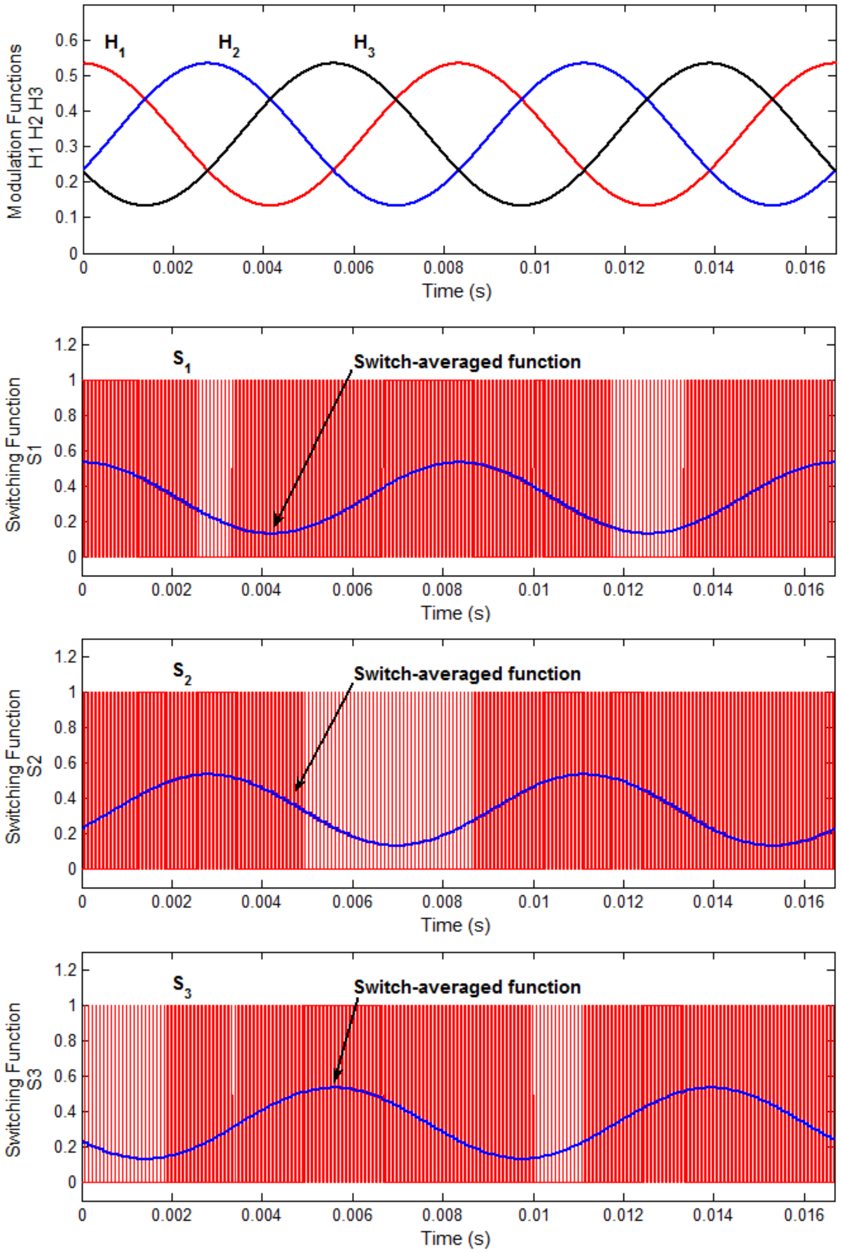


Fig. 17. Modulation functions and switching functions

The switching sequence followed in the Venturini technique is shown in Fig. 18 and is in accordance with Fig. 14

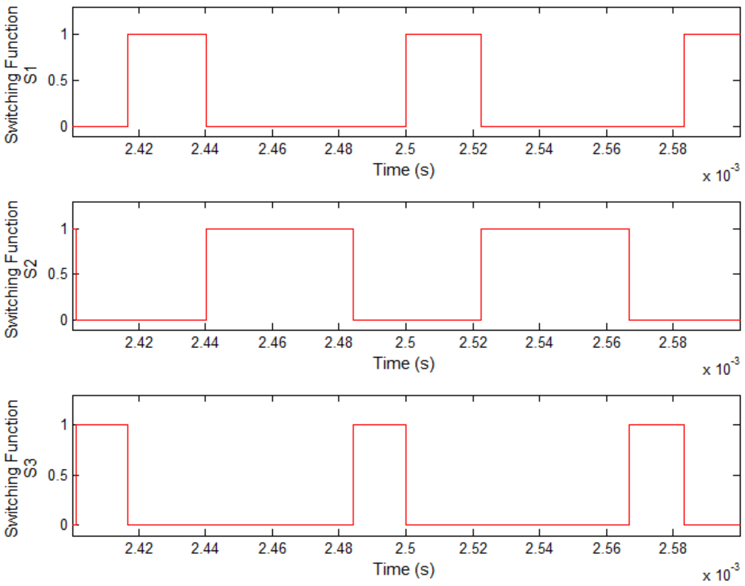


Fig. 18. Switching sequence of the Venturini method

The line-neutral input and output voltages, obtained using (17) are shown in Fig. 19. The switching nature of the output voltages, which sequentially take the values of all three input voltages, is apparent from the figures. It can be seen that the fundamental components of the output voltages, obtained by using the Fast Fourier Transform (FFT), are stepped down from the inputs by a factor of '*q*' (0.3), and are in phase with the corresponding input voltage.

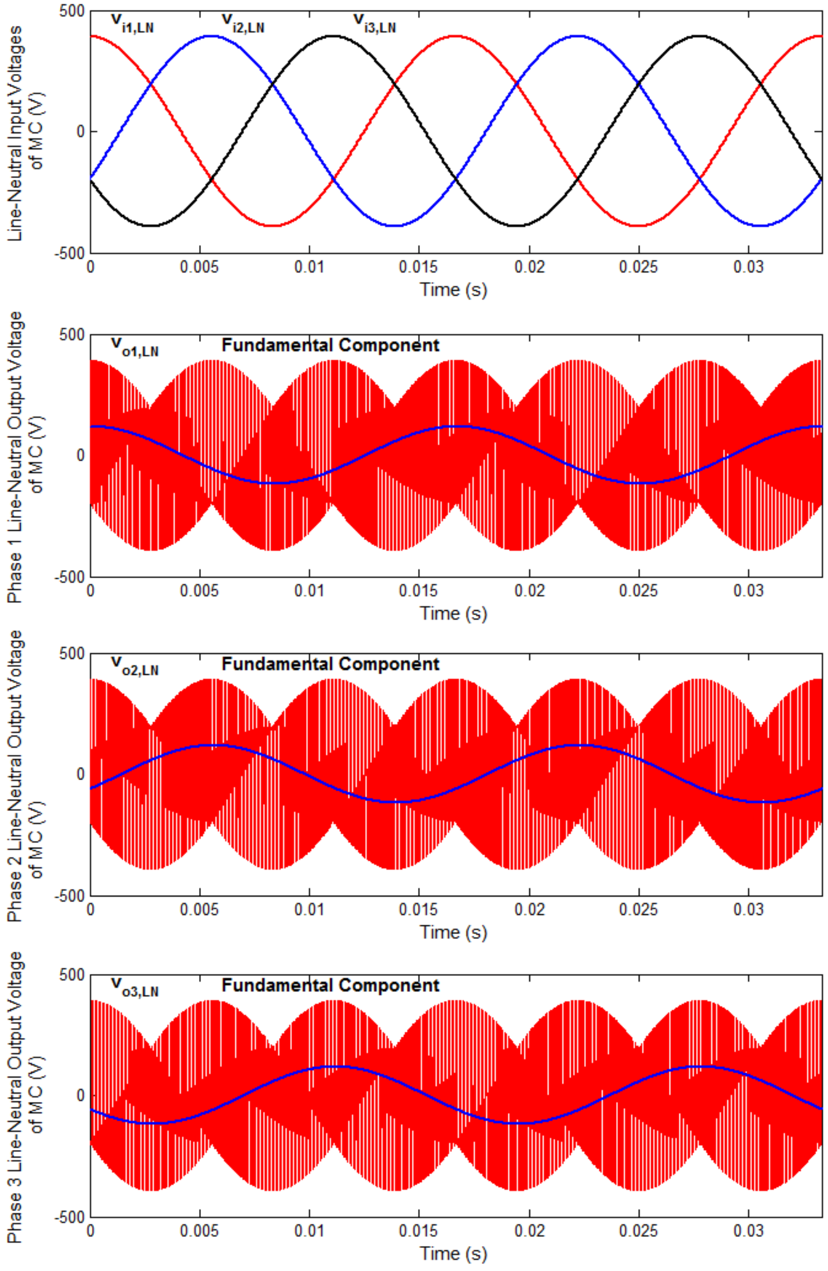


Fig. 19. Line-neutral input voltages and output voltages of the MC

The currents of the MC are shown in Fig. 20. The input currents are seen to be switching between the output currents as indicated by (18). The fundamental components of the input currents are stepped down by a factor of 0.3 when compared to the output currents.

More importantly, each input current has exactly the opposite phase as the corresponding output current, as expected from (13) and (14). Thus, the Venturini modulation method is seen to assert the unique property of phase reversal between the input and output currents.

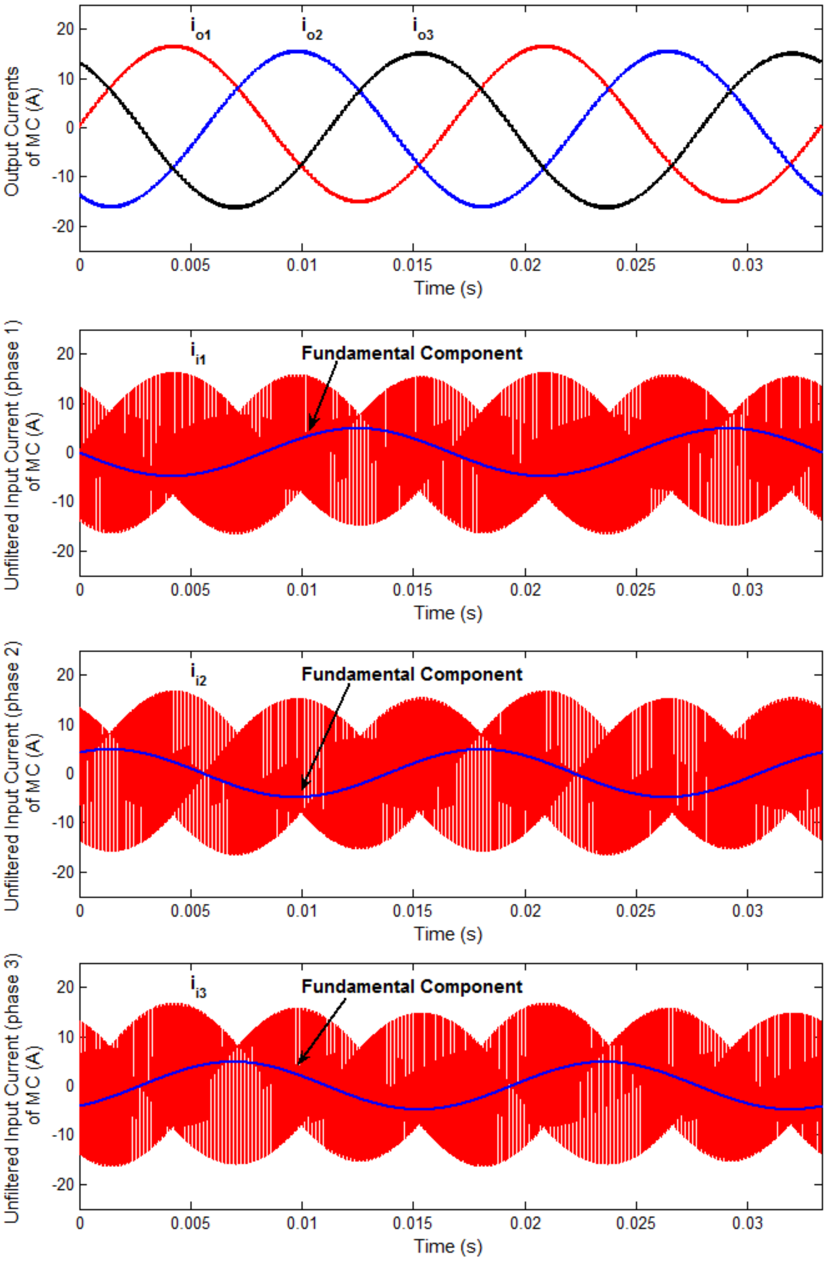


Fig. 20. Output currents and input currents of the MC

The fundamental components of the voltages and currents of one phase of the MC are shown in Fig. 21.

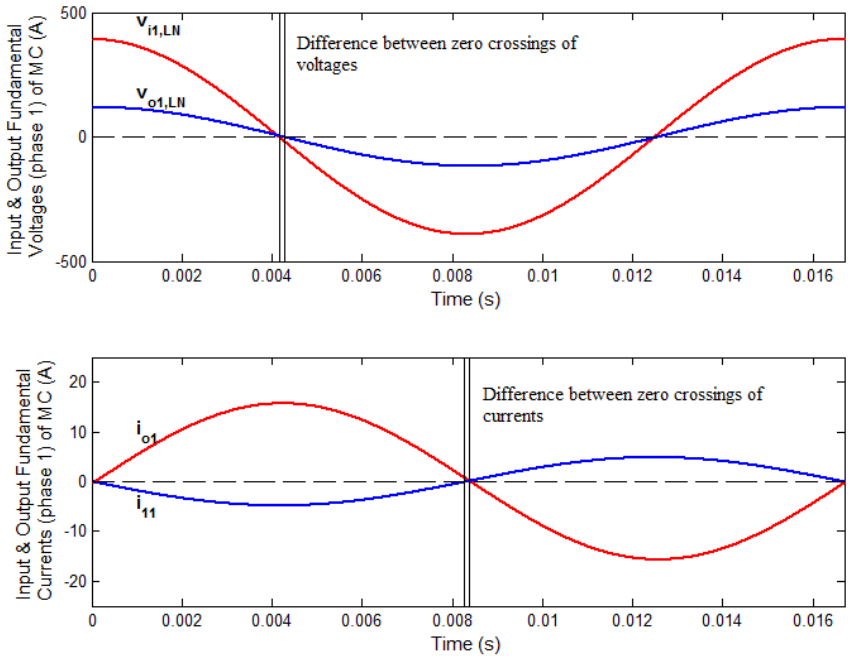


Fig. 21. Accuracy of MC waveforms for switching frequency of 12kHz

The magnitudes and phase angles of the fundamental components of the MC voltages are summarized in TABLE 2. Slight differences between the simulated and expected values can be noted from the table. Also, a closer observation of Fig. 21 shows that the input and output voltages are not exactly in phase, and the input and output currents are not exactly out of phase. This deviation has been found to be attributed to the modeling inaccuracy introduced by the Venturini method while deriving the switching functions from the discrete-time modulation functions (Fig. 13).The difference between the modulation function and its discrete version for a switching frequency of 12 kHz is shown in Fig. 22.

TABLE 2. Comparison of expected and simulated output voltages of the MC

	Magnitude		**Phase**	
	Expected value	Simulated value	Expected value	Simulated value
$v_{o1,LN}$	117.58 V	118.27 V	0°	-1.80°
$v_{o2,LN}$	117.58 V	118.27 V	-120°	-121.80°
$v_{o3,LN}$	117.58 V	118.28 V	120°	118.20°
i_{o1}	15.59 A	15.69 A	-90°	-91.81°
i_{o2}	15.59 A	15.69 A	150°	148.20°
i_{o3}	15.59 A	15.69 A	30°	28.20°
i_{i1}	4.68 A	4.85 A	90°	90.11°
i_{i2}	4.68 A	4.85 A	-30°	-30.11°
i_{i3}	4.68 A	4.84 A	-150°	-150.00°

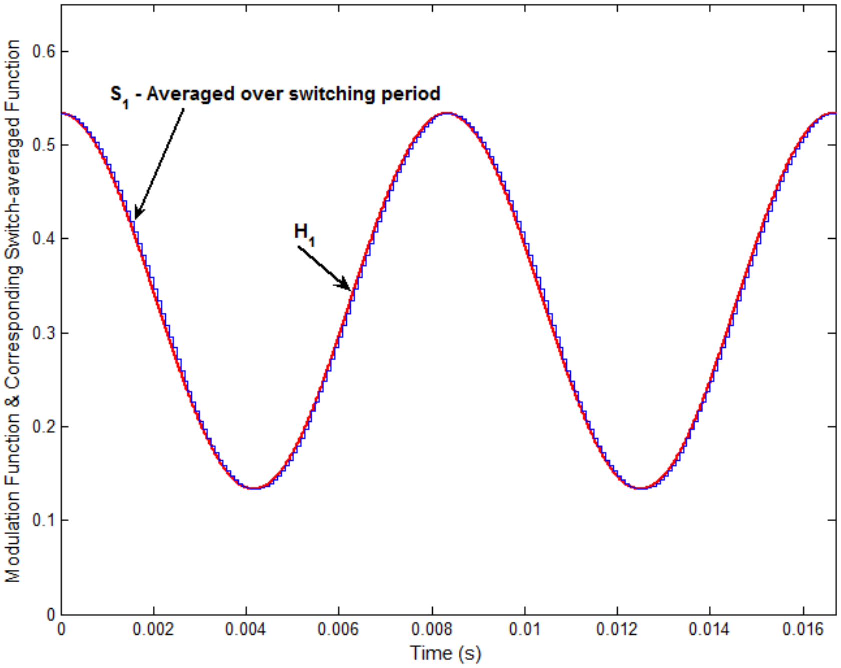


Fig. 22. Comparison between modulation and switch-averaged functions for switching frequency of 12kHz

As mentioned earlier, the discrete versions follow the original modulation functions with greater accuracy as the zero-order hold period T_{disc} is decreased. In other words, as the switching frequency is increased by using larger values of '*N*', the switching functions will more closely model the modulation functions. The modeling inaccuracy is depicted for a switching frequency of 1.2 kHz (N=10) in Fig. 23 and Fig. 24. In Fig. 24, the phase difference between the modulation function H_1 and the geometric mean of the discrete version of H_1 is approximately 0.26 radians. The phase difference in voltages of Fig. 23 is also approximately the same (0.32 radians). Thus, the correlation between the modeling error of the Venturini method and the inaccuracy in the MC waveforms is further ascertained.

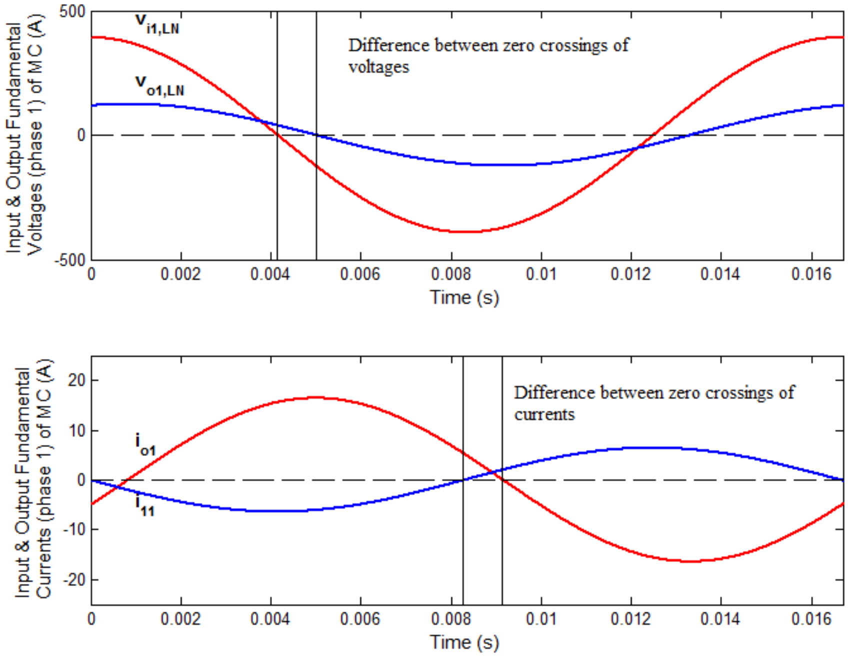


Fig. 23. Accuracy of MC waveforms for switching frequency of 1.2 kHz

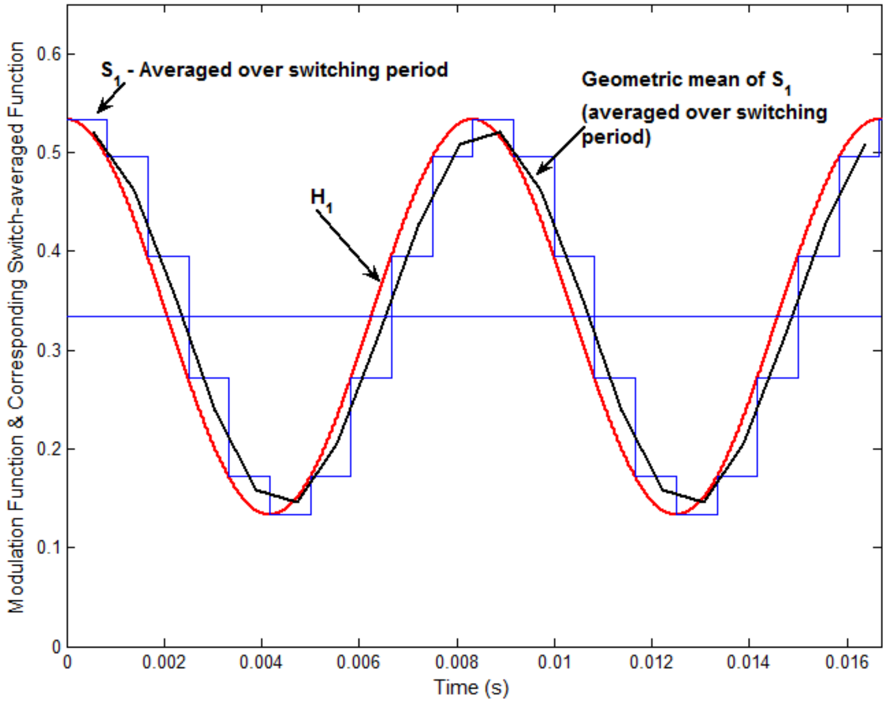


Fig. 24. Comparison between modulation and switching functions for switching frequency of 1.2 kHz

The voltage and current waveforms for a switching frequency of 120 kHz (N=1000), shown in Fig. 25, illustrate near-ideal behavior of the MC. Practically no difference exists between the modulation function and its discrete function as illustrated in Fig. 26.

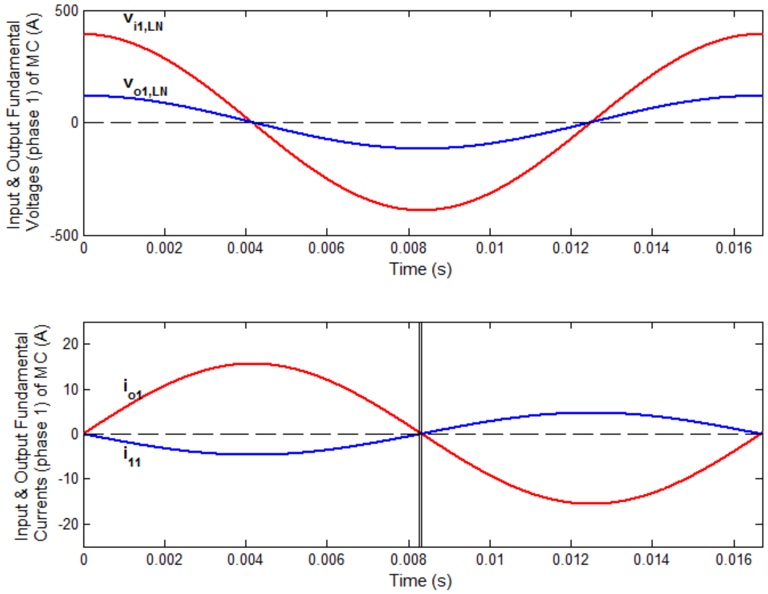


Fig. 25. Accuracy of MC waveforms for switching frequency of 120kHz

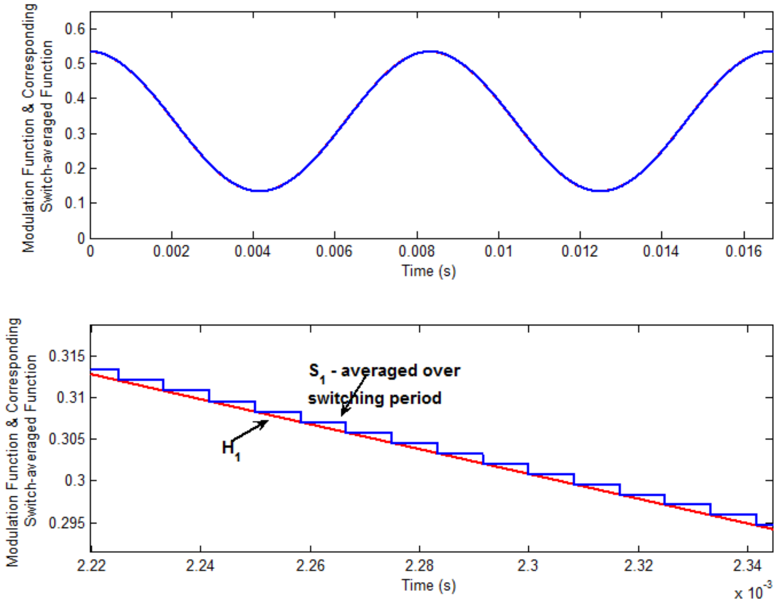


Fig. 26. Comparison between modulation and switch-averaged functions for switching frequency of 120kHz

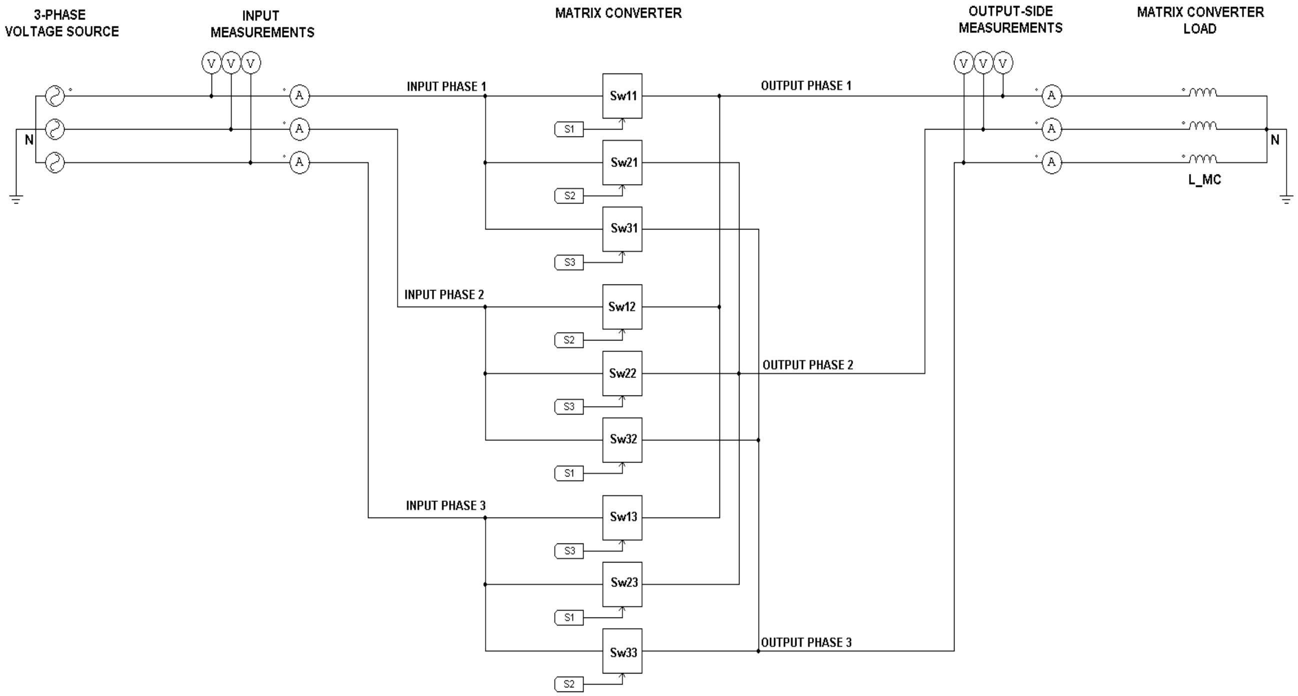


Fig. 27. Schematic of system in PSIM

2. Simulation in PSIM

The system of Fig. 16 was implemented in PSIM as shown in Fig. 27. The control logic for switch signal generation is provided in Appendix B. The line-neutral input and output voltages are shown in Fig. 28.

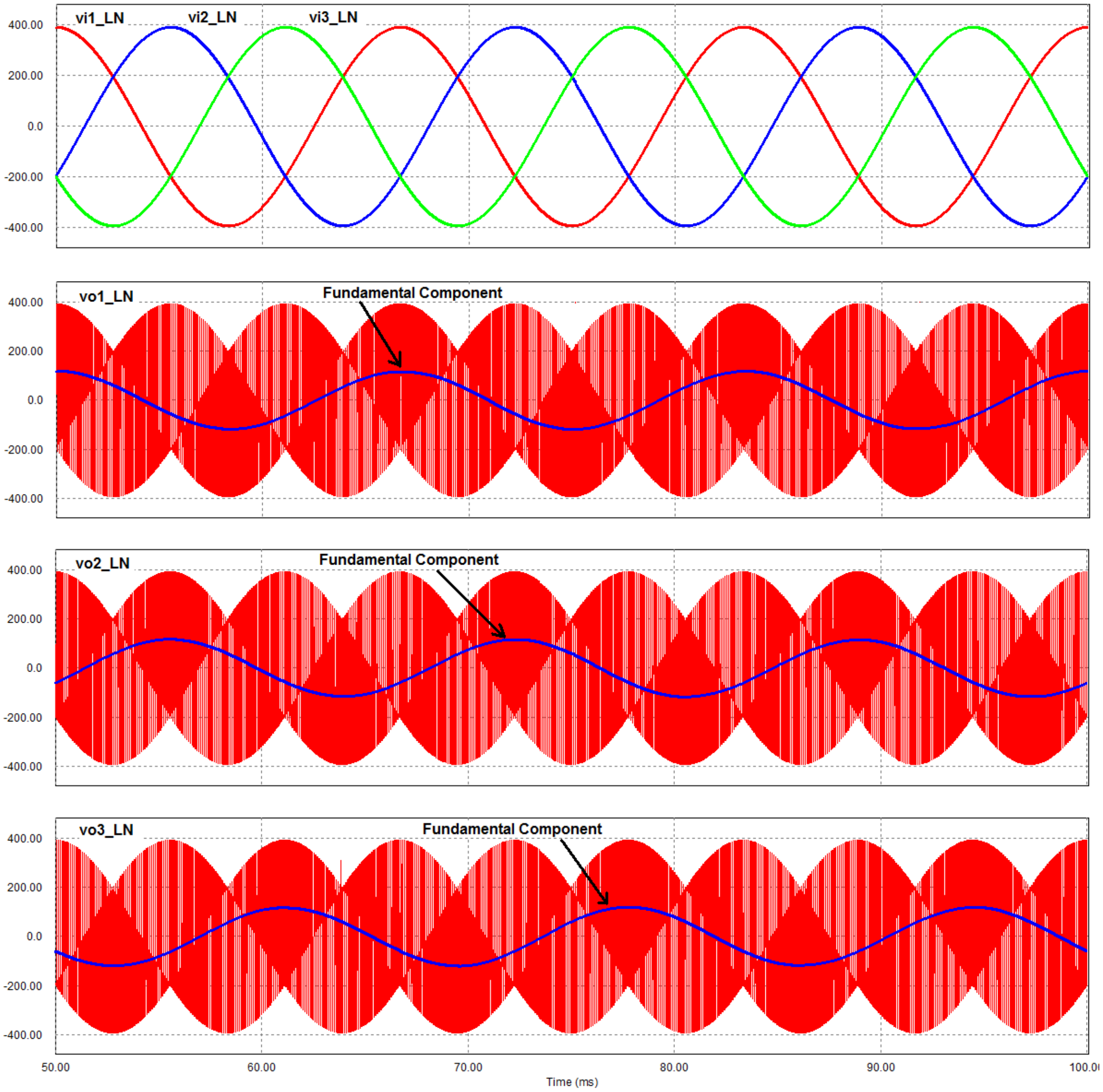


Fig. 28. PSIM simulation - line-neutral input and output voltages of the MC

The input and output currents of the PSIM-simulated system are shown in Fig. 29.

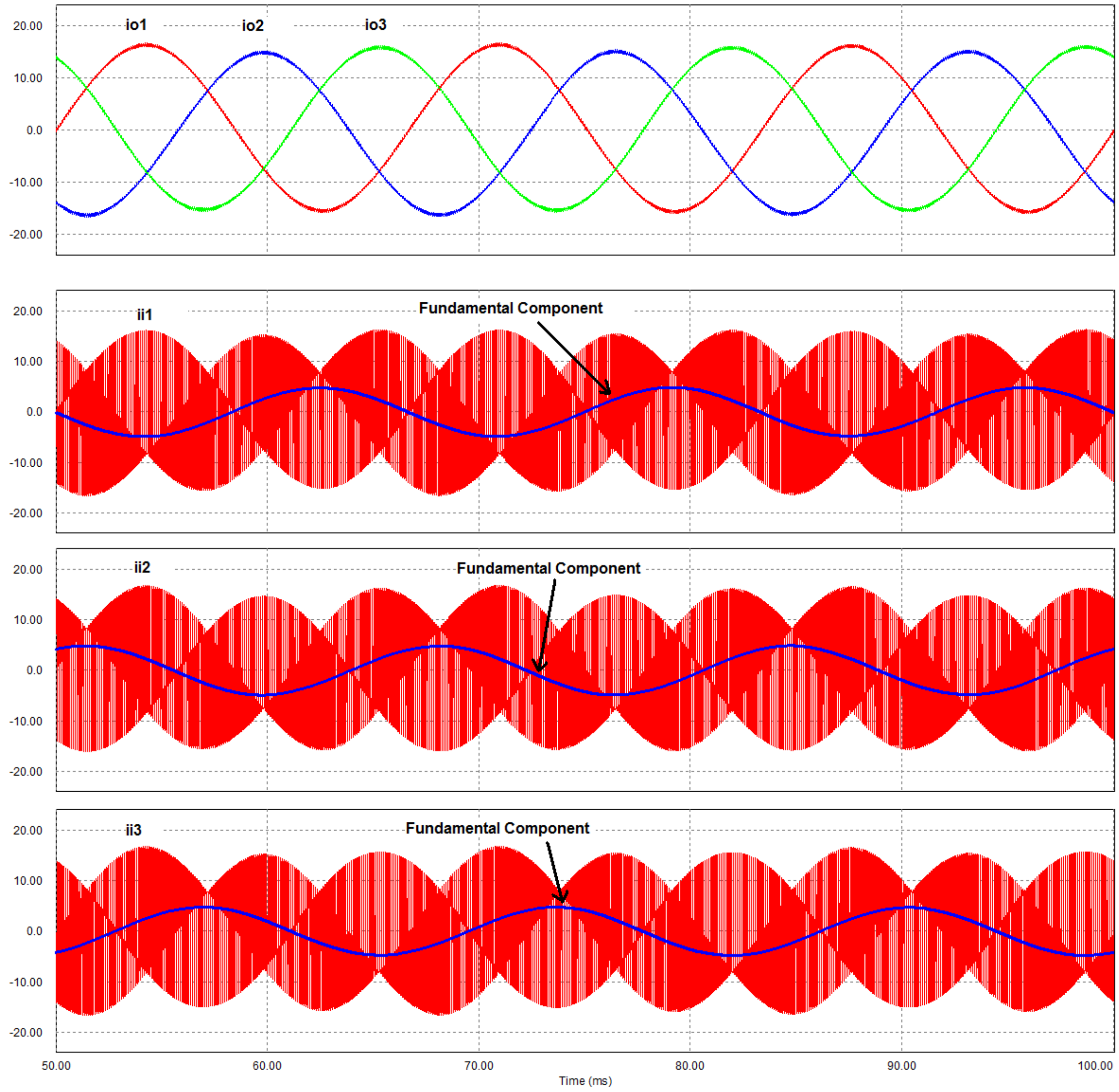


Fig. 29. PSIM simulation - output and input currents of the MC

The fundamental components of the MC voltages and currents are given in TABLE 3. Comparing the values with those of TABLE 3, it can be seen that the PSIM simulations are in close agreement with the MATLAB simulation results (the magnitude values differ by a maximum of ±3.6%, while the phase angle values differ by a maximum of ±2.5%).

TABLE 3. Magnitudes and phase angles of fundamental components of MC waveforms

PARAMETER	**MAGNITUDE**	**PHASE ANGLE**
Line-neutral MC input voltage (phase 1) v_{i1}	391.92 V	-0.02°
Line-neutral MC input voltage (phase 2) v_{i2}	391.92 V	-120.02°
Line-neutral MC input voltage (phase 3) v_{i3}	391.92 V	-119.98°
Line-neutral MC output voltage (phase 1) v_{o1}	114.48 V	-1.76°
Line-neutral MC output voltage (phase 2) v_{o2}	117.31 V	-119.50°
Line-neutral MC output voltage (phase 3) v_{o3}	119.85 V	-118.21°
MC input current (phase 1) i_{i1}	4.93 A	91.12°
MC input current (phase 2) i_{i2}	4.79 A	-31.32°
MC input current (phase 3) i_{i3}	4.68 A	-148.65°
MC output current (phase 1) i_{o1}	15.50 A	-91.20°
MC output current (phase 2) i_{o2}	15.79 A	148.76°
MC output current (phase 3) i_{o3}	15.64 A	27.86°

3. Addition of Input Line Filters

Second-order filters are required between the voltage source and the MC to reduce harmonic content of the currents drawn from the source. Several filter topologies have been discussed in literature [26, 27] some of which are shown in Fig. 30. The filter topology of Fig. 30 (d) has been chosen due to its high damping factor and relatively

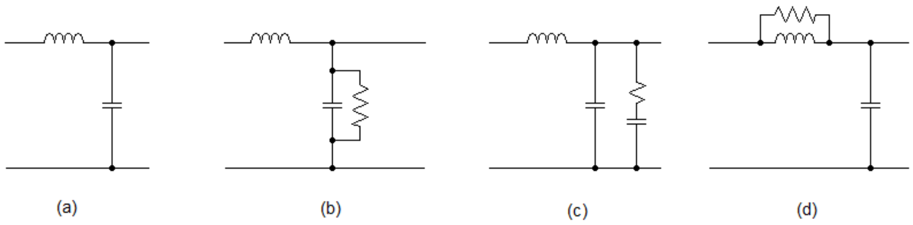


Fig. 30. Some input filter topologies

lower losses. The PSIM schematic of the system with input filters is shown in Fig. 31. Damping resistors connected in parallel with the inductors reduce ringing effects. The cut-off frequency $\omega_{cut\text{-}off}$ of the filter is chosen as 1.2 kHz, which is 1/10th the switching frequency. The filter parameters are chosen according to (24). The damping resistor is chosen to result in a filter damping factor ζ of 0.5. It is then adjusted during simulation to improve the system performance. The resulting filter parameters are given in TABLE 4.

$$\begin{aligned} \omega_{cut-off} &= 1\Big/\sqrt{L_{filter}C_{filter}} \\ \zeta &= \frac{1}{2R_{damp}}\sqrt{\frac{L_{filter}}{C_{filter}}} \end{aligned} \tag{24}$$

TABLE 4. Input line filter parameters

PARAMETER	VALUE
Per-phase inductance L_{filter}	1 mH
Per-phase capacitance C_{filter}	20 μH
Per-phase damping resistor R_{damp}	10 Ω

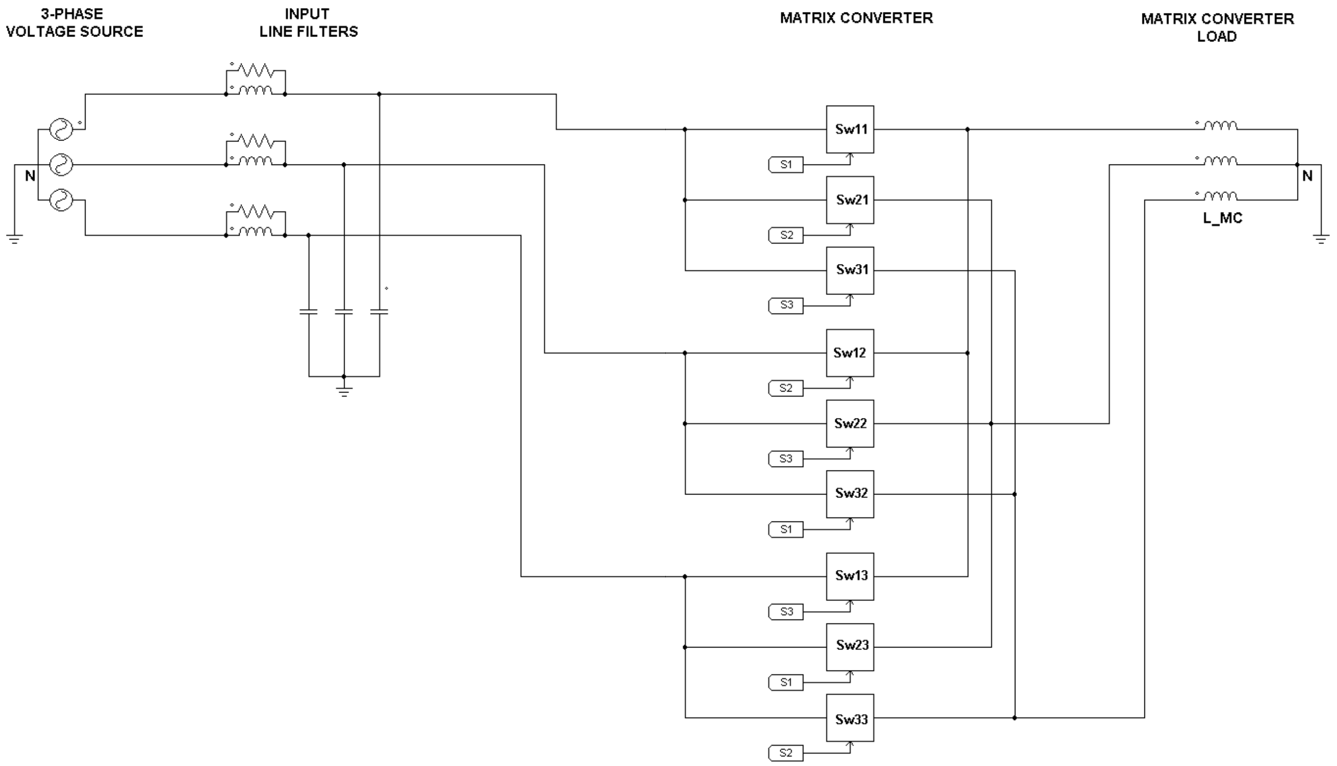


Fig. 31. Schematic of MC system with input line filters in PSIM

The line-neutral voltages of the system are given in Fig. 32. The upper row shows the source voltages, while the second row displays the MC input voltages (after the filter stage). The last row shows the resulting MC output voltage of phase 1. It is seen that the filter contributes to some ringing in the MC input voltages. As the damping resistor is increased, the ringing effect decreases; however, it deteriorates the filtering action of the input filter.

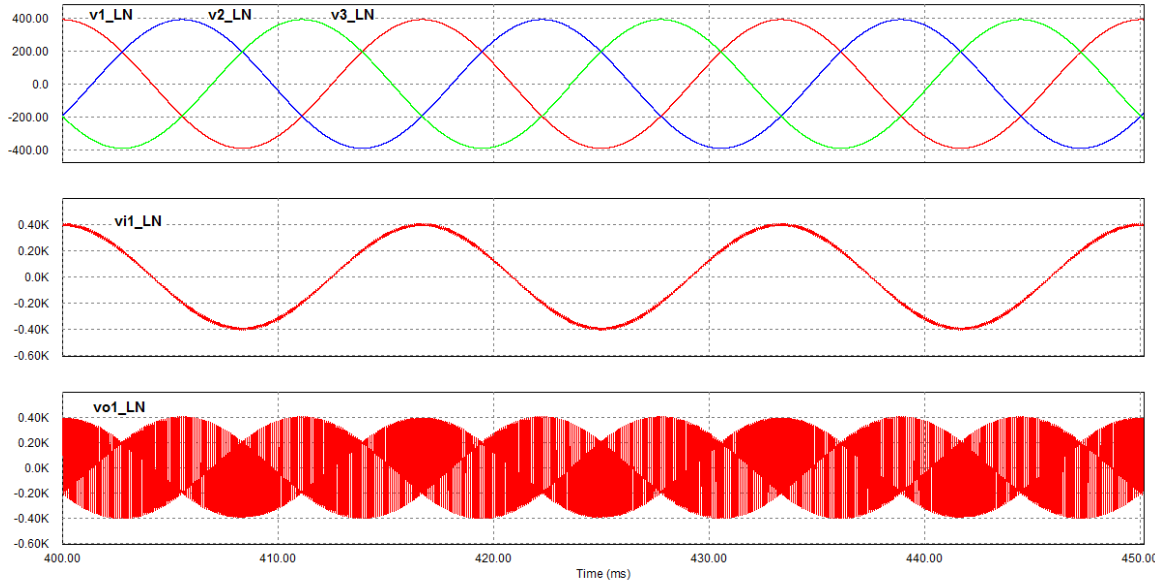


Fig. 32. PSIM simulation with input filters - source, MC input and MC output voltages (phase 1)

The currents of the system are shown in Fig. 33. The top-most row shows the three output currents of the MC. The MC input current of phase 1 is shown in the second row, while the filtered current drawn from the voltage source is given in the last row. The figure shows that appropriate filtering has taken place.

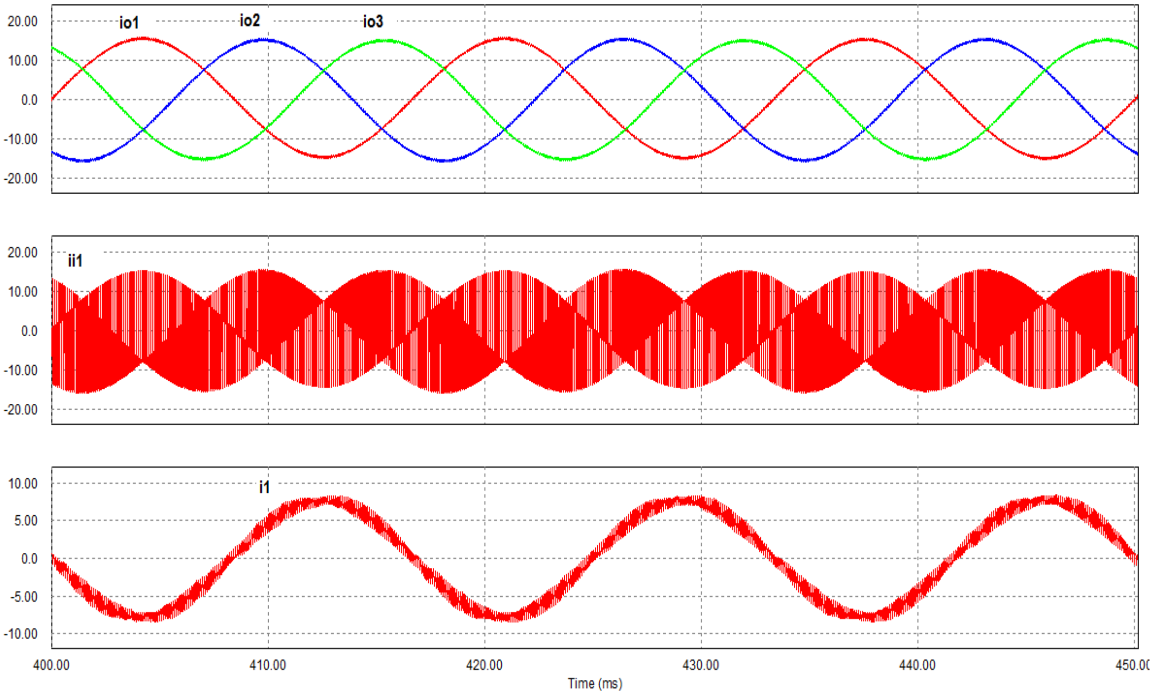


Fig. 33. PSIM simulation with input filters – MC output, MC input and source currents (phase 1)

The magnitudes and phase angles of the fundamental components of the MC waveforms are summarized in TABLE 5. It is seen that the filters have not altered the basic operation of the MC – the current phase reversal property. A comparison with TABLE 3 shows the unfiltered and filtered cases to be in close agreement (magnitudes and angles differ by a maximum of ±3.5% and ±1%, respectively).

TABLE 5. Magnitudes and phase angles of fundamental components of waveforms for the MC system with input filters

PARAMETER	MAGNITUDE	PHASE ANGLE
Line-neutral MC input voltage (phase 1) v_{i1}	394.83 V	-0.04°
Line-neutral MC input voltage (phase 2) v_{i2}	394.78 V	-120.04°
Line-neutral MC input voltage (phase 3) v_{i3}	394.80 V	-119.96°
Line-neutral MC output voltage (phase 1) v_{o1}	110.81 V	-1.36°
Line-neutral MC output voltage (phase 2) v_{o2}	113.79 V	-119.06°
Line-neutral MC output voltage (phase 3) v_{o3}	116.22 V	-118.53°
MC input current (phase 1) i_{i1}	4.77 A	90.88°
MC input current (phase 2) i_{i2}	4.65 A	-31.54°
MC input current (phase 3) i_{i3}	4.54 A	-148.96°
MC output current (phase 1) i_{o1}	15.02 A	-90.97°
MC output current (phase 2) i_{o2}	15.33 A	149.09°
MC output current (phase 3) i_{o3}	15.18 A	28.03°

In conclusion, it is seen that the simulation results support the mathematical analysis of the MC using the Venturini modulation method. The current phase reversal property has been presented and is applied to the problem of VAR compensation in the following chapter.

CHAPTER III

THE PROPOSED VAR COMPENSATOR

The Venturini modulation method for a MC was introduced in Chapter II and modulation functions H_1, H_2 and H_3 were developed as in (10) and (11) which operate the converter such that the phase of currents are reversed between the input and the output. Simulation results of the MC system shown in Fig. 16 further illustrate this property by showing that while the output currents lag the output voltages due to the inductive element, the input currents actually lead the corresponding input voltages. Thus, the inductor on the MC output is made to appear as a capacitor at the MC input by the Venturini modulation method.

This current phase reversal property of the MC finds important application in reactive power load compensation, where capacitors are used extensively to locally provide lagging VARs to loads, despite their serious reliability issues. This thesis proposes a reactive power load compensator consisting of the MC with a 3-phase choke (inductive) element connected at its output terminals. By controlling the modulation index 'q', the MC can be operated to draw variable leading VARs from the source, while the choke draws lagging VARs from the MC output terminals.

A. System Description and Analysis

The layout of a test case ac power system with a lagging load that needs compensation is shown in Fig. 34. The proposed VAR compensator is shunt-connected to the system at a point near the load. Though the source is typically Δ-connected, its wye equivalent has been used to facilitate understanding the phase relationship between phase currents and voltages. The line-neutral ac network voltages are denoted by $v_{1,LN}$, $v_{2,LN}$ and $v_{3,LN}$ and are also equal to the MC input voltages $v_{i1,LN}$, $v_{i2,LN}$ and $v_{i3,LN}$, respectively. The magnitude and frequency of the voltages are denoted by $V_{LN,rms}$ and ω respectively. The line-neutral MC output voltages are given by $v_{o1,LN}$, $v_{o2,LN}$ and $v_{o3,LN}$.

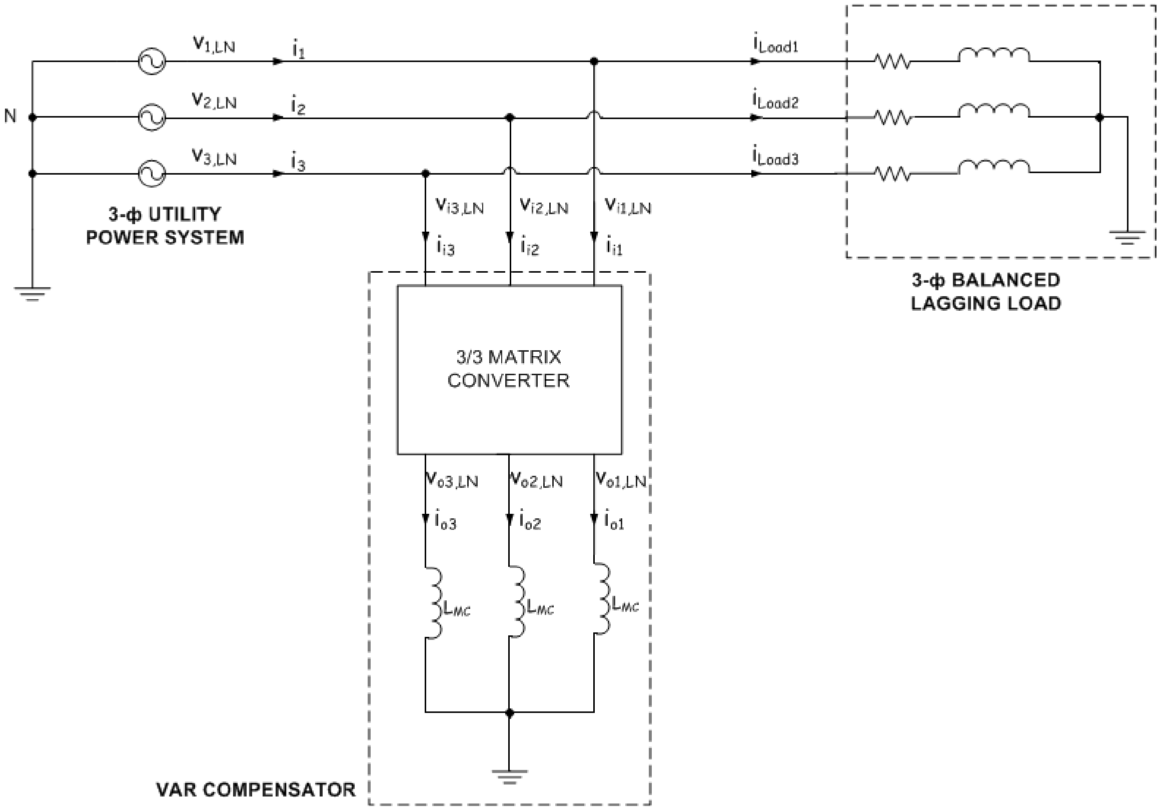


Fig. 34. The proposed VAR compensator

The 3-phase choke element, denoted by L_{MC}, draws currents i_{o1}, i_{o2} and i_{o3} from the output terminals of the MC. The currents drawn by the MC from the system are denoted by i_{i1}, i_{i2} and i_{i3}, while the currents drawn by the load are given by i_{Load1}, i_{Load2} and i_{Load3}. The total currents drawn by the effective load, the original lagging load in parallel with the compensator, are given by i_1, i_2 and i_3.

Let the ac network and MC input voltages be given by (25). Let it be assumed that in order to provide complete load compensation, the MC is operated with a modulation index 'q_{comp}'. Then, from (12), we obtain the MC output voltages as in (26).

$$\begin{aligned} v_{1,LN} &= v_{i1,LN} = \sqrt{2}\times V_{LN,rms}\cos(\omega t) \\ v_{2,LN} &= v_{i2,LN} = \sqrt{2}\times V_{LN,rms}\cos(\omega t - 2\pi/3) \\ v_{3,LN} &= v_{i3,LN} = \sqrt{2}\times V_{LN,rms}\cos(\omega t + 2\pi/3) \end{aligned} \tag{25}$$

$$\Rightarrow \quad \begin{aligned} v_{o1,LN} &= \sqrt{2}\times q_{comp}\times V_{LN,rms}\cos(\omega t) \\ v_{o2,LN} &= \sqrt{2}\times q_{comp}\times V_{LN,rms}\cos(\omega t - 2\pi/3) \\ v_{o3,LN} &= \sqrt{2}\times q_{comp}\times V_{LN,rms}\cos(\omega t + 2\pi/3) \end{aligned} \tag{26}$$

The choke element draws currents that lag the MC output voltages by exactly 90° as given by (27). By virtue of the current reversal property of the Venturini modulation method (13)-(14), the input currents of the MC are then derived as in (28).

$$\begin{aligned} i_{o1} &= \sqrt{2}\times q_{comp}\times\frac{V_{LN,rms}}{\omega L_{MC}}\times\cos(\omega t - \pi/2) \\ i_{o2} &= \sqrt{2}\times q_{comp}\times\frac{V_{LN,rms}}{\omega L_{MC}}\times\cos(\omega t - \pi/2 - 2\pi/3) \\ i_{o3} &= \sqrt{2}\times q_{comp}\times\frac{V_{LN,rms}}{\omega L_{MC}}\times\cos(\omega t - \pi/2 + 2\pi/3) \end{aligned} \tag{27}$$

$$\begin{aligned}
i_{i1} &= \sqrt{2} \times q_{comp}{}^{2} \times \frac{V_{LN,rms}}{\omega L_{MC}} \times \cos(\omega t + \pi/2) \\
i_{i2} &= \sqrt{2} \times q_{comp}{}^{2} \times \frac{V_{LN,rms}}{\omega L_{MC}} \times \cos(\omega t + \pi/2 - 2\pi/3) \\
i_{i3} &= \sqrt{2} \times q_{comp}{}^{2} \times \frac{V_{LN,rms}}{\omega L_{MC}} \times \cos(\omega t + \pi/2 + 2\pi/3)
\end{aligned} \tag{28}$$

The instantaneous input power per-phase is obtained as a product of the input voltages and currents of the compensator

$$\begin{aligned}
p(t)_{per\ phase} &= \left(\sqrt{2} \times V_{LN,rms} \cos(\omega t)\right) \times \left(\sqrt{2} \times q_{comp}{}^{2} \times \frac{V_{LN,rms}}{\omega L_{MC}} \times \cos(\omega t + \pi/2) \right) \\
&= 2 \times \frac{q_{comp}{}^{2} V_{LN,rms}{}^{2}}{\omega L_{MC}} \times \cos(\omega t) \times \cos(\omega t + \pi/2)
\end{aligned} \tag{29}$$

Using standard trigonometric identities, we have

$$\begin{aligned}
p(t)_{per\ phase} &= \frac{q_{comp}{}^{2} V_{LN,rms}{}^{2}}{\omega L_{MC}} \times \left[\cos(2\omega t + \pi/2) + \cos(-\pi/2)\right] \\
&= \frac{q_{comp}{}^{2} V_{LN,rms}{}^{2}}{\omega L_{MC}} \times (-) \sin(2\omega t)
\end{aligned} \tag{30}$$

It is seen that the instantaneous power associated with the MC input has only a negative double frequency component. This indicates that while the MC does not consume any real power, it supplies reactive power to the ac network. The total three-phase reactive power supplied by the MC to the ac network is then given by

$$Q = 3 \times \frac{\left(q_{comp}\right)^{2} \times \left(V_{LN,rms}\right)^{2}}{\omega L_{MC}} \tag{31}$$

So, in order to compensate a load requiring ‘Q’ lagging VARs, the MC must be operated at a modulation index ‘q_{comp}’ which satisfies (31). From this equation, it is seen that the reactive power compensated by the MC increases with the modulation index. Further, lesser the choke inductance L_{MC}, greater is the compensation. The choke value is decided by considering the maximum required compensation at the modulation index limiting value which is 0.5.

B. Simulation Results

The system shown in Fig. 34 was simulated in MATLAB and the parameters used are listed in TABLE 6. The line-to-line voltage (phase voltage) of the Δ-configured ac power grid is chosen as 480 V which results in the line-to-neutral voltage of 277.13 V for the wye-model in Fig. 34. The maximum required reactive power compensation is given by

$$Q_{Load} = P_{Load} \times \tan\left(\cos^{-1}\left(p.f.\right)\right) \tag{32}$$

TABLE 6. System parameters used for simulation

PARAMETER	**VALUE**
Line-to-neutral power grid voltage $V_{LN,rms}$	277.13 V
Angular frequency of power grid voltage ω	2π*60 rad/s
Per-phase resistance of load $R_{Load,\ per\ phase}$	14.75 Ω
Per-phase inductance of load $L_{Load,\ per\ phase}$	29.34 mH
Total rated real power of load P_{Load}	10 kW
Power factor of load *p.f.*	0.8
Total maximum load reactive power requirement Q_{load}	7.5 kVAR
VAR rating of compensator	8.0 kVAR
MC output-side inductance L_{MC}	20 mH
Discrete intervals per period of modulation function N	100
Operating modulation index q_{comp}	0.4954

The VAR rating of the compensator is chosen accordingly. The required L_{MC} is calculated using (31). The chosen '*N*' gives a reasonable switching frequency of 12 kHz. Using the chosen MC parameter values, the modulation index is recalculated using (31).

1. Simulations in MATLAB

The system is simulated in MATLAB as mentioned in Chapter II. The input and output voltages of phase 1 of the MC are given in Fig. 35. The input and output currents of the MC phase 1 are shown in Fig. 36. The phase relationships between the fundamental components of the input and output voltages and currents are illustrated in Fig. 37.

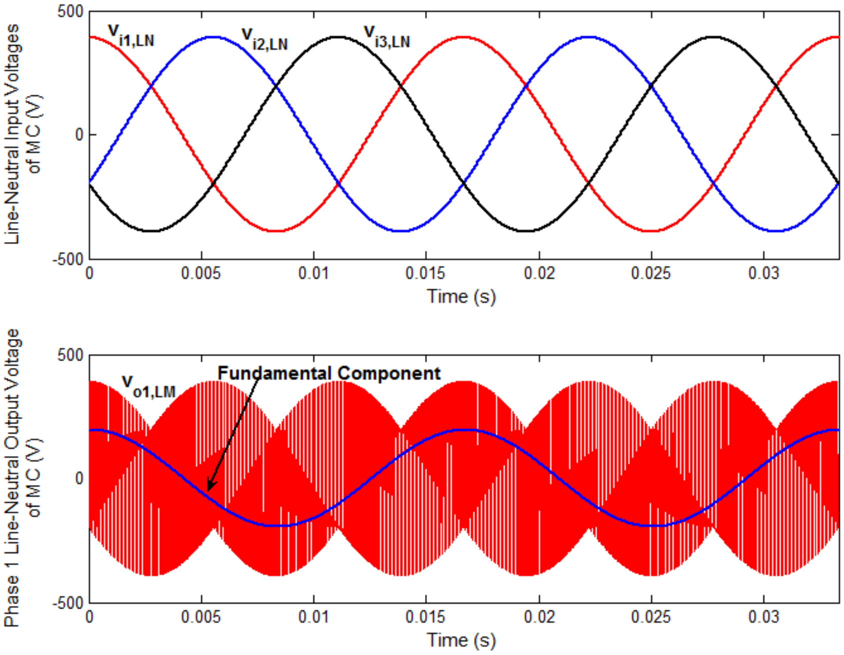


Fig. 35. MATLAB simulation - Input and output voltages (phase 1) of the MC

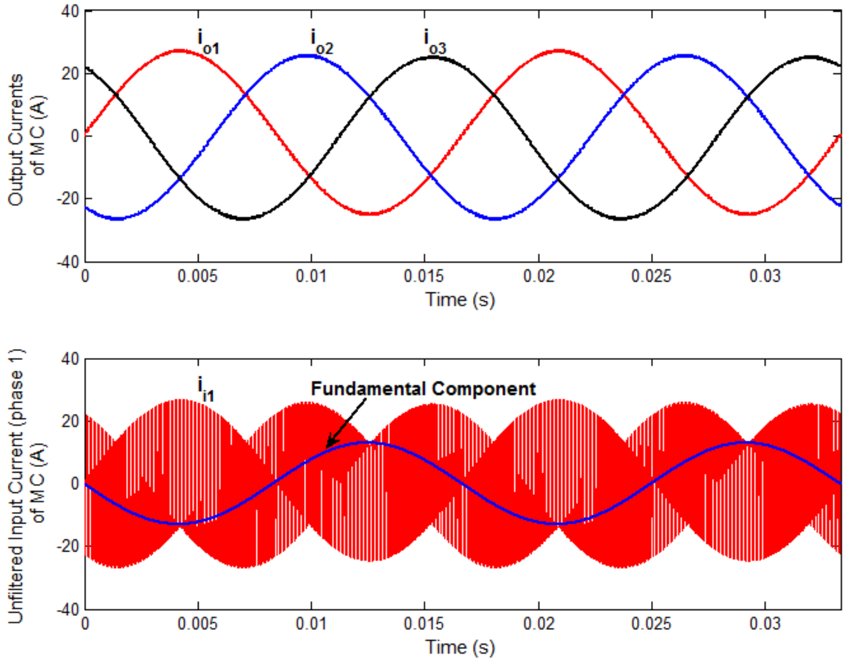


Fig. 36. MATLAB simulation – Output and input currents (phase 1) of the MC

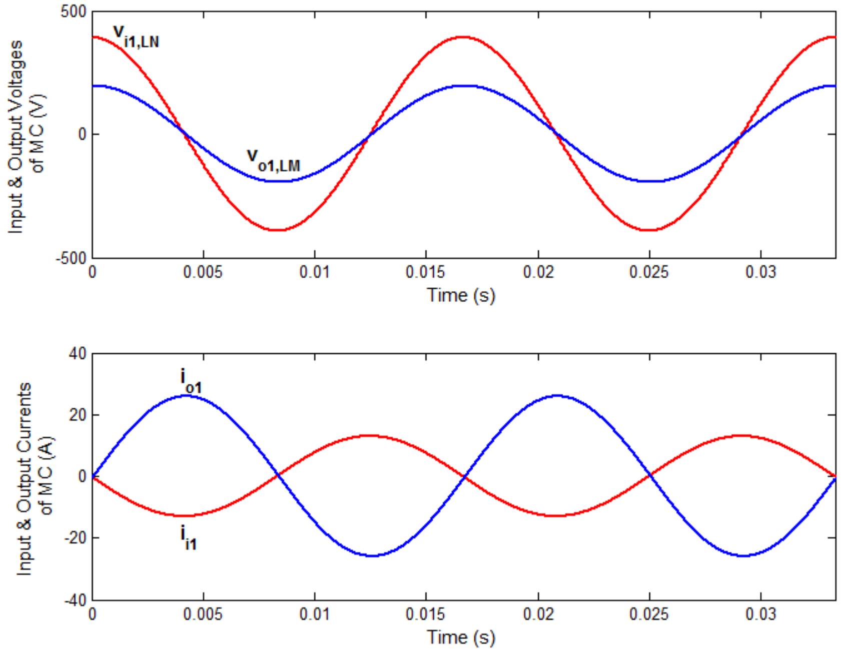


Fig. 37. MATLAB simulation - Phase relationships between voltages and currents of the MC

The simulation result of Fig. 38 shows that reactive power compensation has been achieved. Though the load draws lagging current i_{Load1}, the effective load (the compensator and the original load) draws unity power factor (UPF) current i_1 that is in phase with the ac network voltage.

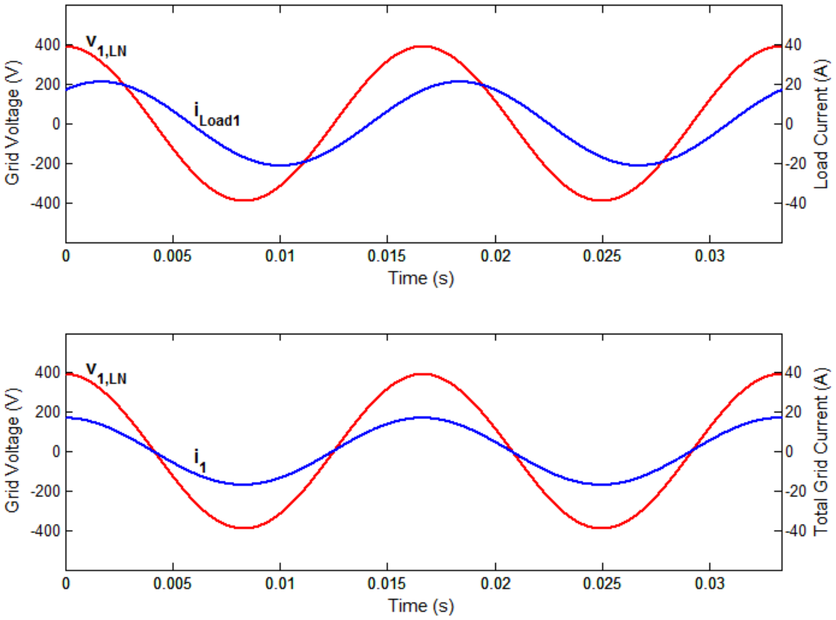


Fig. 38. MATLAB simulation – Lagging load currents and UPF ac network currents showing the achieved reactive power compensation

The variation of reactive power compensation with the modulation index ‘*q*’ for the given compensator parameters is shown in Fig. 39. As expected from (31), the VAR compensation increases with the square of the modulation index. Since there were no losses modeled in the converter, there is no real-power flow. Interestingly, the figure shows a small offset in the relationship between ‘*q*’ and the reactive power compensated. For example, when ‘q’ is zero, the reactive power compensated is non-zero (90 VAR).

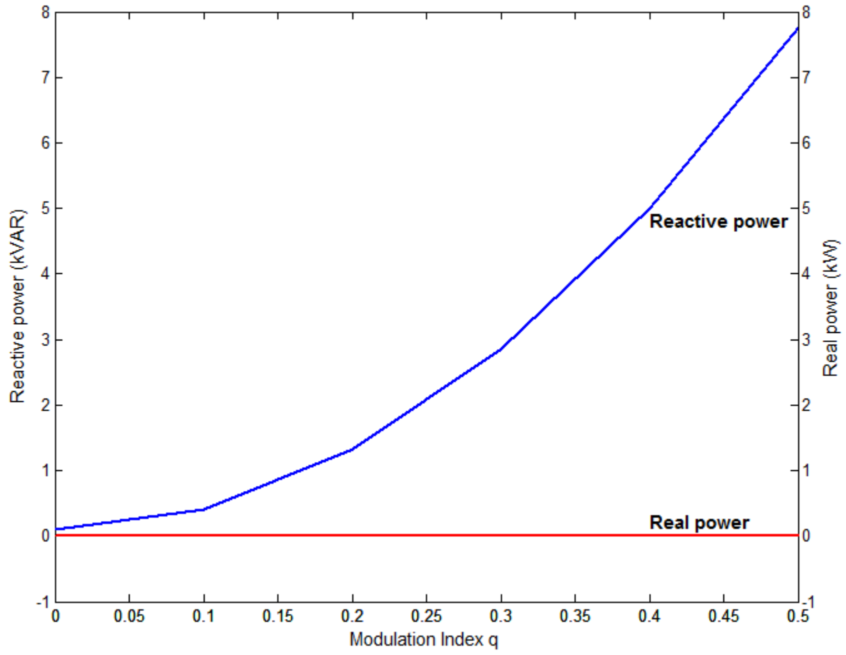


Fig. 39. Variation of lagging reactive power compensated and real power drawn by the MC with modulation index *'q'*

This error, which is the result of the switching functions not exactly implementing the modulation functions, decreases with an increase in the value of '*N*', as explained in Chapter II. Another solution to the error that might be explored in future work is the introduction of a phase shift 'δ' in the modulation functions as in

$$\begin{aligned} H_1 &= \frac{1}{3}\left(1 + 2q\cos\left(2\omega t + \delta\right)\right) \\ H_2 &= \frac{1}{3}\left(1 + 2q\cos\left(2\omega t - 2\pi/3 + \delta\right)\right) \\ H_3 &= \frac{1}{3}\left(1 + 2q\cos\left(2\omega t + 2\pi/3 + \delta\right)\right) \end{aligned} \qquad (33)$$

2. Simulation in PSIM

The ac network system with the lagging load and compensator were simulated in PSIM as shown in Fig. 40. The bi-directional switches of the MC are implemented in the common-collector configuration as shown in Fig. 41. The switching function (S_1, S_2, S_3) generation is the same as for the simulations shown in Chapter II and is explained in

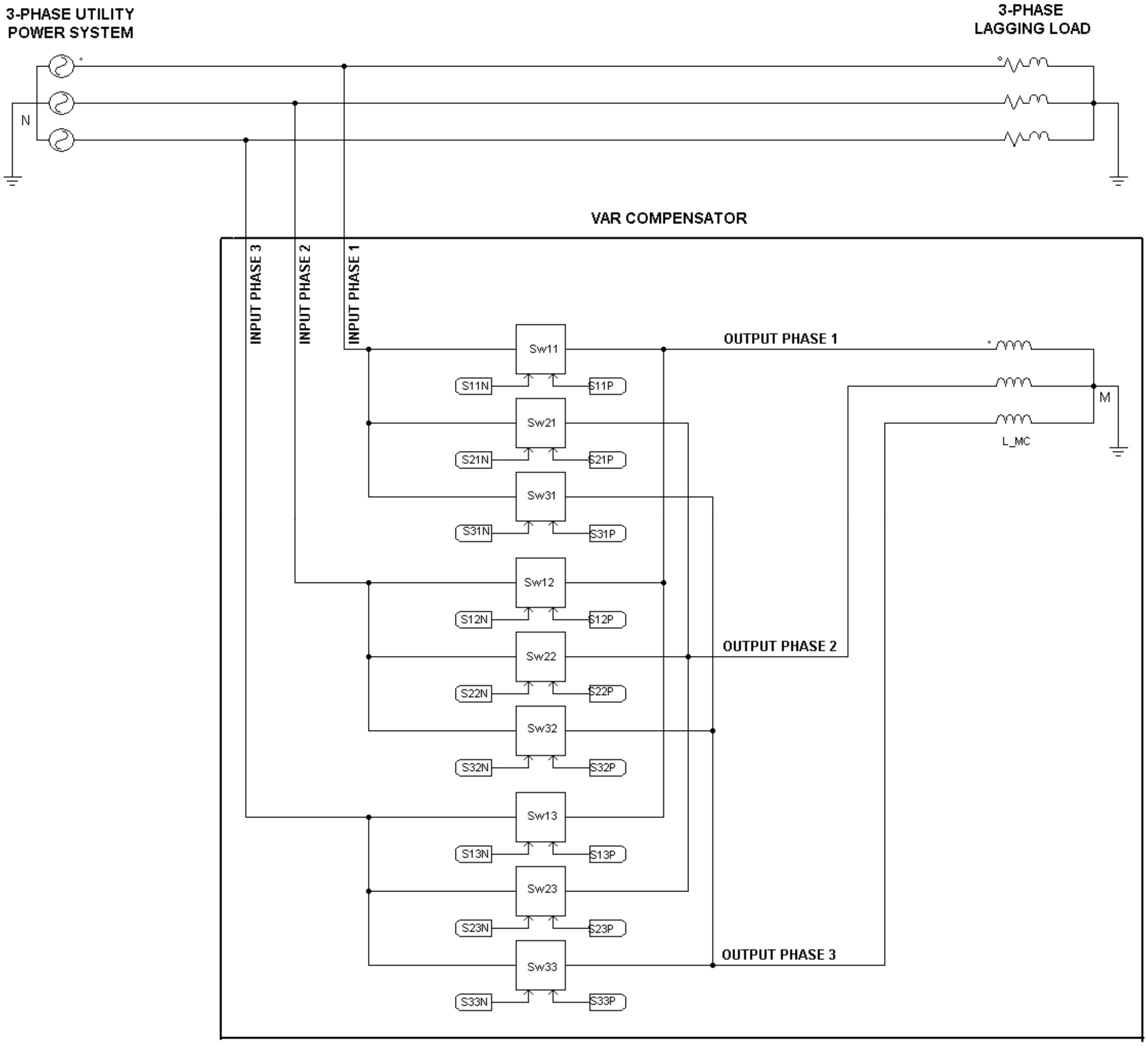


Fig. 40. PSIM schematic of ac network power system with a load and the proposed VAR compensator

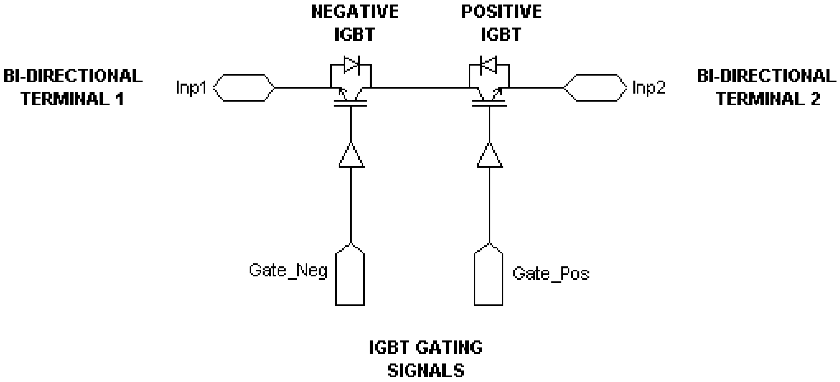


Fig. 41. Bi-directional switch sub-circuit of the MC (common-collector configuration)

Appendix B. A switching sequence known as the four-step commutation is implemented to generate the final 18 gating signals (S11P, S11N, S21P, S21N etc.). The four-step sequence is required to prevent commutation failure in the MC. The PSIM schematic of the control logic is given in Appendix B.

a. Four-step commutation logic

During the commutation of an output current of the MC from one input phase to the other, a safe switching sequence must be implemented so as to prevent an open-circuit on the output and a short-circuit of the input phases. The four-step commutation logic is one of the most popular ways used to ensure this safety. The logic can be easily explained with the help of the example shown in Fig. 42, where the output current is transferred from input phase 1 to input phase 2. Then, S1P and S1N can be termed as the outgoing switches, while S2P and S2N are known as the incoming switches. Let it be assumed that the outgoing and incoming switches are governed by switching functions

S_1 and S_2, respectively. According to the current direction shown, it can be seen that only S1P and S2P carry current during the commutation. Then, the four-step commutation logic results in the switching sequence shown in Fig. 43. The ‘inactive’ S1N is turned off first, followed by the switching on of the incoming ‘active’ S2P. The outgoing switch S1P is then turned off, after which the switch S2N is turned on.

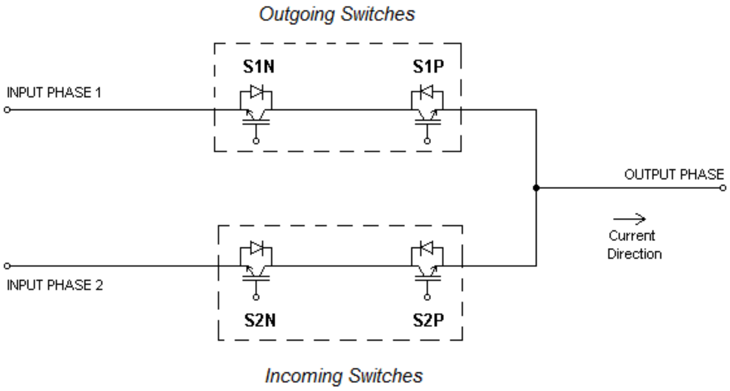


Fig. 42. Example for four-step commutation

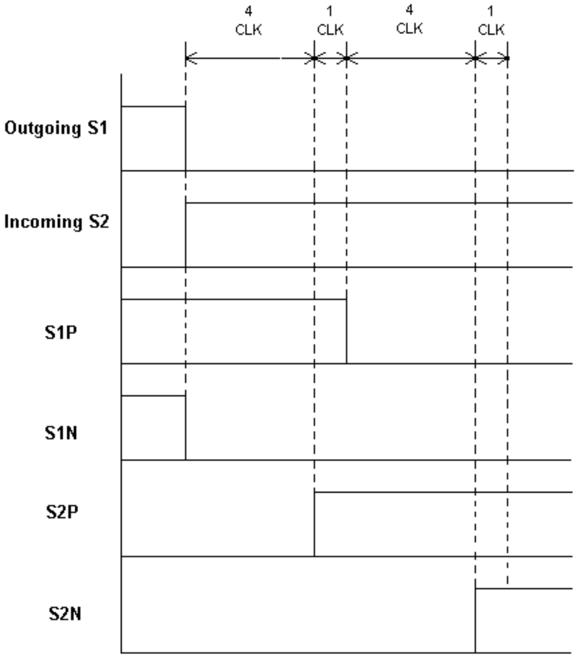


Fig. 43. Switching sequence for example four-step commutation

The time between each step of the commutation is decided based on the required turn-on and turn-off times of the switches, to ensure the safe transfer of current between the switches. The IGBT-based MC module FM35R12KE3 by EUPEC has been used as a reference for the simulation. The IGBTs used in this module have total turn-on and turn-off times of 135 ns and 610 ns, respectively [28]. As PSIM is a fixed time-step software, turn-on and turn-off times of 200 ns and 800 ns have been considered so as to result in reasonable simulation run-times. Accordingly a clock signal (CLK) of time period 200 ns is used as a base for the commutation logic. It is seen that switching signals to the outgoing IGBTs follow the corresponding switching function after a delay of 0 CLK and 5 CLK depending on whether they are carrying current or not. Similarly, the signals to the incoming switches follow their corresponding switching function after a delay of 4 CLK and 9 CLK cycles depending on their current carrying status. Following this logic, the four-step commutation has been implemented in PSIM using shift registers to generate the delayed versions of the switching functions. A 4x1 multiplexer is used to choose between the four delayed versions based on the current carrying status of the IGBTs and whether they are outgoing or incoming. Changes in the current direction or the switching functions during the commutation period can disrupt the switching sequence, causing over-voltages and over-currents in the MC. To prevent such situations, detection of the currents and switching functions is synchronized with an ENABLE signal that has a time period equal to the commutation duration (2 μs).

b. Results of the Simulation

The three switching functions S_1, S_2 and S_3 are shown in Fig. 44 and Fig. 45.

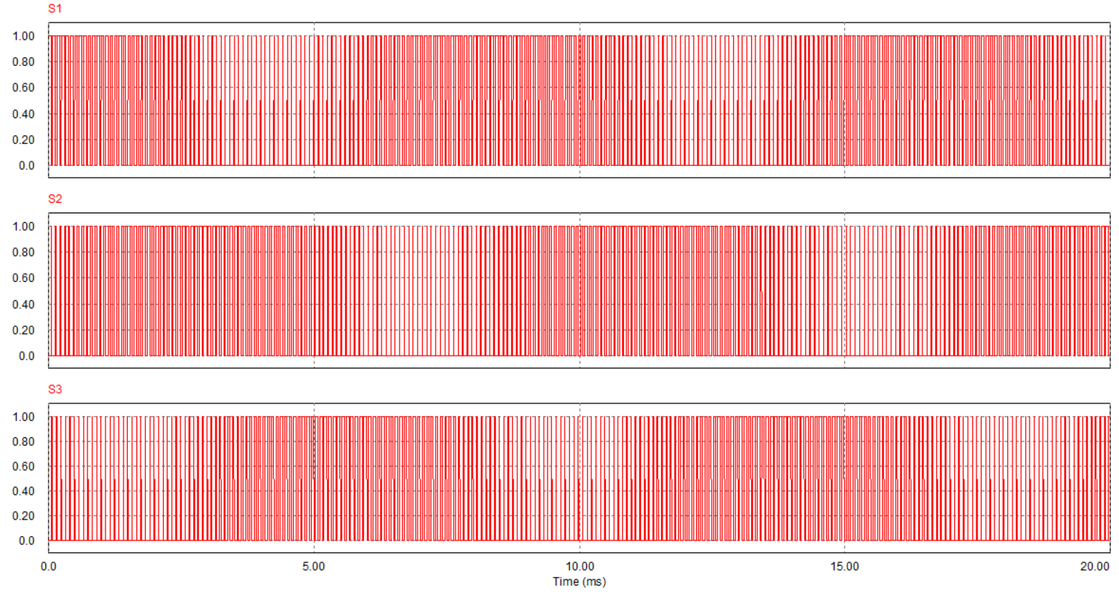


Fig. 44. Switching functions S_1, S_2 and S_3

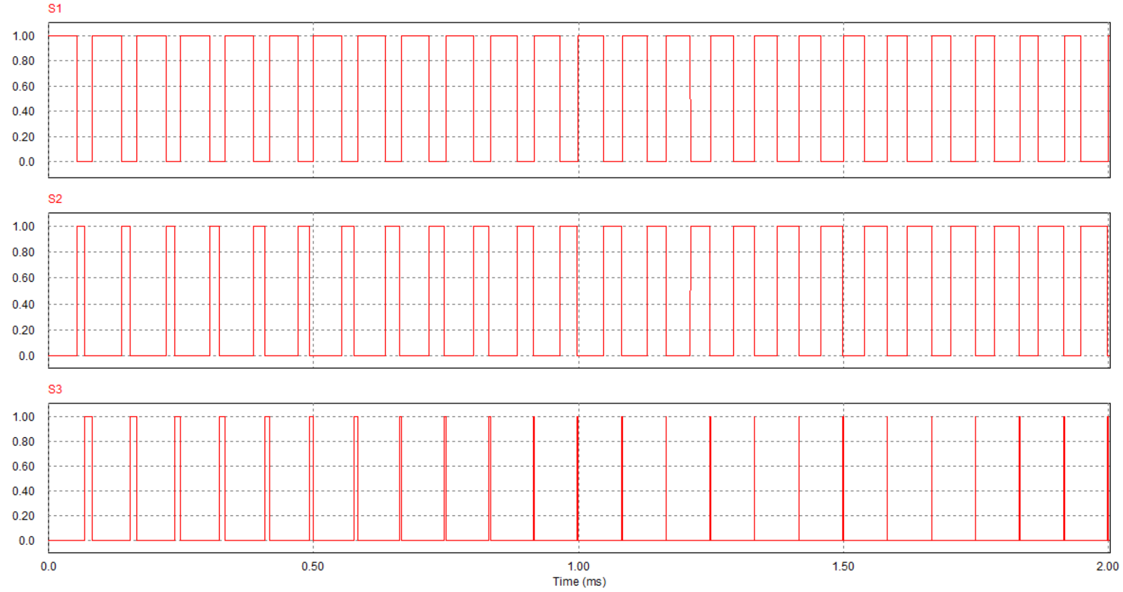


Fig. 45. Switching functions S_1, S_2 and S_3 (zoomed in)

The commutation process is shown in Fig. 46 and Fig. 47 for the cases when the switches are carrying positive and negative currents, respectively.

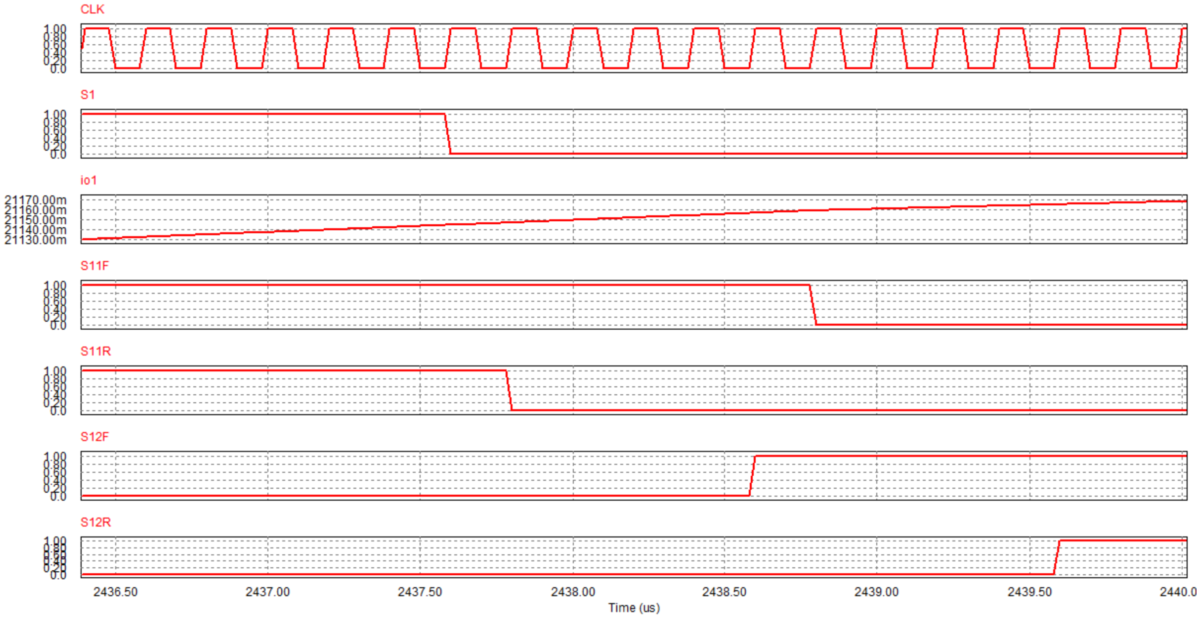


Fig. 46. Four-step commutation between S11P, S11N, S12P and S12N when $i_{o1} > 0$

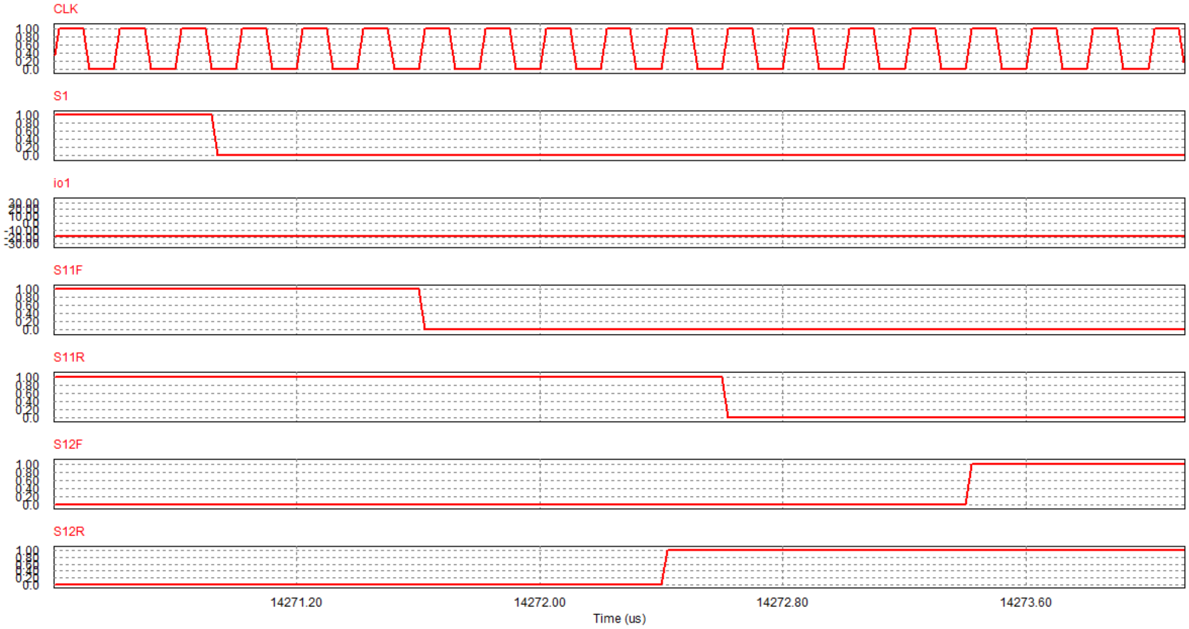


Fig. 47. Four-step commutation between S11P, S11N, S12P and S12N when $i_{o1} < 0$

The function of the ENABLE signal which detects changes in switching signals and current only on its rising edge is shown in Fig. 48. It is seen that the changes in the switching function values and the current directions are asserted on the gating signals only at the rising edge of the ENABLE signal.

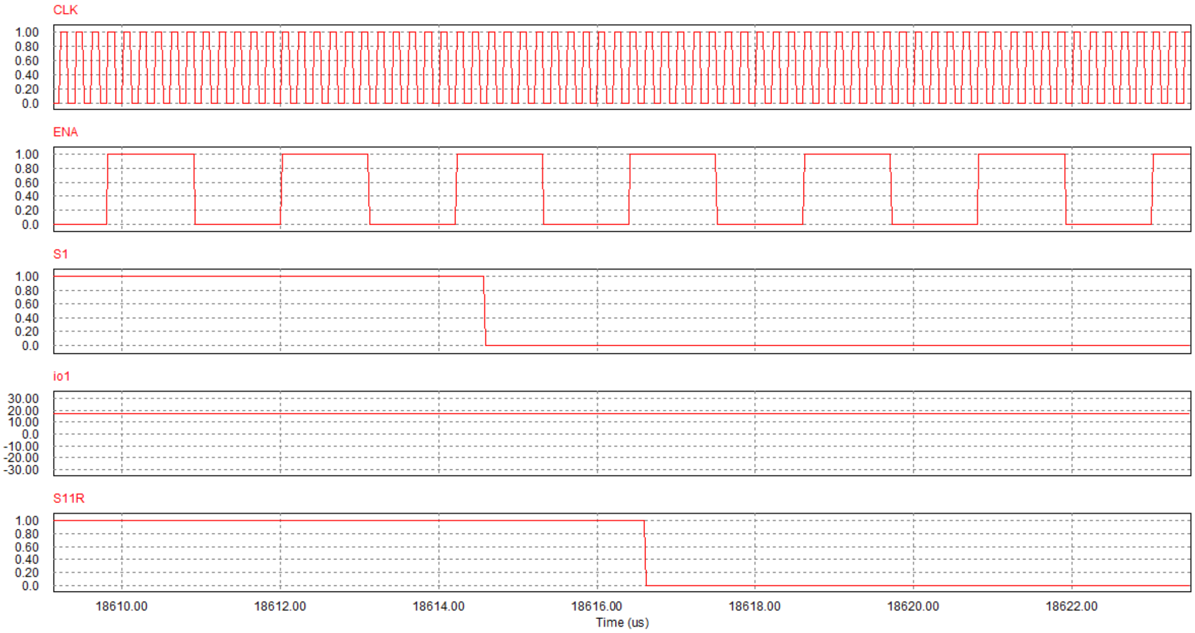


Fig. 48. Function of the ENA (enable) signal

The input and output voltages of phase 1 of the MC obtained using the four-step commutation technique with the Venturini modulation method are shown in Fig. 49. The load current i_{Load1} is shown to lag the ac network voltage by the power factor angle (36.87°) in Fig. 50. The effective load current drawn by the original load and the VAR compensator is also shown. It is seen that this ac network current is in phase with the ac network voltage, indicating that no reactive power is being transmitted from the source to the original load. Thus, reactive power compensation has been achieved by the proposed compensator.

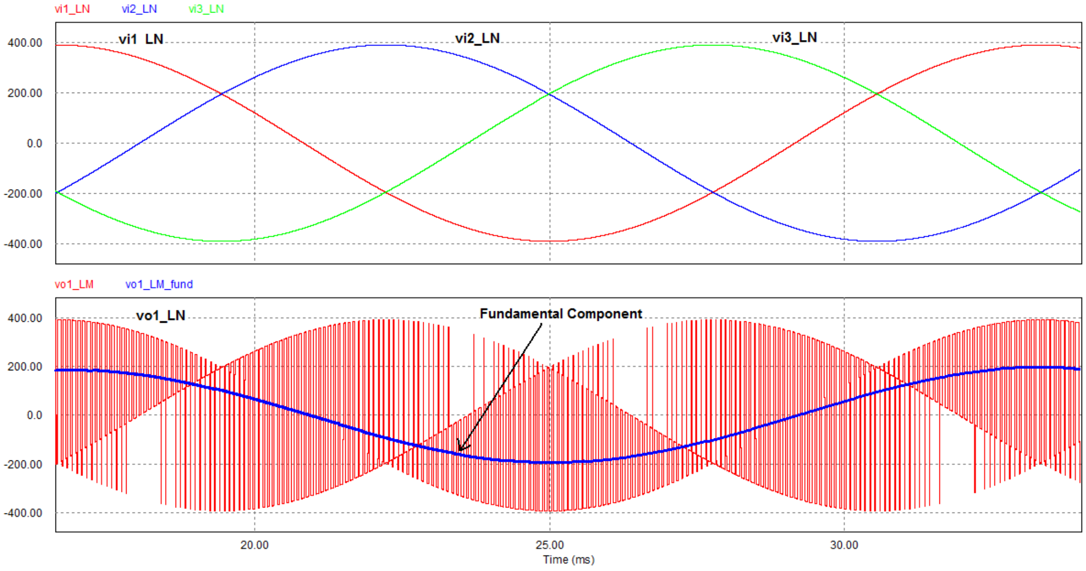


Fig. 49. The input and output voltages (phase 1) of the MC using four-step commutation with Venturini's modulation method

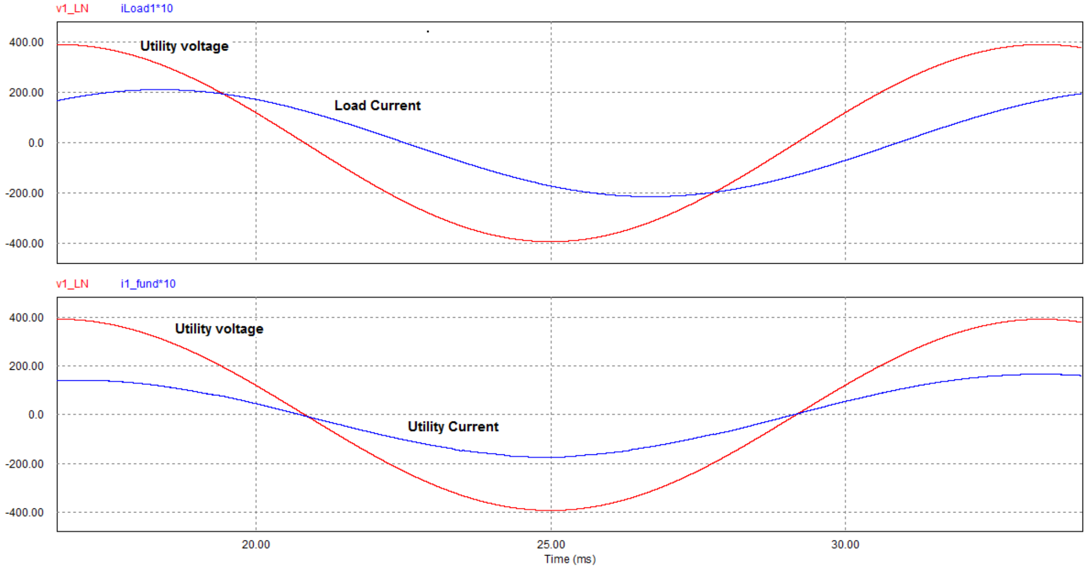


Fig. 50. Ac network current (load + compensator currents) in relation to the ac network voltage, showing that VAR compensation has taken place

It can be noted from the figure that the ac network current is not exactly in phase with the voltage. As explained before, this difference is due to the inaccuracy introduced by the discrete modulation functions and can be eliminated/reduced by increasing the switching frequency, appropriately adjusting the modulation index '*q*' or investigating other modulation methods.

The proposed VAR compensator based on the Venturini modulation method is shown to be an effective solution to reliability issues of traditional capacitor-based compensation techniques. Simulation results strongly support the theoretical proposition, thus establishing the compensator as a viable and promising solution.

CHAPTER IV

DIGITAL IMPLEMENTATION OF CONTROL SIGNALS FOR THE MATRIX CONVERTER

The role of the control logic for the MC is to first generate the basic switching functions S_1, S_2 and S_3 and then use four-step commutation to result in the final 18 switching signals. This chapter presents a hardware implementation of the control logic using a digital signal processor (DSP) and a complex programmable logic device (CPLD) as shown in Fig. 51.

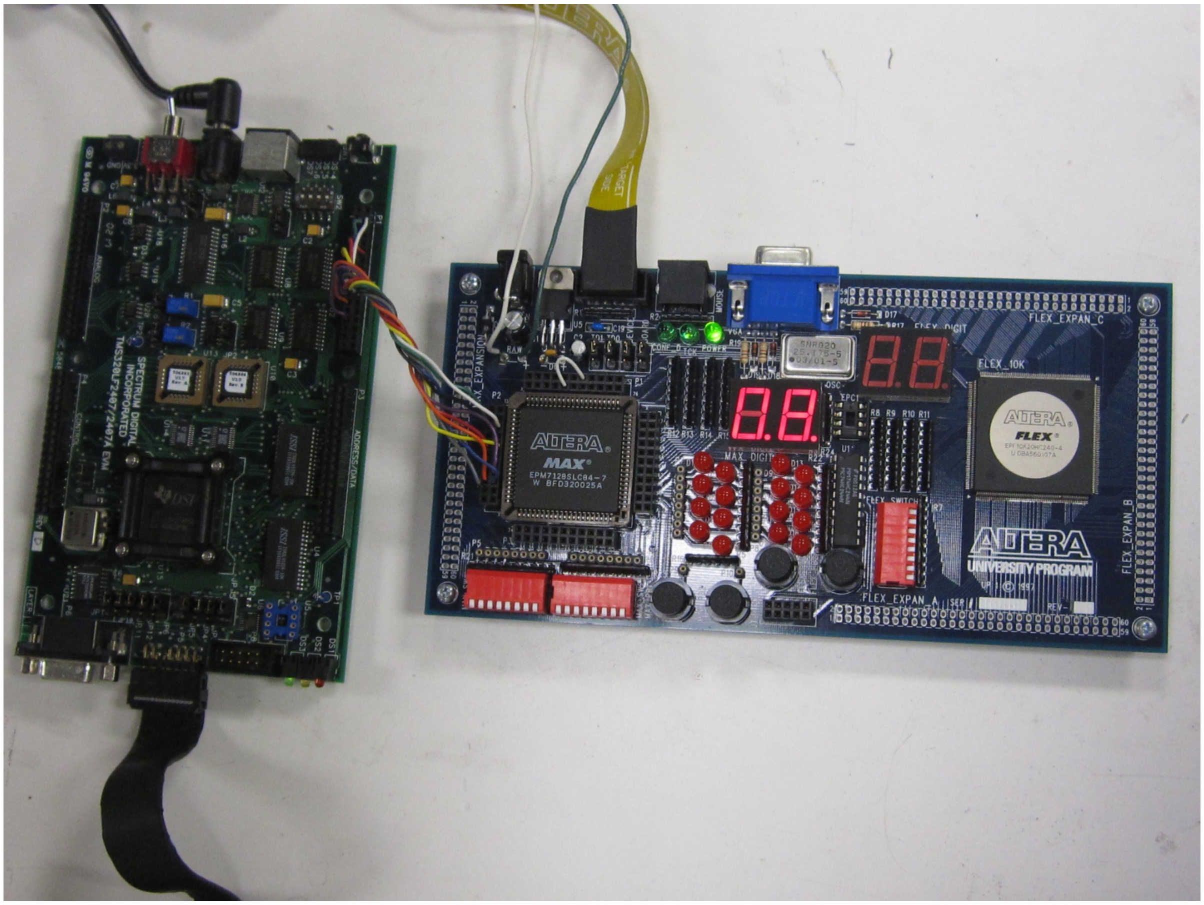


Fig. 51. Hardware set-up of the control logic for the MC

The block diagram of the control logic is given by Fig. 52 and is explained below:

i. DSP: The TMS320LF2407A, a 16-bit fixed point processor that runs at the rate of 40 million instructions per second (MIPS), is designed for high-performance control-based applications and is chosen for the implementation. The DSP is programmed in the C language using the Code Composer Studio software. The main function of the DSP is to generate the three switching functions S_1, S_2 and S_3. As the three functions are mutually exclusive, two of them are found to be sufficient to generate all the switching signals. Two timers (T1 and T2) of the DSP, are used in the compare/PWM (pulse-width modulation) to produce S_1 and S_3, respectively. Look-up tables are used to store the different '*ON*' times given by t_1 and t_3, which are loaded into the timer compare registers using the interrupt system of the DSP. The DSP is also required to generate digital current direction signals that are used by the CPLD in the commutation logic. In the case when the entire compensator is designed in hardware, voltage and current sensor outputs

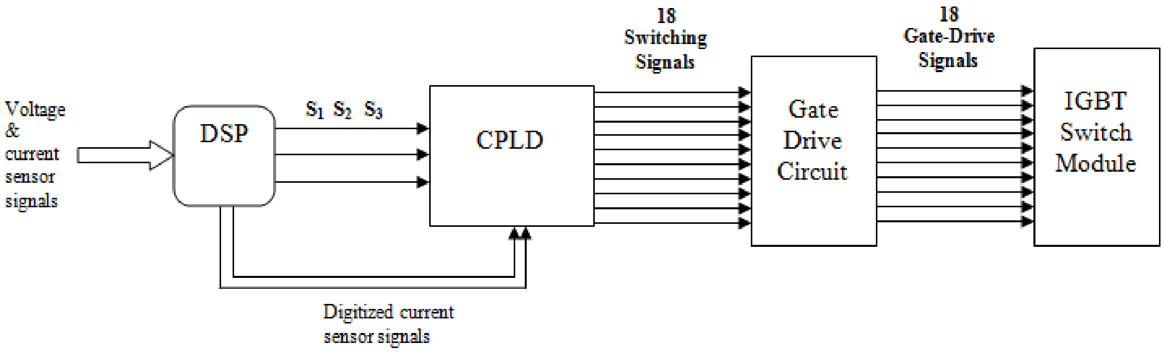


Fig. 52. Block diagram of the digital implementation of control signals for the MC

are converted to digital signals by the DSP's analog-to-digital converter (ADC). However, in the VAR compensator, it is known that the output currents of the MC lag its output/input voltages by 90° due to the choke element at the MC output. So, the current directions have been predicted based on the switching functions which are actually in phase with the MC voltages.

ii. CPLD: The EPM7128SLC84-7 from the Altera MAX7000S family is chosen to implement the commutation logic. It is an EEPROM-based PLD with 128 macro-cells that can provide speed or power optimization. The CPLD is programmed using Quartus-II and its schematic is as shown in Fig. 53. The four-step commutation is implemented using a state-machine, which is found to be more efficient and robust than the logic presented in Chapter III. It does not require the additional ENABLE signal which can induce delays in the switching signals. The clock CLK is externally generated by the DSP using one of its timers. The switch signal outputs are registered using flip-flops triggered by the global clock of the CPLD, to avoid asynchronous glitches in the outputs.

The switches of each output phase of the MC are controlled by the state-machine shown in Fig. 53. From the figure, it is seen that the states 1, 10 and 19 are event-triggered by the falling or rising edges of the incoming S1 and S3, while the other states are pre-determined by the previous states.

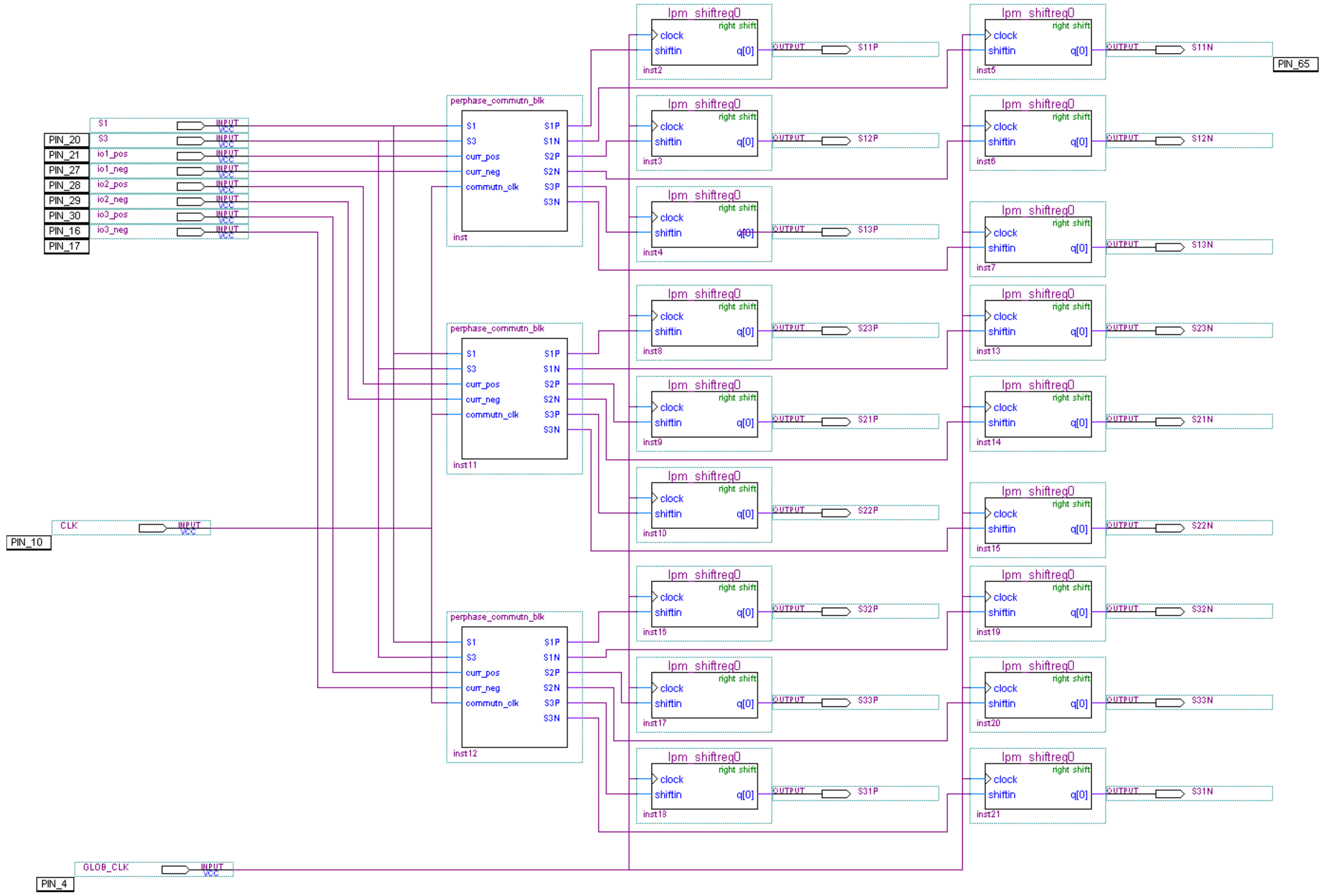


Fig. 53. Schematic of CPLD logic in Quartus

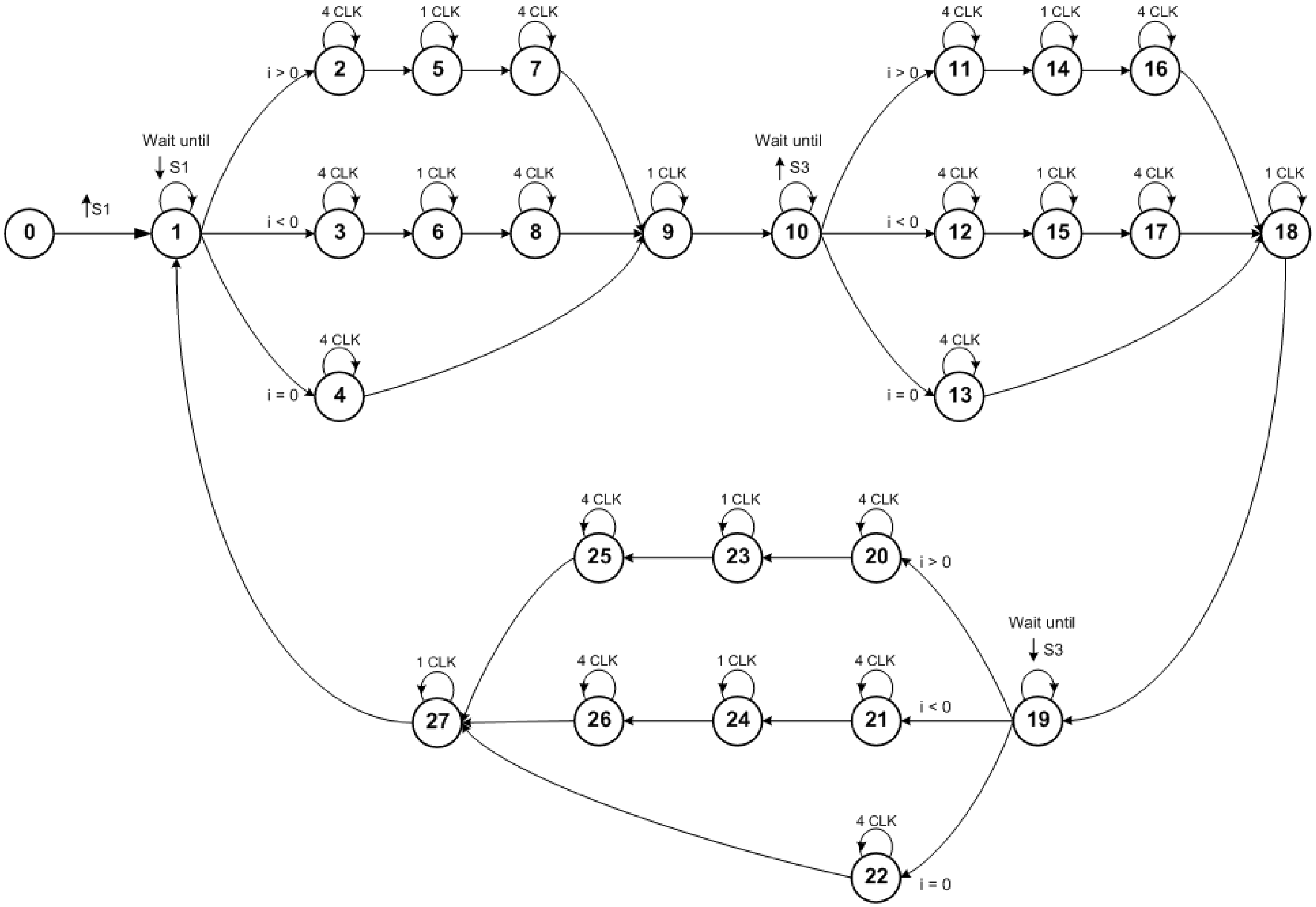


Fig. 54. State machine representation of four-step commutation

The definition of states is provided in TABLE 7, where S1P/S1N, S2P/S2N and S3P/S3N denote the positive and negative switches governed by S_1, S_2 and S_3 respectively.

TABLE 7. State table showing switching signal outputs

STATES	**STATE OUTPUTS**					
	S1P	**S1N**	**S2P**	**S2N**	**S3P**	**S3N**
0	0	0	0	0	0	0
1	1	1	0	0	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	0	0	0	0
5	1	0	1	0	0	0
6	0	1	0	1	0	0
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0
10	0	0	1	1	0	0
11	0	0	1	0	0	0
12	0	0	0	1	0	0
13	0	0	0	0	0	0
14	0	0	1	0	1	0
15	0	0	0	1	0	1
16	0	0	0	0	1	0
17	0	0	0	0	0	1
18	0	0	0	0	1	1
19	0	0	0	0	1	1
20	0	0	0	0	1	0
21	0	0	0	0	0	1
22	0	0	0	0	0	0
23	1	0	0	0	1	0
24	0	1	0	0	0	1
25	1	0	0	0	0	0
26	0	1	0	0	0	0
27	1	1	0	0	0	0

The switching functions obtained as DSP outputs are given by Fig. 55 and Fig. 56. The ‘START’ signal corresponds to the first pulse of the switching functions given by ‘*k=0*’.

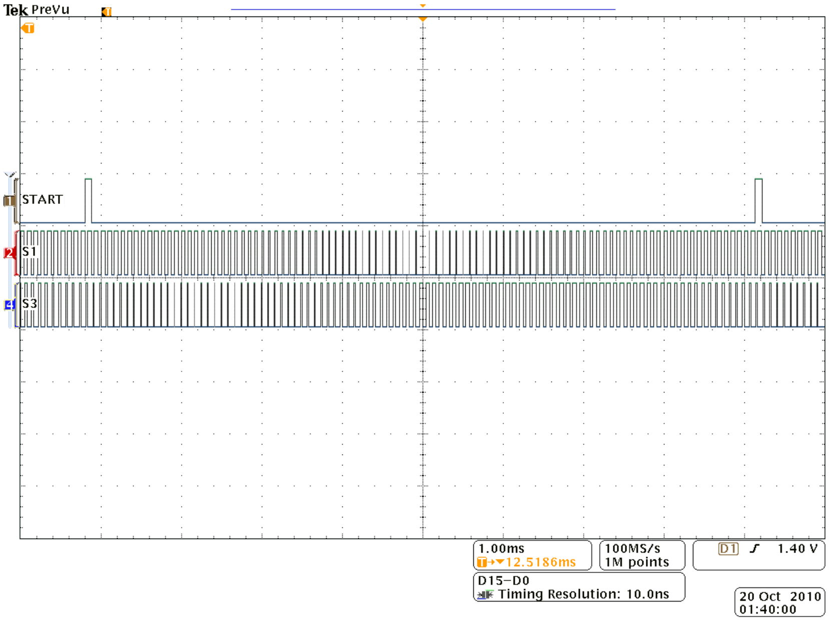


Fig. 55. Switching functions S_1 and S_3 generated by the DSP

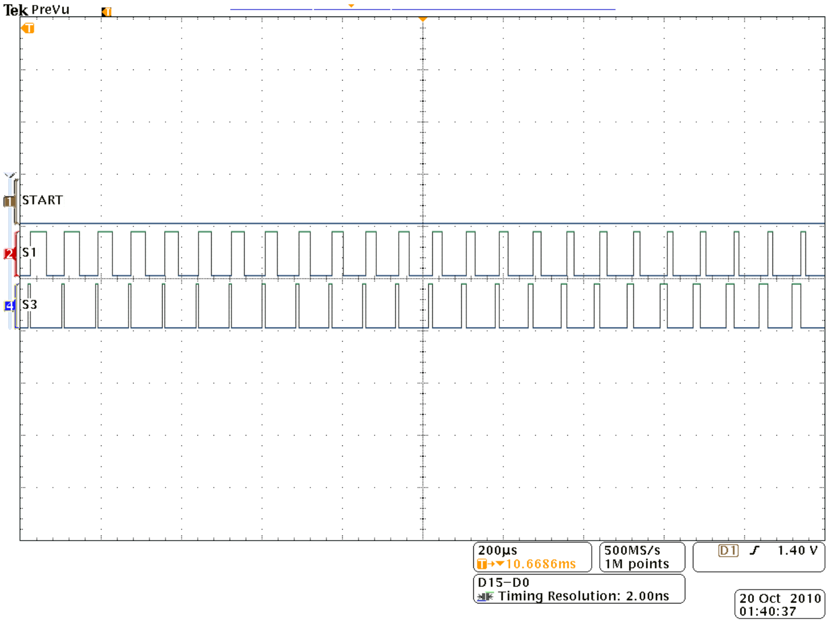


Fig. 56. Zoomed-in image of the switching functions

The current directions are shown in Fig. 57 as generated by the DSP. If only the '*positive*' signal corresponding to an output current is high, the current is positive and if only the '*negative*' signal is high, the current is negative. The current is otherwise zero.

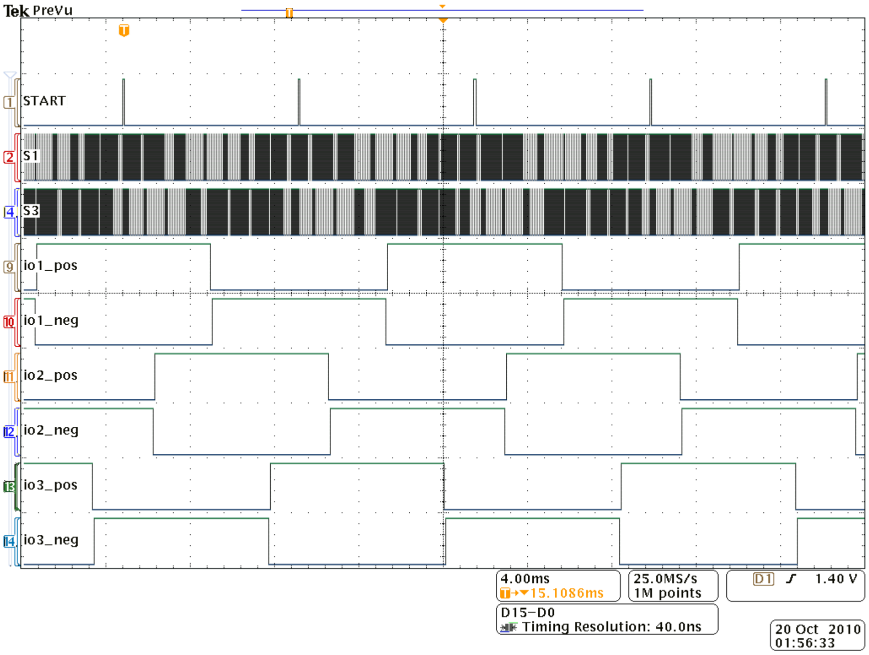


Fig. 57. Current directions generated by the DSP

Screenshots of the switching signals for each MC output phase are shown in Fig. 58, Fig. 59 and Fig. 60.

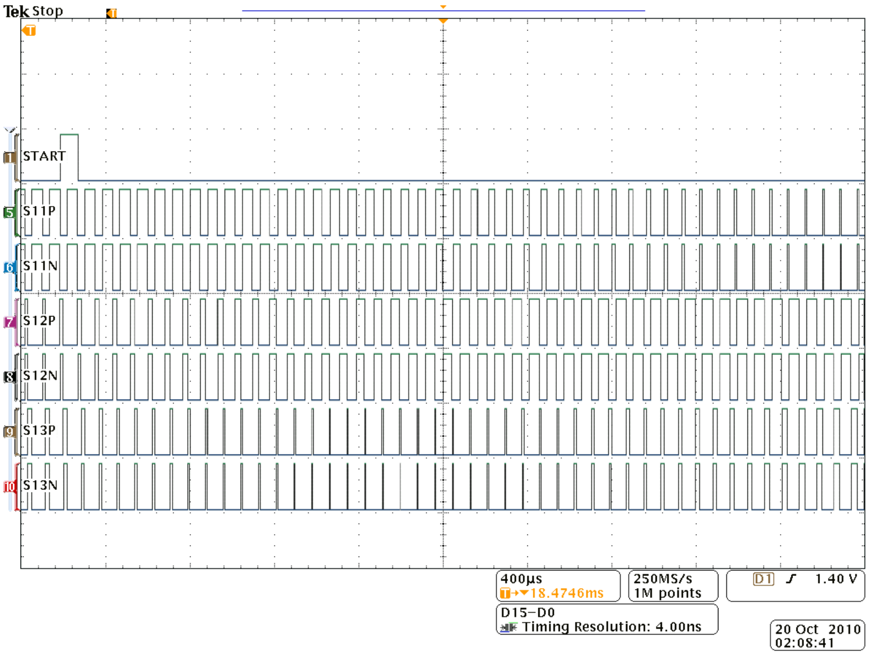


Fig. 58. Switching signals for output phase 1

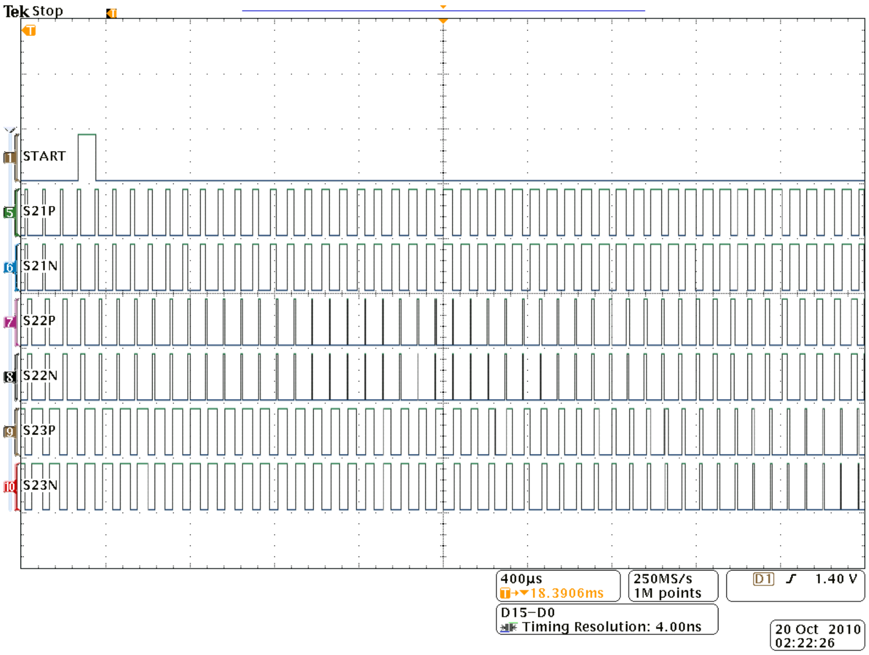


Fig. 59. Switching signals for output phase 2

The commutation sequence for switches S11P, S11N, S12P and S12N, which are governed by the functions S_1 and S_2, is shown in Fig. 61, for positive current i_{o1}.

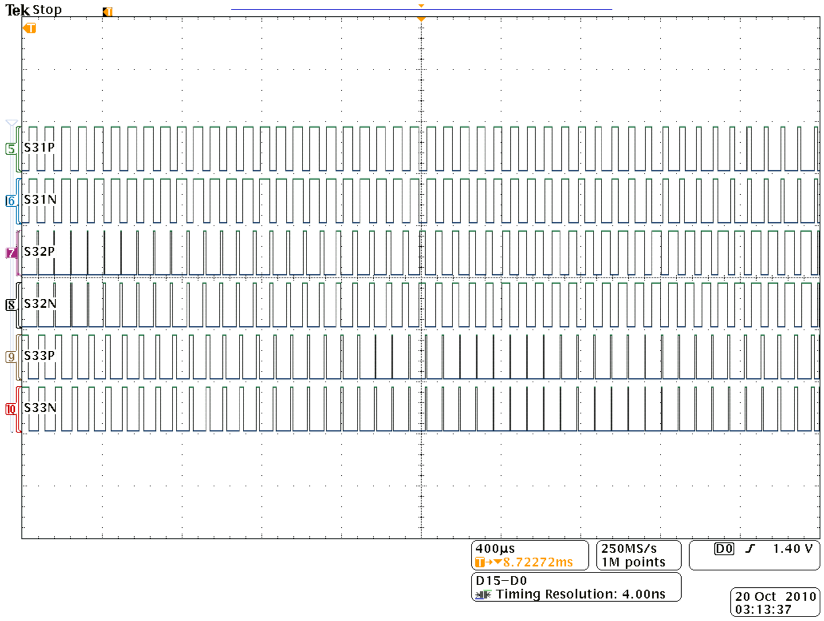


Fig. 60. Switching signals for output phase 3

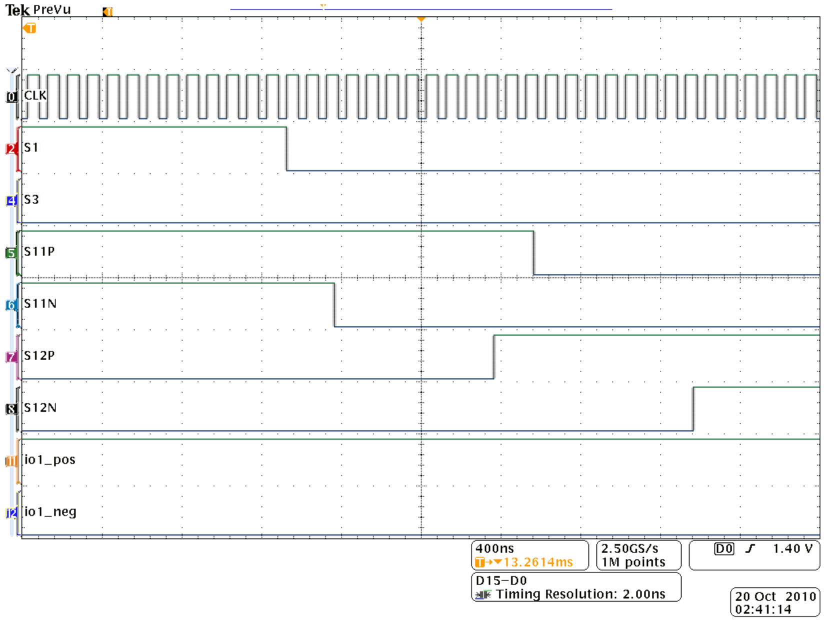


Fig. 61. Commutation sequence for switches governed by S_1 and S_2 for positive output current

Four-step commutation for S11P, S11N, S12P and S12N is shown in Fig. 62 and Fig. 63 for negative and zero output current directions, respectively.

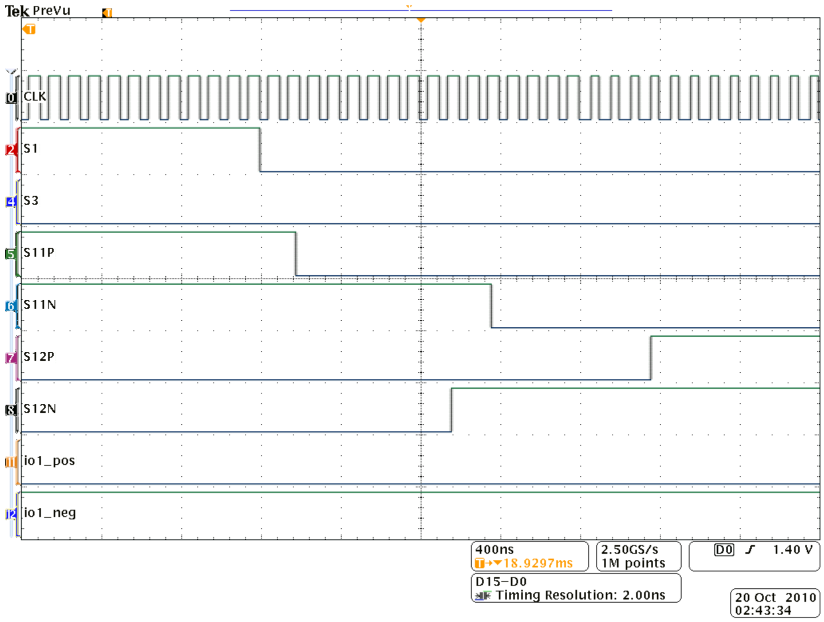


Fig. 62. Commutation sequence for switches governed by S_1 and S_2 for negative output current

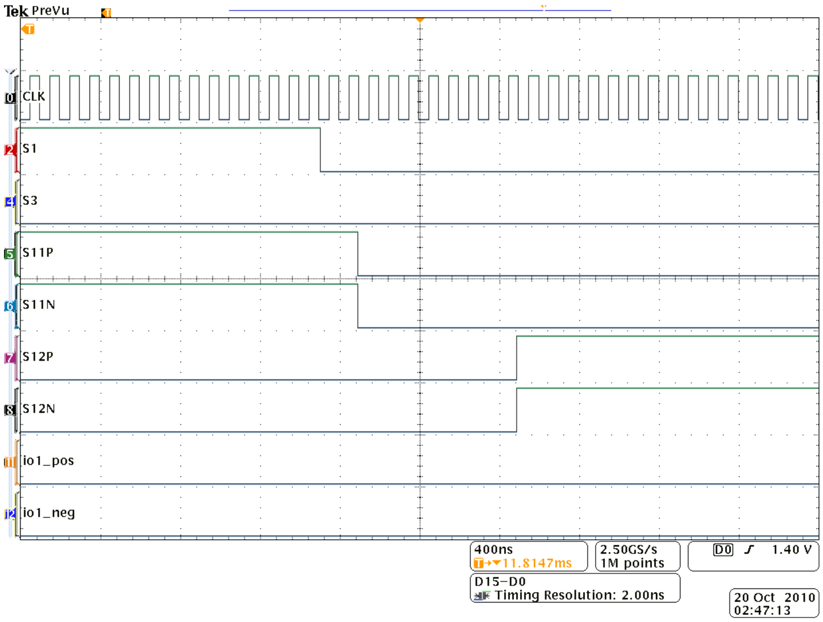


Fig. 63. Commutation sequence for switches governed by S_1 and S_2 for zero output current

The commutations between switches governed by S_2 - S_3 and S_3 - S_1 are shown in Fig. 64 and Fig. 65, respectively

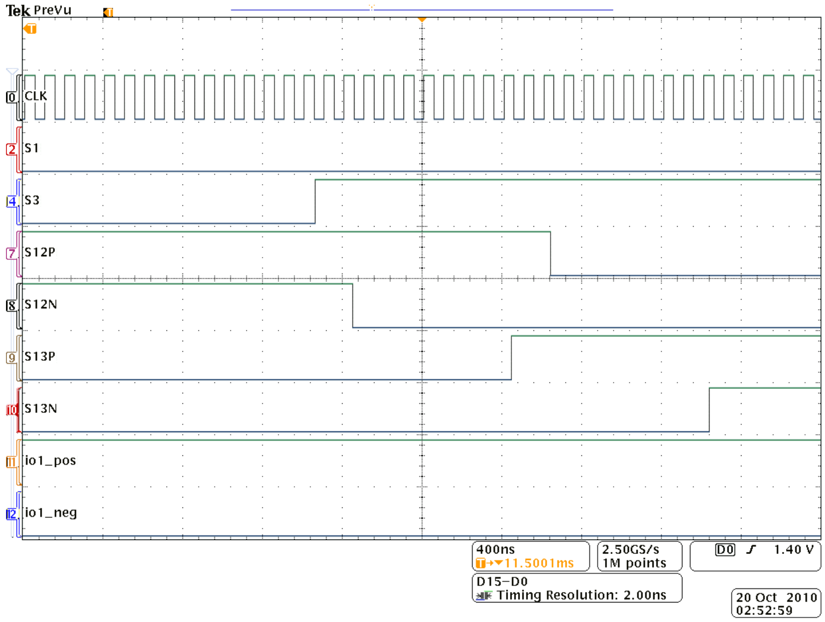


Fig. 64. Commutation sequence for switches governed by S_2 and S_3 for positive output current

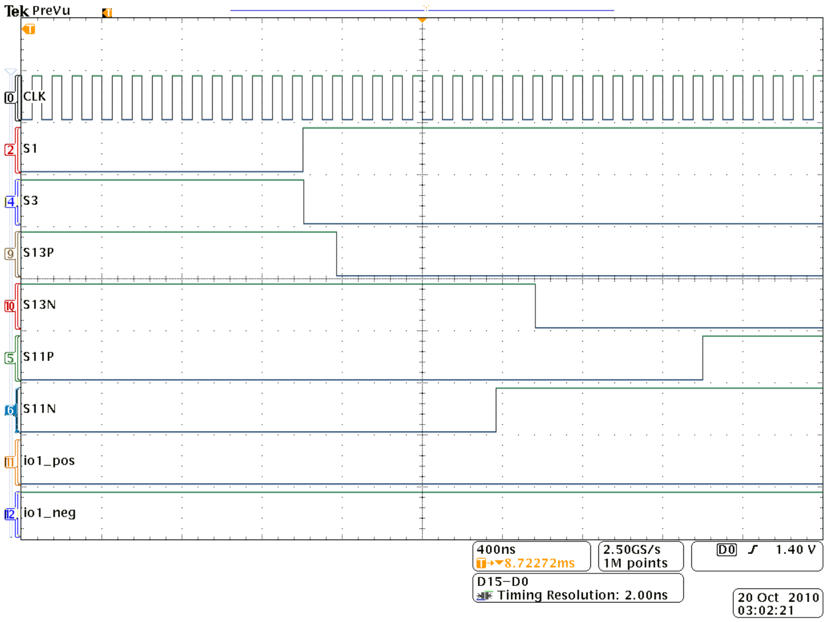


Fig. 65. Commutation sequence for switches governed by S_3 and S_1 for negative output current

A block diagram of the complete hardware implementation of the proposed VAR compensator is given in Fig. 66. The components of the diagram are listed and explained below [17, 29, 30].

i. Matrix converter module: The module FM35R12KE3 by EUPEC, based on 18 common-collector IGBTs, may be used in the implementation [25]. The switches are rated at a collector-emitter voltage of 1200 V and a collector current of 35 A.

ii. Input filters: As seen previously, LC line filters are required to reduce input current distortion. Several factors must be considered while designing these filters such as cut-off frequency, damping factor, over-voltages and over-currents. Also, some lower harmonics might be amplified on adding the filter.

iii. Clamp circuit: The matrix converter must be protected from over-voltages due to input line disturbances and output current faults. Clamp circuits consisting of fast recovery diodes and a capacitor are used to safely disseminate the energy.

iv. Gate-drive circuit: The matrix converter module FM35R12KE3 requires six isolated gate-drive supplies [25] consisting of 60 Hz laminated transformers, linear regulators and optical isolators. The drive circuit must be designed well to have fast response time and excellent signal isolation.

In addition to the above components, excellent noise immunity must be ensured through careful design as the MC has a large number of switches. In addition, the MC is more prone to line unbalances and distortion due to the lack of intermediate energy storage elements. So, techniques that account for disturbances must be employed [30].

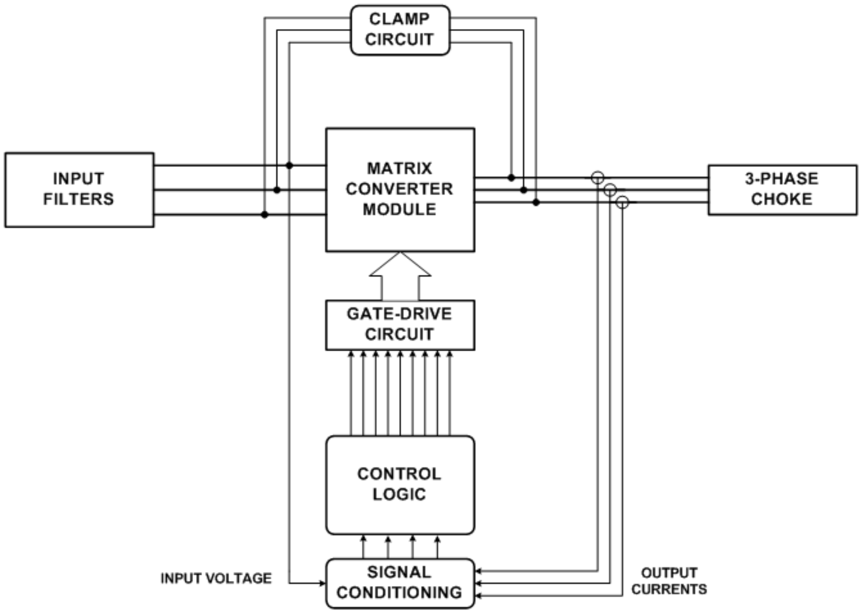


Fig. 66. Block diagram of hardware implementation of the VAR compensator

From the above description, it can be seen that the hardware implementation of the entire VAR compensator is a rigorous process involving several factors. However, its development and design is not essential to this thesis as previous work [29-31] indicate it to be practically possible. On the other hand, the implementation of the control logic in hardware is more critical and relevant to this thesis. The results presented in this chapter show that the Venturini modulation method can be implemented accurately using a DSP and a CPLD. Thus, the proposed VAR compensator has been shown to be a practically feasible solution.

CHAPTER V

CONCLUSION

Reactive power in the ac power system network is attributed to the complex impedance of the loads and the transmission lines. While it is fundamental to the system, reactive power is detrimental to the reliability, efficiency and overall performance of the ac network. This thesis focuses on load-side reactive power compensation, a technique used to manage the reactive power requirement of loads by absorbing or supplying VARs to the ac network. The existing capacitor-based solutions to VAR compensation have been reviewed in detail. The serious reliability issues of capacitors resulting from catastrophic and wear-out failure modes have been explained. Thus, the need to eliminate capacitors from load-side VAR compensation techniques has been established. The thesis proposes a load compensation technique based on the 3-phase ac-to-3-phase ac matrix converter, which uses inductors instead of capacitors, to locally supply VARs to loads.

The Venturini modulation method has been chosen to control the matrix converter as it can enable the converter to operate with the advantageous current phase reversal property. Through this property, though the inductive element of the proposed compensator consumes VARs from the matrix converter, the converter supplies VARs to the ac power system network. Detailed analysis of the operation of the matrix converter using the Venturini method has been taken up in the thesis to establish the current phase reversal property. The application of the matrix converter and inductors to VAR compensation has been justified through mathematical analysis. An expression for the VARs compensated by the proposed system has also been established. Simulation studies carried out in the MATLAB and PSIM environments support the theoretical analysis. Results for a specific case establish that load-side VAR compensation is achieved by the proposed solution as per the initial claims. Associated practical concerns such as input line filters and four-step commutation have also been addressed. Finally, a digital implementation of the switching control signals that uses a DSP and a CPLD has been presented to emphasize the practical feasibility of the proposed topology. The hardware generated results show that the Venturini modulation method can be effectively implemented as in theory and can be further applied to the proposed VAR compensator. Previous work shows the hardware implementation of the entire compensator to be a rigorous but practically feasible process. So, its development is not seen to be critically essential to this thesis. Thus, the proposed capacitor-less VAR compensator has been shown to be an effective and promising solution to the reliability issues of traditional capacitor-based VAR compensation techniques.

Future work in this area might include studies on the hardware implementation of the entire proposed topology and associated practical issues including over-voltage and over-current protection, line-filtering and noise isolation. Additional research maybe conducted on improving the performance characteristics of the matrix converter such as speed of response and harmonic content of voltage and current waveforms. Other modulation techniques might also be explored.

REFERENCES

[1] T. J. E. Miller, *Reactive power control in electric systems*. New York: Wiley, 1982.

[2] J. Dixon, L. Moran, J. Rodriguez, and R. Domke, "Reactive power compensation technologies: state-of-the-art review," *Proceedings of the IEEE,* vol. 93, pp. 2144-2164, Dec. 2005.

[3] H. K. Tyll and F. Schettle, "Historical overview on dynamic reactive power compensation solutions from the begin [sic] of AC power transmission towards present applications," in *Proceedings of the IEEE Power Systems Conference and Exposition (PSCE '09)*, Seattle, WA, 2009, pp. 1-7.

[4] N. G. Hingorani, "FACTS technology - state of the art, current challenges and the future prospects," in *Proceedings of the IEEE Power Engineering Society General Meeting*, Tampa, FL, 2007, pp. 1-4.

[5] L. Gyugyi, "Reactive power generation and control by thyristor circuits," *IEEE Transactions on Industry Applications*, vol. IA-15, pp. 521-532, Sept. 1979.

[6] H. Nomura, K. Fujiwara, and H. Kawakami, "A power factor compensator using a force-commutated cycloconverter," *IEEE Transactions on Industry Applications*, vol. 26, pp. 769-776, Jul./Aug. 1990.

[7] A. M. Imam, D. M. Divan, R. G. Harley, and T. G. Habetler, "Real-time condition monitoring of the electrolytic capacitors for power electronics

applications," in *Proceedings of the Annual IEEE Applied Power Electronics Conference (APEC 2007)*, Anaheim, CA, 2007, pp. 1057-1061.

[8] M. Tarr, "Electrolytic capacitors." [Online]. Accessed Aug. 2010 Available: http://www.ami.ac.uk/courses/topics/0136_ec/index.html

[9] A. S. Farag, C. Wang, T. C. Cheng, G. Zheng, Y. Du, L. Hu, B. Palk, and M. Moon, III, "Failure analysis of composite dielectric of power capacitors used in distribution systems," in *Proceedings of the Electrical Insulation Conference and the Electrical Manufacturing & Coil Winding Conference*, Rosemont, IL, 1997, pp. 557-564.

[10] Catalog No. E1001D (Ver. 2), "Aluminum electrolytic capacitors," Nippon Chemi-Con Corporation, Tokyo, Japan.

[11] W. G. Dunbar, D. L. Schweickart, and J. C. Horwath, "AC capacitor failure mechanisms," in *Proceedings of the Electrical Insulation Conference and the Electrical Manufacturing & Coil Winding Conference*, Cincinnati, OH, 1999, pp. 89-92.

[12] P. Venet, H. Darnand, and G. Grellet, "Detection of faults of filter capacitors in a converter. Application to predictive maintenance," in *Proceedings of the Telecommunications Energy Conference (INTELEC '93)*, Paris, 1993, vol. 2, pp. 229-234.

[13] A. Braham, A. Lahyani, P. Venet, and N. Rejeb, "Recent developments in fault detection and power loss estimation of electrolytic capacitors," *IEEE Transactions on Power Electronics*, vol. 25, pp. 33-43, Jan. 2010.

[14] M. Venturini, "A new sine wave in sine wave out conversion technique eliminates reactive elements," in *Proceedings of the Solid-State Power Conversion Conference (POWERCON 7)*, San Diego, CA, 1980, pp. E3/1 - E3/15.

[15] D. Balakrishnan and R. Balog, "Capacitor-less VAR compensator based on matrix converter," in *Proceedings of the IEEE North American Power Symposium (NAPS 2010)*, Arlington, TX, 2010.

[16] L. Gyugyi and B. R. Pelly, *Static power frequency changers: Theory, Performance and Application*. New York: Wiley, 1976.

[17] P. W. Wheeler, J. Rodriguez, J. C. Clare, L. Empringham, and A. Weinstein, "Matrix converters: a technology review," *IEEE Transactions on Industrial Electronics*, vol. 49, pp. 276-288, Apr. 2002.

[18] A. Alesina and M. Venturini, "Solid-state power conversion: a fourier analysis approach to generalized transformer synthesis," *IEEE Transactions on Circuits and Systems*, vol. 28, pp. 319-330, Apr. 1981.

[19] A. Alesina and M. Venturini, "Intrinsic amplitude limits and optimum design of 9-switches direct PWM AC-AC converters," in *Recordings of the IEEE Power Electronics Specialists Conference (PESC '88)*, Kyoto, Japan, 1988, vol. 2, pp. 1284-1291.

[20] P. D. Ziogas, S. I. Khan, and M. H. Rashid, "Analysis and design of forced commutated cycloconverter structures with improved transfer characteristics,"

IEEE Transactions on Industrial Electronics, vol. IE-33, pp. 271-280, Aug. 1986.

[21] L. Huber and D. Borojevic, "Space vector modulator for forced commutated cycloconverters," in *Recordings of the IEEE Industry Applications Society Annual Meeting*, San Diego, CA, 1989, vol. 1, pp. 871-876.

[22] L. Huber, D. Borojevic, and N. Burany, "Analysis, design and implementation of the space-vector modulator for forced-commutated cycloconvertors," *IEE Proceedings B of Electric Power Applications*, vol. 139, pp. 103-113, Mar. 1992.

[23] D. Casadei, G. Grandi, G. Serra, and A. Tani, "Space vector control of matrix converters with unity input power factor and sinusoidal input/output waveforms," in *Proceedings of the Fifth European Conference on Power Electronics and Applications*, Brighton, 1993, vol. 7, pp. 170-175.

[24] D. Casadei, G. Serra, A. Tani, and L. Zarri, "Matrix converter modulation strategies: a new general approach based on space-vector representation of the switch state," *IEEE Transactions on Industrial Electronics*, vol. 49, pp. 370-381, Apr. 2002.

[25] M. Hornkamp, M. Loddenkötter, M. Münzer, O. Simon, and M. Bruckmann, "EconoMAC the first all-in-one IGBT module for matrix converters," in *Proceedings of the Drivers and Controls and Power Electronics Conference*, London, UK, 2001, pp. 35-39.

[26] R. W. Erickson and D. Maksimovic, *Fundamentals of power electronics*. New York: Kluwer Academic Publishers, 2001.

[27] S. Hongwu, L. Hua, W. Xingwei, and Y. Limin, "Damped input filter design of matrix converter," in *Proceedings of the International Conference on Power Electronics and Drive Systems (PEDS 2009)*, Taipei, 2009, pp. 672-677.

[28] A. Dasgupta, P. Tripathy, and P. S. Sensarma, "Matrix converter as UPFC for transmission line compensation," in *Proceedings of the International Conference on Power Electronics (ICPE '07)*, Daegu, 2007, pp. 1050-1055.

[29] C. L. Neft and C. D. Schauder, "Theory and design of a 30-hp matrix converter," *IEEE Transactions on Industry Applications*, vol. 28, pp. 546-551, May/Jun. 1992.

[30] H. J. Cha, "Analysis and design of matrix converters for adjustable speed drives and distributed power sources," in Ph.D. dissertation, Texas A&M University, College Station, TX, 2004.

[31] S. Ratanpanachote, "Applications of an electronic transformer in a power distribution system," in Ph.D. dissertation, Texas A&M University, College Station, TX, 2004.

APPENDIX A

(a) MATLAB Code to show VAR compensation:

```
%----------------------------------------------------------------
% PROGRAM TO MODEL VAR COMPENSATOR AS APPLIED TO (AC
NETWORK+LOAD) SYSTEM
%----------------------------------------------------------------


global dt Tsim t
global ncycles n
global Vutil_LL_rms Vutil_LN_pk
global V1util_LN V2util_LN V3util_LN
global futil wutil
global fsw
global Vi1_MC_LN Vi2_MC_LN Vi3_MC_LN
global Vo1_MC_LN Vo2_MC_LN Vo3_MC_LN
global Vo1_MC_LN_fund Vo2_MC_LN_fund Vo3_MC_LN_fund
global io1_MC io2_MC io3_MC
global io1_MC_fund io2_MC_fund io3_MC_fund
global ii1_MC ii2_MC ii3_MC
global ii1_MC_fund ii2_MC_fund ii3_MC_fund


%----------------------------------------------------------------
% Time set-up of simulation
%----------------------------------------------------------------
dt = (1/60)*1e-5;           % Time step
ncycles = 2;                % No. of 60Hz fundamental cycles
n = ncycles*2;              % No. of 120Hz switching cycles
```

```
Tsim = ncycles*(1/60);      % Total simulation time
t = 0:dt:Tsim-dt;           % Time vector

%-------------------------------------------------------------
% Setting up the ac network
%-------------------------------------------------------------
Vutil_LL_rms = 480*1000;    % Line-Line RMS ac network voltage
[V]

% Line-Neutral Peak ac network voltage [V]
Vutil_LN_pk = (2^0.5)/(3^0.5) * Vutil_LL_rms;

futil = 60;                 % Fundamental frequency [Hz]
wutil = 2*pi*futil;         % Angular fundamental frequency
                            % [rad/s]

% Line-Neutral ac network voltages (varying with time)
V1util_LN = Vutil_LN_pk * cos(wutil*t);
                            % Taken as 0 reference for
                            % phasor representation
V2util_LN = Vutil_LN_pk * cos(wutil*t - 2*pi/3);
V3util_LN = Vutil_LN_pk * cos(wutil*t + 2*pi/3);
%-------------------------------------------------------------
% Setting up the load: P = 10kW, Q = 7.5kVAR
%-------------------------------------------------------------

Rload = 14.7458;
Lload = 29.3358e-3;
```

```
phiload = atan(Lload*wutil/Rload);

Zload = (Rload^2 + (Lload*wutil)^2)^0.5;


% Determine the currents drawn by the load

iload1 = (Vutil_LN_pk/Zload) * cos(wutil*t - phiload);

iload2 = (Vutil_LN_pk/Zload) * cos(wutil*t - 2*pi/3 - phiload);

iload3 = (Vutil_LN_pk/Zload) * cos(wutil*t + 2*pi/3 - phiload);


%Expected reactive power drawn by load

Pload_expected = (Vutil_LL_rms^2/Zload)*cos(phiload);

Qload_expected = (Vutil_LL_rms^2/Zload)*sin(phiload);


% Defining the load on the MC

% (3-phase 4-wire balanced inductive Y-Load [H])

Lmc_perphase = 0.1169;


% Determination of required modulation index 'q' of the

% compensator to maintain unity power factor of the entire

% system

q = ((Qload_expected*Lmc_perphase*wutil)^0.5)/Vutil_LL_rms;


% Calling function ‘MC’ to calculate the matrix converter

% operation

MC(q,Lmc_perphase);



% Total ac network currents
```

```
iutil1 = iload1 + ii1_MC;

iutil2 = iload2 + ii2_MC;

iutil3 = iload3 + ii3_MC;


% Fundamental components of total ac network currents

iutil1_fund = iload1 + ii1_MC_fund;

iutil2_fund = iload2 + ii2_MC_fund;

iutil3_fund = iload3 + ii3_MC_fund;


% -------------------- PLOTS --------------------------------

% Define custom-plot functions using global variables
```

(b) MATLAB Code for the function ‘MC’:

```
%----------------------------------------------------------------

% FUNCTION TO MODEL STATIC VAR COMPENSATOR BASED ON 3/3 MATRIX

% CONVERTER FOR USER-DEFINED MODULATION INDEX 'q'

%----------------------------------------------------------------


function [] = MC(q,Lmc_perphase)


global dt Tsim t

global ncycles n

global Vutil_LL_rms Vutil_LN_pk

global V1util_LN V2util_LN V3util_LN

global futil wutil

global fsw

global Vi1_MC_LN Vi2_MC_LN Vi3_MC_LN

global Vo1_MC_LN Vo2_MC_LN Vo3_MC_LN
global Vo1_MC_LN_fund Vo2_MC_LN_fund Vo3_MC_LN_fund
global io1_MC io2_MC io3_MC
global io1_MC_fund io2_MC_fund io3_MC_fund
global ii1_MC ii2_MC ii3_MC
global ii1_MC_fund ii2_MC_fund ii3_MC_fund

%----------------------------------------------------------------
% Setting up the MC parameters
%----------------------------------------------------------------
fi = 60;                    % Input frequency (Hz) of MC
fo = 60;                    % MC Output frequency (Hz)
fmc = fi + fo;              % Modulation function frequency =
                            % output+input frequencies of MC (Hz)
wmc = 2*pi*120;             % Modulation function angular
                            % frequency [rad/s]

%----------------------------------------------------------------
% Modulation, Switching and Switch-Averaged Functions
%----------------------------------------------------------------
% Modulation Functions H1, H2, H3
H1 = 1/3 + (2/3)*q*cos(wmc*t);
H2 = 1/3 + (2/3)*q*cos(wmc*t - 2*pi/3);
H3 = 1/3 + (2/3)*q*cos(wmc*t + 2*pi/3);

% ---- Derivation of switching & switch-averaged functions -----

% (a) Switching frequency
Tmc = 1/fmc;                % Modulation function period (s)
```

```
Nsw = 100;                 % Number of switching periods
                           % per modulation function period
Tsw = Tmc/Nsw;             % Switching period (s)
fsw = Nsw*fmc;             % Switching frequency (Hz)

% (b) ON times of the 3 basic switching functions for each
% switching period within a modulation function period
k = 0:Nsw-1;
t1 = (Tsw/3)*(1 + 2*q*cos(wmc*k*Tsw));
t2 = (Tsw/3)*(1 + 2*q*cos(wmc*k*Tsw - 2*pi/3));
t3 = (Tsw/3)*(1 + 2*q*cos(wmc*k*Tsw + 2*pi/3));

% (c) Defining ON times in terms of time step 'dt'
n_Tsw = round(Tsw/dt);
n_t1 = round(t1/dt);
n_t2 = round(t2/dt);
n_t3 = n_Tsw - n_t1 - n_t2;

% (d) Switching & switch-averaged functions over 1 switching
% period Tmc
% Switching functions
S1 = 0;                    % Initializing to 0
S2 = 0;
S3 = 0;
% Switch-averaged functions
Sw_avg1 = 0;               % Initializing to 0
Sw_avg2 = 0;
Sw_avg3 = 0;
```

```
for i = 1:Nsw;
    S1 = [S1 ones(1,n_t1(i)) zeros(1,n_t2(i)) zeros(1,n_t3(i))];
    S2 = [S2 zeros(1,n_t1(i)) ones(1,n_t2(i)) zeros(1,n_t3(i))];
    S3 = [S3 zeros(1,n_t1(i)) zeros(1,n_t2(i)) ones(1,n_t3(i))];
    Sw_avg1 = [Sw_avg1 t1(i)/Tsw * ones(1,n_Tsw)];
    Sw_avg2 = [Sw_avg2 t2(i)/Tsw * ones(1,n_Tsw)];
    Sw_avg3 = [Sw_avg3 t3(i)/Tsw * ones(1,n_Tsw)];
end

S1 = S1(2:length(S1));      % Discarding initial 0
S2 = S2(2:length(S2));
S3 = S3(2:length(S3));
Sw_avg1 = Sw_avg1(2:length(Sw_avg1));
Sw_avg2 = Sw_avg2(2:length(Sw_avg2));
Sw_avg3 = Sw_avg3(2:length(Sw_avg3));

% (e) Switching & switch-averaged functions over 'n' switching
% periods
S1n = 0;                    % Initializing to 0
S2n = 0;
S3n = 0;
Sw_avg1n = 0;
Sw_avg2n = 0;
Sw_avg3n = 0;

for i = 1:n;
    S1n = [S1n S1];
    S2n = [S2n S2];
    S3n = [S3n S3];
    Sw_avg1n = [Sw_avg1n Sw_avg1];
    Sw_avg2n = [Sw_avg2n Sw_avg2];
    Sw_avg3n = [Sw_avg3n Sw_avg3];
end

S1n = S1n(2:length(S1n));  % Discarding initial 0
S2n = S2n(2:length(S2n));
S3n = S3n(2:length(S3n));
Sw_avg1n = Sw_avg1n(2:length(Sw_avg1n));
Sw_avg2n = Sw_avg2n(2:length(Sw_avg2n));
Sw_avg3n = Sw_avg3n(2:length(Sw_avg3n));

%----------------------------------------------------------------
% Output voltages of MC
%----------------------------------------------------------------

% Math-derived line-neutral output voltages of MC: Using
% H1,H2,H3
Vo1_MC_LN_math = H1.*Vi1_MC_LN + H2.*Vi2_MC_LN + H3.*Vi3_MC_LN;
Vo2_MC_LN_math = H2.*Vi1_MC_LN + H3.*Vi2_MC_LN + H1.*Vi3_MC_LN;
Vo3_MC_LN_math = H3.*Vi1_MC_LN + H1.*Vi2_MC_LN + H2.*Vi3_MC_LN;

% Switched line-neutral output voltages of MC: Using S1n,S2n,S3n
Vo1_MC_LN = S1n.*Vi1_MC_LN + S2n.*Vi2_MC_LN + S3n.*Vi3_MC_LN;
Vo2_MC_LN = S2n.*Vi1_MC_LN + S3n.*Vi2_MC_LN + S1n.*Vi3_MC_LN;
Vo3_MC_LN = S3n.*Vi1_MC_LN + S1n.*Vi2_MC_LN + S2n.*Vi3_MC_LN;
```

```
% Switch-averaged line-neutral output voltages of MC:
% Using Sw_avg1n,Sw_avg2n,Sw_avg3n
Vo1_MC_LN_swavg = Sw_avg1n.*Vi1_MC_LN + Sw_avg2n.*Vi2_MC_LN ...
    + Sw_avg3n.*Vi3_MC_LN;
Vo2_MC_LN_swavg = Sw_avg2n.*Vi1_MC_LN + Sw_avg3n.*Vi2_MC_LN ...
    + Sw_avg1n.*Vi3_MC_LN;
Vo3_MC_LN_swavg = Sw_avg3n.*Vi1_MC_LN + Sw_avg1n.*Vi2_MC_LN ...
    + Sw_avg2n.*Vi3_MC_LN;

%----------------------------------------------------------------
% Output currents of MC
%----------------------------------------------------------------
% Initialization of currents based on steady state values
io1_MC_0 = (q*Vi_LN_MC_pk/(wi*Lmc_perphase))*cos(-pi/2);
io2_MC_0 = (q*Vi_LN_MC_pk/(wi*Lmc_perphase))*cos(-pi/2-2*pi/3);
io3_MC_0 = (q*Vi_LN_MC_pk/(wi*Lmc_perphase))*cos(-pi/2+2*pi/3);

% Time Integration of voltage across the load inductor
io1_MC = (1/Lmc_perphase)*cumtrapz(t,Vo1_MC_LN) + io1_MC_0;
io2_MC = (1/Lmc_perphase)*cumtrapz(t,Vo2_MC_LN) + io2_MC_0;
io3_MC = (1/Lmc_perphase)*cumtrapz(t,Vo3_MC_LN) + io3_MC_0;

%----------------------------------------------------------------
% Switched input currents of MC: Using S1n,S2n,S3n
%----------------------------------------------------------------
ii1_MC = S1n.*io1_MC + S2n.*io2_MC + S3n.*io3_MC;
ii2_MC = S2n.*io1_MC + S3n.*io2_MC + S1n.*io3_MC;
```

```
ii3_MC = S3n.*io1_MC + S1n.*io2_MC + S2n.*io3_MC;

%-----------------------------------------------------------------
% Fundamental components of voltages and currents
%-----------------------------------------------------------------
% Use function 'freqdom':
% [fund_mag fund_ph] = freqdom(signal,sampling time)

% Output voltages
[Vo1_MC_LN_fundmag Vo1_MC_LN_fundph] = freqdom(Vo1_MC_LN,dt);
[Vo2_MC_LN_fundmag Vo2_MC_LN_fundph] = freqdom(Vo2_MC_LN,dt);
[Vo3_MC_LN_fundmag Vo3_MC_LN_fundph] = freqdom(Vo3_MC_LN,dt);

Vo1_MC_LN_fund = Vo1_MC_LN_fundmag*cos(wi*t + Vo1_MC_LN_fundph);
Vo2_MC_LN_fund = Vo2_MC_LN_fundmag*cos(wi*t + Vo2_MC_LN_fundph);
Vo3_MC_LN_fund = Vo3_MC_LN_fundmag*cos(wi*t + Vo3_MC_LN_fundph);

% Output currents
[io1_MC_fundmag io1_MC_fundph] = freqdom(io1_MC,dt);
[io2_MC_fundmag io2_MC_fundph] = freqdom(io2_MC,dt);
[io3_MC_fundmag io3_MC_fundph] = freqdom(io3_MC,dt);

io1_MC_fund = io1_MC_fundmag*cos(wo*t + io1_MC_fundph);
io2_MC_fund = io2_MC_fundmag*cos(wo*t + io2_MC_fundph);
io3_MC_fund = io3_MC_fundmag*cos(wo*t + io3_MC_fundph);


% Input currents
```

```
[ii1_MC_fundmag ii1_MC_fundph] = freqdom(ii1_MC,dt);
[ii2_MC_fundmag ii2_MC_fundph] = freqdom(ii2_MC,dt);
[ii3_MC_fundmag ii3_MC_fundph] = freqdom(ii3_MC,dt);

ii1_MC_fund = ii1_MC_fundmag*cos(wi*t + ii1_MC_fundph);
ii2_MC_fund = ii2_MC_fundmag*cos(wi*t + ii2_MC_fundph);
ii3_MC_fund = ii3_MC_fundmag*cos(wi*t + ii3_MC_fundph);

%---------------------------------------------------------------
% Real & Reactive Power drawn by the MC from the ac network at
% fundamental frequency
%---------------------------------------------------------------
% Per-phase and total real power drawn by the MC
P1_MC = 0.5 * Vutil_LN_pk * ii1_MC_fundmag * ...
    cos(0-ii1_MC_fundph);
P2_MC = 0.5 * Vutil_LN_pk * ii2_MC_fundmag * ...
    cos(-2*pi/3-ii2_MC_fundph);
P3_MC = 0.5 * Vutil_LN_pk * ii3_MC_fundmag * ...
    cos(2*pi/3-ii3_MC_fundph);
P_MC = P1_MC + P2_MC + P3_MC;

% Per-phase and total reactive power drawn by the MC
% Lagging VARs: +ve, Leading VARs: -ve
Q1_MC = 0.5 * Vutil_LN_pk * ii1_MC_fundmag * ...
    sin(0-ii1_MC_fundph);
Q2_MC = 0.5 * Vutil_LN_pk * ii2_MC_fundmag * ...
    sin(-2*pi/3-ii2_MC_fundph);
Q3_MC = 0.5 * Vutil_LN_pk * ii3_MC_fundmag * ...
```

```
    sin(2*pi/3-ii3_MC_fundph);
Q_MC = Q1_MC + Q2_MC + Q3_MC;
Q_MC_theor = -1.5*((q*Vutil_LN_pk)^2)/(wutil*Lmc_perphase);
```

(c) MATLAB Code for the function ‘freqdom’ to determine the fundamental components of MC voltages & currents:

```
function [fund_mag, fund_ph] = freqdom(signal,timestep)
Fs = 1/timestep;            % Sampling frequency
x = signal;                 % Time domain signal
L = length(x);              % Length of signal
F = fft(x);

% Positive spectrum 0:L/2 or 0:(L-1)/2
if(mod(L,2)==0)
    k = L/2;
else
    k = (L-1)/2;
end
F = F(1:(k+1));             % Positive spectrum
FFT_mag = abs(F)/(L/2);     % Magnitude of spectrum with
                            % de-scaling
FFT_ph = angle(F);          % Phase of spectrum in radians
freq = (Fs/(2*k))*(0:k);    % Frequency axis
i_fund = 60*k/(Fs/2) + 1;   % Index corresponding to fundamental
fund_mag = FFT_mag(i_fund);
fund_ph = FFT_ph(i_fund);
```

APPENDIX B

The software version used is PSIM Professional Network Version 8.0.3.400.

(a) The PSIM element 'Simplified C block' has been used to generate the basic three switching functions – Refer to Fig. 33

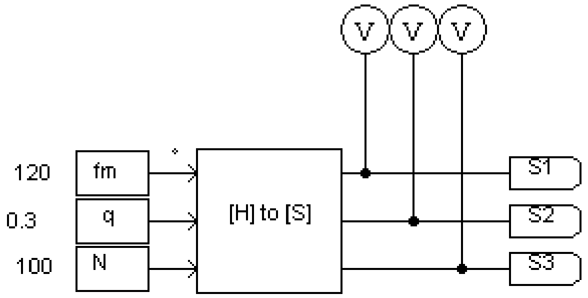


Fig. 67. PSIM C block used for generation of S_1, S_2 and S_3

Content of the C block:

```
int K, K1, H1, H2, H3, N;
double time, fm, Tseq, q, pi, t1, t2, t3, wm, tm, phi;

pi = 3.1459;

{
      fm = x1;
      tm = 1/fm;
      N = x3;
      q = x2;
      wm = 2*pi*fm;
      Tseq = 1/(N*fm);
      phi = 0;
}

{
      time = t;
      K = floor(time/Tseq);
      K1 = K % N;
}

{
      t1 = (Tseq/3) * (1 + 2*q*cos(K1*Tseq*wm - phi));
      t2 = (Tseq/3) * (1 + 2*q*cos(K1*Tseq*wm - 2*pi/3 - phi));
      t3 = (Tseq/3) * (1 + 2*q*cos(K1*Tseq*wm + 2*pi/3 - phi));
}

{
      if( t < (K*Tseq + t1) )
            H1 = 1;
      else
            H1 = 0;
      if( t >= (K*Tseq + t1) && t < (K * Tseq + t1 + t2) )
            H2 = 1;
      else
            H2 = 0;
      if( t >= (K * Tseq + t1 + t2) && t < ((K+1) * Tseq) )
            H3 = 1;
      else
            H3 = 0;
}

 y1 = H1;
 y2 = H2;
 y3 = H3;
```

(b) PSIM Logic used for four-step commutation: Refer to Fig. 49

– The basic three switching functions are used to derive the 18 switching signals S11P, S11N, S12P, S12N etc.

– ‘CLK’ has a period of 200 ns, while ENA has a period equal to 2200 ns.

– The signal ‘Sw_curr’ is 1 when current flows through the corresponding switch and is 0 otherwise. It is obtained from the sensed output currents using appropriate ‘simplified C code’ and is synchronized with the signal ENA

– The ‘Edge Detector’ uses the simplified C block to detect the rising and falling edges of the switching functions

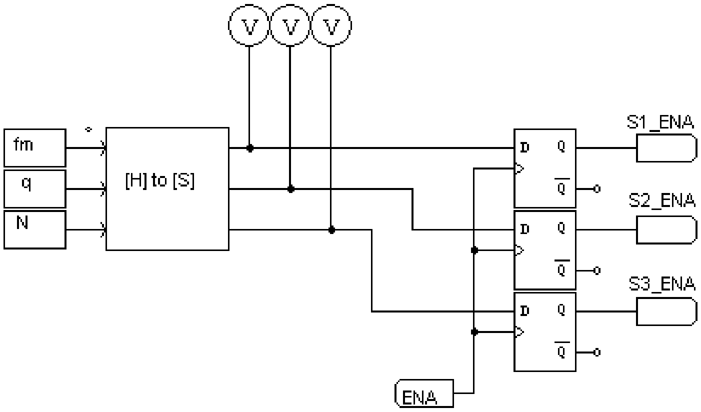


Fig. 68. Generation of the basic three switching functions

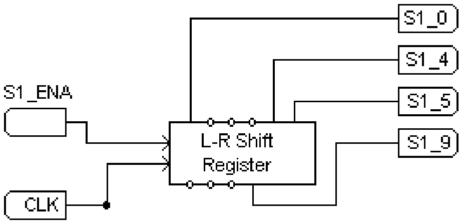


Fig. 69. Generation of delayed versions of the switching functions

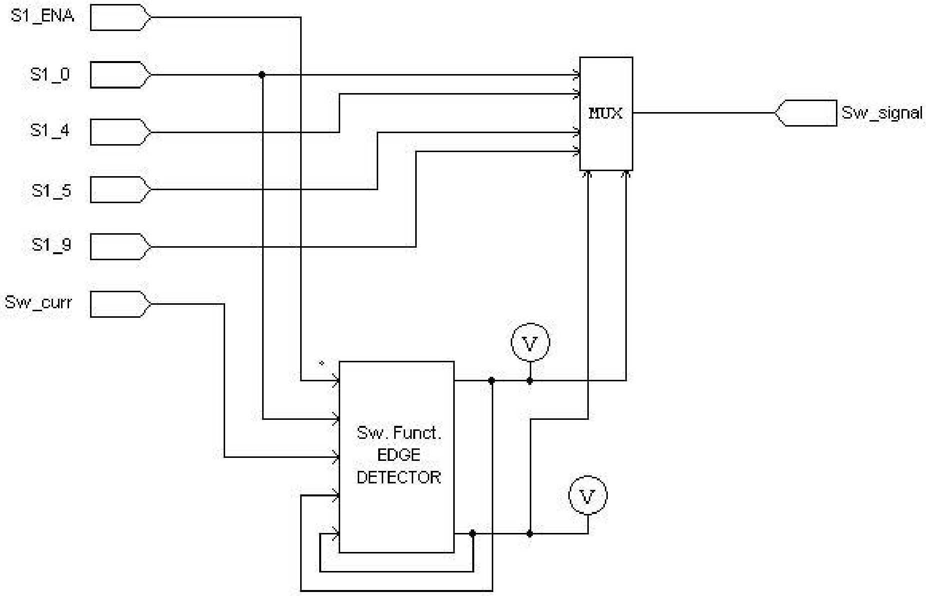


Fig. 70. Generation of the switching signal

APPENDIX C

a) DSP Code : Generation of switching functions S1, S2, S3 and generation of output current directions

```
/* Header file for register declarations */
#include "regs2407.h"

/* Look-up tables: ON-times ‘t1’ and ‘t3’ */
int n_t1[100] = {
      2000,1998,1993,1984,1972,1956,1938,1915,1890,1862,

      1830,1796,1759,1720,1678,1634,1587,1539,1490,1438,

      1386,1332,1278,1223,1167,1111,1055,1000,945,890,

      836,784,733,683,635,589,545,503,463,426,

      392,361,332,307,285,266,250,238,229,224,

      222,224,229,238,250,266,285,307,332,361,

      392,426,463,503,545,589,635,683,733,784,

      836,890,945,1000,1055,1111,1167,1223,1278,1332,

      1386,1438,1490,1539,1587,1634,1678,1720,1759,1796,

      1830,1862,1890,1915,1938,1956,1972,1984,1993,1998 };

int n_t3[100] = {
      2667,2714,2760,2803,2844,2882,2919,2952,2982,3010,

      3034,3055,3073,3088,3099,3107,3110,3111,3108,3101,

      3092,3078,3062,3042,3018,2992,2962,2931,2896,2857,

      2817,2774,2730,2683,2634,2584,2533,2480,2425,2370,

      2315,2260,2203,2148,2093,2038,1983,1930,1878,1827,

      1778,1730,1685,1641,1600,1562,1526,1492,1462,1435,

      1410,1389,1371,1357,1346,1339,1334,1334,1337,1343,

      1352,1366,1383,1403,1426,1452,1482,1515,1550,1587,

      1628,1670,1715,1761,1810,1861,1913,1966,2019,2074,
      2129,2185,2241,2296,2352,2407,2461,2514,2567,2617 };

int k = 0;
int m = 0;

/* Dummy function to trap spurious interrupts */
void dummy_int(void)
{
      while(1)
      {
            WDKEY = 85;             /*0x0055h; */
            WDKEY = 170;            /*0x00AAh;*/
      }
}

/* Function to service the T1CINT interrupt */
interrupt void timer_int(void)
{

      if(k<99)                      /* For N pulses, k<(N-1) */
      {
            k = k + 1;
            PEDATDIR = 65527;       /* FFF7h - OFF for k<>0 */
      }
      else
      {
            k = 0;
            PEDATDIR = 65535;       /* FFFFh - ON for k = 0 */
      }

      T1CMPR = n_t1[k];             /* Updating new on-times
                                    (k=0 to 100) of S1,S2,S3 */
      T2CMPR = n_t3[k];

      EVAIFRA = 65535;              /* FFFFh - Reset all EVA
                                    GroupA flags */
      asm(" clrc INTM");
      return;
}

/* Function to service the T2CINT interrupt */
interrupt void curr_int(void)
{
    if(m<199)
      m = m + 1;
    else
      m = 0;

    if(m>=0 && m<17)
      PADATDIR = 65390;     /* FF6Eh */
    else if(m==17)
      PADATDIR = 65382;     /* FF66h */
    else if(m>17 && m<50)
```

```
     PADATDIR = 65398;      /* FF76h */
    else if(m==50)
     PADATDIR = 65366;      /* FF56h */
    else if(m>50 & m<83)
     PADATDIR = 65367;      /* FF57h */
    else if(m==83)
     PADATDIR = 65303;      /* FF17h */
    else if(m>83 & m<117)
     PADATDIR = 65431;      /* FF97h */
    else if(m==117)
     PADATDIR = 65415;      /* FF87h */
    else if(m>117 & m<150)
     PADATDIR = 65423;      /* FF8Fh */
    else if(m==150)
     PADATDIR = 65422;      /* FF8Eh */
    else if(m>150 & m<183)
     PADATDIR = 65454;      /* FFAEh */
    else if(m==183)
     PADATDIR = 65326;      /* FF2Eh */
    else if(m>183 & m<200)
     PADATDIR = 65390;      /* FF6Eh */

    EVAIFRB = 65535;        /* FFFFh - Reset all EVA
                            GroupA flags */
    asm(" clrc INTM");
    return;
}

/* Main Function */
void main(void)
{
/* ------------Initialization---------------- */
     asm(" setc INTM");     /* Disable interrupts */
     asm (" clrc CNF");     /* Configure block B0 to
                            data memory */
     IMR = 0;               /* 0000h - Mask interrupts at
                            core level */
     IFR = 65535;           /* FFFFh - Clear interrupt flags
                            at core level */
     WDCR = 111;            /* 006Fh - Disable WD Timer */
     SCSR1 = 13;            /* 000Dh - Clear Illegal Addr bit
                            & Enable EVA,EVB clock inputs */
     WDKEY = 85;            /* 0055h - Reset WD counter */
     WDKEY = 170;           /* 00AAh */
     WSGR = 0;              /* 0000h - Set wait-state generator
                            for - 0 wait states */
     MCRA = 65535;          /* FFFFh - Assign primary
                            functionality to PortA/B Pins */
     MCRB = 65535;          /* FFFFh - Assign primary
                            functionality to PortC/D Pins */
     MCRC = 65535;          /* FFFFh - Assign primary
                            functionality to PortE/F Pins */

/* ------------End of Initialization---------- */
/* ----------Start of Actual Code------------- */

     /* Timer 1 - generates S1              at P1, pin 12 */
     /* Timer 2 - generates S3              at P1, pin 13 */
     /* Timer 4 - generates CPLD clock      at P4, pin 28 */

     /* Configuration of GPTCON A/B for timers 1,2,3,4 */
     GPTCONA = 73;         /* 0049h - Enable compare outputs,
                           T2 active high,
                           T1 active low operation */
     GPTCONB = 70;         /* 0046h - Enable compare outputs,
                           T4 active low,
                           T3 active high operation */


     /* Set timer period registers */
     T1PR = 3332;          /* Tdisc/DSP_clk -1 = 40M/12k - 1 */
     T2PR = 3332;
     T4PR = 3;             /* CPLD_clk/DSP_clk - 1
                           = 100n/25n - 1 */

     /* Reset all timer counter registers */
     T1CNT = 0;
     T2CNT = 0;
     T4CNT = 0;

     /* Set Timer 4 compare value & enable operation */
     T4CMPR = 2;           /* On-time = 50% time period */

     T4CON = 4174;         /* 104Eh - Enable compare,
                           up-counting mode, enable timer */

     /* Timers 1 & 2 compare interrupts set-up */
     EVAIMRA = 0;          /* 0000h - Disable all EVA
                           GroupA interrupts */
     EVAIFRA = 65535;      /* FFFFh - Reset all EVA
                           GroupA flags */
     EVAIMRB = 0;
     EVAIFRB = 65535;

     EVAIMRA = 256;        /* 0100h - Enable T1CINT */
     EVAIMRB = 2;          /* 0002h - Enable T2CINT */
     IMR = 6;              /* 0006h - Unmask INT2(T1CINT),
                           INT3(T2CINT) level interrupts */
     asm(" clrc INTM");    /* Clear INTM bit */

     /* S1 & S3 generation */

     T1CMPR = n_t1[k];     /* Setting initial on-times (k=0)
                           of S1, S2, S3 */
     T2CMPR = n_t3[k];
```

```
      T2CON = 4294;          /* 10C6h - Start T2 with T1, all T1
                             features */
      T1CON = 4166;          /* 1046h - Enable compare, compare
                             reload condition, up-counting mode,
                             enable timer */

      /* Tracking k = 0 */
      MCRC = 65527;          /* FFF7h - IOPE3 (P1, pin 11) */
      PEDATDIR = 65535;      /* FFFFh - ON for k = 0 */

      MCRA = 65286;          /* FF06h - IOPA 7-6 4-3 5-0 */
      PADATDIR = 65390;      /* FF6Eh */


while(1)
      {
            ;
      }

}
```

b) VHDL code for state machine : ‘perphase_commutn_blk’ (refer to Fig. 53)

```
library ieee;
use ieee.std_logic_1164.all;

entity perphase_commutn_blk is port(

      S1, S3      : in std_logic;  -- Basic 3 switching functions
      curr_pos    : in std_logic;  -- Output phase current
                                   -- 1 if curr>0, 0 otherwise
      curr_neg    : in std_logic;  -- 1 if curr<0, 0 otherwise
      commutn_clk: in std_logic;   -- CPLD Clock
                                   -- (freq = 25.175MHz)

      -- Positive & Negative switch signals governed by S1_blank
      S1P, S1N    : out std_logic;
      -- Positive & Negative switch signals governed by S2_blank
      S2P, S2N    : out std_logic;
      -- Positive & Negative switch signals governed by S3_blank
      S3P, S3N    : out std_logic );

end perphase_commutn_blk;

architecture behavioral of perphase_commutn_blk is
      signal present_state,next_state: integer range 0 to 27 :=0;
      signal edge_detect: integer range 0 to 4 :=0;
      signal wait_count: integer range 0 to 8 :=1;
      signal wait_clk: integer range 0 to 8 :=1;
      signal commutn_state: integer range 0 to 1 :=0;
      signal S1_prev,S3_prev: std_logic :='0';

begin
```

```
state_assign: process(present_state, edge_detect, curr_pos,
                      curr_neg)
begin
     case present_state is

           when 0     =>    S1P <= '0'; S1N <= '0';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            if (edge_detect=4) then
                                 next_state <= 1;
                            else
                                 next_state <= 0;
                            end if;
                            wait_clk <= 1;
                            commutn_state <= 0;

           when 1     =>    S1P <= '1'; S1N <= '1';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            next_state <= 25;
                            wait_clk <= 1;
                            commutn_state <= 1;

           when 2     =>    S1P <= '1'; S1N <= '0';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            next_state <= 5;
                            wait_clk <= 8;
                            commutn_state <= 1;

           when 3     =>    S1P <= '0'; S1N <= '1';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            next_state <= 6;
                            wait_clk <= 8;
                            commutn_state <= 1;

           when 4     =>    S1P <= '0'; S1N <= '0';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            next_state <= 9;
                            wait_clk <= 8;
                            commutn_state <= 1;

           when 5     =>    S1P <= '1'; S1N <= '0';
                            S2P <= '1'; S2N <= '0';
                            S3P <= '0'; S3N <= '0';

                            next_state <= 7;
```

```
                 wait_clk <= 2;
                 commutn_state <= 1;

when 6     =>    S1P <= '0'; S1N <= '1';
                 S2P <= '0'; S2N <= '1';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 8;
                 wait_clk <= 2;
                 commutn_state <= 1;

when 7     =>    S1P <= '0'; S1N <= '0';
                 S2P <= '1'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 9;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 8     =>    S1P <= '0'; S1N <= '0';
                 S2P <= '0'; S2N <= '1';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 9;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 9     =>    S1P <= '0'; S1N <= '0';
                 S2P <= '1'; S2N <= '1';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 26;
                 wait_clk <= 1;
                 commutn_state <= 1;

when 10    =>    S1P <= '0'; S1N <= '0';
                 S2P <= '1'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 13;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 11    =>    S1P <= '0'; S1N <= '0';
                 S2P <= '0'; S2N <= '1';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 14;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 12    =>    S1P <= '0'; S1N <= '0';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';
```

```
                    next_state <= 17;
                    wait_clk <= 8;
                    commutn_state <= 1;

when 13    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '1'; S2N <= '0';
                    S3P <= '1'; S3N <= '0';

                    next_state <= 15;
                    wait_clk <= 2;
                    commutn_state <= 1;

when 14    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '1';
                    S3P <= '0'; S3N <= '1';

                    next_state <= 16;
                    wait_clk <= 2;
                    commutn_state <= 1;

when 15    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '0';
                    S3P <= '1'; S3N <= '0';

                    next_state <= 17;
                    wait_clk <= 8;
                    commutn_state <= 1;

when 16    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '0';
                    S3P <= '0'; S3N <= '1';

                    next_state <= 17;
                    wait_clk <= 8;
                    commutn_state <= 1;

when 17    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '0';
                    S3P <= '1'; S3N <= '1';

                    next_state <= 27;
                    wait_clk <= 1;
                    commutn_state <= 1;

when 18    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '0';
                    S3P <= '1'; S3N <= '0';

                    next_state <= 21;
                    wait_clk <= 8;
                    commutn_state <= 1;

when 19    =>       S1P <= '0'; S1N <= '0';
                    S2P <= '0'; S2N <= '0';
                    S3P <= '0'; S3N <= '1';
```

```
                 next_state <= 22;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 20    =>    S1P <= '0'; S1N <= '0';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 1;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 21    =>    S1P <= '1'; S1N <= '0';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '1'; S3N <= '0';

                 next_state <= 23;
                 wait_clk <= 2;
                 commutn_state <= 1;

when 22    =>    S1P <= '0'; S1N <= '1';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '1';

                 next_state <= 24;
                 wait_clk <= 2;
                 commutn_state <= 1;

when 23    =>    S1P <= '1'; S1N <= '0';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 1;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 24    =>    S1P <= '0'; S1N <= '1';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 next_state <= 1;
                 wait_clk <= 8;
                 commutn_state <= 1;

when 25    =>    S1P <= '1'; S1N <= '1';
                 S2P <= '0'; S2N <= '0';
                 S3P <= '0'; S3N <= '0';

                 if (edge_detect=1) then
                       if (curr_pos='1') then
                             next_state <= 2;
                       elsif (curr_neg='1') then
                             next_state <= 3;
```

```
                                 else
                                       next_state <= 4;
                                 end if;
                            else
                                 next_state <= 25;
                            end if;

                            wait_clk <= 1;
                            commutn_state <= 0;

           when 26    =>    S1P <= '0'; S1N <= '0';
                            S2P <= '1'; S2N <= '1';
                            S3P <= '0'; S3N <= '0';

                            if (edge_detect=2) then
                                 if (curr_pos='1') then
                                       next_state <= 10;
                                 elsif (curr_neg='1') then
                                       next_state <= 11;
                                 else
                                       next_state <= 12;
                                 end if;
                            else
                                 next_state <= 26;
                            end if;

                            wait_clk <= 1;
                            commutn_state <= 0;

           when 27    =>    S1P <= '0'; S1N <= '0';
                            S2P <= '0'; S2N <= '0';
                            S3P <= '1'; S3N <= '1';

                            if (edge_detect=3) then
                                 if (curr_pos='1') then
                                       next_state <= 18;
                                 elsif (curr_neg='1') then
                                       next_state <= 19;
                                 else
                                       next_state <= 20;
                                 end if;
                            else
                                 next_state <= 27;
                            end if;

                            wait_clk <= 1;
                            commutn_state <= 0;

     end case;

end process state_assign;

state_clocked: process(commutn_clk)
begin
```

```
          if rising_edge(commutn_clk) then
               if (wait_count=wait_clk) then
                    -- Moving to next state
                    present_state <= next_state;     .
                    wait_count <= 1;
               else
                    present_state <= present_state;
                    wait_count <= wait_count + 1;
               end if;

               -- Detecting falling/rising edges of input
               -- switching signals.
               if(commutn_state=1 and edge_detect/=0) then
                    edge_detect <= edge_detect;
               else
                    if(S1_prev='1' and S1='0') then
                    -- falling(S1_blank) & rising(S2_blank)
                         edge_detect <= 1;
                    elsif(S3_prev='0' and S3='1') then
                    -- falling(S2_blank) & rising(S3_blank)
                         edge_detect <= 2;
                    elsif(S3_prev='1' and S3='0') then
                    -- falling(S3_blank) & rising(S1_blank)
                         edge_detect <= 3;
                    elsif(S1_prev='0' and S1='1') then
                    -- only rising(S1_blank)
                         edge_detect <= 4;
                    else
                    -- no edge detected
                         edge_detect <= 0;
                    end if;
               end if;

               S1_prev <= S1;
               S3_prev <= S3;

          end if;

     end process state_clocked;
end behavioral;
```

VITA

Name: Divya Rathna Balakrishnan

Address: Department of Electrical Engineering
c/o Dr. Robert Balog
Texas A&M University,
College Station, TX 77843-3128

Email Address: divyarathna.b@gmail.com

Education: B.Tech., Electrical & Electronics Engineering,
National Institute of Technology, Karnataka, India, 2008

M.S., Electrical Engineering, Texas A&M University, 2010

Publications: D. Balakrishnan and R. Balog, "Capacitor-less VAR compensator based on matrix converter," in *Proceedings of the IEEE North American Power Symposium NAPS 2010*, Arlington, TX, 2010.

Job position: Electrical Engineer (Grade B)
Hatch Mott MacDonald Inc., Houston, TX