Matrix converter technology in doubly-fed induction generators for wind generators

Benjamin J. Harris
University of Wollongong

Matrix Converter Technology in Doubly-Fed Induction Generators for Wind Generators

A thesis submitted in fulfilment of the

requirements for the award of the degree

Master of Engineering (Research)

from

University of Wollongong

by

Benjamin J Harris, BE(ELEC)

School of Electrical, Computer and Telecommunications Engineering

April 2009

To the special people in my life. Thank you for your support.

Acknowledgements

I would like to express my sincere thanks to my parents, sister, family, friends and the princess for their constant support. They have been invaluable in providing me with the strength and confidence to pursue my dreams which has lead to this work coming to fruition. This I am sure will continue to be beneficial in a long and prosperous engineering career. I would also like to give a special thank you to Miss Eva Cheng for her help with this project. I send my best wishes to her in completing her own studies.

My thesis supervisors, Associate Professor Sarath Perera and Dr. Duane Robinson have given me a wealth of much needed advice and guidance throughout this study. This work would not have come to realisation without their tireless efforts and patience. I am highly appreciative of the extraordinary supervision that they have both provided and for this I must also express my deepest thanks.

Certification

I, Benjamin James Harris declare that this thesis, submitted in fulfilment of the requirements for the award of Master of Engineering (Research), in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is entirely my own work unless otherwise referenced or acknowledged. This manuscript has not been submitted for qualifications at any other academic institute.

Benjamin James Harris

Abstract

Wind generator technologies have been widely researched and documented. Modern wind generator systems are now being implemented with an output power of up to 5 MVA. They adopt power electronic control systems which allow the system to operate with a wide range of turbine shaft speeds, as well as with Maximum Power Point Tracking (MPPT) algorithms to maximise the energy delivery to the power system.

Despite the extensive development, maximum power transfer from the total available power in the wind gusts to the grid is limited by the mechanical and electrical losses in the generation process. A Doubly-Fed Induction Generator (DFIG) uses a two stage power converter process in the rotor circuit which contributes to the electrical losses in the system. By adopting a Matrix Converter (MC), the electrical losses may be reduced as the power conversation is conducted as a single stage process. This thesis presents existing variable speed generator technologies with a focus on DFIG systems. Based on this research, an alternative design for a DFIG control system using an MC is developed and analysed.

The work begins with the presentation of existing DFIG systems using back-to-back PWM converters connected between the rotor circuit and the grid. The study examines MC technology and the application of current commutation techniques to MC systems. A non-ideal MC model and commutation controller is developed in the PSCAD® / EMTDC^{TM} environment. Simulation and analysis are conducted on the MC system connected to a passive load to determine its suitability for application in DFIG systems.

The MC is connected to the rotor circuit of the DFIG system in the simulation and analysis is carried out to investigate the viability of the system. From this the MC excited DFIG is extended with the development of a hybrid Maximum Power Point Tracking (MPPT) algorithm. The wind generator system is tested using pseudo-random wind speeds with varying wind gusts. The results of the simulation are presented and compared with existing MPPT technologies to assess the overall performance of the system in relation to existing technologies.

Finally, the MC excited DFIG control system is adapted to provide reactive power compensation to the power system for the regulation of voltage in distribution networks. Testing of the system using a variable load connected to a common bus bar and single transmission line is conducted. Observations from the simulation show that VAr compensation from the DFIG system does reduce voltage fluctuations in power systems.

List of Principal Symbols

P, Q	Active, Reactive Power
v_d, v_q	Direct, Quadrature Voltage
i_d, i_q	Direct, Quadrature Current
λ	Flux linkage or Tip-speed ratio
i_{ms}	Stator magnetizing current
L_s, L_r	WRIM stator, rotor self-inductance
L_{ls}, L_{lr}	WRIM stator, rotor leakage inductance
L_m	WRIM magnetizing inductance
X_s, X_r	WRIM stator, rotor self-reactance
X_{ls}, X_{lr}	WRIM stator, rotor leakage reactance
X_m	WRIM magnetizing reactance
r_s, r_r	Stator, Rotor resistance
$\omega_e, \omega_r, \omega_{slip}$	Electrical, Rotor, Slip angular velocity
P_t	Mechanical Power output of a wind turbine
T_t	Mechanical Torque of a wind turbine
C_p	Coefficient of power of a wind turbine
ρ	Air density
R	Radius of a wind turbine
V_w	Wind velocity
β	Pitch angle
f_s, t_s	Switching frequency / time
d_x	Duty Cycle of SSV_x
P	Poles
p	Pole Pairs

Subscripts, Superscripts

d, q	Direct, Quadrature axis values
A, B, C	Utility grid phase values
a, b, c	Output phase values
s, r	stator, rotor
$'$	Stator referred value
$*$	Reference value
e	Synchronous value

Acronyms

2QSW	Two Quadrant Switch
4QSW	Four Quadrant Switch
AC	Alternating Current
ASG	Adjustable Speed Generator
DC	Direct Current
DFIG	Doubly-Fed Induction Generator
EMTDC	Electromagnetic Transient Program for Direct Current
FFT	Fast Fourier Transform
FIR	Finite Impulse Response (filter)
FSG	Fixed Speed Generator
HF	High Frequency
MC	Matrix Converter
MPPT	Maximum Power Point Tracking
PCC	Point of Common Coupling
PSCAD® / EMTDC^{TM}	Power Systems Computer Aided Design
PWM	Pulse Width Modulation

RMS	Root Mean Square
SSCC	Semi-Soft Commutation Controller
SSV	Stationary Switch Vector
STATCOM	Static Synchronous Compensator
SVC	Static VAr Compensator
SVM	Space Vector Modulation
SVMC	Space Vector Modulation Controller
THD	Total Harmonic Distortion
VSG	Variable Speed Generator
VSI	Voltage Source Inverter
VSR	Voltage Source Rectifier
VVVF	Variable Voltage / Variable Frequency
WRIM	Wound Rotor Induction Machine

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of the Problem

The Doubly Fed Induction Generator (DFIG) has been widely used and researched in wind generator applications. The primary benefit of this system over other generator configurations is that the power electronic converters in the system need only to convert power to and from the rotor windings of the Wound Rotor Induction Machine (WRIM). This translates to a converter power rating of approximately 25% of the total generator power rating. However, these power converters in a DFIG system usually rely on a back-to-back DC link configuration to produce the AC-AC conversion. This project is an investigation into the feasibility of using a Matrix Converter (MC) to conduct the AC-AC power conversion in the rotor circuit.

While past work such as [1] has investigated the use of an MC to excite the rotor in a DFIG, the research has been based on the use of ideal switching devices in the MC. When working with the assumption of ideal switching devices many of the complexities involved with non-ideal power electronic components are not explored. Hence the viability of constructing an MC excited DFIG has not definitively estab-

lished.

1.2 Thesis Objectives

In this project wind generator technologies and their operation are investigated. Wound Rotor Induction Machine (WRIM) modelling and analysis are introduced to provide a comprehensive study into the control of the Doubly Fed Induction Generator (DFIG). Associated control systems are explored with respect to their use in optimal power tracking systems. Also independent decoupled active and reactive power control in DFIG systems are researched and analysed.

The research focuses on the Matrix Converter (MC) and existing technologies available in relation to the modulation, control and capabilities of these power converters. Switching devices and commutation strategies will be investigated and critically reviewed. A non-ideal MC model and control system will be developed in a simulation environment such that the system can supply variable voltage, variable frequency output allowing power flow control to be observed.

The viability of using an MC in a DFIG system is explored by implementing the MC model into the rotor circuit of a WRIM model to create a complete DFIG model. Testing using a number of conditions will be conducted to assess the overall performance of the developed system. The results of which will be critically analysed and from this conclusions and recommendations made.

1.3 Methodologies

Firstly, the matrix converter (MC) and modulation strategies are simulated using ideal components in MATLAB®. The system is assessed and is then simulated in the PSCAD® / EMTDC^{TM} simulation environment to represent the behaviour of non-ideal components. A current commutation controller is developed for use in PSCAD® / EMTDC^{TM} to ensure that the MC is able to function effectively. Power flow control in an MC is simulated in PSCAD® / EMTDC^{TM} and analysis is conducted by exporting the results into MATLAB® for post processing.

The WRIM model is tested in PSCAD® / EMTDC^{TM} using free acceleration tests. A pump load is also applied to the machine to illustrate the characteristics of the machine under loaded conditions. From this the DFIG control theory investigated in the literature review is evaluated using an ideal voltage source in the rotor circuit of the WRIM. This is extended by applying the MC model to the rotor circuit of the WRIM model replacing the ideal voltage sources thus creating a complete DFIG model. The model is then simulated under a variety of conditions and the results are analysed using Fast Fourier Transforms to determine the Total Harmonic Distortion in the MATLAB® environment. The analysis is presented in the thesis with conclusions and recommendations for future work.

1.4 Thesis Outline

The contents of the remaining chapters of this work are as follows. In Chapter 2 a comprehensive literature review is presented on wind generator technologies with emphasis on Doubly Fed Induction Generators (DFIG) together with the control methods being analysed. Modelling techniques of Wound rotor induction machines

(WRIM) are reviewed to establish the basis for DFIG control. An overview of previous work into Matrix Converter (MC) technology is reviewed. A review of relevant information is conducted in Chapter 3 which forms the basis of the research conducted in the thesis.

Chapter 4 shows the development of an MC system using non-ideal component models. Commutation control methods are adopted to facilitate the reliable operation of the MC. The results obtained from the MC system including the commutation control system are presented.

Chapter 5 examines the theory of DFIG control by applying the methods researched in the literature review. The MC model using non-ideal component models is applied to the DFIG system to regulate the rotor currents and to provide overall power control. The results are analysed in various modes of operation to show the viability of the system. A Maximum Power Point Tracking Algorithm is developed an applied to the MC excited DFIG with the results shown. A reactive power compensation algorithm is developed to provide the mechanism for voltage regulation in power systems. The system is subjected to a variable load which acts as a flicker source via a common bus. The reactive power compensation algorithm is actuated to show that the MC excited DFIG is able to provide voltage regulation to power systems.

Finally, in Chapter 6 using collated results from the research presented in the previous chapters, conclusions and recommendations are made. This will set the foundation for further research in future studies.

Chapter 2

Literature Review

2.1 Wind Power Generators

There are many different variations in the configuration of wind generators. However, they can be divided into two categories; Fixed Speed Generators (FSG) and Adjustable (or Variable) Speed Generators (ASG). The most common types of wind generator configuration are classified as:

1. Danish Concept (FSG)
2. Direct-in-Line (ASG)
3. Doubly-Fed Induction Generator (ASG)

The operation and characteristics of these wind generator configurations are outlined Sub-Sections 2.1.1, 2.1.2 and 2.1.3.

2.1.1 Danish Concept

In the early days of wind power, generators were built employing a configuration that is known as the *Danish Concept* [2], which is an example of a Fixed Speed Generator.

This method of conversion used a squirrel-cage induction machine with the stator directly connected to the grid. Having the stator directly connected to the grid the speed of the generator had to be regulated. This is achieved using advanced pitch control techniques. This is an effective method of capturing wind power, however, the major disadvantage of this system is much of the available power in the wind gust would be wasted due to the fact that the generator is unable to vary its speed to match the optimal power transfer characteristic of the wind turbine. This technique is still used in low-power implementations [3]. This is due to its limitations in its ability to effectively transfer the wind power and to adjust to sudden changes in wind speed.

The problems experienced in the Danish Concept are overcome by the use of adjustable speed generators (ASG). This variety of generators use power electronic devices to allow the generator to turn with a variable speed to achieve optimal power tracking. In using an ASG, the generator is able to operate over a range of speeds which allows the system to capture the wind energy approximately 10% more efficiently than the Danish Concept generators [3]. ASG systems offer improved power quality as the torque pulsations are reduced and power variations are thus eliminated [3]. Another advantage of ASG systems is that the pitch control system can be vastly simplified as they are not primarily responsible for the control of the generator output.

2.1.2 Direct-in-Line

ASG systems can vary in the way they are implemented. One implementation is the *Direct-in-Line* generator shown in Figure 2.1. The Direct-in-Line system incorporates a squirrel-cage induction machine which is connected to the grid via an AC-AC power converter. The primary drawback of the Direct-in-Line implementation is that the

AC-AC converters used must be large enough to handle the total power rating of the generator, i.e. the converters in the system must be rated at greater than 1.0pu. Also the low-pass filters used to remove the HF switching components in the current waveforms must increased in size to cope with the larger magnitude in unwanted switching harmonics [2]. For these reasons the size of the generator that is used is in the system is limited.

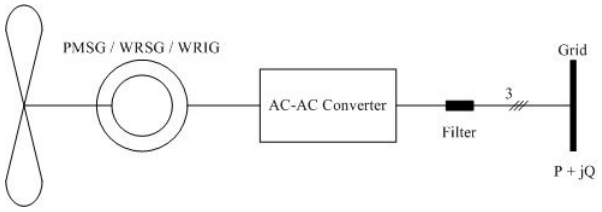


Figure 2.1: Direct-in-line wind turbine system

2.1.3 Doubly-Fed Induction Generator Theory

The Doubly-Fed Induction Generator (DFIG) is an ASG system which overcomes many of the downfalls of the Danish Concept and the Direct-in-Line methods of wind power generation. As a result more energy is extracted from the wind and delivered to the power grid.

DFIG uses a Wound Rotor Induction Machine (WRIM) with the stator directly connected to the grid. The rotor is excited with an AC-AC power converter which is connected between the rotor and the grid. This system is known as a slip recovery system or more commonly as a *Scherbius Drive* as shown in Figure 2.2. The DFIG concept is the preferred method of wind power generator implementation in modern-high powered systems. This is due to the AC-AC power converter rating is

approximately 25% of the total system power [2, 3, 4] as it only converts the *slip power* of the system [5]. This translates to a large reduction in the size of the power converters and thus the size of the High Frequency (HF) filters used in the converters is also vastly reduced. Furthermore, Scherbius drive systems are approximately 2-3% more efficient than the Direct-in-Line system [2]. This is due to reduced switching losses in the AC-AC power converters as they transfer less power between the grid and the generator. The speed range of DFIG is typically limited to ±33% [2] of the synchronous speed. As the Scherbius drive system resolves many of the problems found in the Danish and Direct-in-Line concepts, higher capacity generating systems can be developed. Typical modern high-power generators are rated up to 5 MW.

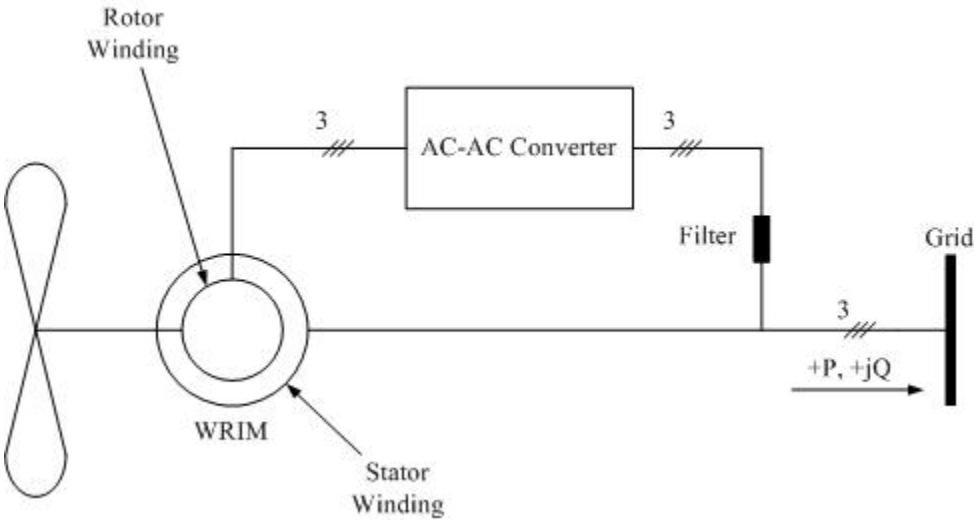


Figure 2.2: Doubly-Fed Induction Generator wind turbine system

The power converters are configured in a two stage conversion process which requires a capacitor to regulate the DC link voltage between the two converters as shown in Figure 2.3. The DFIG rotor speed can be controlled to achieve maximum power extraction as the wind speed changes through the regulation of the rotor current.

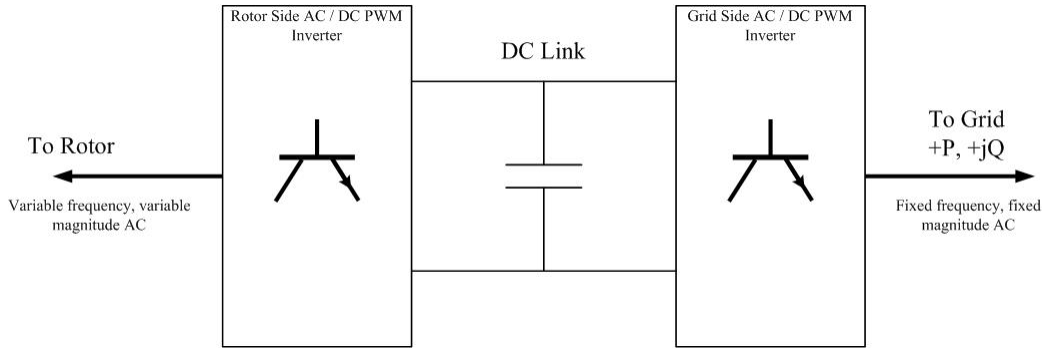


Figure 2.3: Back-to-Back PWM Inverters in a DFIG

2.2 Control of Doubly-Fed Induction Generators

In a DFIG system the supply/grid side and rotor side PWM power converters are both vector controlled. This allows for independent control of the generator torque and the rotor excitation of the rotor side converter and also allows for independent control of the displacement factor between voltage and current of the supply side converter. It is shown in [3] that the synchronous reference frame of the rotor side vector controller can be linked to the rotor flux or stator voltage space vector. However, typically the vector controller is linked to the stator flux vector as shown in [6, 7, 1]. For the purposes of this thesis, the DFIG is stator flux vector controlled.

The speed range of the system is typically restricted to $\pm 33\%$ of synchronous speed [2]. This restriction is acceptable as the energy that is extracted at low wind speeds is relatively small. As the slip speed increases the efficiency of the system decreases. This is caused by switching losses in the rotor circuit power converters as the system moves further away from synchronous speed and the rotor current increases. Rotor iron and frictional losses also add to these losses which decrease the efficiency of the system [6].

2.2.1 Control of DFIG Rotor Side PWM Converter

The objective of the DFIG rotor side PWM converter is to provide decoupled control of the stator active and reactive power by means of rotor current regulation [5]. This is achieved by controlling the rotor currents in a synchronously rotating reference frame with the d-axis aligned with the stator flux vector position. This is often referred to as a stator flux vector controller.

To align the synchronously rotating reference frame, the stator flux linkage vector is calculated using a Park Transform as defined in Appendix A. The stator flux vector is found using (2.1), (2.2) and (2.3).

$$\lambda_{ds} = \int (v_{ds} - r_s i_{ds}) dt \tag{2.1}$$

$$\lambda_{qs} = \int (v_{qs} - r_s i_{qs}) dt \tag{2.2}$$

$$\theta_{\lambda s} = \tan^{-1} \frac{\lambda_{qs}}{\lambda_{ds}} \tag{2.3}$$

In [6] it is shown that by employing the value $\theta_{\lambda s}$ found in (2.3) the stationary reference frame can be transformed into a synchronously rotating the reference frame by using the Clarke Transformation as shown in Appendix A, (A.3). Orienting the d-axis with the stator flux vector (2.4) is obtained.

$$\lambda_s^e = \lambda_{ds}^e + j\lambda_{qs}^e = \lambda_{ds}^e = L_m i_{ms}^e \tag{2.4}$$

λ_{qs}^e is equal to zero since the d-axis of the reference frame is aligned with the

stator flux linkage vector. Assuming all values used are in per unit form the voltage equation for the rotor of the WRIM is given by (2.5) and (2.6).

$$v_{dr}^e = r_r i_{dr}^e + \frac{d}{dt}(L_r i_{dr}^e + L_m i_{ds}^e) - \omega_{slip}(L_r i_{qr}^e + L_m i_{qs}^e) \quad (2.5)$$

$$v_{qr}^e = r_r i_{qr}^e + \frac{d}{dt}(L_r i_{qr}^e + L_m i_{qs}^e) + \omega_{slip}(L_r i_{dr}^e + L_m i_{ds}^e) \quad (2.6)$$

where

$$\omega_{slip} = w_e - w_r$$

$$L_s = L_{ls} + L_m$$

$$L_r = L_{lr} + L_m$$

The active and reactive stator power output is given by (2.7) and (2.8) respectively.

$$P_s = -\frac{3}{2}\omega_e \frac{L_m^2}{L_s} i_{ms}^e i_{qr}^e \quad (2.7)$$

$$Q_s = \frac{3}{2}\omega_e \frac{L_m^2}{L_s} i_{ms}^e (i_{ms}^e - i_{dr}^e) \quad (2.8)$$

The magnetizing current (i_{ms}^e) is close to constant and as such using (2.7) the active power output of the stator (P_s) is proportional to i_{qr}^e. Therefore, P_s can be

controlled by regulating i^e_{qr}. Also using (2.8) the reactive power output of the stator (Q_s) can be controlled by regulating i^e_{dr}. Using (2.5) and (2.6), the voltage equations (2.9) and (2.10) can be extracted.

$$v^{e'}_{dr} = r_r i^e_{dr} + \frac{d}{dt}(L_r i^e_{dr} + L_m i^e_{ds}) \tag{2.9}$$

$$v^{e'}_{qr} = r_r i^e_{qr} + \frac{d}{dt}(L_r i^e_{qr} + L_m i^e_{qs}) \tag{2.10}$$

The values i^e_{qr} and i^e_{qr} are obtained using a closed loop system and they are processed by PI controllers to obtain the reference values v^e_{qr} and v^e_{dr} [6]. To compensate for the cross-coupling effect of the d-q components, decoupling components are added using (2.11) and (2.12).

$$v^e_{drc} = -\omega_{slip}(L_r i^e_{qr} + L_m i^e_{qs}) \tag{2.11}$$

$$v^e_{qrc} = \omega_{slip}(L_r i^e_{dr} + L_m i^e_{ds}) \tag{2.12}$$

Using (2.9), (2.10), (2.11) and (2.12) the reference values v^{e*}_{qr} and v^{e*}_{dr} are derived as shown by (2.13), (2.14), (2.15) and (2.16).

$$v^{e*}_{dr} = v^{e'}_{dr} - \omega_{slip}(L_r i^e_{qr} + L_m i^e_{qs}) \tag{2.13}$$

$$v^{e*}_{qr} = v^{e'}_{qr} + \omega_{slip}(L_r i^e_{dr} + L_m i^e_{ds}) \tag{2.14}$$

$$v^{e*}_{dr} = v^{e'}_{dr} + v^e_{drc} \tag{2.15}$$

$$v^{e*}_{qr} = v^{e'}_{qr} + v^e_{qrc} \tag{2.16}$$

The reference values v^{e*}_{dr} and v^{e*}_{qr} are transformed back into three phase stationary

reference frame voltages and passed onto the rotor circuit PWM power converter controllers to apply a voltage to the rotor windings of the WRIM to regulate rotor current. This is to enable effective decoupled control of P_s and Q_s. The block diagram of the DFIG rotor side controller is shown in Figure 2.4.

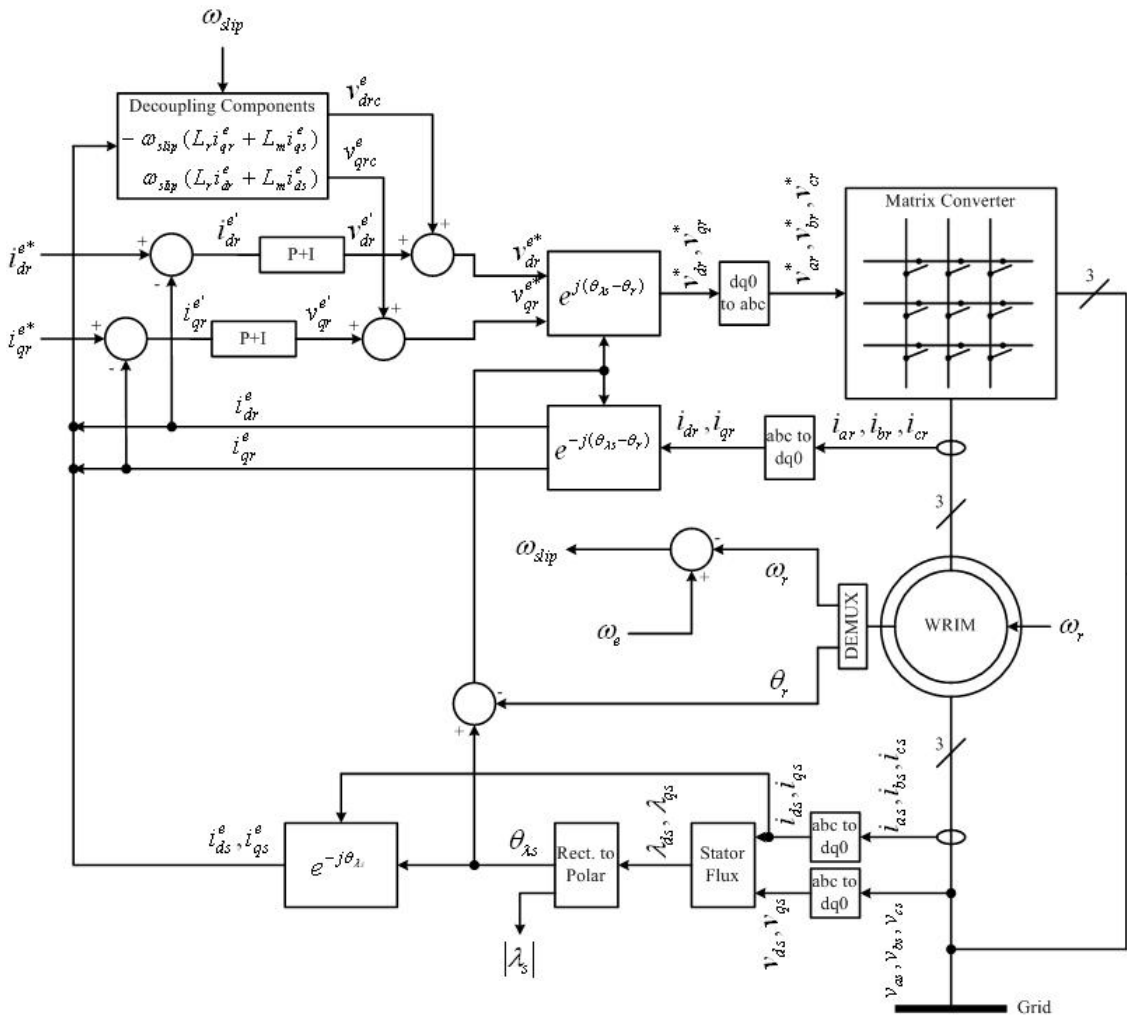


Figure 2.4: DFIG control system

2.3 Matrix Converter Theory

The three phase to three phase Matrix Converter (MC) is an AC-AC power converter which consists of nine bi-directional four quadrant power electronic switching devices arranged as a 3×3 matrix together with a low-pass filter. The switching devices are used to modulate the output voltage waveform with a variable voltage and variable frequency (VVVF) while drawing a sinusoidal input current. The MC switching devices are arranged so that any of the output lines can be connected to any of the input phases as shown in Figure 2.5. Depending on the Pulse Width Modulation (PWM) scheme that is adopted, the bi-directional switches are closed for a calculated length of time resulting in a voltage pulse at the output. Using a low pass filter the high frequency (HF) switching components are removed and the input current waveform is synthesized to the desired magnitude and frequency. Two common PWM methods used with a MC are Venturini and Space Vector modulation.

The MC has many advantages over other AC-AC converters. The primary advantages of the MC are:

- It is a single stage converter which does not have any large energy storage elements in the circuit. This means that the MC can be packaged into a smaller enclosure.
- It can be modulated to produce a variable sinusoidal output voltage and draws a sinusoidal input current.
- Its input displacement factor is controllable and is independent of the load that is connected to the output.
- It is a four-quadrant switching system which is capable of bi-directional power

flow.

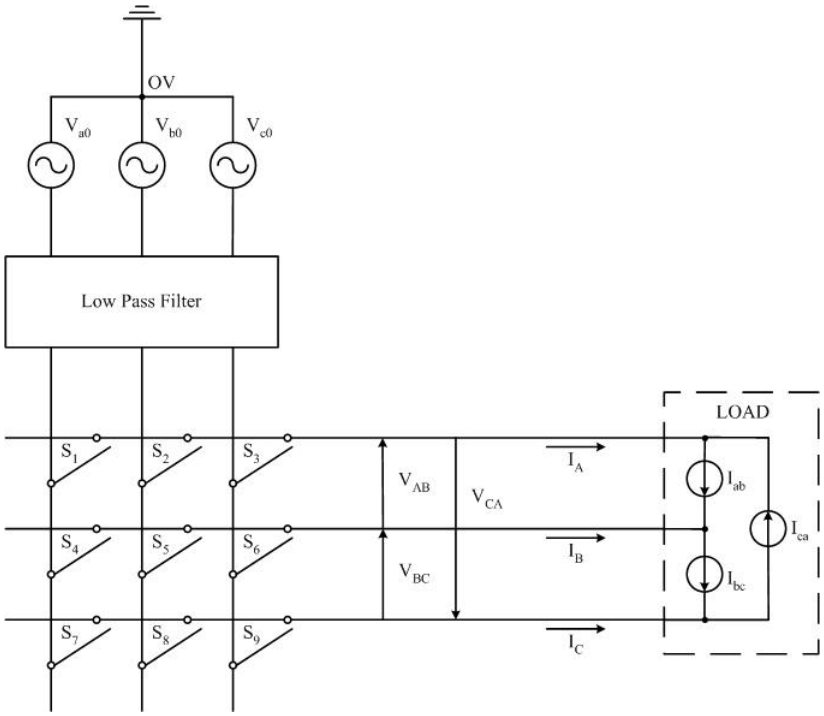


Figure 2.5: Ideal Matrix Converter circuit

The MC is a forced commutation converter and there are no *freewheel* paths for current to flow. For this reason the commutation strategy must reliably control the flow of current throughout the system which is explained in Section 2.7.

2.4 Matrix Converter Simulation

Simulation of matrix converters in a digital environment has been extensively researched [10-18]. When non-ideal component models are used, the complexity in control increases. A matrix converter is comprised of nine bi-directional switches that require precise timed control signals to modulate the output voltage and input

current waveforms. When ideal switches are used in the switching matrix, it is assumed that the switch can change from the *on* state to the *off* state instantaneously and vice-versa. When such a device is assumed, the commutation process that maintains the continuity of the current through the devices will not be disrupted and hence the simulation will continue with minimised transient conditions that would otherwise potentially cause the system to fail. Furthermore, the devices have a zero on-state resistance and an infinite off-state resistance; the forward bias voltage drop is also neglected in such a model.

Matrix converter simulations that are based on non-ideal components are complicated. The bi-directional switches require time to change from the on-state to the off-state and vice versa. Consequently, the commutation process must be controlled using a commutation scheme as outlined in Section 2.7, thus introducing a latency time into the response of the system. The system dynamics are altered by the finite off-state resistance (though it may be relatively large as compared to the on-state resistance), and the on-state resistances that exist within the switches. The forward bias voltage drop further changes the system dynamics.

2.5 Space Vector Modulation

Space Vector Modulation (SVM) is a form of pulse width modulation that is based on the two-phase representation of three phase quantities. Some of the advantages that SVM has over other PWM techniques are:

- A wide linear modulation range [15].
- It has improved Total Harmonic Distortion (THD) characteristics as much of

the disturbance is centred on the switching frequency.

- The switching frequency is much greater than the input supply fundamental frequency and thus it is possible to remove the high frequency switching components using a low pass filter.
- It is easily implemented in digital applications [15].

For these reasons, SVM is adopted for use in this study.

SVM requires careful analysis of the input waveforms and many mathematical operations to synthesize the desired output. To simplify the analysis of SVM in an MC it has been shown in [9] that an equivalent circuit can be adopted. The circuit consists of a Voltage Source Rectifier (VSR) and a Voltage Source Inverter (VSI) which are joined by a fictitious DC link as shown in Figure 2.6. SVM is applied to the two circuits independently to produce the desired input current and output voltage waveforms.

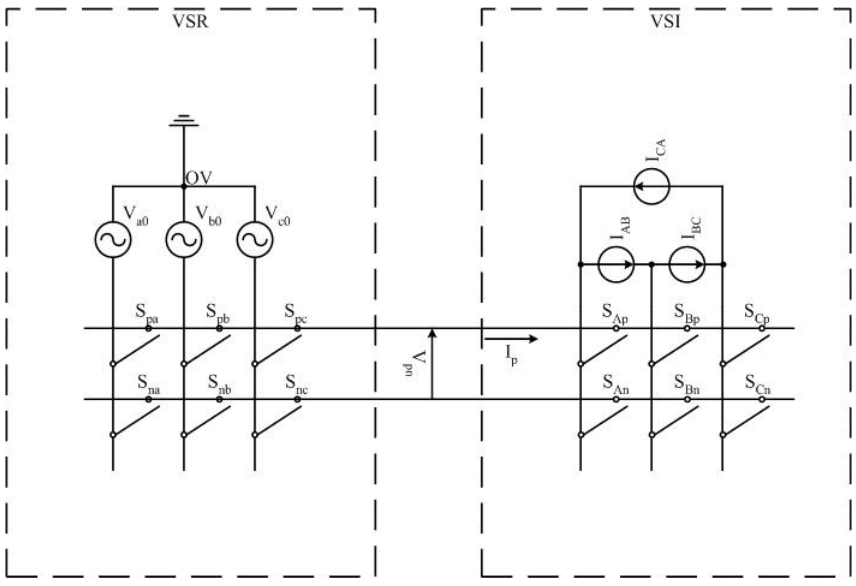


Figure 2.6: Equivalent circuit of an MC using a VSR and VSI circuit

2.5.1 Voltage Source Inverter

To apply SVM on the Voltage Source Inverter (VSI) it assumed that V_{pn} in Figure 2.6 is a constant DC voltage.

$$V_{pn} = E_{DC}$$

Using the Park Transform (as shown in Appendix A.3) it is possible to obtain an output voltage space vector v_{ol}. In the VSI circuit of Figure 2.6 it can be seen that the 6 switches connect the output phases to either the $+v_{pn}$ rail or the 0V rail of the DC link. The 6 switches provide $2^6 = 64$ possible switching combinations. However, many of the combinations are invalid due to short circuit conditions developed across the DC link or an open circuit condition of an inductive load. If these undesirable combinations are removed it is found that there are $2^3 = 8$ valid switching combinations. If the output value E_{DC} is substituted for each of the switching combinations and the transform equation in (A.6) is applied the output line-line and transform values are obtained, which are listed in Table 2.1. Using the notation of 1 to signify the output phase connected to the positive rail of the DC link and 0 for the negative rail, the valid switch combinations of the VSI that are produced are shown in Table 2.1.

From Table 2.1 it can be seen that each switching combination results in a vector which is known as a Stationary Switch Vector (SSV). Each SSV can be plotted on an $\alpha + j\beta$ plane (shown in Appendix B.2) which has six SSVs equally spaced apart at an angle of $\frac{\pi}{3}$ radians. There are also two *zero* vectors at the origin of the plane which correspond to an output line-line value of 0V.

SSV	**A**	**B**	**C**	$\mathbf{v_{AB}}$	$\mathbf{v_{BC}}$	$\mathbf{v_{CA}}$	$\mathbf{v}_\alpha$	$\mathbf{v}_\beta$
V_1	1	0	0	E_{DC}	0	$-E_{DC}$	E_{DC}	$\frac{1}{\sqrt{3}}E_{DC}$
V_2	1	1	0	0	E_{DC}	$-E_{DC}$	0	$\frac{2}{\sqrt{3}}E_{DC}$
V_3	0	1	0	$-E_{DC}$	E_{DC}	0	$-E_{DC}$	$\frac{1}{\sqrt{3}}E_{DC}$
V_4	0	1	1	$-E_{DC}$	0	E_{DC}	$-E_{DC}$	$-\frac{1}{\sqrt{3}}E_{DC}$
V_5	0	0	1	0	$-E_{DC}$	E_{DC}	0	$-\frac{2}{\sqrt{3}}E_{DC}$
V_6	1	0	1	E_{DC}	$-E_{DC}$	0	E_{DC}	$-\frac{1}{\sqrt{3}}E_{DC}$
V_7	0	0	0	0	0	0	0	0
V_8	1	1	1	0	0	0	0	0

Table 2.1: VSI Line-Line and Vector Output Values

The output waveforms are modulated through the selection of the different SSVs during every switching period t_s. Despite the output vector V_{oL} being a constantly changing vector (in magnitude and phase), it can be assumed to be stationary over the switching period as the switching period is much smaller (typically between $50\mu s$ and $200\mu s$) compared to the modulated period. As v_{oL} is constantly changing and rarely will it directly correspond to any of the eight SSVs, it becomes necessary to approximate the vector through applying a number SSVs for a calculated period.

During every switching period the sector that the output vector is located in is determined. From this the upper and lower SSVs that bind the sector and θ_{oL} are ascertained as shown in Figure 2.7. Using the sine rule the duty cycle d_n of the upper, lower and zero vectors is calculated as given by (2.17) - (2.19).

$$d_k = m_v \sin(\frac{\pi}{3} - \theta_{oL}) \tag{2.17}$$

$$d_l = m_v \sin(\theta_{oL}) \tag{2.18}$$

$$d_0 = 1 - d_k - d_l \tag{2.19}$$

where

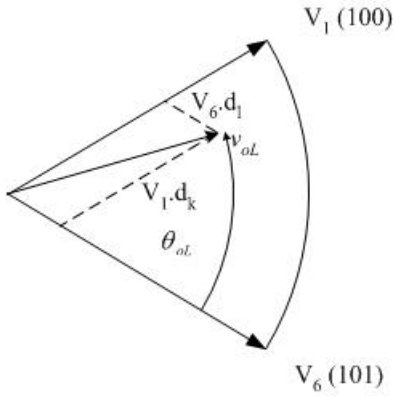


Figure 2.7: SSV selection in a VSI circuit

$$
\begin{aligned}
0 &\leqslant m_v = \frac{|v_{oL}|}{E_{DC}} \leqslant 1 \\
d_x &= \frac{T_x}{t_s} \\
x &\in \{k, l\} \\
0 &\leqslant \theta_{oL} = \angle v_{oL} - \angle V_k \leqslant \frac{\pi}{3}
\end{aligned}
$$

From the duty cycles of the selected SSVs the *on* time of the SSV can be calculated by multiplying the duty cycle by the switching period. The different SSVs are applied in the switching sequence for the calculated times and this produces the desired output. This process is repeated during every switching period and this results in the output waveforms being modulated.

2.5.2 Voltage Source Rectifier

To apply SVM to the Voltage Source Rectifier (VSR) it is assumed that I_p in Figure 2.6 is a constant direct current.

$$I_p = I_{DC}$$

Using the Park Transform as shown in Appendix A.4 it is possible to obtain an input current space vector i_{il}. From here the VSR analysis is similar to the VSI system. The main conditions that must be satisfied is that the three phase input voltage source is not short circuited and the DC link which acts as a constant load current is not subjected to an open circuit situation. Thus, there are $3^2 = 9$ valid switching combinations of the VSR. If the input value I_{DC} is substituted for each of the switching combinations and the transform in (A.10) is applied, then the input phase current values and the transform values are obtained as listed in Table 2.2. Using the notation of a,b, or c to signify the input phase that is connected to the $+V_{pn}$ rail 0V rail of the DC link, the valid switch combinations of the VSI are produced. This is also shown in Table 2.2.

SSV	**$+\mathbf{V_{pn}}$ rail**	**0V rail**	$\mathbf{i_a}$	$\mathbf{i_b}$	$\mathbf{i_c}$	$\mathbf{i}_\alpha$	$\mathbf{i}_\beta$
I_1	a	c	I_{DC}	0	$-I_{DC}$	I_{DC}	$\frac{1}{\sqrt{3}}I_{DC}$
I_2	b	c	0	I_{DC}	$-I_{DC}$	0	$\frac{2}{\sqrt{3}}I_{DC}$
I_3	b	a	$-I_{DC}$	I_{DC}	0	$-I_{DC}$	$\frac{1}{\sqrt{3}}I_{DC}$
I_4	c	a	$-I_{DC}$	0	I_{DC}	$-I_{DC}$	$-\frac{1}{\sqrt{3}}I_{DC}$
I_5	c	b	0	$-I_{DC}$	I_{DC}	0	$-\frac{2}{\sqrt{3}}I_{DC}$
I_6	a	b	I_{DC}	$-I_{DC}$	0	I_{DC}	$-\frac{1}{\sqrt{3}}I_{DC}$
I_7	a	a	0	0	0	0	0
I_8	b	b	0	0	0	0	0
I_9	c	c	0	0	0	0	0

Table 2.2: VSR circuit phase current and vector input values

From Table 2.2 it can be seen that each switching combination results in a vector which is known as a Stationary Switch Vector (SSV). If this was plotted in the $\alpha + j\beta$ domain it can be seen that six separate SSVs and three zero vectors exist for the VSR

circuit as shown in Appendix B.3.

Similar to the VSI, the adjacent SSVs of the input current vector are determined for each switching period, and the duty cycles are calculated using (2.20) - (2.22).

$$d_m = m_i \sin(\frac{\pi}{3} - \theta_{iL}) \tag{2.20}$$

$$d_n = m_i \sin(\theta_{iL}) \tag{2.21}$$

$$d_0 = 1 - d_m - d_n \tag{2.22}$$

where

$$\begin{aligned} 0 &\leqslant m_i = \frac{|i_{iL}|}{I_{DC}} \leqslant 1 \\ d_x &= \frac{T_x}{t_s} \\ x &\in \{m, n\} \\ 0 &\leqslant \theta_{iL} = \angle i_{iL} - \angle I_m \leqslant \frac{\pi}{3} \\ \angle i_{iL} &= \angle v_{iL} + \phi_i \end{aligned}$$

The input current vector (i_{iL}) displacement angle from the the voltage vector (v_{iL}) is shown as ϕ_i in the equations.

Using the duty cycle values obtained from (2.20) - (2.22) the *on* time for each SSV can be calculated. This process is repeated during every switching period. The input current vector position is determined using the input voltage vector and the desired displacement factor.

2.5.3 SVM of a Matrix Converter

To apply SVM to an MC, the VSI and VSR circuits are combined to produce switching combinations and duty cycles that will synthesize the required input current and output voltage waveforms. As the VSR is able to produce a constant DC voltage source and the VSI is capable of providing a constant direct load current, the two systems can be mutually linked. Using Tables 2.1 and 2.2 the valid switching combinations and SSVs are determined. This is done by substituting the connected input and output phases for each SSV in the VSI and VSR. Once the input/output phase connection is known for each SSV, the switching combination is then ascertained. Using the switch index notation shown in Figure 2.5 an SSV-Switch Combination *lookup* table is obtained as shown in Appendix B.1. Note that in Table B.1 there are three zero vectors which are possible in any switching sequence. Also each vector appears twice in the table (e.g SSV pair $I_1V_1 = I_4V_4$). As a result of this there are 21 different valid switch combinations used by the matrix converter.

The duty cycles for the MC SSVs are derived from the simultaneous solution of (2.17) - (2.19) and (2.20) - (2.22). Assuming $m_i = 1$ in (2.20) - (2.22) it has been shown in [9] that the resulting duty cycles are:

$$d_{km} = d_k d_m = m \sin(\frac{\pi}{3} - \theta_{oL}) \sin(\frac{\pi}{3} - \theta_{iL}) = \frac{T_{km}}{t_s} \quad (2.23)$$

$$d_{lm} = d_l d_m = m \sin(\theta_{oL}) \sin(\frac{\pi}{3} - \theta_{iL}) = \frac{T_{lm}}{t_s} \quad (2.24)$$

$$d_{kn} = d_k d_n = m \sin(\frac{\pi}{3} - \theta_{oL}) \sin(\theta_{iL}) = \frac{T_{kn}}{t_s} \quad (2.25)$$

$$d_{ln} = d_l d_n = m \sin(\theta_{oL}) \sin(\theta_{iL}) = \frac{T_{ln}}{t_s} \quad (2.26)$$

$$d_0 = 1 - d_{km} - d_{lm} - d_{kn} - d_{ln} = \frac{T_0}{t_s} \quad (2.27)$$

where

$$\begin{aligned} |v_{oL}|_{max} &= \frac{\sqrt{3}}{2}|v_{iL}|\cos\phi_i \\ m &= \frac{2|v_{oL}|}{\sqrt{3}|v_{iL}|\cos\phi_i} \end{aligned}$$

The duty cycles and switching combinations are calculated during every switching period. The optimal switching sequence as shown by [9] is:

$$T_{km} \rightarrow T_{lm} \rightarrow T_{ln} \rightarrow T_{kn} \rightarrow T_0$$

This switching sequence is optimised such that the switch state transitions are minimized. It has also been shown in [8] that a symmetric switching sequence can be adopted to further reduce the total harmonic distortion (THD). This results in the following switching sequence:

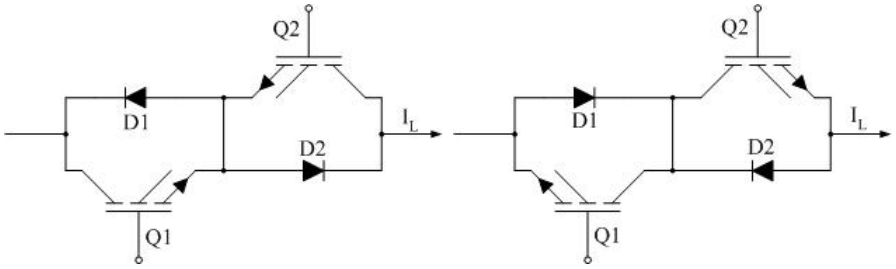
$$T_{km} \rightarrow T_{lm} \rightarrow T_{ln} \rightarrow T_{kn} \rightarrow T_0 \rightarrow T_{kn} \rightarrow T_{ln} \rightarrow T_{lm} \rightarrow T_{km}$$

The nine switch signals are passed onto the current commutation controller for the application of the gate signals to the power electronic devices within the MC circuit. The methods of commutation control will vary depending on the commutation strategy adopted which is elaborated on in Section 2.7.

2.6 Bi-Directional Four Quadrant Switches

The three phase - three phase Matrix Converter (MC) system is constructed using nine bi-directional four-quadrant switches (4QSW) arranged in a 3×3 matrix. Currently, there are no 4QSWs that are commercially available. As such, a 4QSW is produced

using two anti-parallel two-quadrant switches (2QSW). The preferred configurations of the 4QSW are the common emitter and common collector circuits which are shown in Figure 2.8(a) and 2.8(b), respectively. This is due to the reverse bias blocking characteristic of the anti-parallel switch which allows the direction of current flow to be controlled by the gate signals to the switching devices [12]. The switching losses are the same for either arrangement. However, the gate signal supply requirements are different for the two configurations.



(a) Common Emitter Configuration

(b) Common Collector Configuration

Figure 2.8: Four Quadrant Switch Configuration

Assuming the common emitter configuration is being used, current is able to flow in the direction of I_L if Q1 is gated. If the polarity of the voltage applied to the system is reversed or the gate signal on Q1 is removed, current will cease to flow. The converse is true if Q2 is gated. Similarly for the common collector configuration it is possible to explicitly control the direction of current flow.

There is another configuration which provides four-quadrant switching control. This arrangement uses a full-wave diode bridge and a single power electronic switch as shown in Figure 2.9 [17]. The advantage of this arrangement is that it only uses

one solid state switch to provide commutation and hence only one isolated gate drive power supply is required to drive each commutation cell. The major disadvantage of this circuit is that the current flow direction cannot be explicitly controlled, which is needed in many current commutation control schemes. For this reason the diode bridge arrangement is rarely utilised in MC switching arrays.

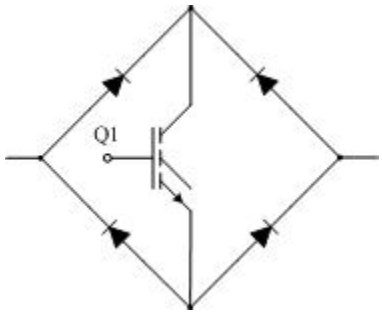


Figure 2.9: Full Bridge 4QSW

All different bi-directional switching configurations use a gate-controlled power electronic switching device. As the gated switching device provides the actuation in the modulation strategy to the output lines of the MC, a device should be carefully selected such that it is able to meet the demands of the application. There are many different gated switching devices which may be used in the matrix converter. These are:

1. Insulated Gate Bi-Polar Thyristor (IGBT)
2. Metal Oxide Semiconductor Field Effect Transistor (MOSFET)
3. MOS control Thyristor (MCT)
4. Integrated Gate Commutated Thyristor (IGCT)

Selection of the actuation device is usually based on switching frequency, voltage and load current demands. As the MC may be supplying a highly inductive load with a high switching frequency a MOSFET or an IGBT would be the most suitable devices for this application.

2.7 Current Commutation Techniques

Current commutation is the process of transferring the current flowing through a device (power electronic switch) to another different device. Commutation in the Matrix Converter (MC) is difficult to achieve as there is no natural freewheel path for the current to flow. With other conventional AC-AC converters, there are natural freewheel paths which allow stored energy in inductive circuits to be dissipated. This saves the switching devices from excessive voltage transients which can lead to destruction of the device. In the MC these avenues do not exist and they cannot be incorporated into the design without compromising the functionality of the switching matrix. Therefore, the commutation process in an MC is actively controlled.

With an MC there are two constraints for operation which must be conformed to, in order to ensure that the switching devices in the MC are not destroyed. These are:

1. Each output phase must be connected to an input phase at all times. This is due to AC drive system loads that are usually highly inductive circuits. To suddenly break the path of the current will result in a high transient voltage which will result in the destruction of the switching device.

2. Each output phase must only be connected to one input phase at all times. This is because if an output phase is connected to more than one input phase a line-line short circuit of the input will result. This will cause a large current to

flow through the switching device and will ultimately result in the destruction of the device. This is known as a shoot-through condition [11].

There are a number of different commutation strategies such as:

- Dead-Time Commutation
- Overlap Commutation, and
- Semi-Soft Commutation [11, 13, 18]

Although dead-time and overlap commutation schemes are outlined in Appendix B.1, they are not adopted in this study. This is because they momentarily violate the rules for conduction in an MC for a brief period during every change in switch combination.

2.7.1 Two Step Semi-Soft Commutation

Semi-soft commutation is more favourable than dead-time and overlap commutation as the rules of operation in an MC are not violated thus eliminating the need to use snubber circuits within the switching matrix. There are a couple of variations to semi-soft commutation, however, all these methods rely on the same principle of controlling the direction of current flow. For this reason a bi-directional switch which is able to explicitly control current flow direction is adopted to allow for this method to function correctly. A switch cell composed of two IGBTs and two diodes connected in antiparallel to form a four quadrant switch (4QSW), as shown in Figures 2.8(a) and 2.8(b), may be used as current flow direction and can be controlled by gating one of the IGBTs or vise versa.

As the output phase current increases above a predetermined threshold level during an AC wave half cycle the conducting device is detected and enabled via the gate

drive signal. The second device is disabled by the removal of the gate drive signal. As the current level approaches zero and will attempt to reverse the direction of flow, the following steps are performed by the commutation controller:

1. The current magnitude falls below the predetermined threshold level (towards zero)

2. Both actuating devices are enabled with the gate drive signal within the conducting 4QSW cell

3. Current magnitude increases in the opposite polarity and exceeds the threshold level

4. The non-conducting device in the 4QSW cell is disabled with the removal of the gate drive signal

When the switching combination is changed by the modulation scheme the following sequence of events occurs:

1. The predetermined incoming non-conducting 4QSW cell is determined by the modulation scheme. The actuating device within the incoming 4QSW cell which will allow the current to flow in the same direction is enabled by the gate drive signal. At this point in time two devices in two separate 4QSW cells will be conducting. A short circuit of the grid condition is avoided because the two enabled devices are in a forward bias condition and this will stop short circuit current from flowing.

2. The predetermined outgoing 4QSW cell actuating device is disabled by the removal of the gate drive signal. Commutation hand-over is completed.

A potential problem with this method is that at current magnitudes below the predetermined threshold level when two devices are enabled within the 4QSW cell, an accurate current flow direction measurement cannot be achieved using large magnitude current transducers. This can result in errors in determining the non-conducting device within the 4QSW cell. To overcome this the current direction can be determined by the measurement of the voltage drop across the devices in the 4QSW cell. Sophisticated control logic is needed for this method of commutation. Any logical errors may result in the device being destroyed and hence the system will fail.

2.8 Summary

In this chapter, technologies available for wind generation were introduced and compared. The reasons for using a DFIG systems as opposed to other methods of generating electrical power were highlighted. The DFIG model and operation was analysed to show how it is controlled in sub-synchronous and super-synchronous modes as a four quadrant system. The use of back-to-back PWM inverters connected to the rotor windings of a WRIM that allows for four quadrant power flow within the circuit and control of the entire system was briefly described.

Matrix converter theory was explored as a possible alternative for use in the rotor circuit of DFIG systems. The matrix converter was shown to have four quadrant control capabilities with varying voltage and frequency. Such properties are required in the rotor circuit to control DFIG systems. The different modulation methods, giving specific attention to Space Vector Modulation method of PWM, and commutation control within the matrix converter were also explored.

Chapter 3

Background Material

3.1 DFIG Operation

A DFIG is a four quadrant system which is capable of supplying and consuming active and reactive power when connected to a utility grid. The primary function of a DFIG is to perform as a generator and as such it is assumed that positive active and reactive power flow is delivered to the grid as shown in Figure 2.2. Using this convention the DFIG will generally not operate in the two *motoring* quadrants where active power is drawn from the grid. While the net power output of the DFIG typically corresponds to the two regenerative quadrant modes, the power converters within the system will operate in all of the four quadrants. The DFIG has two modes of operation: sub-synchronous and super-synchronous modes. The mode of operation is dependent on the angular speed of the rotor with respect to the electrical grid synchronous speed.

3.1.1 Sub-Synchronous Mode

A DFIG operates in Sub-Synchronous Mode if the rotor angular speed is slower than the electrical grid synchronous speed. In sub-synchronous mode real power is drawn from the grid and delivered to the rotor windings via the AC-AC power converters.

Assuming the convention that real and reactive power flowing into the grid is positive, the negative power flow through the power converters (i.e. into the rotor windings) induces an electromagnetic torque in the rotor. The torque produced is in the same direction as the machine rotation which is in addition to the mechanical torque which is supplied by the wind via the wind turbine.

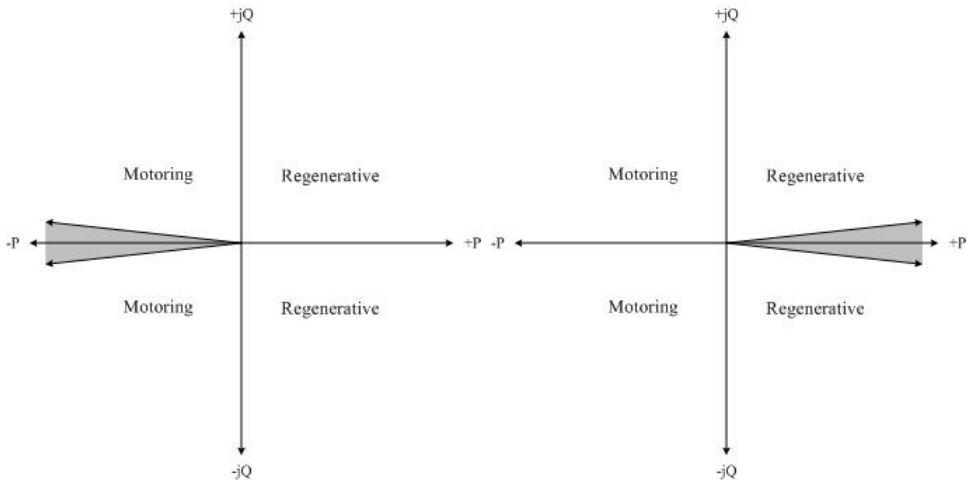
The majority of the reactive power from the DFIG is generally supplied by the stator windings of the machine and as such the displacement factor of the grid side converter is negligible. Assuming the power flow is measured between the grid side converter and the grid Point of Common Coupling (PCC) the resulting P,Q relation is produced as shown in Figure 3.1(a) by the shaded region. Note that the region of operation for the rotor circuit is within the two Motoring quadrants of the diagram while the overall net output of the generator is in the regenerative quadrants.

3.1.2 Super-Synchronous Mode

When the rotor angular speed is greater than the electrical grid speed the DFIG will operate in Super-Synchronous Mode. In super-synchronous mode, power is generated in the rotor circuit (in addition to power generated in the stator) and is delivered to the grid via the AC-AC power converters. An electromagnetic torque is induced in the rotor that acts as a load that resists the direction of rotation as a result of real power being drawn from the rotor circuit.

Similar to Sub-Synchronous mode operation, the displacement factor of the grid side converter is typically a small value and is negligible. By measuring the power flow between the grid side converter and the grid point of common coupling (PCC), the resulting P,Q relation that is produced is as shown in Figure 3.1(b) by the shaded

region which is located within the two Regenerative quadrants of the P/Q Diagram.



(a) Sub-Synchronous Mode (b) Super-Synchronous Mode

Figure 3.1: Power Flow through the AC - AC power converters of a DFIG

3.2 Wound Rotor Induction Machine Analysis

The Wound Rotor Induction Machine (WRIM) is the device used for converting mechanical power into electrical power in a DFIG system. Using the Park Transformation as shown in Appendix A a Direct, Quadrature and Zero *(dq0)* WRIM model can be developed. The per phase equivalent circuit is shown in Figure 3.2 [19]. The model is suitable for analysis in arbitrary reference frames. The corresponding voltage equations are given by (3.1) and (3.2) [19].

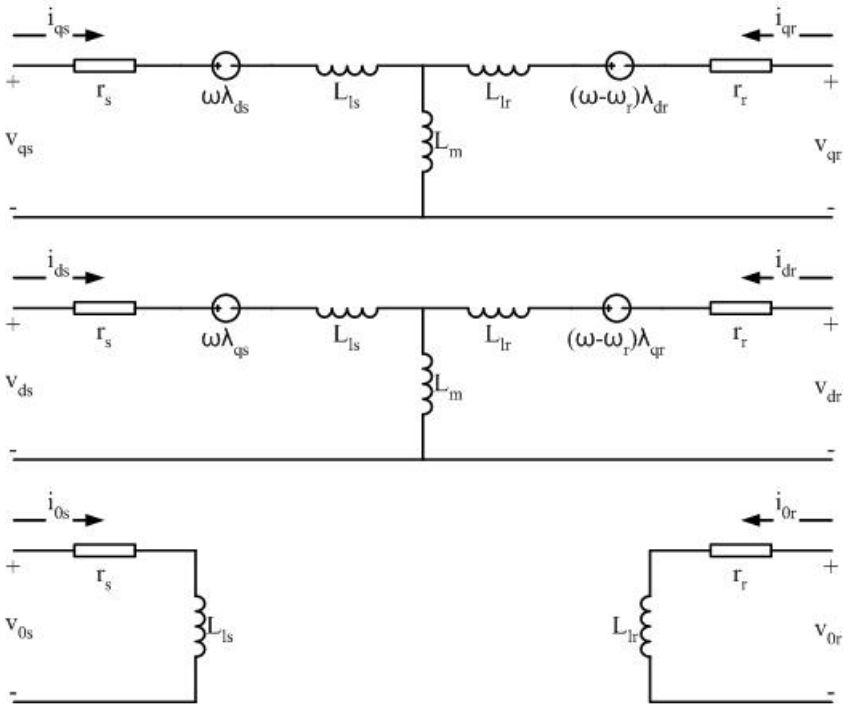


Figure 3.2: WRIM equivalent circuit in an arbitrary reference frame

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{0s} \\ v_{qr} \\ v_{dr} \\ v_{0r} \end{bmatrix} = \begin{bmatrix} \frac{r_s X_r}{D} + \frac{p}{\omega_b} & \frac{\omega}{\omega_b} & 0 & -\frac{r_s X_m}{D} & 0 & 0 \\ -\frac{\omega}{\omega_b} & \frac{r_s X_r}{D} + \frac{p}{\omega_b} & 0 & 0 & -\frac{r_s X_m}{D} & 0 \\ 0 & 0 & \frac{r_s}{X_{ls}} + \frac{p}{\omega_b} & 0 & 0 & 0 \\ -\frac{r_r X_m}{D} & 0 & 0 & \frac{r_r X_s}{D} + \frac{p}{\omega_b} & \frac{\omega-\omega_r}{\omega_b} & 0 \\ 0 & -\frac{r_r X_m}{D} & 0 & -\frac{\omega-\omega_r}{\omega_b} & \frac{r_r X_s}{D} + \frac{p}{\omega_b} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{r_r}{X_{lr}} + \frac{p}{\omega_b} \end{bmatrix} \begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda_{qr} \\ \lambda_{dr} \\ \lambda_{0r} \end{bmatrix} \tag{3.1}$$

where

$$\begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda_{qr} \\ \lambda_{dr} \\ \lambda_{0r} \end{bmatrix} = \frac{1}{D} \begin{bmatrix} X_r & 0 & 0 & -X_m & 0 & 0 \\ 0 & X_r & 0 & 0 & -X_m & 0 \\ 0 & 0 & \frac{D}{X_{ls}} & 0 & 0 & 0 \\ -X_m & 0 & 0 & X_s & 0 & 0 \\ 0 & -X_m & 0 & 0 & X_s & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{D}{X_{lr}} \end{bmatrix} \tag{3.2}$$

$$\begin{aligned} X_s &= X_{ls} + X_m \\ X_r &= X_{lr} + X_m \\ D &= X_s X_r - X_m^2 \\ p &= \frac{d}{dt} \end{aligned}$$

X_{lr}, X_{ls} and X_m are the per unit rotor leakage, stator leakage and magnetizing reactance respectively and λ is the flux linkage. Subscripts q,d and 0 denote the quadrature, direct and zero sequence, respectively.

Chapter 4

Matrix Converter Development

4.1 Ideal Matrix Converter Simulation

The work presented in this chapter covers the development of a non-ideal MC model to be implemented in a DFIG system such that the viability of the system, can be analysed and assessed. Connecting a non-ideal matrix converter to a load such as a wound rotor induction machine will introduce added factors to the dynamic system and may cause the system to function incorrectly. The inclusion of these quantities to the matrix converter can also impact the input filter design. Despite the incorporation of a well designed commutation scheme, transients may be emphasized by the non-ideal nature of the system. Hence, the filter has to be designed to reduce injected frequency components.

As an initial assessment, the simulation of the SVM in an MC using ideal switching devices has been undertaken using MATLAB. The input of the MC is connected to a $415V_{L-L}$ 50 Hz infinite busbar and the system converts the output to a synthesized $300V_{L-L}$ 20 Hz waveform modulated with a switching frequency of 5kHz, as shown in Figure 4.1. A filter is used on the output signal to remove high frequency pulses and

the fundamental frequency of 20 Hz can be observed. The phase shift between the filtered output and the SVM output waveforms is as a result of the FIR filter phase characteristic at 20 Hz.

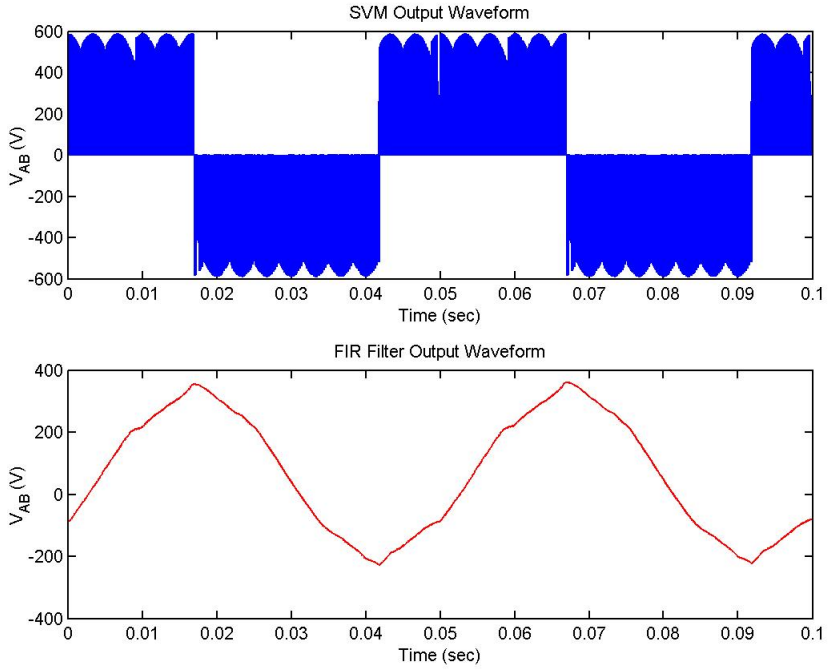


Figure 4.1: SVM voltage output obtained using ideal switching devices

Figure 4.2 shows a zoomed view of the SVM output pulses with varying magnitudes of the output pulses. The individual output pulses are a result of the different input phases being selected by the switching combinations. This simulation shows a SVM scheme can be theoretically applied to an MC using ideal switching devices and is able to synthesize output voltage waveforms of a different voltage and frequency.

The viability of an MC in a DFIG system can be assessed by adopting non-ideal component models and using SVM theory. PSCAD® / EMTDC^{TM} has been selected as the simulation environment, as the package has many incorporated models

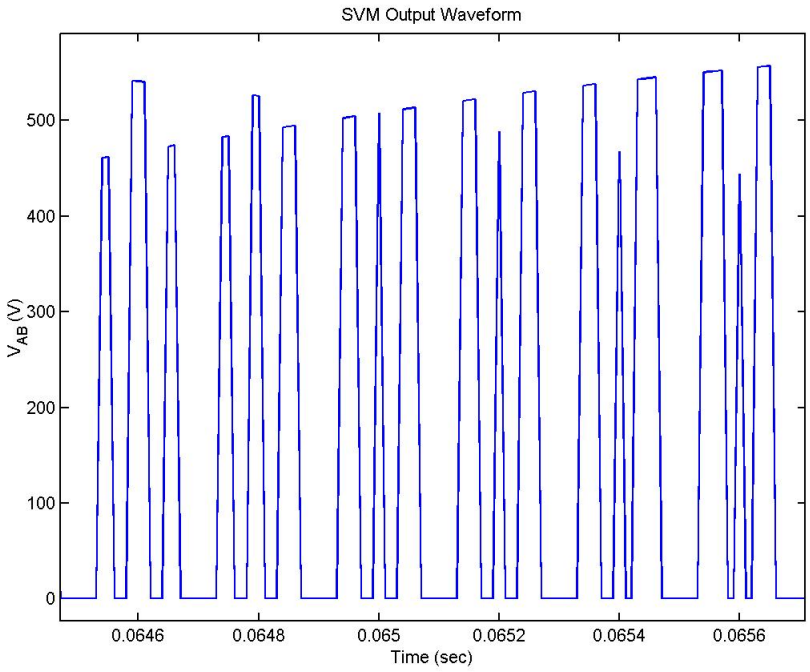


Figure 4.2: Close-up time scale view of the SVM voltage output

and adopts an interpolation interface algorithm which improves the accuracy of the simulations.

4.2 Bi-Directional Four Quadrant Switch Cell Verification

A four quadrant bi-directional switch (4QSW) cell is simulated in PSCAD® / EMTDC^{TM} so that it can be used to form the switch matrix of an MC. A 4QSW cell consists of two IGBTs and two diodes connected in an anti-parallel common emitter configuration, as shown in Figure 4.3. The V-I characteristics of the IGBT and diode employed in the simulation are shown in Figures 4.4(a) and 4.4(b), respectively. As outlined in Section 2.6, it is possible to explicitly control the direction of current flow in the 4QSW cell, depending on which IGBT's control signal is applied. This application of control signals, known as forced commutation, makes four quadrant power flow control possible.

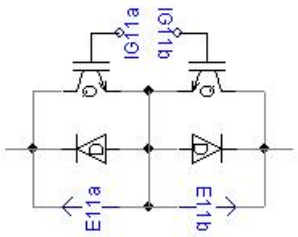
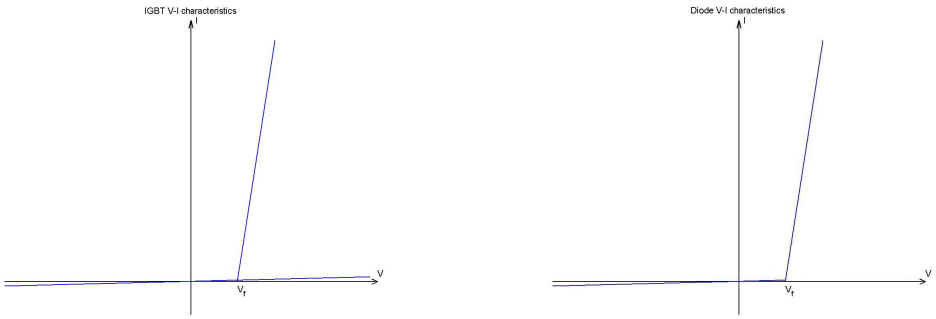


Figure 4.3: Four Quadrant Switch Cell using Common Emitter configuration in the PSCAD® / EMTDC^{TM} environment

The diode and IGBT parameters in the 4QSW cell model are set as per Table 4.1. Using the circuit shown in Figure 4.5, testing on the operation and biasing conditions of the 4QSW cell was conducted, with results and gate signals shown in Figure 4.6. The source input voltage is shown in the top plot and the line current through the 4QSW is shown in the second plot. As the two IGBTs are switched between the different devices, the varying polarities of the current flow can be seen to be dependent on the biasing of the enabled device(s). When only IGBT11a is gated, positive current flow is allowed to flow while the negative component is blocked, and when the IGBT11b is enabled the opposite is also true. At $t = 0.15$s, both IGBTs



(a) IGBT V-I characteristic (b) Diode V-I characteristic

Figure 4.4: V-I Characteristics of Four Quadrant Switch Components

are gated and the current is allowed to flow in both directions. The voltage across the IGBTs is shown in the third plot: when the device is gated the current across the device drops towards zero.

	IGBT	**Diode**
On-Resistance (R_{on})	0.01Ω	0.01Ω
Off-Resistance (R_{off})	1.0MΩ	1.0MΩ
Forward Bias Voltage Drop (V_f)	2.5V	1.0V

Table 4.1: 4QSW cell component parameters

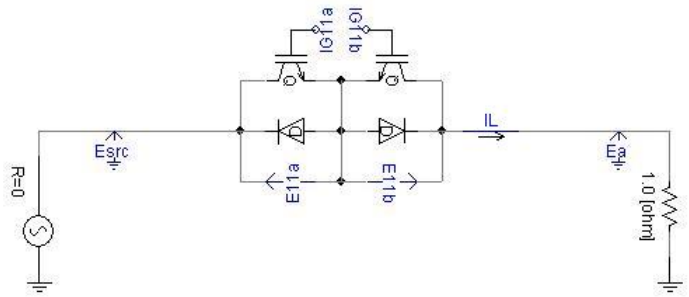


Figure 4.5: Simulation circuit of 4QSW cell in PSCAD® / EMTDC^{TM} environment

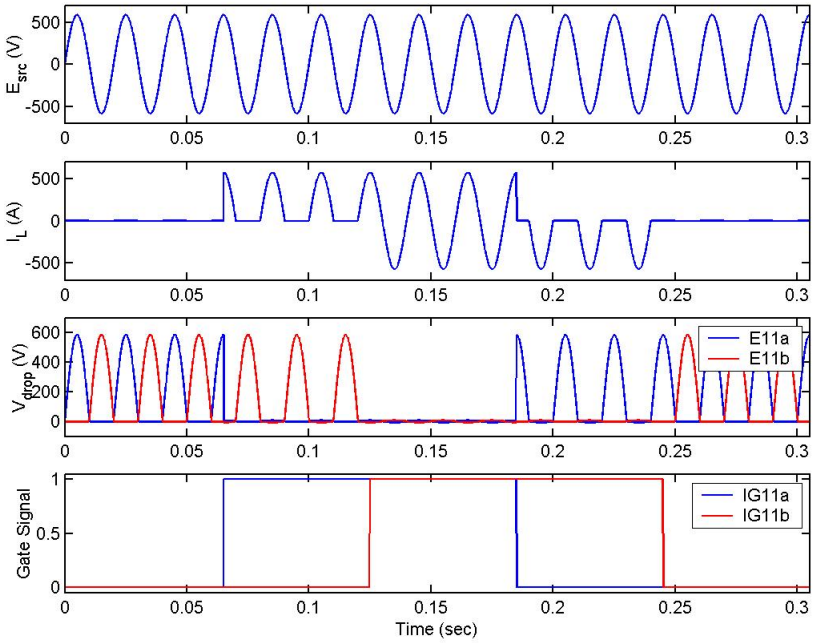


Figure 4.6: 4QSW cell simulation results

Using the results in Figure 4.6, it can be seen that it is possible to control current flow and its direction through the engagement of the two IGBT gate drive signals (IG11a and IG11b) in the 4QSW cell. The forward bias voltage drop of the conducting devices in the 4QSW cell, E11a and E11b, cannot be observed in Figure 4.6 because of the relatively large scale used. When both IGBTs are being gated during the period of $0.125 \leqslant t \leqslant 0.185$s, depending on the direction of current flow, the voltage drops and E11a and E11b alternate between -6.72 and +8.22V as shown in Figure 4.7. This is due to the combined voltage drop of the forward bias voltage and the On-Resistance R_{on} conduction voltage drop. From this it can be seen that the bias voltage drop across the devices in the 4QSW cell can be used to detect the current direction as outlined in Section 4.2.1. Also it can be concluded the common emitter 4QSW cell with semi-soft commutation is suitable for use in the matrix converters.

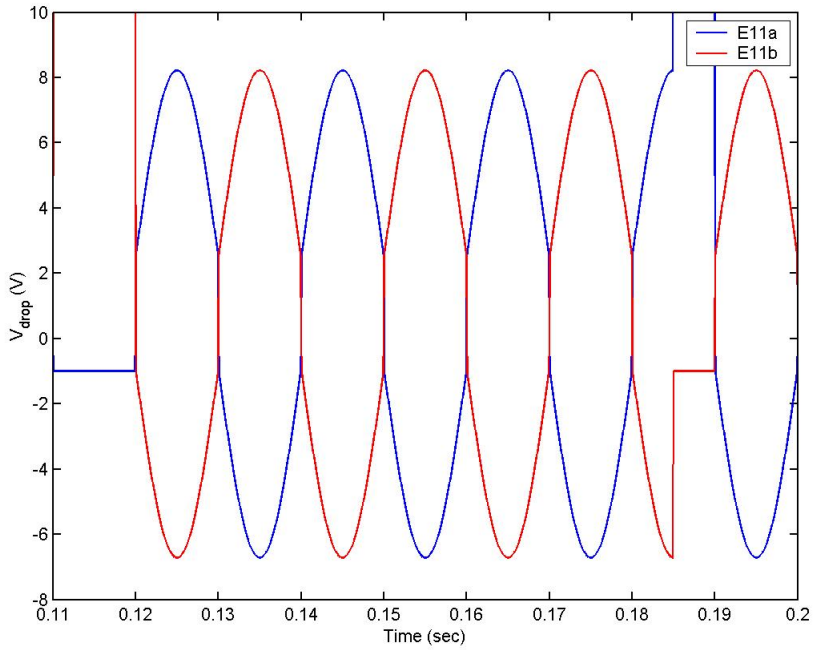


Figure 4.7: Close-up view of 4QSW cell simulation results

4.2.1 Current Direction Detection

At low current levels it can be difficult to accurately detect the direction of current flow using a current transducer. As stated in Section 2.7, it is a necessity when adopting the Two-Step Current Commutation technique to make precise current direction measurements in order for the system to function correctly. It is possible to determine the current direction at small magnitudes by measuring the voltage drop across the devices in the 4QSW cell [13].

If the forward bias voltage drop of a diode is equal to V_f^{diode} and the conducting forward bias voltage drop of a IGBT is V_f^{IGBT}, it can be shown that the voltages across a 4QSW will differ depending on the direction of current flow. Consider Figure 4.8, where the current flow is in the direction shown by I_L.

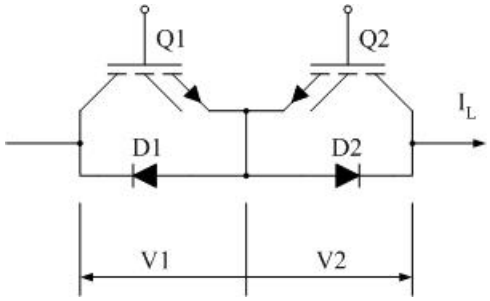


Figure 4.8: Common Emitter Four Quadrant Switch Cell

Assuming the on-state resistance of the IGBT and the diode are R_{on}^{IGBT} and R_{on}^{diode}, the off-state resistances are R_{off}^{IGBT} and R_{off}^{diode}, respectively, and $R_{on}^{IGBT} \ll R_{off}^{IGBT}$ and $R_{on}^{diode} \ll R_{off}^{diode}$. Then when Q1 and Q2 both have the gate control signal applied and are conducting, the voltages V1 and V2 will be as in (4.1) and (4.2).

$$V1 = V_f^{IGBT} + I_L R_{on}^{IGBT} \quad (4.1)$$

$$V2 = -V_f^{diode} - I_L R_{on}^{diode} \quad (4.2)$$

If the current flows in the opposite direction such that $I_L = -I_L$, then the voltages are also reversed. From (4.1) and (4.2) it can be seen that the voltage drop across the devices will increase proportionally as current increases. Also, as the current decreases towards zero the voltage drop will approach the forward bias voltage of the device. This relation can be used as an accurate method of detecting the direction of current flow through a 4QSW cell. Therefore, current flow direction detection is possible for all current levels rather than depending on a current transducer.

4.3 Two-Step Semi-Soft Commutation Controller (SSCC) Development

Semi-Soft commutation is a necessary process in an MC to maintain current continuity so that large destructive voltage transients do not occur. As the MC is often connected to an inductive load the current must not be interrupted. Such a situation will result in a sudden change in current in a relatively small period of time, and will result in large voltage transients. Commutation is a process which requires accurate measurements of current flow and time to ensure that the constraints of conduction within an MC, as outlined in Section 2.7, are not violated. The consequences of violating the rules may be catastrophic, which can result in the destruction of the MC and the external systems that are connected.

The semi-soft commutation controller (SSCC) model is developed in the PSCAD®

/ EMTDC^{TM} environment for use in the MC system. The system is a finite state machine with a *flow diagram* consisting of twenty-five complex states which are shown in Appendix C (along with the Fortran developed source code). The SSCC model relies on a number of parameters which are:

Current change settling time is the minimum time that the modulated output current waveform can change polarity. This adds hysteresis to the time that the current can change direction after the last polarity change. It prevents the system from becoming unstable during transients.

Device make connection time is the minimum time that the IGBT gated signal can be applied and the device begins to conduct. This parameter governs the period of overlap time in a commutation.

Device break connection time is the minimum time in which the IGBT can stop the current from flowing after the gate signal is removed. This parameter ensures that no IGBT is gated within the specified time after a gate signal is removed from another IGBT that will cause a *shoot-through* condition.

Current change threshold is the parameter that governs the maximum current level that both IGBTs are gated within a 4QSW cell. This is to ensure that a current direction change occurs smoothly and continuity is maintained. If the absolute value of the current is greater than this parameter, only the conducting IGBT gate signal is applied.

Each SSCC is developed to control one output phase by switching the three input phases. For a 3×3 matrix converter, three SSCCs are needed to control all the 4QSW cells. There are two outputs from the SSCC model, each with a dimension of three channels, as shown in Figure 4.9. One output supplies gate signals for the positive

conducting IGBTs in the three 4QSW cells and the other supplies the negative IGBTs. Combined, there are six output channels that coordinate three 4QSW cells as per the two-step semi-soft commutation scheme. There are two inputs into the model, with one used for the three switch signals and the other used for current measurement.

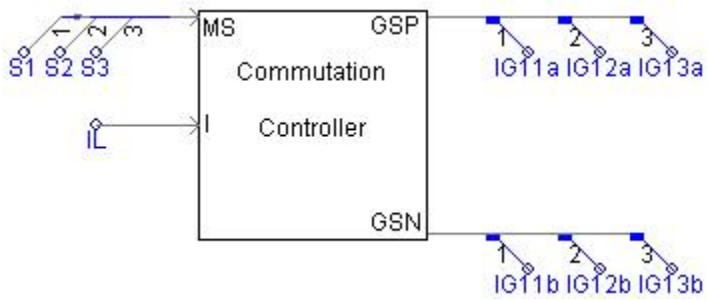


Figure 4.9: SSCC model in PSCAD® / EMTDC^{TM} environment

To verify the functionality of the SSCC model a simulation exercise has been conducted using the circuit shown in Figure 4.10. The objective of the simulation is to confirm that the SSCC is able to perform commutation between the three input phases and also allow current to change polarity while maintaining continuity in the output current. The resulting output waveforms and six IGBT gate signals are shown in Figure 4.11.

Figure 4.11 also shows the different states of the SSCC model. Having developed the SSCC as a finite state machine ensures that there are no output errors which may cause a constraint violation as each state has only one possible output. The model can only move to another state when governing conditions are satisfied. Figure 4.12 shows a plot with a zoomed timescale of the commutation process between IG11a and IG12a where the overlap in the gate signals is shown. Figure 4.13 shows a current polarity change. When the absolute value of the output current drops below the

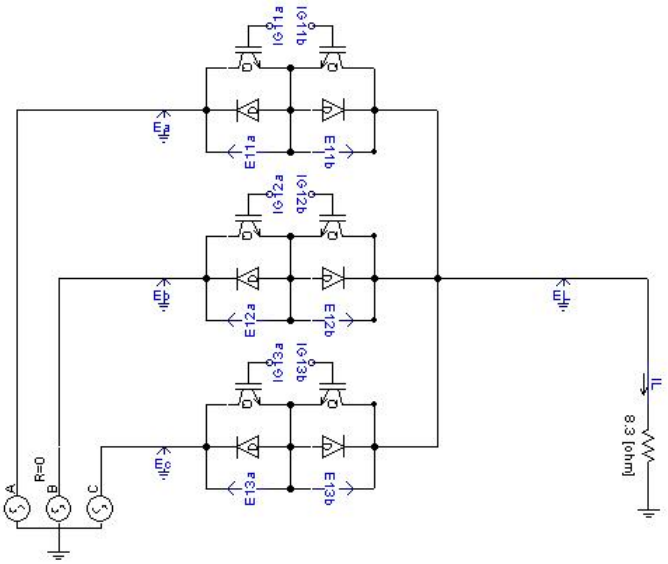


Figure 4.10: SSCC model simulation circuit in PSCAD® / EMTDC^{TM} environment

current change threshold (which is 100mA), in this case both IGBTs (IG13a and IG13b) are gated. When the current increases past the threshold current just the conducting IGBT (IG13b) is gated and the system returns to steady state.

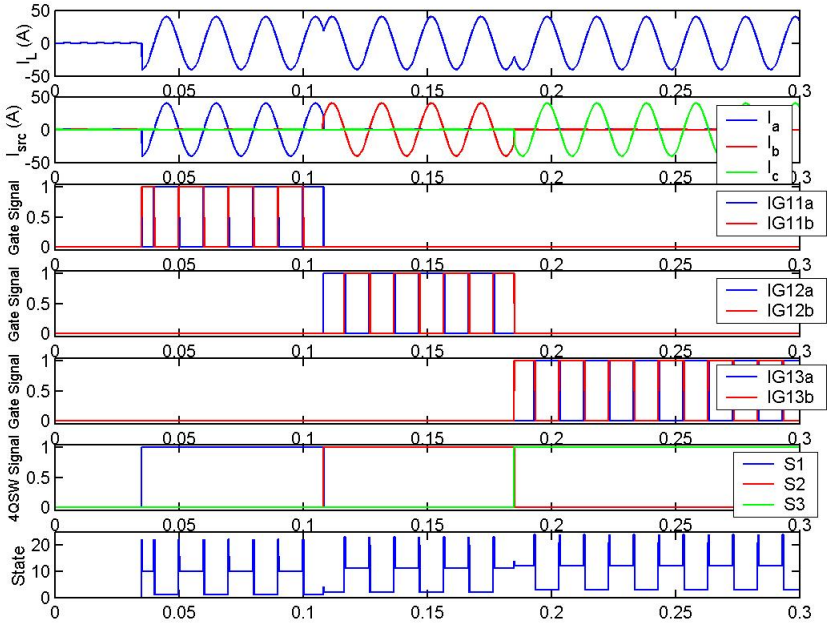


Figure 4.11: SSCC model simulation results

Using these results it can be concluded that the SSCC model is capable of performing current commutation in a three phase input to single phase output converter. Current continuity is maintained throughout the simulation thus avoiding any destructive transients from occurring.

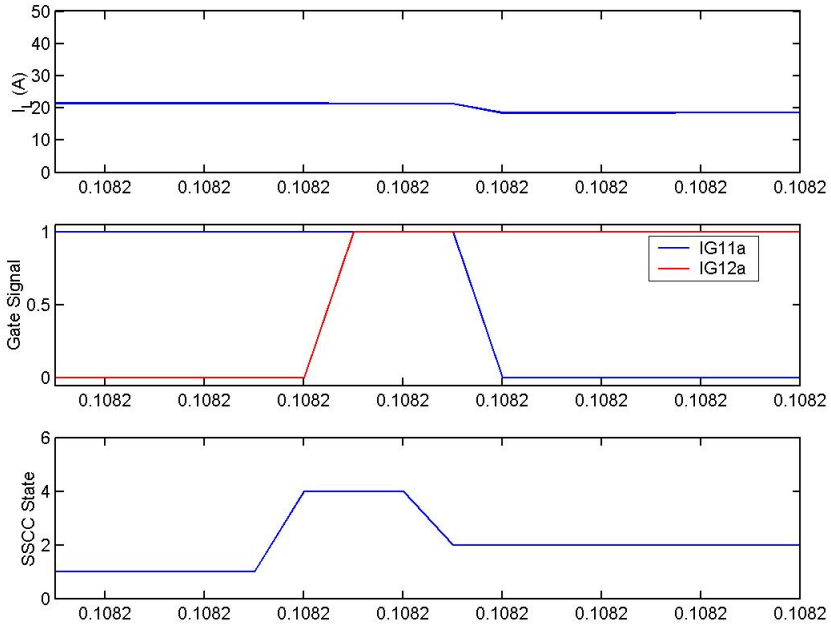


Figure 4.12: SSCC commutation overlap

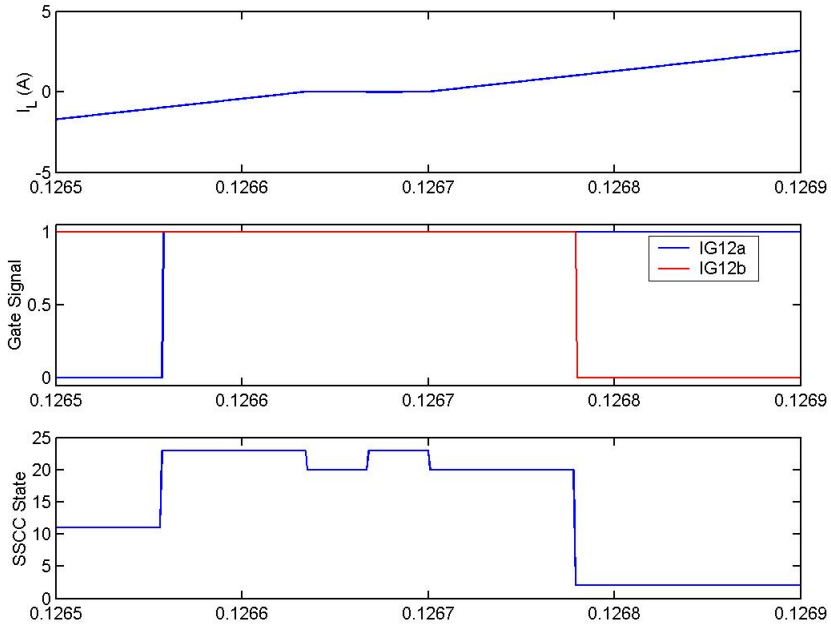


Figure 4.13: SSCC current polarity change

4.4 Matrix Converter Filtering

To prevent high frequency switching harmonics being injected into the grid as a result of modulation, a second order Low Pass (LP) filter is applied at the input of the switching matrix. The unwanted harmonics are centred around the switching frequency of 20 kHz which is much greater than the fundamental frequency of 50 Hz. Hence the low order harmonics are relatively small in magnitude. This means that the size of the filter may be vastly reduced. A typical LP filter used in an MC application is shown in Figure 4.14. The resistor in the filter also adds damping to the circuit to prevent oscillations.

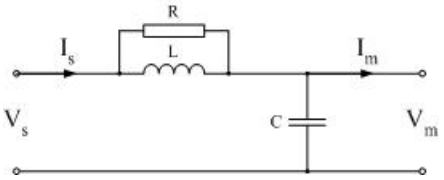


Figure 4.14: Matrix Converter Input Low Pass Filter

The transfer function of the LP filter is

$$\frac{I_s(s)}{I_m(s)} = \frac{Ls + R}{RLCs^2 + Ls + R} \tag{4.3}$$

The unwanted current harmonics are centred around the switching frequency which is $\omega_s = 125.67 \times 10^3$ rad/s, and the grid supply frequency is $\omega_e = 314.2 \leqslant$ rad/s. The filter design must remove the harmonic current components while minimizing the effect on the fundamental current component. Based on (4.3) the poles for the filter are given by (4.4).

$$s = \frac{-L \pm \sqrt{L^2 - 4R^2LC}}{2RLC} \tag{4.4}$$

The pole placement is selected to be approximately midway between the fundamental waveform and the switching harmonics. In this case a pole placement of approximately 16×10^3 rad/s is established. Also it is preferable to select component values such that the system is slightly *underdamped*. The damping ratio of the system can be found using (4.5).

$$\zeta = \frac{1}{2RC\sqrt{\frac{1}{LC}}} \tag{4.5}$$

where $0 \leqslant \zeta < 1$ is underdamped, $\zeta = 1$ is critically damped and $\zeta > 1$ is overdamped [20]. The other consideration is the resonant natural frequency of the filter which should not correspond to the switching frequency of the MC as this will cause the system to become unstable. For this reason it is ideal to select the resonant frequency to be midway between the fundamental grid frequency and the switching frequency. The resonant frequency can be found by using (4.6).

$$\omega_n = \sqrt{\frac{1}{LC}} \tag{4.6}$$

With these considerations and using (4.3) - (4.6), the following filter values in Table 4.2 are selected.

R	4Ω
L	4mH
C	70μF
ω_n	1.8898×10^3 rad/s

Table 4.2: Second order LP filter parameters

From these values a Bode plot can be generated which shows the relative gain (dB) versus frequency (rad/s), as shown in Figure 4.15.

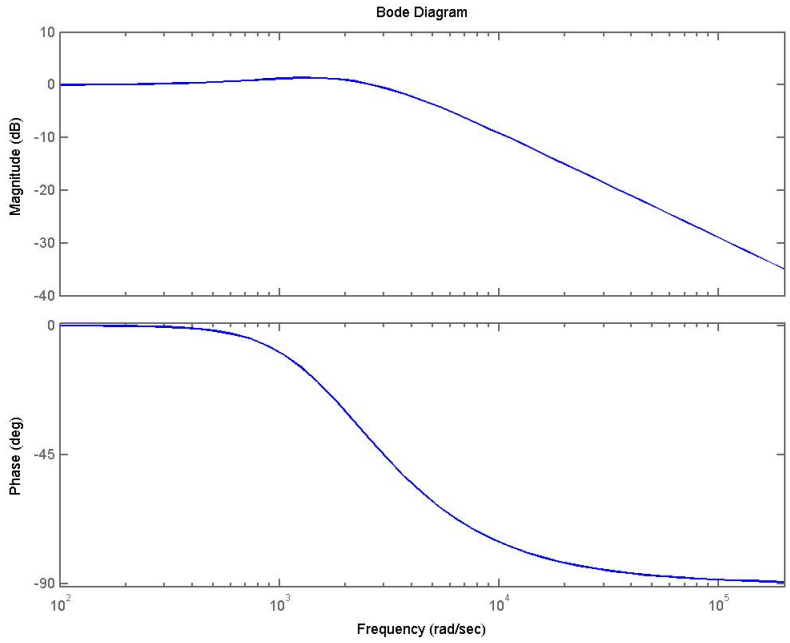


Figure 4.15: Second order low pass filter Bode plot

It can be seen that the filter attenuates the current at frequencies near the switching frequency and has a relatively small effect on the fundamental frequency. Also the phase change at the fundamental is relatively small. This is a desirable quality as the MC is designed such that it is able to control the input current displacement factor and any phase shift would further complicate the design process.

With these considerations, the low pass filter outlined in this section with the defined parameters is selected as the input filter for the matrix converter. The output does not require any filtering as the inductive load that the MC is connected to performs the low pass filtering of the output current waveforms.

4.5 Space Vector Modulation Development

The Space Vector Modulation Controller (SVMC) is designed to produce the switch combination signals at the relevant time for the nine 4QSWs of the matrix converter (MC) as per Section 2.5. The SVM method that has been verified on the MC with the ideal switches is applied to a MC with non-ideal component models. The SVMC controls the 4QSWs via the semi-soft commutation controller (SSCC) which then produces eighteen gate signals to control IGBTs within the nine 4QSWs as shown in Figure 4.16.

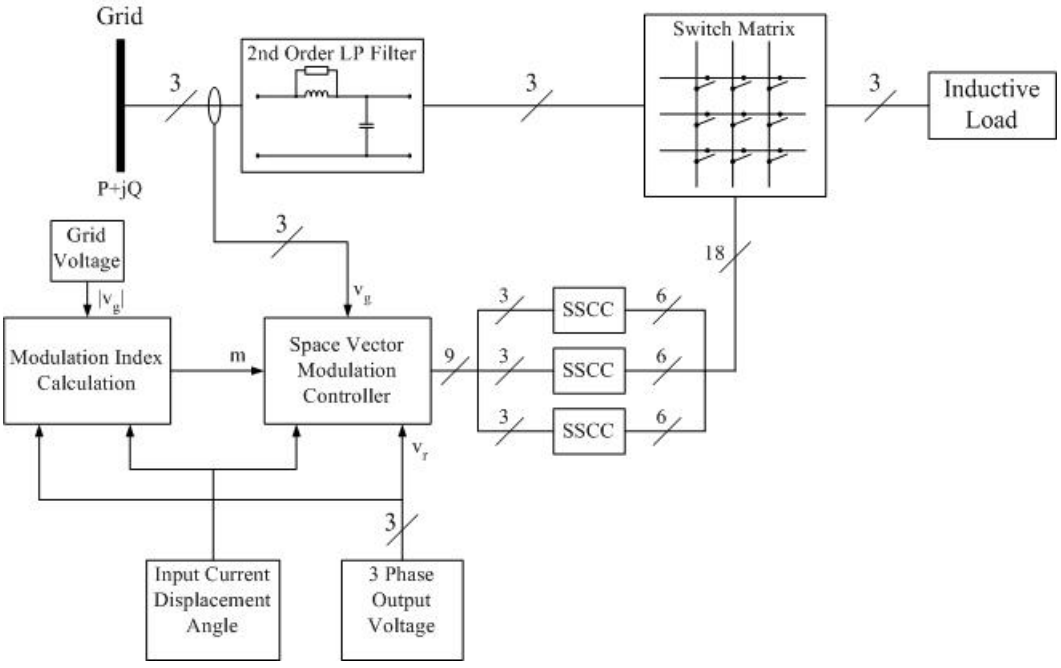


Figure 4.16: Matrix Converter block diagram

The modulation index that is used in the SVMC is calculated by a separate block. This is done so that the modulation index can be externally controlled if needed depending on the control requirements of the system. The modulation index is calculated as a piecewise function:

$$m = \begin{cases} 0, & \frac{2|v_{oL}|}{\sqrt{3}|v_{iL}|\cos\phi_i} \leqslant 0 \\ \frac{2|v_{oL}|}{\sqrt{3}|v_{iL}|\cos\phi_i}, & 0 < \frac{2|v_{oL}|}{\sqrt{3}|v_{iL}|\cos\phi_i} < 1 \\ 1, & \frac{2|v_{oL}|}{\sqrt{3}|v_{iL}|\cos\phi_i} \geqslant 1 \end{cases} \tag{4.7}$$

Use of the piecewise function (4.7) ensures that the modulation index stays within the linear voltage range of the MC. The input voltage vector magnitude used in the modulation index is determined by the input nominally rated RMS value of the grid voltage which is set at the commencement of the simulation, as indicated by Figure 4.16. This is done to ensure stability of the system during voltage transients. The SVMC applies the SVM algorithm during every switching period incorporating the following steps:

1. Read output voltage waveform values
2. Read input current waveform values
3. Calculate output voltage space vector
4. Calculate input current space vector
5. Calculate Modulation Index
6. Calculate SSV pairs
7. Calculate θ_{iL} and θ_{oL}
8. Calculate duty cycles $d_{km}, d_{lm}, d_{kn}, d_{ln}$ and d_0
9. Calculate switch times from duty cycles
10. Derive switching combinations from SSV pairs for each switch time
11. Apply switching combinations at relevant switch times

The SSCC adds latency time to the application of the switch combinations. To allow for this a time hysteresis or deadband which is set in the SVMC is added to the minimum time the controller applies a combination. This allows the SSCC to apply the gate signals effectively. If the duty time of an SSV is less than the deadband time, the SSV is not applied and the time is redistributed to the other SSVs to be applied during the switching cycle.

As there are three zero vectors, the selection of the vector that is applied depends on the dominant input phase in the previous switch combination. Every SSV combination uses no more than two input phases to generate the output three phase voltage. This means that in each SSV one of the input phases is applied to two output phases. Once the dominant input phase is determined, the zero vector that uses the same input phase is applied. This means that only one output phase connection needs to be changed. The other advantage is that if a zero vector is needed for an extended period of time, the same zero switch combination is applied. This ensures that the number of switch state changes is minimized. The SVMC code used for this is given in Appendix D. The input parameters of the SVMC PSCAD® / EMTDC^{TM} model (Figure 4.17) are:

1. Switching Frequency (Hz);
2. Deadband Time (s); controls the minimum time a SSV combination is actuated.
3. Switching Cycle (Asymmetric / Symmetric); changes between Asymmetric and Symmetric switching cycle.

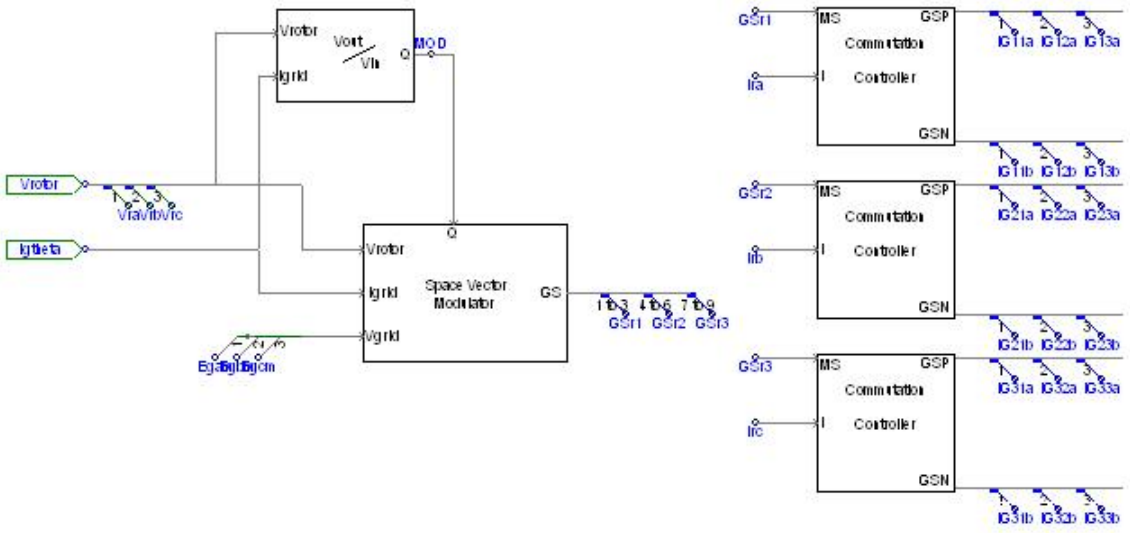


Figure 4.17: Matrix Converter control block

4.6 Matrix Converter Evaluation

A simulation with a passive load connected to the output of the MC was carried out to evaluate the SVM control system and MC. An inductive load consisting of a series connected 6Ω resistor and a 10mH inductor was applied to the output of the MC as shown in Figure 4.18. A test to evaluate the variable frequency functionality of the MC was conducted and the results are shown in Figure 4.19.

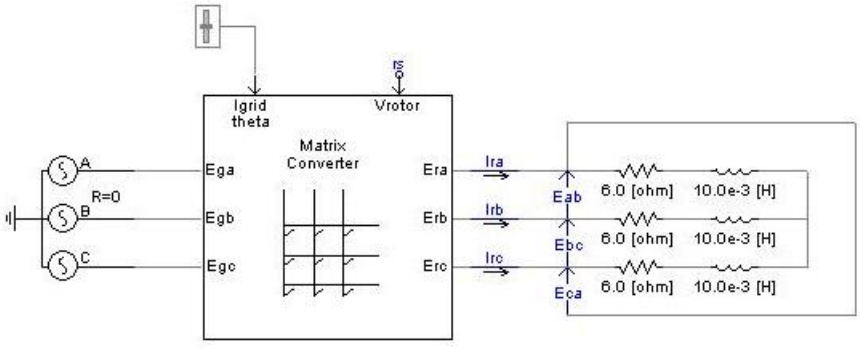


Figure 4.18: Matrix Converter passive load simulation circuit

The waveform in Figure 4.19 was generated using a switching frequency of 20kHz

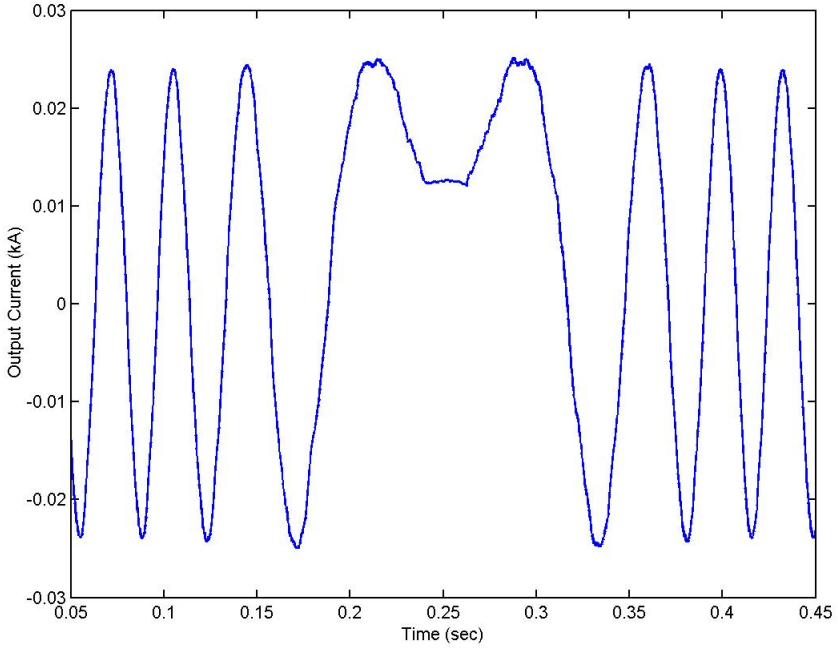


Figure 4.19: MC 30Hz to -30Hz frequency variation output current waveform

and the modulation index set to 0.86. The MC output frequency was ramped from 30Hz to 0Hz and then to -30Hz, which gives a negative sequence waveform during $0.1 \leqslant t \leqslant 0.4$ s. In another test at a constant output frequency of 10Hz and with the modulation index set to 0.86, the calculated input and output current Total Harmonic Distortion (THD) was found to be approximately 5% and 8%, respectively. The input current and output current waveforms for this test are shown in Figures 4.20 and 4.21. When the modulation index was reduced to 0.1 the input current THD and output current THD was observed to be 6% and 100%, respectively.

The input current displacement angle was also adjusted to 0.524 rads and -0.524 rads in Figures 4.22 and 4.23, respectively. This shows that the input power factor can be adjusted and is independent of the power factor of the load connected

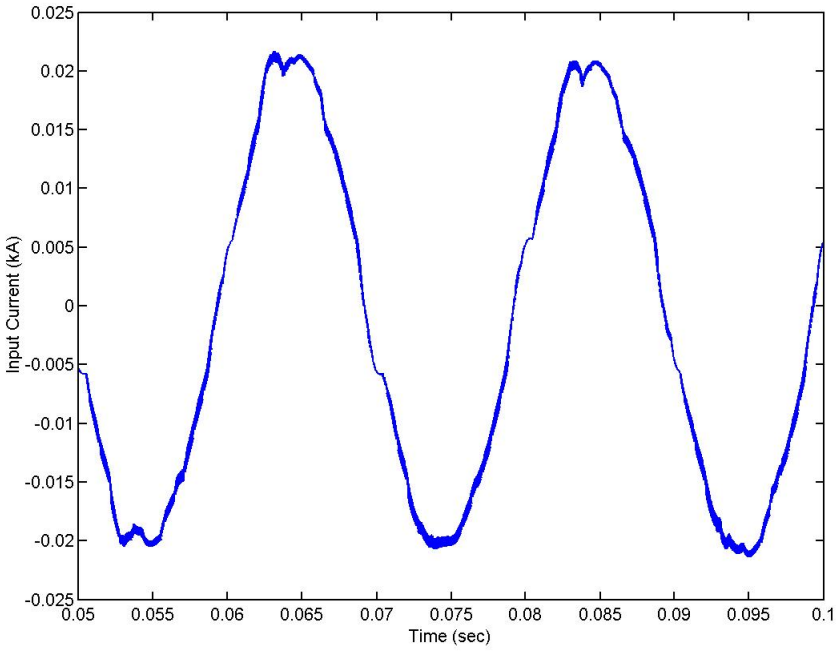


Figure 4.20: 50Hz input current waveform with $f_s = 20$kHz

to the output of the MC.

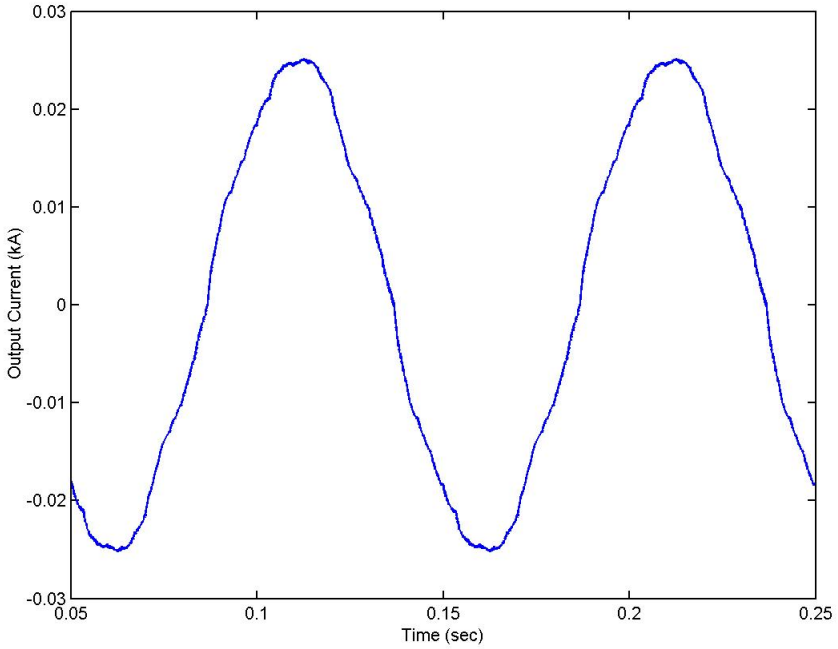


Figure 4.21: 10Hz output current waveform with $f_s = 20$kHz

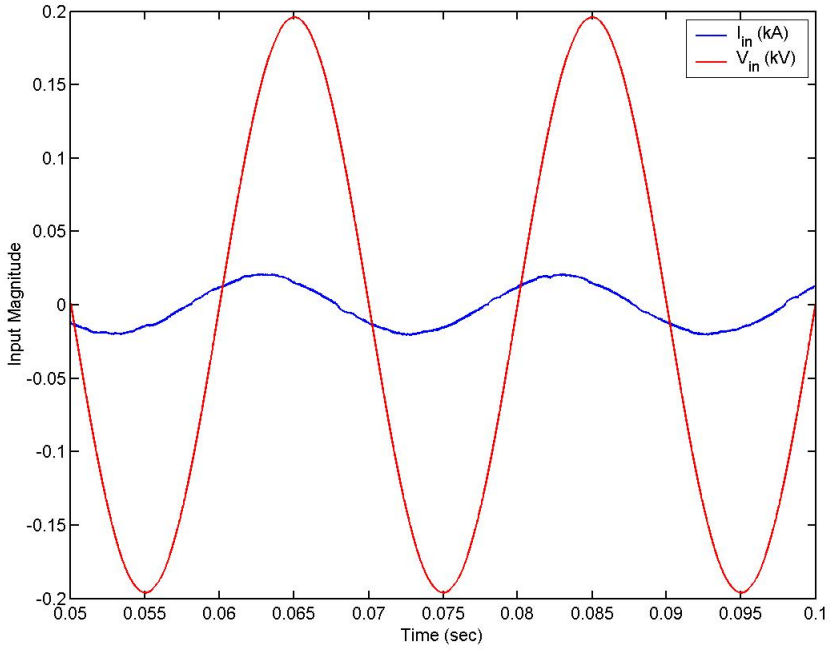


Figure 4.22: 50Hz input current and input voltage $\phi_i = 0.524$ rads

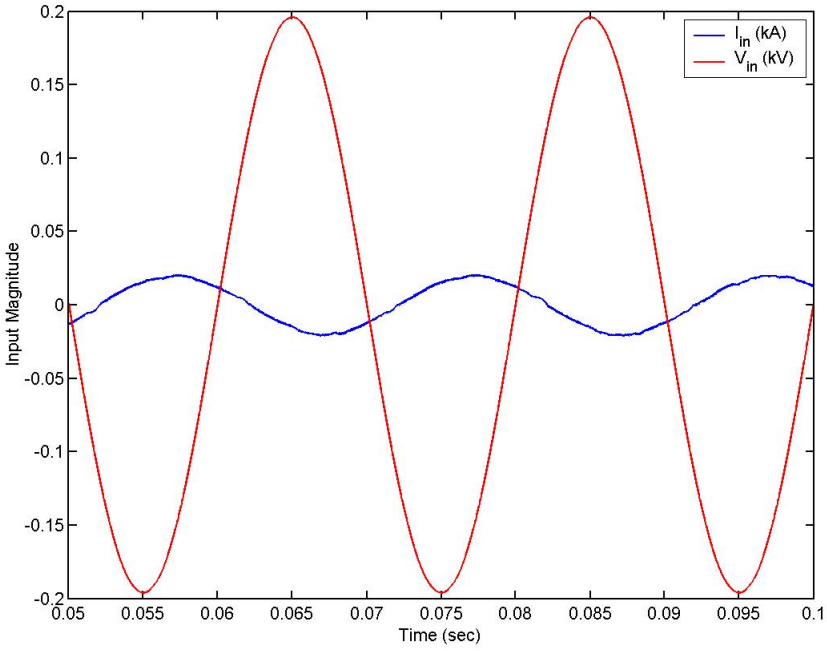


Figure 4.23: 50Hz input current and input voltage $\phi_i = -0.524$ rads

Using a switching frequency of 2kHz and a modulation index set to 0.86, the input current waveform in Figure 4.24 and the output waveform in Figure 4.25 were obtained. Frequency spectrum plots of these waveforms were generated and the calculated input THD was found to be 7% and the output current THD was 8%. These results can be further improved by using a higher switching frequency than 2kHz. The single-sided Fast Fourier Transform (FFT) analysis was obtained as shown in Figures 4.26 for input current and 4.27 for output current. This illustrates that as the switching frequency decreases the modulated waveform THDs increase. Figure 4.28 gives the FFT plot of the output voltage, which has a THD of 39%. The high frequency components are evident in the output voltage frequency spectrum. The corresponding voltage waveforms are shown in Figures 4.29 and a close-up view in 4.30. This shows that the inductive load on the output removes many of the high frequency components in the voltage waveform to reduce the THD of the output current, as is highlighted in Section 4.4.

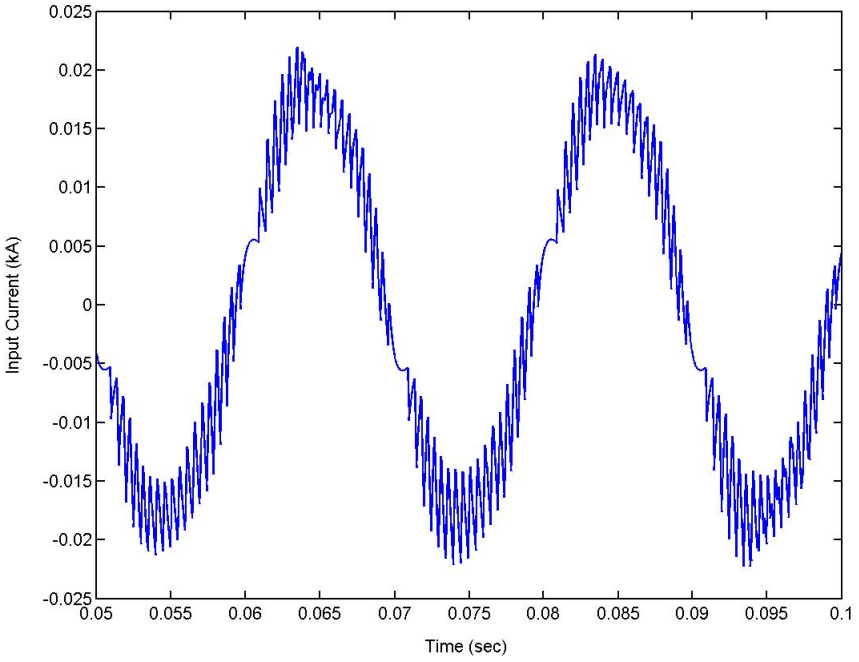


Figure 4.24: 50Hz input current waveform with $f_s = 2$kHz

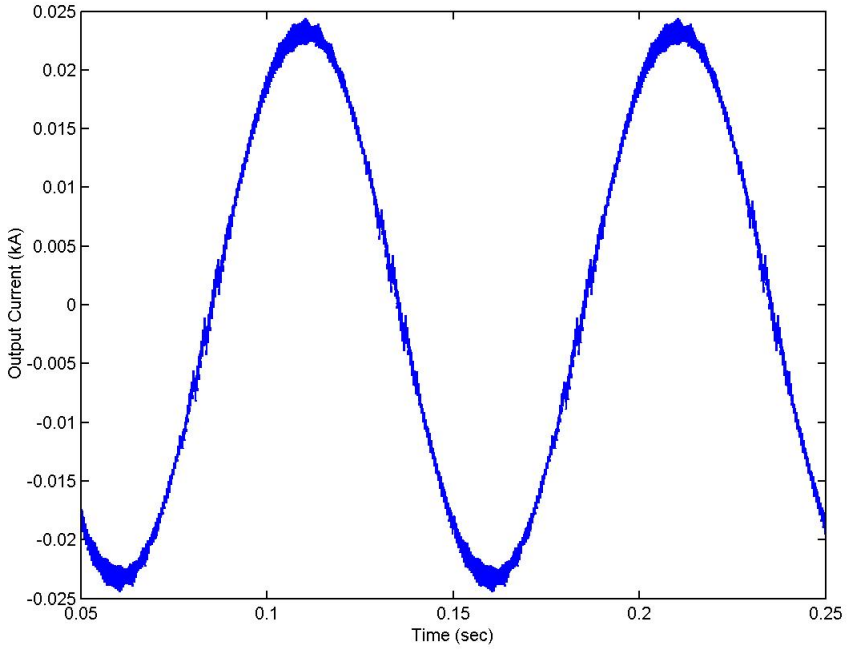


Figure 4.25: 10Hz output current waveform with $f_s = 2$kHz

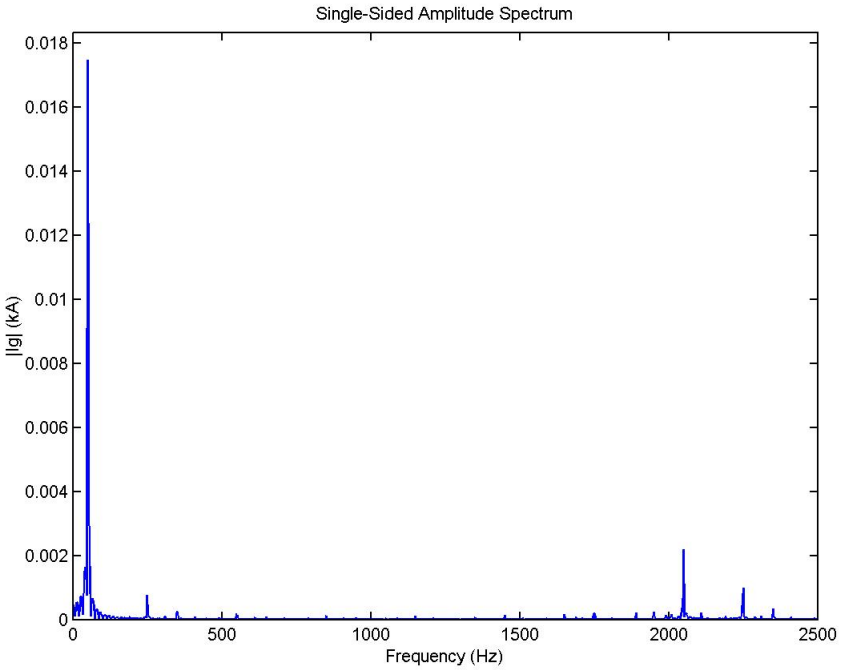


Figure 4.26: 50Hz input current frequency spectrum with $f_s = 2$kHz

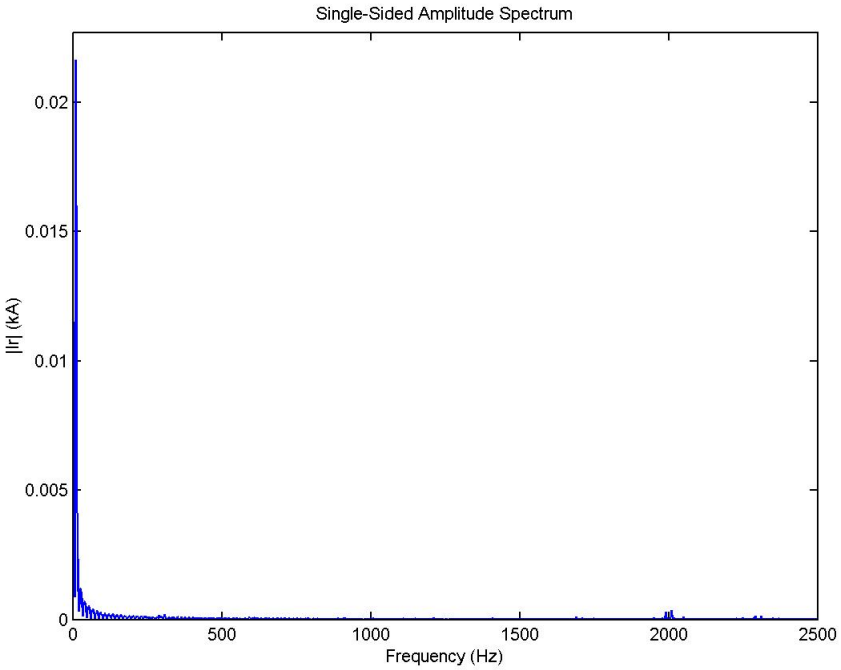


Figure 4.27: 10Hz output current frequency spectrum with $f_s = 2$kHz

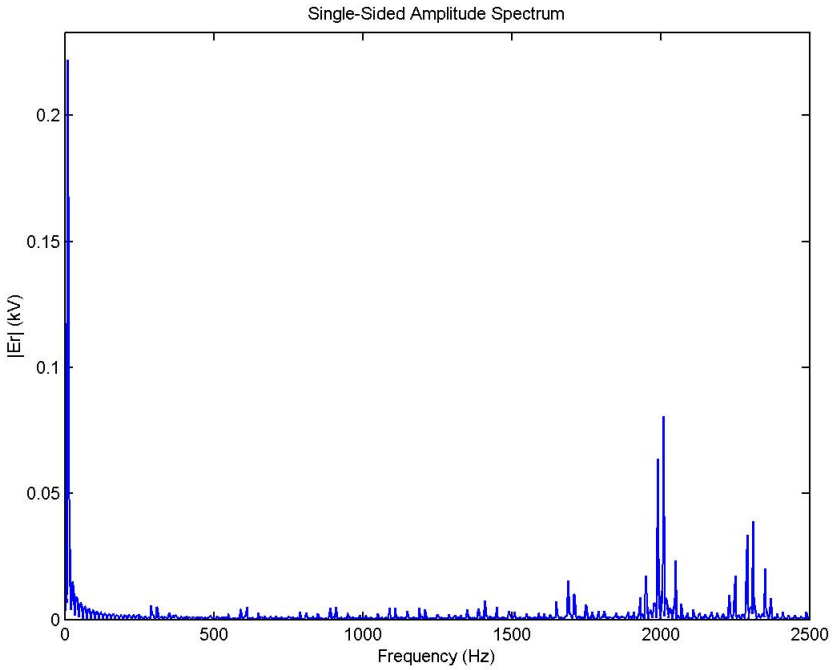


Figure 4.28: 10Hz output voltage frequency spectrum with $f_s = 2$kHz

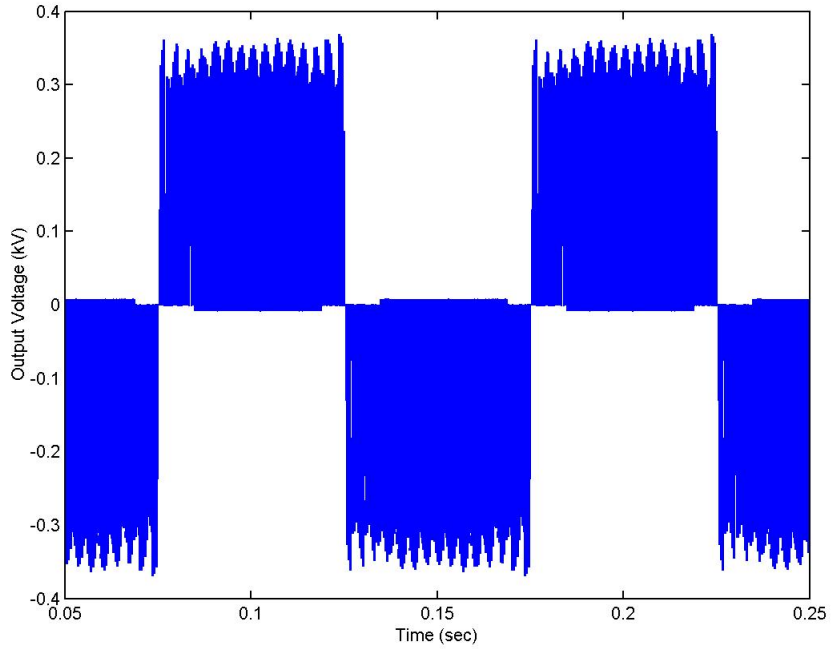


Figure 4.29: 10Hz output voltage waveform with $f_s = 2$kHz

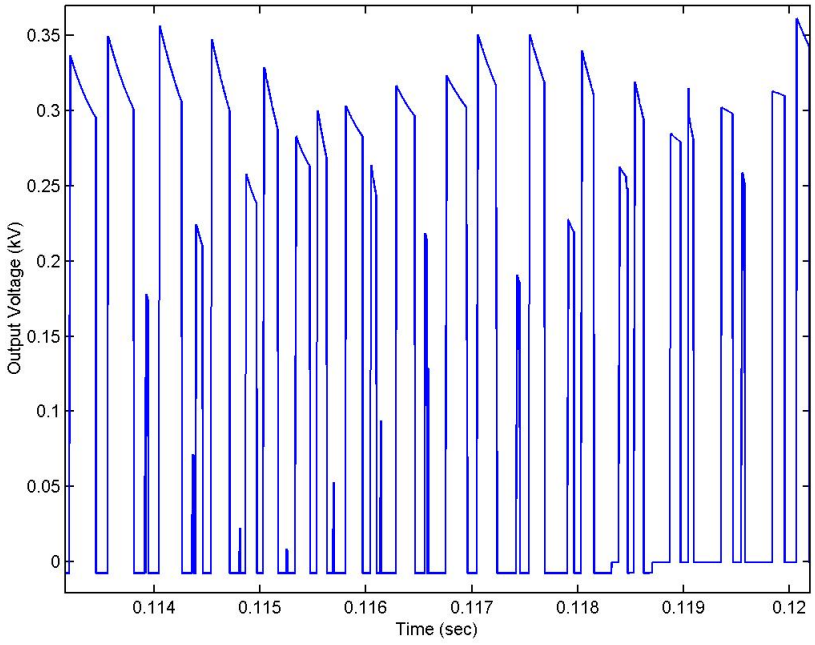


Figure 4.30: Close-up of the 10Hz output voltage waveform

The final simulation conducted on the MC was an Asynchronous link test to illustrate the regenerative capabilities of the MC. An independent three phase AC source was connected as the load via inductors. Inverting mode in the MC can be achieved via a number of different avenues which reverse phases the MC output voltage or input current, with a negative magnitude difference in the MC output voltage with respect to the independent source. The system was tested in inverting mode with the resulting input voltage and current waveforms shown in Figure 4.31. These results prove that the MC is capable of four quadrant power flow control.

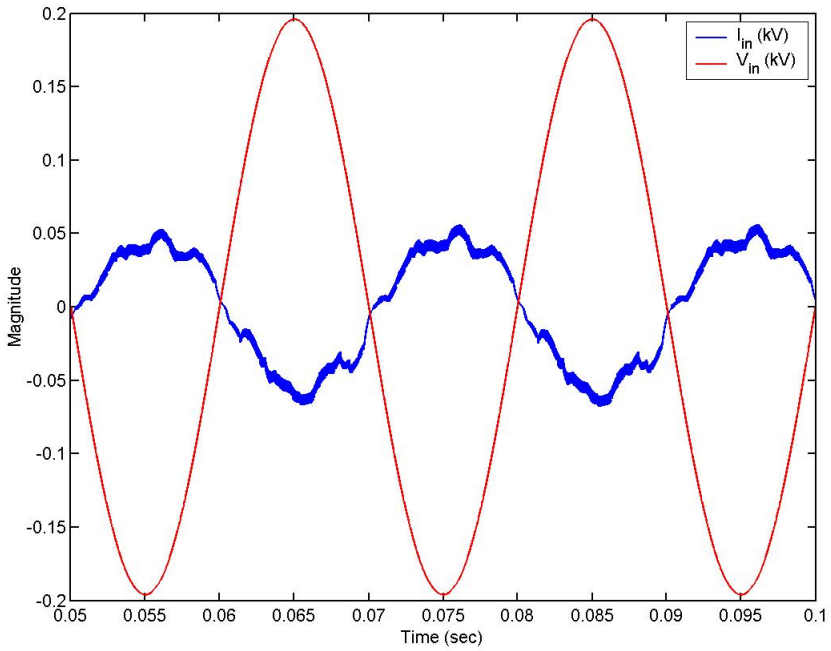


Figure 4.31: 50Hz input current and input voltage $\phi_i = 3.1416$ rads (Inverting Mode)

4.7 Summary

The development process of a non-ideal MC model was outlined in this chapter. SVM PWM theory was verified using ideal component models, and IGBT and diode models V-I characteristics were examined and then applied in a common-emitter 4QSW configuration. The switch characteristics show that it is capable of explicit current direction control which makes it suitable to be used in a four quadrant power converter.

A semi-soft commutation controller (SSCC) was developed and simulated to show that current continuity is maintained throughout all conditions found in an MC system. The results showed that the SSCC is capable of effective current commutation, however, it did incur a latency time throughout its operation between states. The latency time found was considered to be negligible and will not detract from the operation of an MC. A second order low pass filter was also derived to remove unwanted input current switching harmonics from the waveform.

The space vector modulation controller was also developed to be used in conjunction with the SSCC. Simulations of the MC system which includes the 4QSW, SSCC, low pass filter and SVMC, have been applied to a passive load. Results of the simulations were observed covering voltage, frequency and input current displacement factor. The THD of input and output current was calculated and the frequency spectrum shown. Increasing the switching frequency of the system was shown to reduce the harmonic distortion of the output current. The system was also tested in regenerative mode to prove that four quadrant power flow is possible.

Chapter 5

Matrix Converter Excited DFIG Development

5.1 Wound Rotor Induction Machine Model

As outlined in Section 3.1, the DFIG requires a four quadrant power converter in the rotor circuit for effective operation in both sub-synchronous and super-synchronous modes. In Chapter 4, a four quadrant Matrix Converter (MC) was developed and simulations were conducted to verify the power flow capabilities required for the DFIG to achieve effective control. In this chapter the development of the MC excited DFIG system is outlined.

In the PSCAD® / EMTDC^{TM} library there are a number of developed and tested models which can be used in the simulation package. One of these models is a Wound Rotor Induction Machine (WRIM) which is needed in the DFIG system to convert the mechanical power of the generator into electrical power.

To verify the PSCAD® / EMTDC^{TM} WRIM model a series of tests were per-

formed and the results obtained were compared to data referenced from [19]. The first test was a Free acceleration simulation with the rotor circuit short circuited on the PSCAD® / EMTDC^{TM} WRIM model. The parameters for a 1.68 MVA WRIM are detailed in [19] and shown in Appendix G, Table G.1. Using the 1.68 MVA WRIM parameters in the model, the free acceleration simulation produced the torque-speed curve shown in Figure 5.1. Stator and rotor current measurements have also been obtained and are shown in Figures 5.2 and 5.3, respectively. The 1.68 MVA PSCAD® / EMTDC^{TM} model was also tested with a pump load simulation. The pump load is characterized by the equation $T_L = k\omega_r^2$ where $k = \frac{T_{rated}}{\omega_{rated}^2}$ which is 0.985pu for the 1.68 MVA WRIM. The torque speed curve produced by the pump load simulation is shown in Figure 5.4. The RMS stator current obtained during the simulation is shown in Figure 5.5. From this it can be seen that the steady state stator current is 441.9A. This is approximately 105% of the rated current defined in Table G.1. The torque speed curve given in [19] and the WRIM model match.

The PSCAD® / EMTDC^{TM} WRIM model was tested using smaller machine of 7.5kW. The parameters were obtained from [1], and are given in Appendix G, Table G.2. A free acceleration simulation was performed on the model and the torque speed curve was obtained as shown in Figure 5.6. From these results it can be seen that the model has been widely tested and provides good agreement with the differential equations for a WRIM as used in [19].

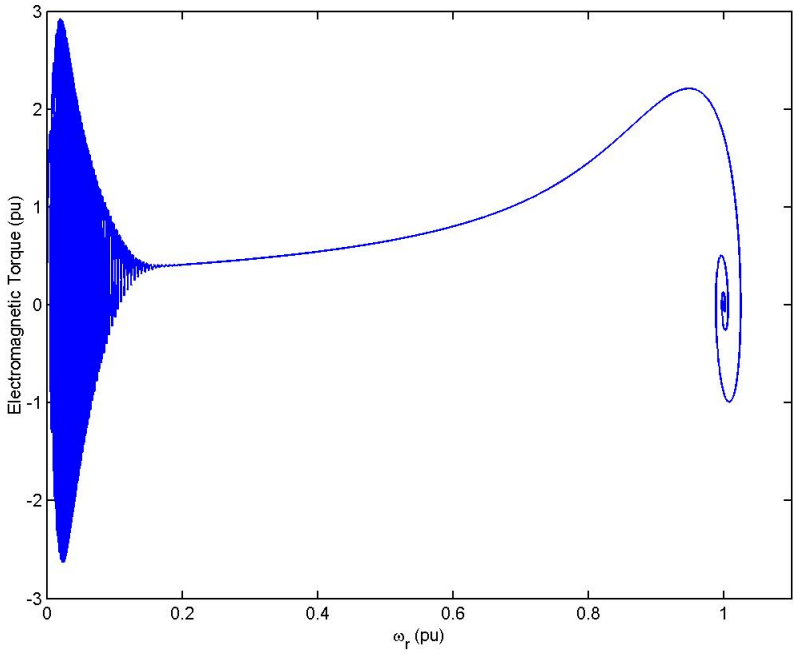


Figure 5.1: 1.68 MVA WRIM torque - speed curve

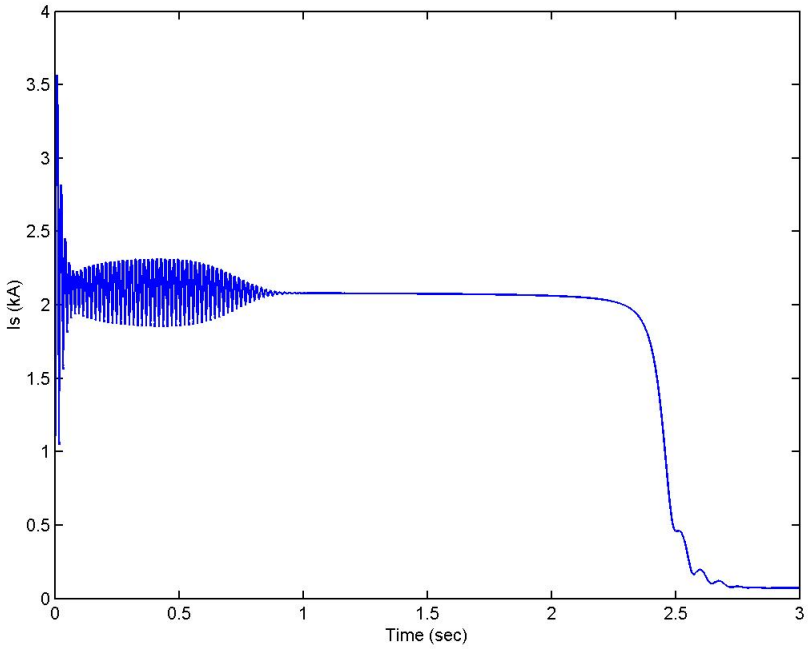


Figure 5.2: 1.68 MVA WRIM stator current

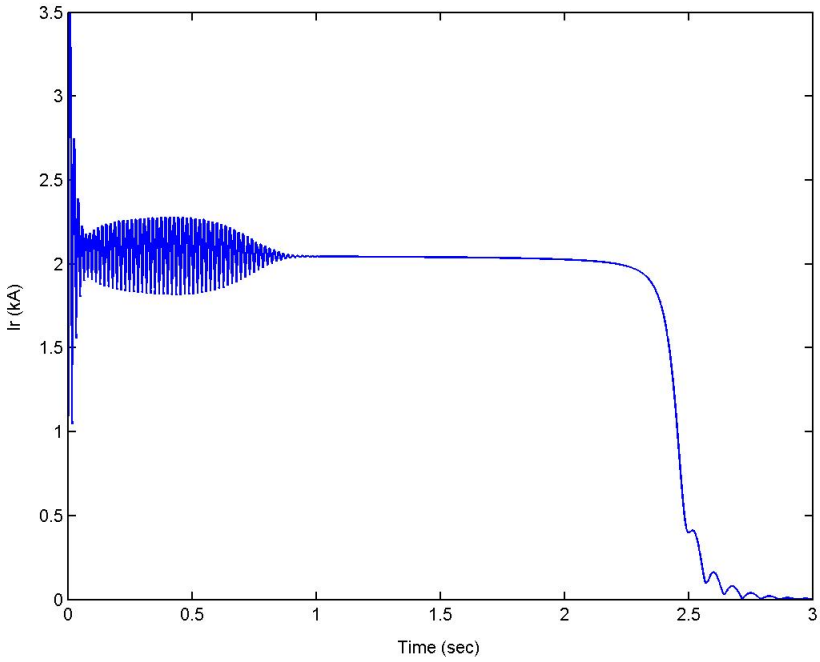


Figure 5.3: 1.68 MVA WRIM rotor current

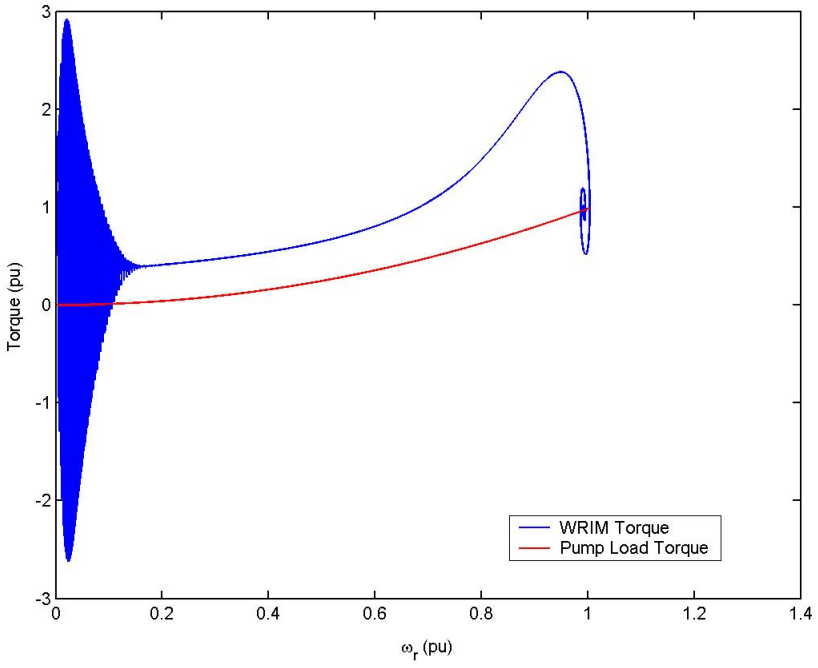


Figure 5.4: 1.68 MVA WRIM with pump load; electromagnetic and load torque vs speed

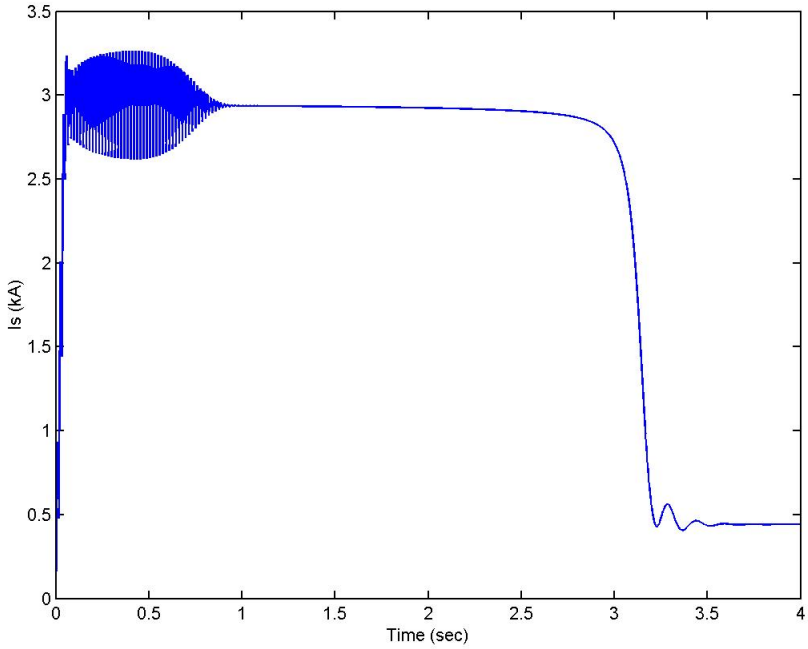


Figure 5.5: 1.68 MVA WRIM with pump load stator current

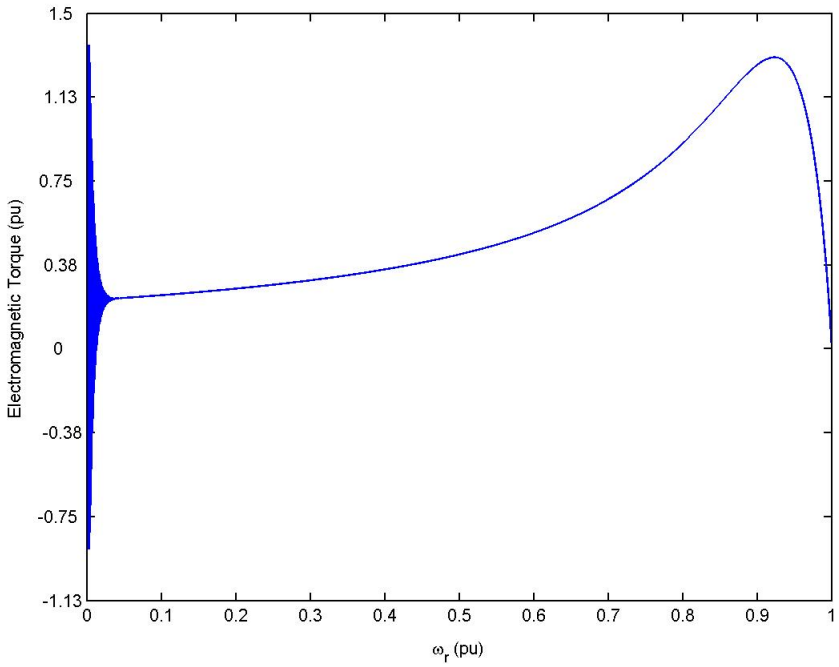


Figure 5.6: 7.5kW WRIM torque - speed curve

5.2 Doubly-Fed Induction Generator Control

The DFIG active and reactive output is controlled through the regulation of rotor currents in the stator flux reference frame. The control theory outlined in Section 2.2.1 shows that independent active and reactive control is possible. The aim of this thesis is to investigate the viability of achieving this level of control using a matrix converter to perform current regulation in the rotor circuit.

The decoupling components are given by (2.11) and (2.12) to provide effective decoupled control between i^e_{qr} and i^e_{dr}. To prove the decoupled control in the stator flux reference frame, an ideal voltage source is applied to the rotor circuit and the simulation was conducted using different generator speeds and current reference values. The WRIM used in the simulation was the 1.68 MVA machine tested in Section 5.1. The DFIG was subjected to step changes to i^e_{qr} and i^e_{dr}. The synchronous quadrature and direct rotor currents i^e_{qr} and i^e_{dr}, during the step change in i^{e*}_{qr} are shown in Figure 5.7, and the corresponding rotor phase current in the stationary reference frame is in Figure 5.8. Figure 5.9 shows the synchronous quadrature and direct rotor currents when the system is subjected to a step change of i^{e*}_{dr}. From these results it can be seen that the decoupled DFIG controller functions correctly with an ideal voltage source in the rotor circuit.

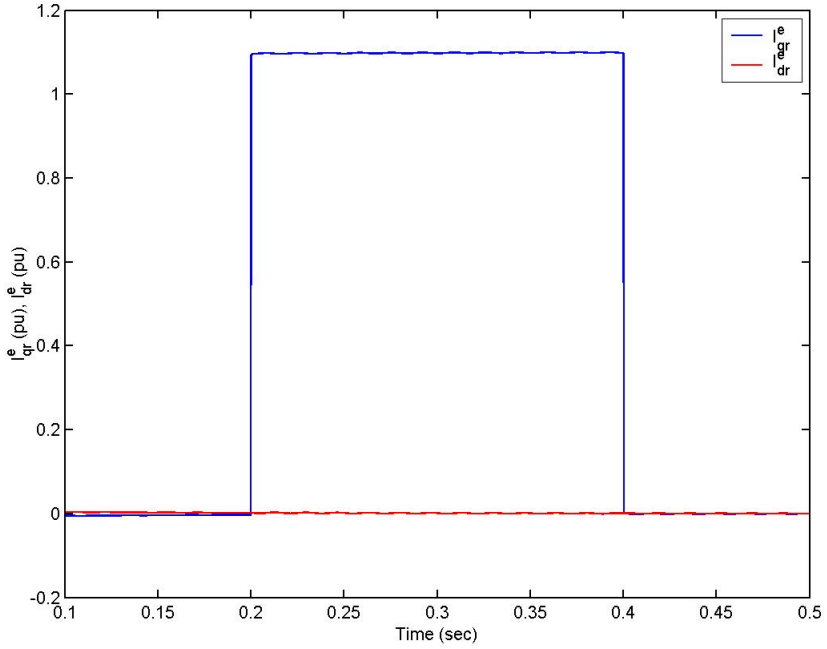


Figure 5.7: Synchronous rotor current during step change to i^e_{qr} with ideal voltage source in rotor circuit

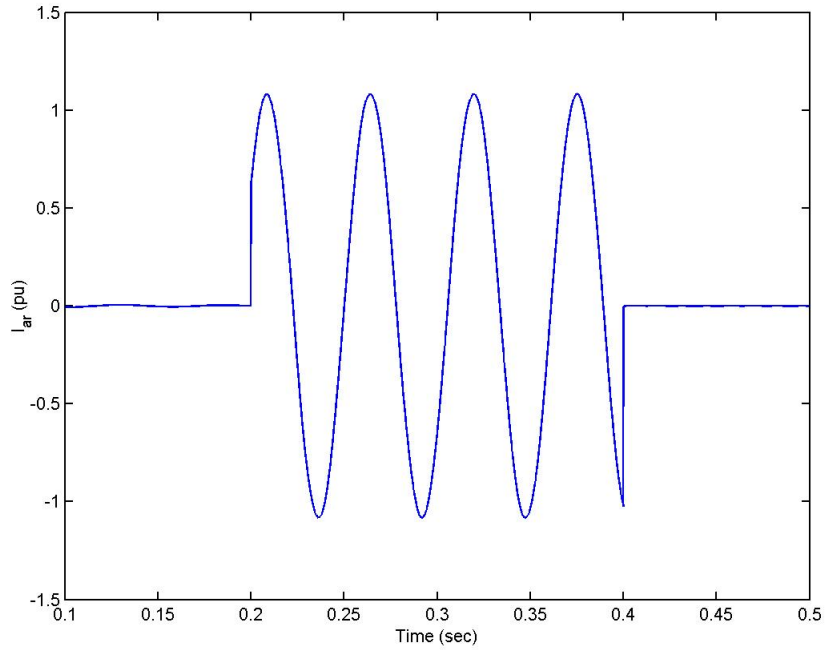


Figure 5.8: Non-synchronous rotor current during step change to i^e_{qr} with ideal voltage source in rotor circuit

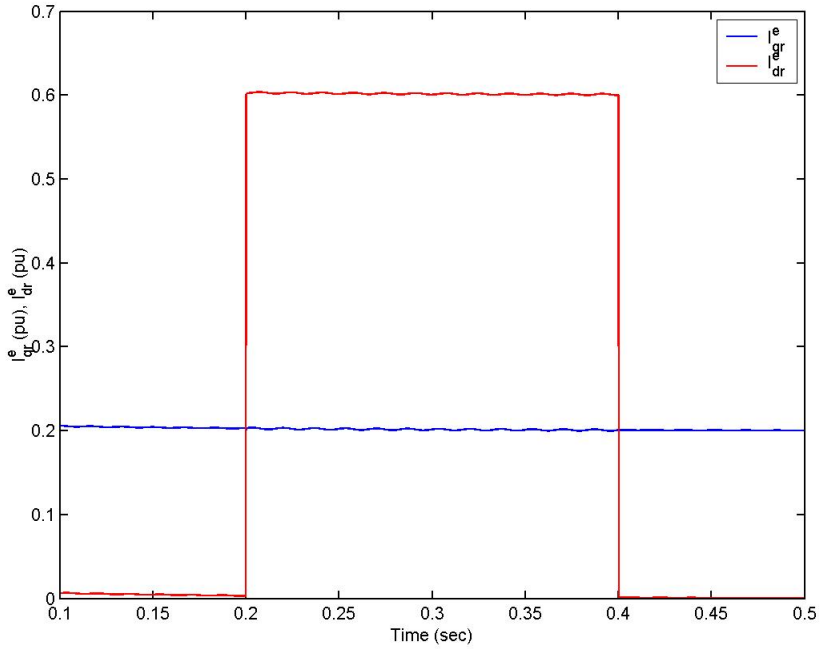


Figure 5.9: Synchronous rotor current during step change to i^e_{dr} with ideal voltage source in rotor circuit

The control system that is used for the DFIG system is shown as shown Section 2.2.1 in Figure 2.4. The shaft speed of the WRIM model is directly set to simplify the system for the purposes of minimizing the calculation time in the simulations.

The feedback loop calculates the d-q rotor currents and shifts them into the stator flux reference frame relative to the rotor via the $e^{-j\theta}$ block. From there the error value for the rotor currents are calculated and are passed through PI controllers with a proportional gain of 30 and integral time of 0.0005 to produce voltage errors which were obtained and found by Ziegler-Nichols method. The errors are compensated for by using the decoupling components and are then shifted back into the stationary reference frame. The three phase current values are then calculated and passed to the matrix converter control system.

Based on this model, simulations have been conducted to evaluate the effectiveness of the MC to regulate the rotor current in a DFIG system. The system is subjected to step changes in the reference values i^e_{qr} and i^e_{dr} and also a ramp change in w_r. These results are tabled in Subsections 5.2.2 to 5.2.3.

5.2.1 1.68 MVA DFIG Step Change in i^e_{qr}

In this simulation the DFIG system has been subjected to a step change in i^e_{qr} from 0 to 1.1pu. The reference values i^e_{dr} was set to 0pu with the generator shaft speed (ω_r) set to 1.3pu. In Figure 5.10 the step response in the synchronous reference frame of i^e_{qr} and i^e_{dr} and the active and reactive output are shown, while Figure 5.11 displays the corresponding rotor phase current. Figure 5.12 shows the output current of the DFIG supplied to the grid and the grid voltage. In this plot the increase in output current and phase change is evident, as a result of the step change of i^e_{qr}. Also, the active

power output of the DFIG increases significantly while the reactive power remains unchanged with the step change in i^e_{qr}.

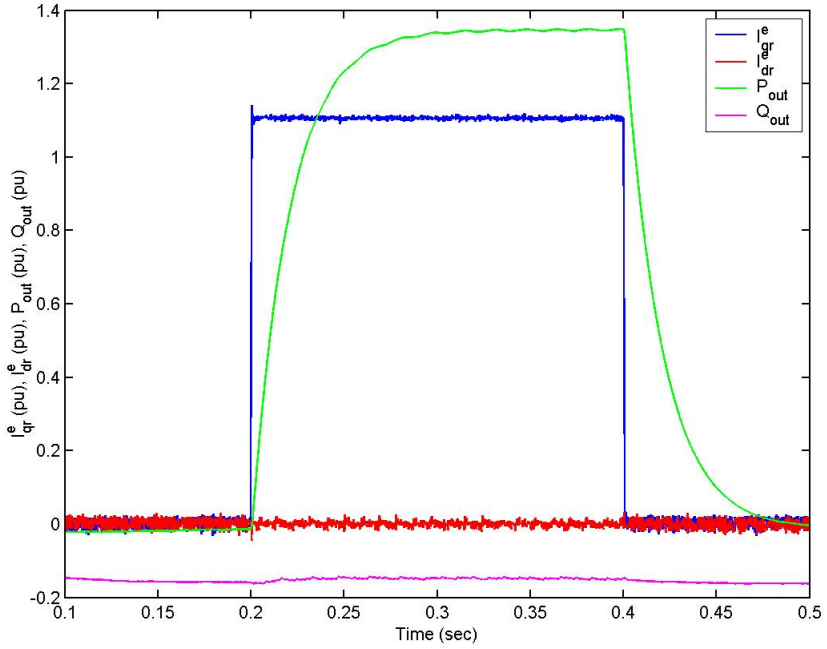


Figure 5.10: 1.68 MVA DFIG synchronous rotor current and power output during a step change to i^e_{qr}

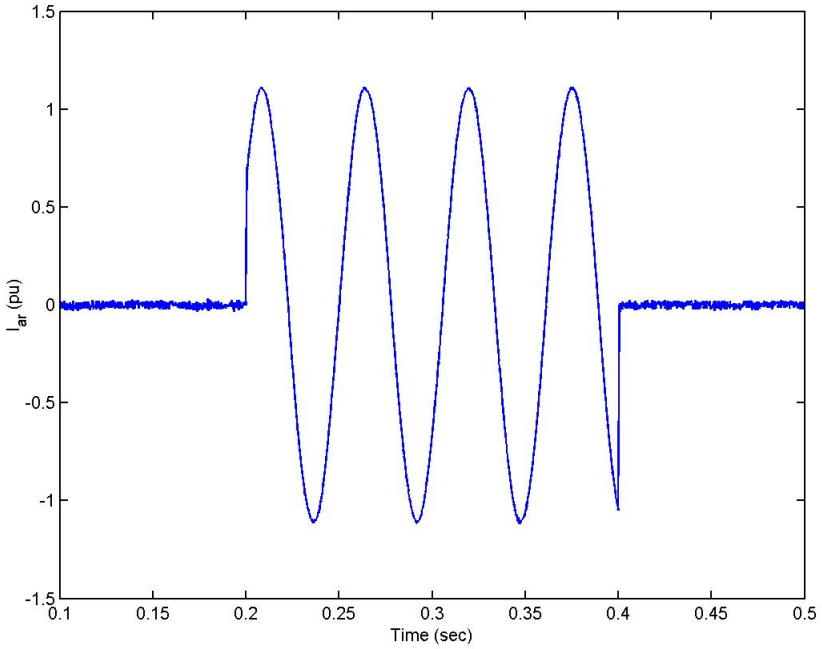


Figure 5.11: 1.68 MVA DFIG rotor phase current during a step change to i^e_{qr}

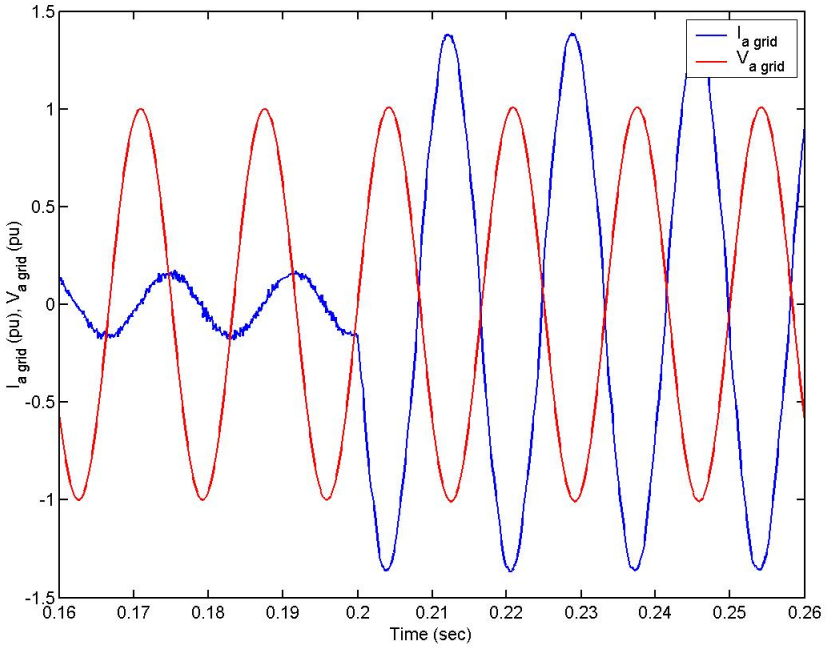


Figure 5.12: 1.68 MVA DFIG grid current and grid voltage during a step change to i^e_{qr}

5.2.2 1.68 MVA DFIG Step Change in i^e_{dr}

This simulation imposed a step change in i^e_{dr} from 0 to 0.6pu. The reference values i^e_{qr} was set to 0.2pu and the generator shaft speed (ω_r) set to 1.3pu. In Figure 5.13 the step response of i^e_{qr} and i^e_{dr} in the synchronous reference frame with the active and reactive output are shown. The reactive power output of the generator has a large increase with the step change of i^e_{dr}. Figure 5.14 shows the rotor phase current during the same period and Figure 5.15 is a zoomed in plot of the stator current with respect to the grid voltage. During the step change in i^e_{dr} the phase change in the output grid current is evident.

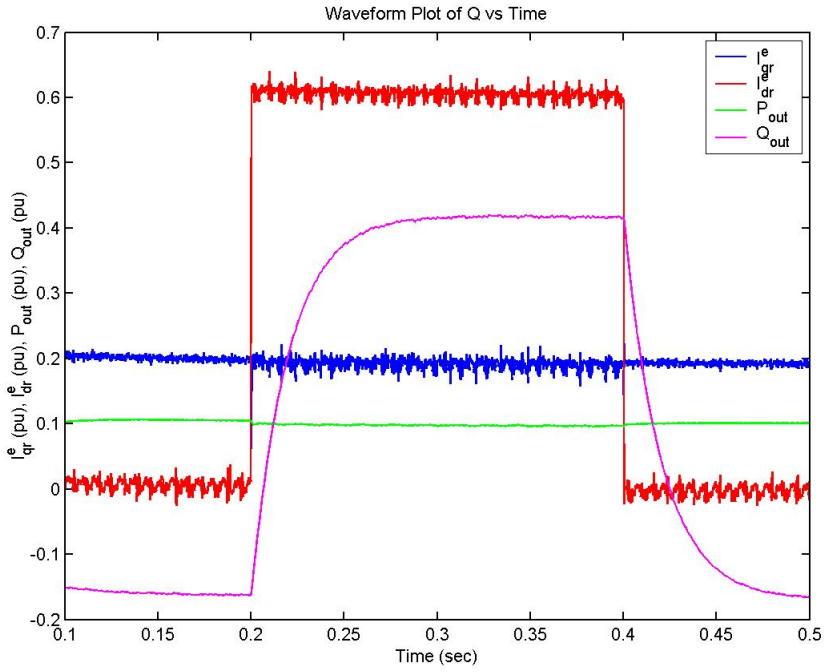


Figure 5.13: 1.68 MVA DFIG synchronous rotor current and power output during a step change to i^e_{dr}

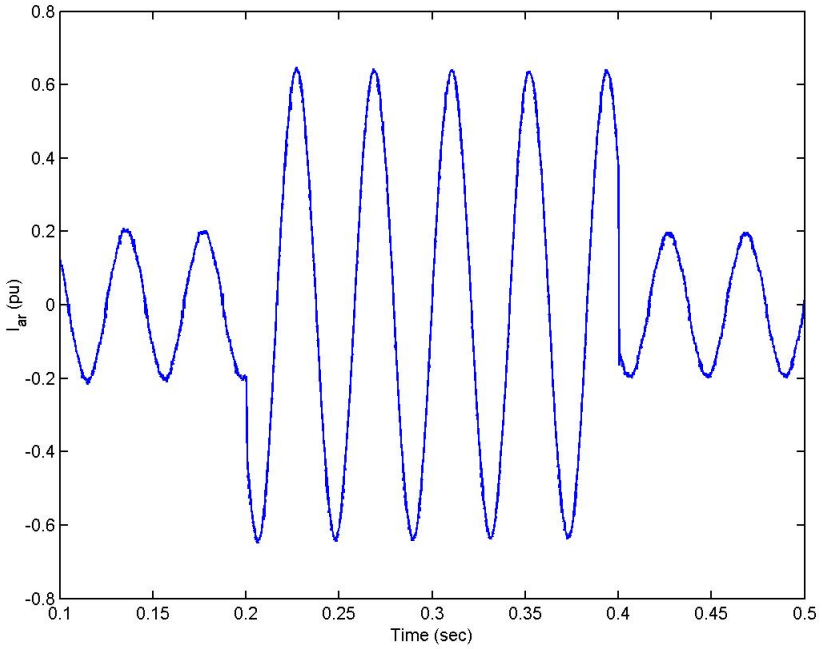


Figure 5.14: 1.68 MVA DFIG rotor current during a step change to i_{dr}^{e}

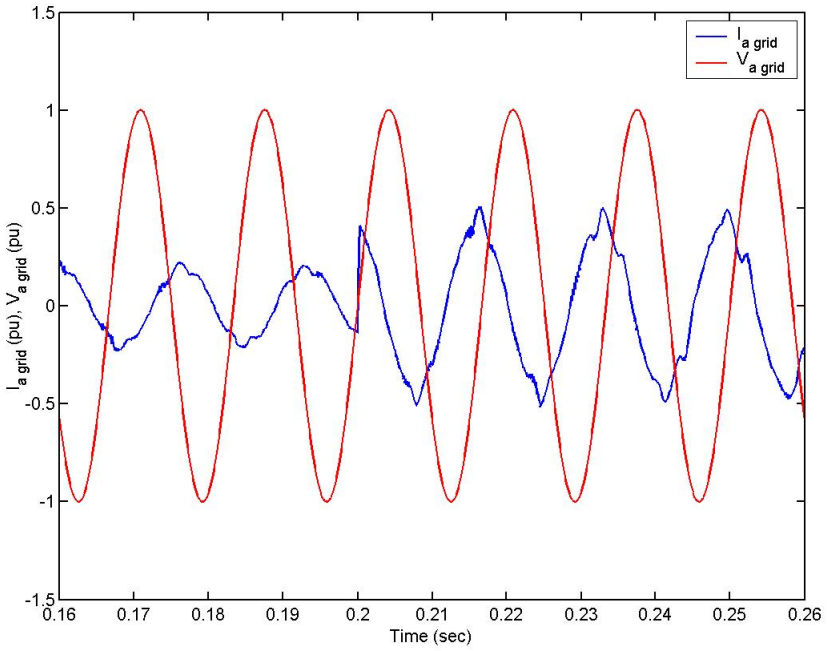


Figure 5.15: 1.68 MVA DFIG grid current grid and voltage during a step change to i_{dr}^{e}

5.2.3 1.68 MVA DFIG Ramp Change in ω_r

For this simulation the DFIG is subjected to a constant ramp change in generator shaft speed (ω_r) from 0.88pu to 1.12pu over 5s. The input reference values i^e_{qr} and i^e_{dr} were held constant at 0.9pu and 0.2pu, respectively. Figure 5.16 gives the rotor phase current and generator slip speed ω_{slip} during the ramp change. When the slip speed is between -0.12pu and zero (i.e. super-synchronous mode) the rotor current shows a negative sequence current characteristic as the rotor is supplying power to the grid. For slip speeds above zero when the generator is in sub-synchronous mode the rotor draws power from the grid.

In Figure 5.17 the power through the DFIG system is plotted as a function of generator slip speed (ω_{slip}). In super-synchronous mode the power being supplied to the grid through the matrix converter is evident. At synchronous speed the matrix converter power is approximately -0.0185pu which would indicate that power is being drawn from the grid. Ideally at synchronous speed the current being fed to the rotor should be zero. However, the discrepancy can be explained by the conduction and switching losses in the matrix converter as well as the copper losses which add internal resistive components to the WRIM. Also, the rotor current changes from a positive to negative sequence while the frequency approaches and then passes through zero. During the ramp change the grid current THD was found to be 4.8% and the single sided spectrum produced from the FFT is shown in Figure 5.18. From these results it can be seen that decoupled active and reactive power control in a 1.68 MVA DFIG is achieved.

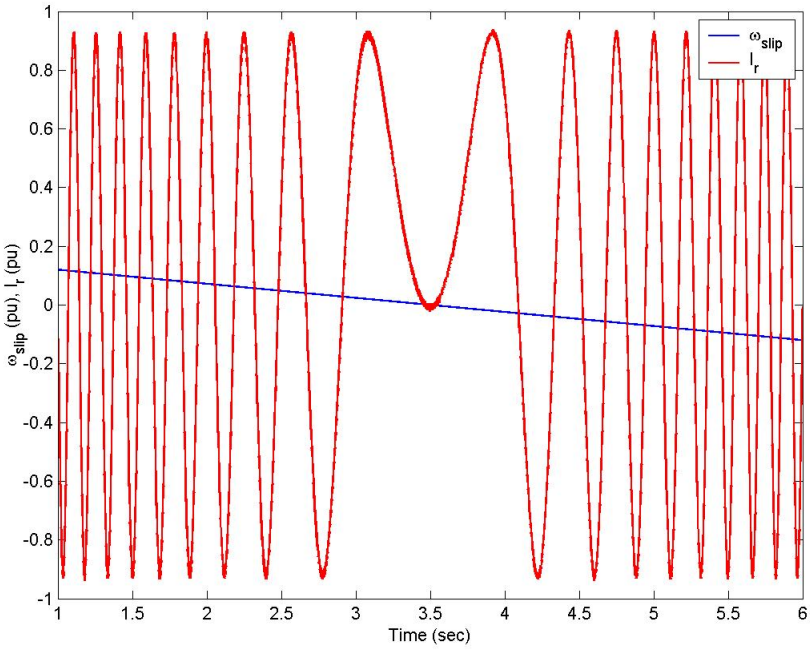


Figure 5.16: 1.68 MVA DFIG slip speed and rotor phase current during a ramp change in ω_r

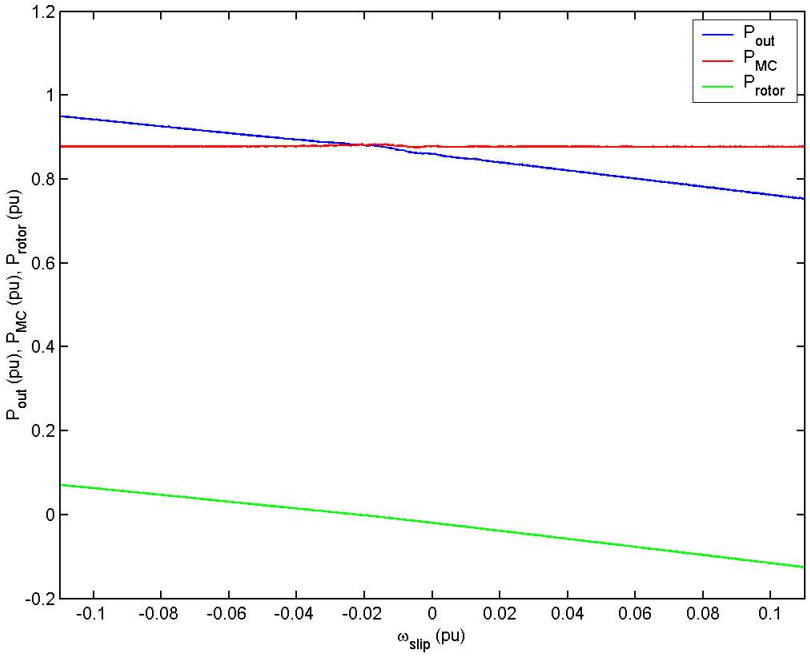


Figure 5.17: 1.68 MVA DFIG power distribution in DFIG during a ramp change in ω_r

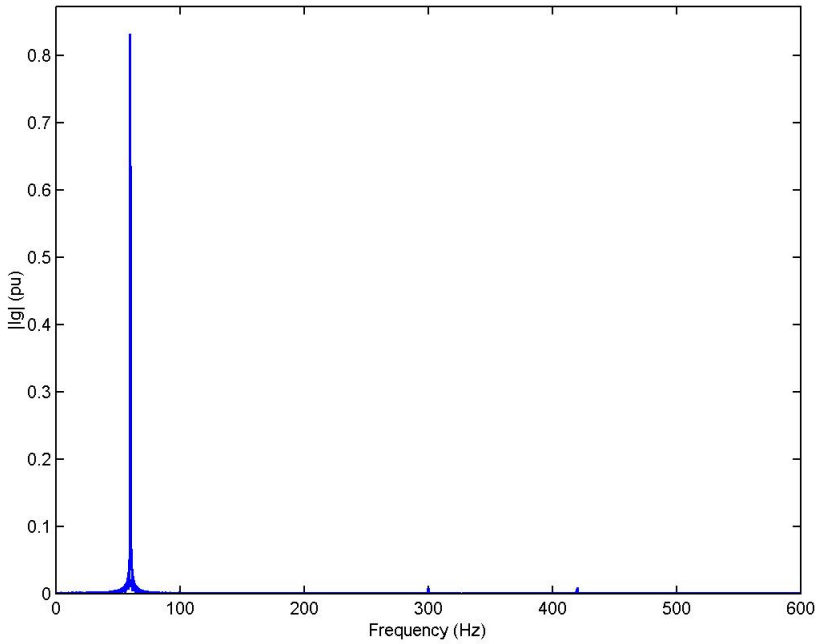


Figure 5.18: FFT spectrum of 1.68 MVA DFIG output grid current during ramp change to ω_r

5.2.4 Simulation of a 7.5kW DFIG

Simulations similar to Subsections 5.2.2 to 5.2.3 were also conducted on a DFIG using a 7.5kW WRIM. Figure 5.19 shows the step response to i^e_{qr} and the active power output once again increases significantly. In Figure 5.20 the results obtained from the step response of i^e_{dr} are shown. The changes in reactive power are minor while the active power output experiences a large increase. The DFIG is subjected to a ramp change in ω_r as shown in Figure 5.21, which plots the resulting rotor phase current. Similar to Subsection 5.2.3, the rotor current changes from a positive to negative sequence while the frequency approaches and then passes through zero. Based on these results it can be seen that decoupled active and reactive power control has been achieved in a non-ideal MC excited DFIG for a 7.5kW system.

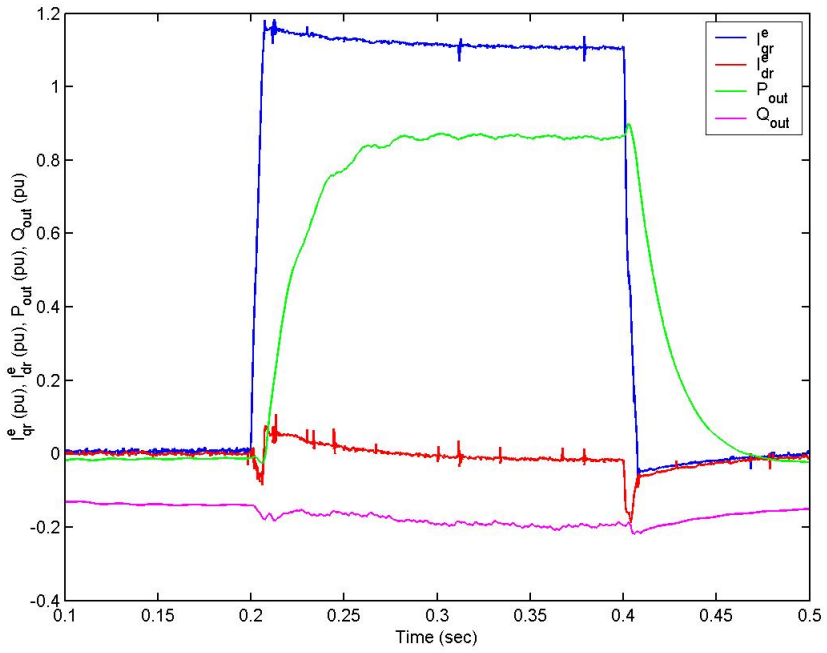


Figure 5.19: 7.5kW DFIG synchronous quadrature and direct rotor current, active and reactive power output during a step change to i^e_{qr}

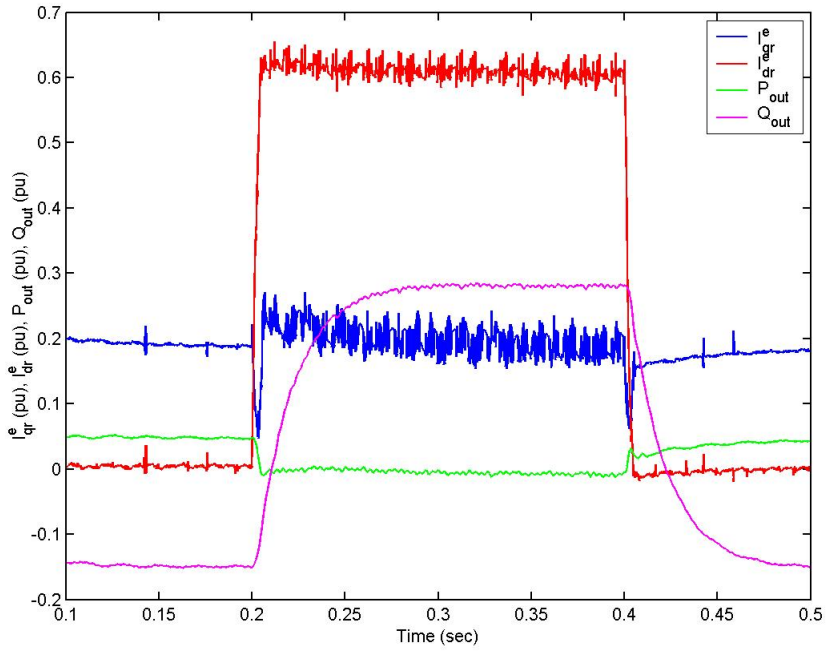


Figure 5.20: 7.5kW DFIG synchronous quadrature and direct rotor current, active and reactive power output during a step change to i^e_{dr}

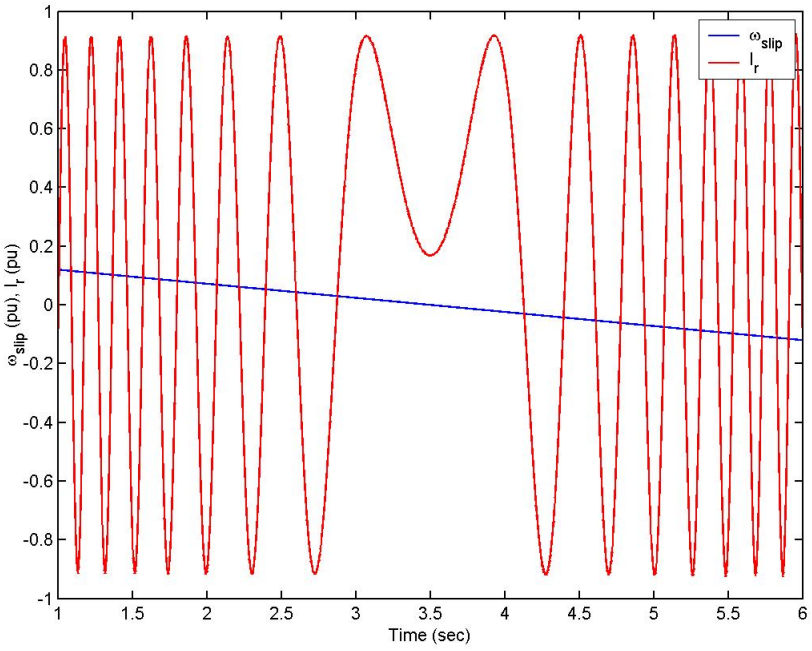


Figure 5.21: 7.5kW DFIG slip speed and rotor phase current during ramp change to ω_r

5.3 Maximum Power Point Tracking Algorithm

The objective of the maximum power tracking is to extract the maximum amount of power from the wind that is available to the generator at any given time. This is done by varying the rotational velocity of the turbine to suit the wind velocity such that the maximum amount of power can be extracted from the wind. For a horizontal-axis wind turbine the mechanical power (P_t) that is available is given by (5.1).

$$P_t = \frac{1}{2} C_P(\lambda, \beta) \rho \pi R^2 V_w^3 \tag{5.1}$$

where C_P is the coefficient of power, ρ is the air density, R is the radius of the wind turbine and V_w is the wind velocity. For convenience it is desirable to relate V_w to ω_t using a single variable λ, which is given by (5.2).

$$\lambda = R \frac{\omega_t}{V_w} \tag{5.2}$$

The relationship between C_P, λ and β (blade pitch angle) can be found experimentally for a given turbine at a particular wind velocity. From this the turbine speed ω_t can be referred to the generator side of the gear box ω_r and the P_t-ω_r characteristic can be derived for various wind velocities as shown by Figure 5.22.

The optimal power output P_{opt} curve for a given shaft speed is derived which follows the relationship (5.3).

$$P_{opt} = K_{opt} \omega_r^3 \tag{5.3}$$

Using the value P_{opt} and substituting it into (2.7) and rearranging, (5.4) can be derived.

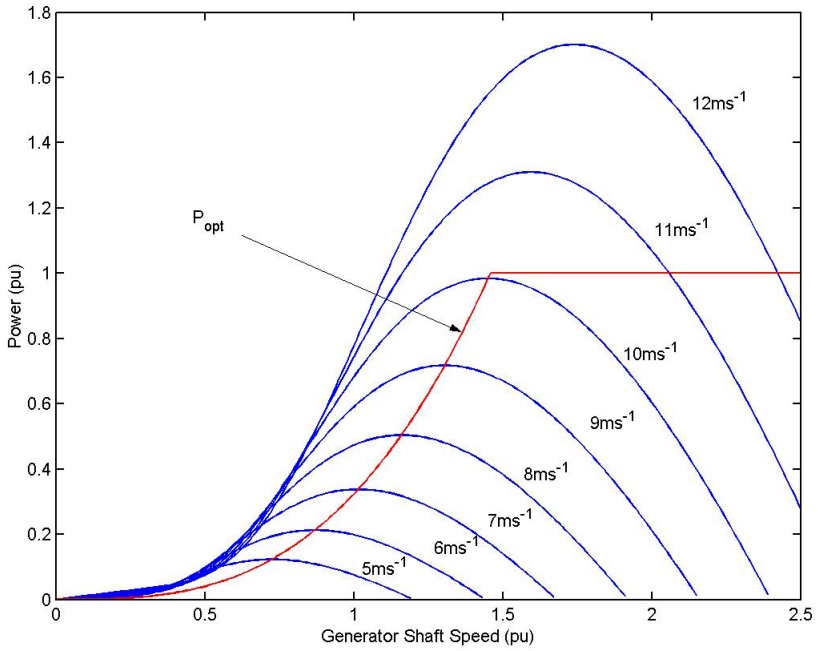


Figure 5.22: Wind Turbine Characteristics

$$i_{qr}^{e*} = \frac{P_{opt}}{-\frac{3}{2}\omega_e \frac{L_m^2}{L_s} i_{ms}^e} \tag{5.4}$$

From this the reference value for i_{qr}^{e*} is used in the DFIG rotor side control system, as outlined in Section 2.2.1. Between the *cut-in* wind velocity and rated wind velocity the DFIG follows the curve P_{opt}. When the wind velocity is greater than the rated wind velocity, the generator shaft speed is limited either via pitch control or using stall point control.

The drawback of implementing such a method of maximum power transfer is that it relies on the experimental wind turbine characteristic results to obtain the optimal shaft speed. If the characteristic of the turbine should change throughout its service life the results are no longer up to date and the optimal power capture

will not be achieved. Also, this system relies on accurate wind speed measurements from an anemometer, which will introduce further complications in its response as the anemometer is also a dynamic system.

5.3.1 Improved Hill Climb Algorithm

An alternative method for tracking the maximum power transfer shaft speed is called the *Hill Climb* algorithm [21] - [22]. The principle of this method is as the generator shaft speed changes the change in power is measured and $\frac{dP_{in}}{d\omega_r}$ is calculated. It can be seen that from Figure 5.22 that:

$$\begin{cases} \frac{dP_{in}}{d\omega_r} < 0 & \text{if } \omega_r < \omega_{opt} \\ \frac{dP_{in}}{d\omega_r} = 0 & \text{if } \omega_r = \omega_{opt} \\ \frac{dP_{in}}{d\omega_r} > 0 & \text{if } \omega_r > \omega_{opt} \end{cases} \tag{5.5}$$

which can be used to find ω_{opt}.

In this study the input power from the wind turbine is found by measuring the torque and shaft speed as:

$$P_{in} = T_t \omega_r \tag{5.6}$$

Where T_t and ω_r are the turbine torque and angular velocity respectively. The calculated input power is passed to the Maximum Power Point Tracking Controller (MPPTC) through a critically damped second order low-pass filter. This is done to remove the power variations arising as a result of sporadic wind speed changes. The characteristic frequency of the filter adopted is 0.004Hz, which was found by Ziegler-Nichols method. This was found to be the optimal setting for the filter to remove

the unwanted wind gust power variations which would cause errors in the control logic system. The generator shaft speed is also fed into the MPPTC via a low-pass filter to calculate $\frac{dP_{in}}{d\omega_r}$. This is because the wind variations also changes the shaft speed, however, it is damped by the generator inertia. The characteristic frequency of the shaft speed filter is 0.05Hz and the damping ratio is 0.8, which means that it is underdamped and reduces the response time of the system.

Inside the MPPT controller the minimum time between changes in shaft speed is limited in order to allow for the system to respond and effective measurements can be made. In this system the minimum time for shaft speed change is limited to 30s. This provides the system with just enough time to settle before the control logic makes a decision to change the shaft speed: this delay prevents erroneous logic decisions. Concurrently, $\frac{dP_{in}}{dt}$ is calculated to determine if the system has settled sufficiently prior to changes in shaft speed. Once the change in input power has settled to a satisfactory value, the optimal value was found to be 0.00025pu. The measurements for $\frac{dP_{in}}{d\omega_r}$ are calculated by the controller and from this the logic determines whether to increase or decrease the shaft speed.

If $\frac{dP_{in}}{d\omega_r} < 0$ then the generator shaft speed is decreased by a predetermined step which is 0.02pu in the case of this system. If $\frac{dP_{in}}{d\omega_r} > 0$ then the generator shaft speed is increased by the same amount. The MPPT controller constantly changes the shaft speed as the system relies in a change to determine whether the optimal shaft speed is being achieved.

A wind turbine model has been developed for use in the simulation of the system and is based on (5.7). The parameters of the wind turbine model used in the sim-

ulation is given in Appendix G, Table G.3. A PSCAD® / EMTDC^{TM} wind model which produces pseudo-random, sporadic wind speed values is connected to the wind turbine model. The wind model produces varying wind gusts around a predetermined average wind speed.

$$\begin{aligned} P_t &= \frac{1}{2}C_P(\lambda,\beta)\rho\pi R^2 V_w^3 \\ C_P(\lambda,\beta) &= c_1(\frac{c_2}{\lambda_i} - c_3\beta - c_4)e^{\frac{-c_5}{\lambda_i}} + c_6\lambda \\ \frac{1}{\lambda_i} &= \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \end{aligned} \tag{5.7}$$

where $c_1 = 0.5176$, $c_2 = 116$, $c_3 = 0.4$, $c_4 = 5$, $c_5 = 21$ and $c_6 = 0.0068$ [23]

The wind turbine model produces a power output which is converted into a torque value using (5.6) rearranged with torque as the subject of the equation. The torque value is passed to the WRIM model, and the control system for the DFIG is altered to control the shaft speed of the generator with i_{qr}^{e*} and a PI controller, as shown in Figure 5.23. The proportional gain and integral time for the controller were set to 50 and 0.01, respectively, which was found by Ziegler-Nichols method.

For the simulations conducted in this study, a 1.7MVA DFIG excited with a matrix converter was adopted. The switching frequency of the matrix converter was reduced from 20kHz to 2kHz and solution time step of the simulation increased to 50μs. This was done to reduce processing time as the simulation length is in the order of 30 minutes. The simulation results yielded from a mean wind velocity of 8m/s are shown in Figure 5.24. From these results it can be seen that the MPPT controller settles around the mean shaft speed of approximately 1.2pu. This speed corresponds

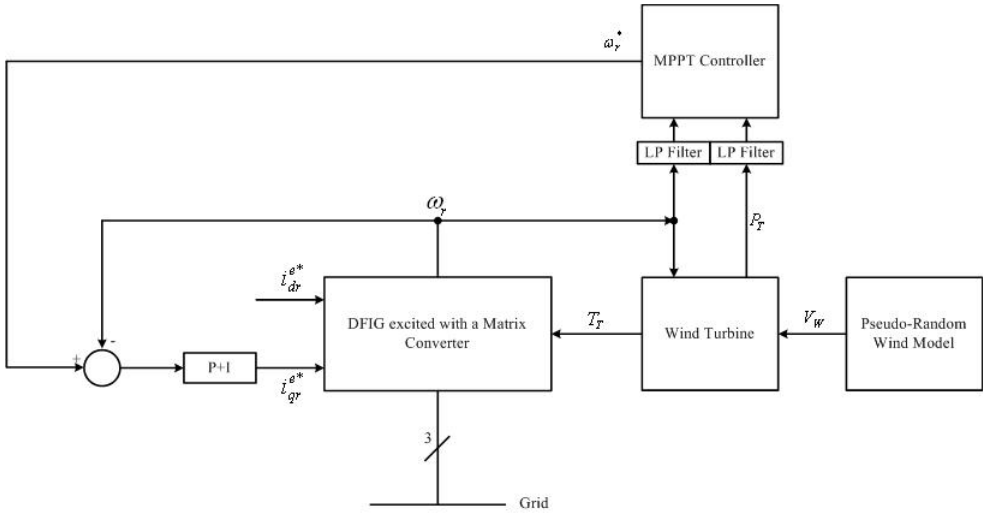


Figure 5.23: Maximum power point tracking control system

to the optimal power curve for the turbine used in the simulation at a wind speed of 8m/s as shown in Figure 5.22. Therefore, the MPPT controller is capable of tracking the optimal power point. There are variations in the turbine speed which are due to the wind fluctuations. In spite of this the low pass filtering and controlled response of the MPPTC allows the system to achieve optimal levels of energy extraction at the available wind speed.

Using the results obtained from the simulation, a calculation of total power transfer efficiency of the MC excited DFIG system with the developed MPPT controller is performed. The calculation is based on (5.8) where $E_{generator}$ and E_{wind} is the total energy from the generator and wind, respectively.

$$\eta(\%) = \frac{E_{generator}}{E_{wind}} \times 100\% \tag{5.8}$$

The energy available in the wind gust can be calculated by using (5.9) and the energy output of the wind generator is found using (5.10).

$$E_{wind} = \frac{1}{2}\pi\rho R^2 V_w^3 t \tag{5.9}$$

$$E_{generator} = \int P_{generator} dt \tag{5.10}$$

Based on these relations, the total wind energy available to the generator during a period of time, and the total generator energy output during that same period can be calculated. Based on the results gathered using an average wind speed of 8m/s, the total energy transfer of the generator was found to be 23.6%. This calculation was taken from when the system had reached a steady operating speed of approximately 1.2pu, which corresponded to a period of $1600 \leq t \leq 2400$ s. These results show that this system possesses superior power transfer characteristics over existing wind generator systems. It should be noted that the test was based on one average wind speed with sporadic wind gusts. For a more definitive measurement of power transfer efficiency the system should be tested over a wider range of wind speeds and for a longer period of time.

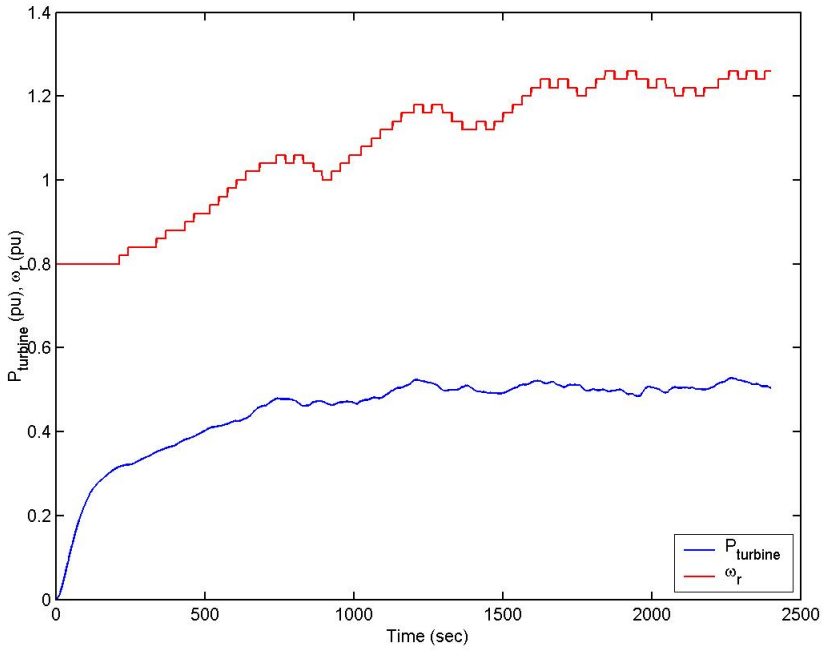


Figure 5.24: Variation of Wind Turbine Power and Generator Speed over time

5.4 DFIG Reactive Power Compensation for Voltage Regulation

As shown in Section 5.2, the matrix converter excited DFIG is capable of decoupled active and reactive power control. The active power component is used to control the speed of the generator through the MPPT controller so that maximum wind energy capture is possible, as detailed in Section 5.3.1. The reactive power component of the DFIG control system is capable of being controlled independent of active power and wind speed.

It is proposed in this section that the reactive power component of the DFIG is used to regulate the bus voltage that the system is connected to. Similar to Static VAr Compensators (SVC) or STATCOMS, the DFIG can inject or absorb reactive power from the grid to regulate system voltage. This can also be used to alleviate the effects of voltage fluctuations where the grid is weak.

From the well known flicker curve [24] it can be seen that the maximum sensitivity of flicker is approximately 1000 rectangular changes per minute; this corresponds to a frequency of approximately 8Hz. The perception of flicker is greatly reduced as the frequency increases and is negligible at 25Hz. Based on this, if reactive power compensation is injected to regulate the voltage and returns the voltage to nominal in a response time of less than 0.04s, then the perception of the flicker can be reduced.

The proposed reactive power compensation loop consists of an RMS voltage measurement and a second order low pass filter with a characteristic frequency of 0.4Hz to obtain a nominal reference voltage. This reference value does not move with high

frequency variations as found in flicker, however, it will respond to voltage changes arising from the tap changes in the transformers which are significantly slower. The nominal reference voltage is used to produce an error value which is passed to a PI controller with a proportional gain and integral time of 4.0 and 0.0001s, respectively. This control system is shown in Figure 5.25.

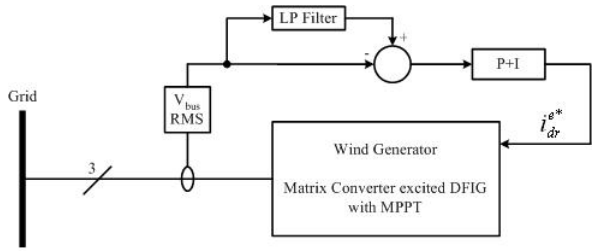


Figure 5.25: Matrix Converter excited DFIG with Reactive Power Compensation

To simulate the response of the reactive power compensation loop the system is connected to a simple power system with variable load. The grid consisted of a simple overhead transmission line with a length of approximately 20km connected to a bus bar with the VAr compensation DFIG of 1.7MVA. A constant base load of 1.2MW and 0.6MVAr and a step variable load of 0.8MW and 0.8MVAr was also connected to the bus bar as shown in Figure 5.26.

To allow the power system to settle first, the fluctuating load circuit breaker was not closed until 10.0s into the simulation. The fluctuation frequency was set to 0.5Hz to show the step response of the reactive power compensation loop inside the Matrix Converter excited DFIG wind generator. The reactive power compensation was stopped at 16.0s to demonstrate the difference in the bus bar voltage. Results obtained from the simulation are shown in Figure 5.27.

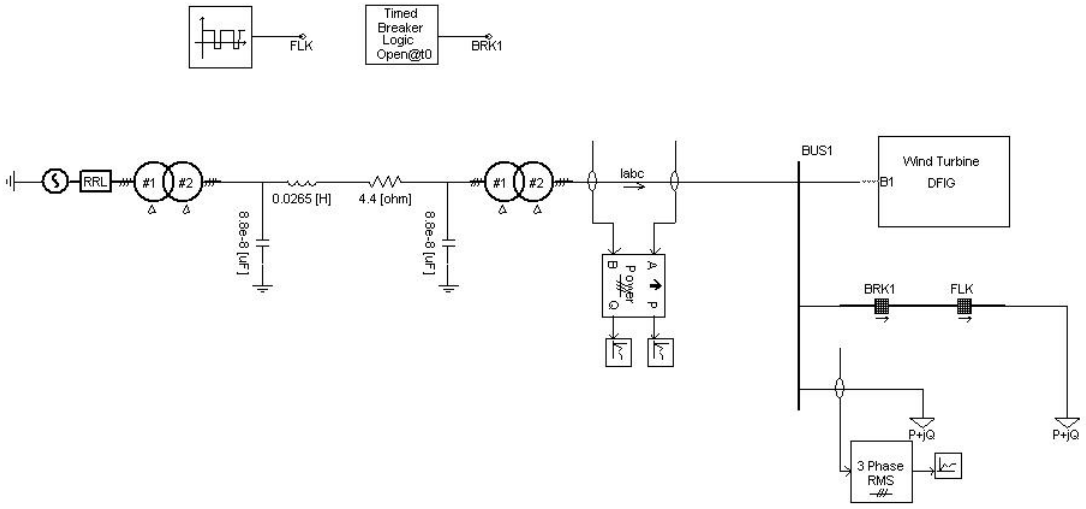


Figure 5.26: Power System Simulation

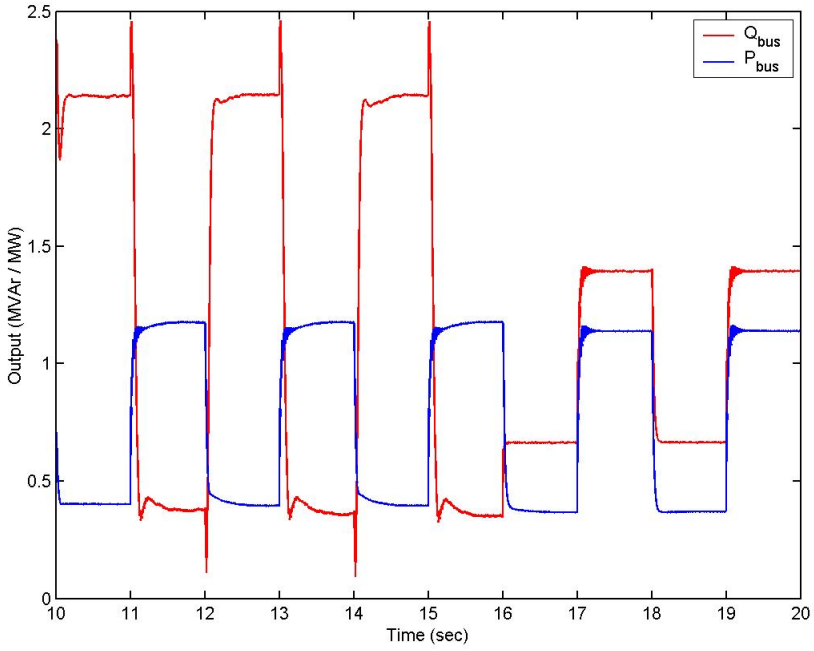


Figure 5.27: PCC Bus Bar Active and Reactive Power

In Figure 5.27 the resultant power flow in the transmission line model can be seen. The intermittent load causes the voltage fluctuations which are then compensated using the reactive power available in the DFIG, as seen in Figures 5.28 and 5.29.

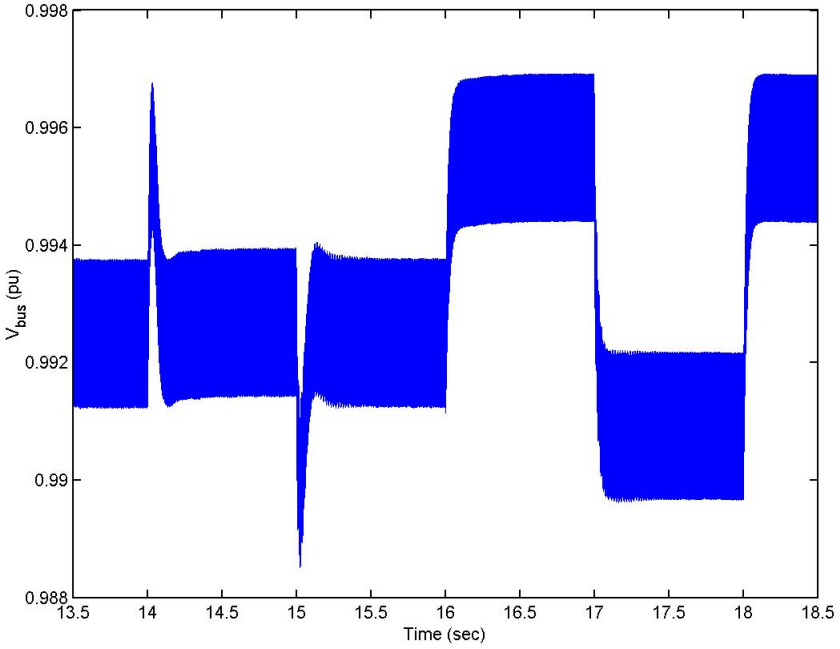


Figure 5.28: PCC Bus Bar Voltage

The voltage fluctuations without compensation can be seen in Figure 5.28 at 16s$<$ $t \leqslant$ 20s. The response time for the compensation to return the voltage to a nominal value is approximately 0.1s. This will not completely mitigate voltage fluctuations, however, it does significantly improve the voltage regulation. The results show that the effectiveness of a single wind generator is small. This is because the generator is supplying close to the maximum available reactive power compensation during each fluctuation. The voltage regulation can be significantly improved if the system is applied to multiple wind generators such as that found in a wind farm.

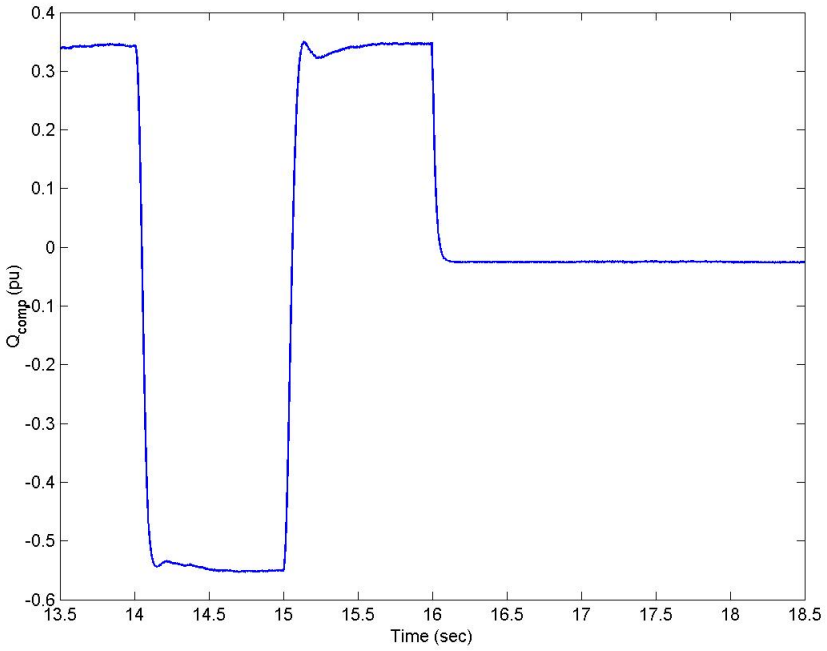


Figure 5.29: Reactive Power Compensation provided by DFIG

5.5 Summary

This chapter provided a description of WRIM modelling and its verification. The WRIM model and data from [19] was used to verify the model given in PSCAD® / EMTDC^{TM}. The simulations showed that the model performs as expected and provides a good approximation of a WRIM. Hence, the WRIM model was found to be suitable for use in DFIG modelling. The DFIG control loop was developed around the WRIM model using ideal voltage sources for rotor excitation to test the theory outlined in Section 2.2. The results gathered from the subsequent simulations showed that the WRIM and DFIG control loop functioned as expected. The step response of the system was illustrated with decoupled active and reactive power control.

The matrix converter system developed in Chapter 4 was incorporated into the DFIG system in place of the ideal voltage sources. Simulations showing the step responses and ramp response of generator speed were conducted on the 1.7MVA and 7.5kW DFIG system. From the results obtained it can be seen that the matrix converter is an effective means of exciting the rotor circuit of the WRIM. Decoupled active and reactive power control was achieved in the matrix converter excited DFIG model. The power flow through the system was shown for sub-synchronous and super-synchronous modes. In sub-synchronous mode the power was shown to flow into the rotor and as the speed of the generator increases into the super-synchronous region power was extracted from the rotor as well as the stator. The transition through the two modes of operation was seamless showing an output current THD value not exceeding 4.8%.

In Section 5.3.1 the control loop of the DFIG was extended to allow for speed control of the DFIG with a torque being applied on the rotor shaft of the WRIM model.

A wind turbine model was developed and applied with a pseudo-random PSCAD® / EMTDC^{TM} wind model. An improved Hill Climb algorithm was developed which used low pass filter measurements to calculate $\frac{dP_{in}}{d\omega_r}$ to determine the optimal generator speed for maximum wind energy capture. The improvements incorporated into the Maximum Power Point Tracking Controller (MPPTC) are that the system waits for a minimum time and also uses measurements of $\frac{dP_{in}}{dt}$ to determine if the system has settled before any speed variations occur. The advantage of the MPPTC is that experimental testing of the wind turbine is not necessary prior to commissioning. The system is adaptive throughout the service life of the generator and the optimum energy capture speed is achieved. The system was simulated with a mean wind velocity of 8m/s and the optimal generator speed of approximately 1.2pu was reached. This showed that the system is effective in optimal energy capture in wind generator systems. The total power transfer efficiency of the developed generator system was calculated to be 23.6% at an average wind speed of 8m/s.

Voltage regulation through reactive power compensation was simulated in Section 5.4. The matrix converter excited DFIG was connected to a simple power system via the PCC bus bar which also supplied a constant base load and a variable load. The control loop was developed using a PI controller to reduce the steady state error and also improve the system response. Simulation results gathered showed that the system was capable of regulating the system voltage. They also showed that the system can be used to mitigate the effects of voltage fluctuations. Further improvements may be found by incorporating the control system on multiple wind generators as found in a wind farm.

Chapter 6

Conclusions and Recommendations

6.1 Summary of Findings

The principle objectives of this research was to investigate the viability of a Matrix Converter (MC) excited Doubly-Fed Induction Generator (DFIG) to supply decoupled active and reactive power to a utility grid. To achieve the research objectives the work focused on two main areas of investigation: the control of a MC using non-ideal component models and decoupled control of a DFIG in wind generator applications.

In the area of MC control the research outlined the application of Space Vector Modulation (SVM) in a three phase to three phase MC. It also explored the different four quadrant switch (4QSW) cell configurations that can be used in an MC. The complexities involved in MC commutation control were investigated and a two-step semi-soft commutation method was presented. MC filter design was shown with a second order low pass filter modelled for use in simulations.

SVM theory was applied to an ideal MC for analysis and then to a non-ideal model with the semi-soft commutation controller. The results of the non-ideal model

showed that the matrix converter is capable of providing variable frequency and variable voltage control to a passive load. Furthermore, the results of the passive load simulations showed that the MC developed relatively small total harmonic distortion (THD) in the input and output current waveforms with values of 5.6% and 8.1%, respectively, with an output frequency of 10Hz. The MC model was also shown to be able to perform four quadrant power flow control in an asynchronous link configuration. Input current displacement angle control was illustrated with all the relevant waveforms.

Wind generator topologies and technologies were briefly explored in this research. However, the main focus of work in regards to the wind generator aspect of this research is on the decoupled active and reactive control of a DFIG. The investigation commenced with DFIG power flow theory and the different modes of operation (sub-synchronous and super-synchronous). Also, an investigation into WRIM theory was conducted to set the foundation for the analysis of DFIG control in the stator flux reference frame. It was shown that decoupled active and reactive power control can be achieved in a DFIG when rotor currents are regulated in the stator flux reference frame.

WRIM models were tested using free acceleration simulations and also a simulation with a pump load characteristic, with the rotor circuit short circuited. The results were compared to those given in [19], which verified that the model was a good approximation of a WRIM and was suitable for implementation into a DFIG system. The DFIG control theory was simulated on the WRIM model using ideal voltage sources. This verified that the DFIG may be controlled in the stator flux reference frame which set the basis for applying the developed non-ideal MC model to the rotor

circuit for simulation and analysis. The results of the simulation showed that the MC excited DFIG was able to perform decoupled active and reactive power control. A frequency spectrum analysis and current THD calculation revealed that the MC excited DFIG injected only a very small level of distortion into the grid.

From the results produced it can be concluded that the MC excited DFIG is suitable for use in a wind generator application. It is capable of supplying active and reactive power to the utility grid over a range of generator speeds. It has been shown to function well with both a large (1.7MVA) and a small (7.5kW) WRIM.

The research continued into Maximum Power Point Tracking control algorithms in Section 5.3. An improved Hill Climb algorithm was developed which uses measurements of changes in power and rotor shaft speed to determine the optimal shaft speed. By analyzing the calculated value of $\frac{dP_{in}}{d\omega_r}$ the controller determines if the speed should be increased or decreased accordingly. The advantage of the system is that it is adaptive to the turbine characteristic throughout the service life of the system. Simulation results showed that the system was capable of achieving the optimal rotor shaft speed for maximum energy capture. The developed MC excited DFIG with the MPPT controller was found to achieve a total power transfer efficiency of 23.6%.

Taking the system to the next step of development in Section 5.4, a reactive power compensation loop for voltage regulation was developed. As the reactive power output of the generator was independent of the active power and wind speed, the rotor excitation could be used to control the generator output to mitigate voltage fluctuations. The system was applied to a simple power system with a base and variable load. The simulation results indicated that the system could provide effective reactive

power compensation, and voltage fluctuations were reduced with a step response time of approximately 0.1s. The compensation was limited by the size of the generator, but this can be improved by applying the same control strategy to multiple generators as found in a wind farm.

6.2 Recommendations for Future Research

The next area of research for an MC excited DFIG could be an investigation into the application of the system in an *islanded* system. The research would require some minor alterations to the rotor control loop. The isolated DFIG system cannot rely on the grid to provide the power for excitation of the rotor circuit. Further research in the area of system stability when subjected to unbalanced conditions should be conducted as this was beyond the scope of this project. The effects of matrix converter excited DFIG to power system stability in fault conditions requires some further work.

Ultimately, future research should aim towards the construction of a real MC excited DFIG. The size of the system should be in the order of approximately 7.5kW to prove the system viability. The DFIG could be subjected to over and under voltage conditions and also unbalanced conditions. The resulting data would provide an excellent basis for the commercial implementation of the MC excited DFIG. For wind generators to gain a greater share of the generation market, further work in these areas is required.

Appendix A

Park and Clarke Transformations

A.1 Park Transformation [25]

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ \cos(\theta - 120^\circ) & \sin(\theta - 120^\circ) & 1 \\ \cos(\theta + 120^\circ) & \sin(\theta + 120^\circ) & 1 \end{bmatrix} \begin{bmatrix} v_q \\ v_d \\ v_0 \end{bmatrix} \tag{A.1}$$

Inverse relationship is given by

$$\begin{bmatrix} v_q \\ v_d \\ v_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos(\theta - 120^\circ) & \cos(\theta + 120^\circ) \\ \sin\theta & \sin(\theta - 120^\circ) & \sin(\theta + 120^\circ) \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \tag{A.2}$$

A.2 Clarke Transformation [25]

Transform from stationary reference frame d-q to synchronously rotating reference frame d^e-q^e

$$v_q^e = v_q \cos\theta - v_d \sin\theta \tag{A.3}$$
$$v_d^e = v_q \sin\theta + v_d \cos\theta$$

Transform from synchronously rotating reference frame d^e-q^e to stationary reference frame d-q

$$v_q = v_q^e \cos\theta + v_d^e \sin\theta \tag{A.4}$$
$$v_d = -v_q^e \sin\theta + v_d^e \cos\theta$$

A.3 SVM Input Voltage Park Transform

$$v_{ol} = \frac{2}{3}(v_{AB} + v_{BC}e^{j120^\circ} + v_{CA}e^{-j120^\circ}) \tag{A.5}$$

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} v_{AB} \\ v_{BC} \\ v_{CA} \end{bmatrix} \tag{A.6}$$

$$v_\alpha = \frac{2}{3}[v_{AB} - \frac{1}{2}(v_{BC} + v_{CA})] \tag{A.7}$$

$$v_\beta = \frac{1}{\sqrt{3}}[v_{BC} - v_{CA}] \tag{A.8}$$

where

$$v = v_\alpha + jv_\beta$$

A.4 SVM Output Current Park Transform

$$i_{il} = \frac{2}{3}(i_a + i_b e^{j120^\circ} + i_c e^{-j120^\circ}) \tag{A.9}$$

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{A.10}$$

$$i_\alpha = \frac{2}{3}[i_a - \frac{1}{2}(i_b + i_c)] \tag{A.11}$$

$$i_\beta = \frac{1}{\sqrt{3}}[i_b - i_c] \tag{A.12}$$

where

$$i_{il} = i_\alpha + j i_\beta$$

Appendix B

Matrix Converter Concepts

B.1 Venturini Pulse Width Modulation

Venturini modulation uses a mathematical relationship to calculate the period for which each switching device is to be gated over the switching period. There are two solutions derived from the Venturini method. One solution produces an *in-phase* component where the output follows the input. The other solution yields a reverse phase displacement at the input. A combination of the two solutions provides a means for input displacement factor control [12] as shown by the equation below.

$$[M(t)] = \alpha_1[M_1(t)] + \alpha_2[M_2(t)] \tag{B.1}$$

where

$$
\begin{aligned}
\alpha_1 + \alpha_2 &= 1 \\
\left[M_1(t)\right] &= \begin{bmatrix} 1+2q\cos(\omega_{m1}t) & 1+2q\cos(\omega_{m1}t-\frac{2\pi}{3}) & 1+2q\cos(\omega_{m1}t-\frac{4\pi}{3}) \\ 1+2q\cos(\omega_{m1}t-\frac{4\pi}{3}) & 1+2q\cos(\omega_{m1}t) & 1+2q\cos(\omega_{m1}t-\frac{2\pi}{3}) \\ 1+2q\cos(\omega_{m1}t-\frac{2\pi}{3}) & 1+2q\cos(\omega_{m1}t-\frac{4\pi}{3}) & 1+2q\cos(\omega_{m1}t) \end{bmatrix} \\
\left[M_2(t)\right] &= \begin{bmatrix} 1+2q\cos(\omega_{m2}t) & 1+2q\cos(\omega_{m2}t-\frac{2\pi}{3}) & 1+2q\cos(\omega_{m2}t-\frac{4\pi}{3}) \\ 1+2q\cos(\omega_{m2}t-\frac{2\pi}{3}) & 1+2q\cos(\omega_{m2}t-\frac{4\pi}{3}) & 1+2q\cos(\omega_{m2}t) \\ 1+2q\cos(\omega_{m2}t-\frac{4\pi}{3}) & 1+2q\cos(\omega_{m2}t) & 1+2q\cos(\omega_{m2}t-\frac{2\pi}{3}) \end{bmatrix} \\
\omega_{m1} &= (\omega_o - \omega_i) \\
\omega_{m1} &= -(\omega_o - \omega_i) \\
q &= \frac{V_{out}}{V_{in}}
\end{aligned}
$$

Each element in $[M(t)]$ from (B.1) corresponds to the duty cycle for each switching device within the MC switching matrix such that $m_{ij} = \frac{t_{ij}}{T_s}$, where T_s is the switching period of the PWM pattern. Applying the matrix $[M(t)]$ to the switching matrix the following result is found:

$$
\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \left[M(t)\right] \begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \tag{B.2}
$$

$$
\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \left[M(t)\right] \begin{bmatrix} M_{Aa}(t) & M_{Ba}(t) & M_{Ca}(t) \\ M_{Ab}(t) & M_{Bb}(t) & M_{Cb}(t) \\ M_{Ac}(t) & M_{Bc}(t) & M_{Cc}(t) \end{bmatrix} \begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \tag{B.3}
$$

where $\begin{bmatrix} v_A(t) & v_B(t) & v_C(t) \end{bmatrix}^T$ represents the grid input voltage waveforms and $\begin{bmatrix} v_a(t) & v_b(t) & v_c(t) \end{bmatrix}^T$ corresponds to the output voltage waveforms.

The input to output transfer ratio of this modulation technique can only reach a maximum of approximately 86%. If needed, the transfer ratio can be extended by the implementation of a transformer on the input of the MC. However, the relatively compact size of the converter circuit is lost by the use of a bulky three phase transformer.

B.2 SVM VSI Output Voltage SSVs

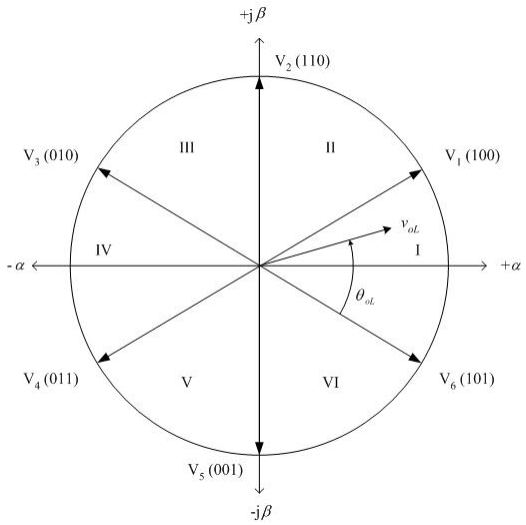


Figure B.1: VSI circuit output on an $\alpha + j\beta$ plane

B.3 SVM VSR Input Current SSVs

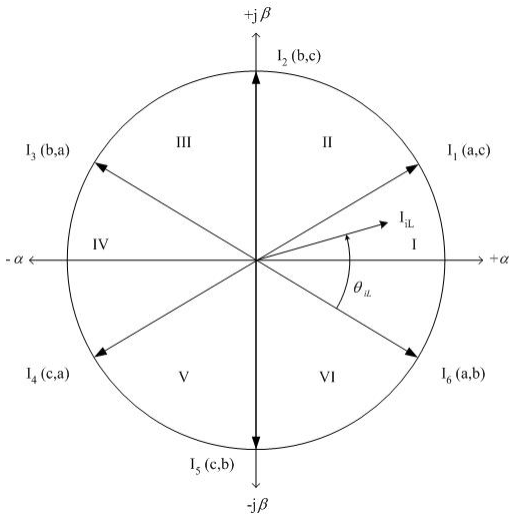


Figure B.2: VSR circuit input on an $\alpha + j\beta$ plane

SSV Pair	Switch Combination	Input Current			Output Voltage			i_{iL} Vector		v_{oL} Vector	
		I_a	I_b	I_c	V_{AB}	V_{BC}	V_{CA}	i_α	i_β	v_α	v_β
I_1V_1	S_1, S_6, S_9	i_A	0	$-i_A$	$-v_{ca}$	0	v_{ca}	i_A	$\frac{1}{\sqrt{3}}i_A$	$-v_{ca}$	$-\frac{1}{\sqrt{3}}v_{ca}$
I_1V_2	S_1, S_4, S_9	$-i_C$	0	i_C	0	$-v_{ca}$	v_{ca}	$-i_C$	$-\frac{1}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{ca}$
I_1V_3	S_3, S_4, S_9	i_B	0	$-i_B$	v_{ca}	$-v_{ca}$	0	i_B	$\frac{1}{\sqrt{3}}i_B$	v_{ca}	$-\frac{1}{\sqrt{3}}v_{ca}$
I_1V_4	S_3, S_4, S_7	$-i_A$	0	i_A	v_{ca}	0	$-v_{ca}$	$-i_A$	$-\frac{1}{\sqrt{3}}i_A$	v_{ca}	$\frac{1}{\sqrt{3}}v_{ca}$
I_1V_5	S_3, S_6, S_7	i_C	0	$-i_C$	0	v_{ca}	$-v_{ca}$	i_C	$\frac{1}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{ca}$
I_1V_6	S_1, S_6, S_7	$-i_B$	0	i_B	$-v_{ca}$	v_{ca}	0	$-i_B$	$-\frac{1}{\sqrt{3}}i_B$	$-v_{ca}$	$\frac{1}{\sqrt{3}}v_{ca}$
I_2V_1	S_2, S_6, S_9	0	i_A	$-i_A$	v_{bc}	0	$-v_{bc}$	0	$\frac{2}{\sqrt{3}}i_A$	v_{bc}	$\frac{1}{\sqrt{3}}v_{bc}$
I_2V_2	S_2, S_5, S_9	0	$-i_C$	i_C	0	v_{bc}	$-v_{bc}$	0	$-\frac{2}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{bc}$
I_2V_3	S_3, S_5, S_9	0	i_B	$-i_B$	$-v_{bc}$	v_{bc}	0	0	$\frac{2}{\sqrt{3}}i_B$	$-v_{bc}$	$\frac{1}{\sqrt{3}}v_{bc}$
I_2V_4	S_3, S_5, S_8	0	$-i_A$	i_A	$-v_{bc}$	0	v_{bc}	0	$-\frac{2}{\sqrt{3}}i_A$	$-v_{bc}$	$-\frac{1}{\sqrt{3}}v_{bc}$
I_2V_5	S_3, S_6, S_8	0	i_C	$-i_C$	0	$-v_{bc}$	v_{bc}	0	$\frac{2}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{bc}$
I_2V_6	S_2, S_6, S_8	0	$-i_B$	i_B	v_{bc}	$-v_{bc}$	0	0	$-\frac{2}{\sqrt{3}}i_B$	v_{bc}	$-\frac{1}{\sqrt{3}}v_{bc}$
I_3V_1	S_2, S_4, S_7	$-i_A$	i_A	0	$-v_{ab}$	0	v_{ab}	$-i_A$	$\frac{1}{\sqrt{3}}i_A$	$-v_{ab}$	$-\frac{1}{\sqrt{3}}v_{ab}$
I_3V_2	S_2, S_5, S_7	i_C	$-i_C$	0	0	$-v_{ab}$	v_{ab}	i_C	$-\frac{1}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{ab}$
I_3V_3	S_1, S_5, S_7	$-i_B$	i_B	0	v_{ab}	$-v_{ab}$	0	$-i_B$	$\frac{1}{\sqrt{3}}i_B$	v_{ab}	$-\frac{1}{\sqrt{3}}v_{ab}$
I_3V_4	S_1, S_5, S_8	i_A	$-i_A$	0	v_{ab}	0	$-v_{ab}$	i_A	$-\frac{1}{\sqrt{3}}i_A$	v_{ab}	$\frac{1}{\sqrt{3}}v_{ab}$
I_3V_5	S_1, S_4, S_8	$-i_C$	i_C	0	0	v_{ab}	$-v_{ab}$	$-i_C$	$\frac{1}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{ab}$
I_3V_6	S_2, S_4, S_8	i_B	$-i_B$	0	$-v_{ab}$	v_{ab}	0	i_B	$-\frac{1}{\sqrt{3}}i_B$	$-v_{ab}$	$\frac{1}{\sqrt{3}}v_{ab}$
I_4V_1	S_3, S_4, S_7	$-i_A$	0	i_A	v_{ca}	0	$-v_{ca}$	$-i_A$	$-\frac{1}{\sqrt{3}}i_A$	v_{ca}	$\frac{1}{\sqrt{3}}v_{ca}$
I_4V_2	S_3, S_6, S_7	i_C	0	$-i_C$	0	v_{ca}	$-v_{ca}$	i_C	$\frac{1}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{ca}$
I_4V_3	S_1, S_6, S_7	$-i_B$	0	i_B	$-v_{ca}$	v_{ca}	0	$-i_B$	$-\frac{1}{\sqrt{3}}i_B$	$-v_{ca}$	$\frac{1}{\sqrt{3}}v_{ca}$
I_4V_4	S_1, S_6, S_9	i_A	0	$-i_A$	$-v_{ca}$	0	v_{ca}	i_A	$\frac{1}{\sqrt{3}}i_A$	$-v_{ca}$	$-\frac{1}{\sqrt{3}}v_{ca}$
I_4V_5	S_1, S_4, S_9	$-i_C$	0	i_C	0	$-v_{ca}$	v_{ca}	$-i_C$	$-\frac{1}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{ca}$
I_4V_6	S_3, S_4, S_9	i_B	0	$-i_B$	v_{ca}	$-v_{ca}$	0	i_B	$\frac{1}{\sqrt{3}}i_B$	v_{ca}	$-\frac{1}{\sqrt{3}}v_{ca}$
I_5V_1	S_3, S_5, S_8	0	$-i_A$	i_A	$-v_{bc}$	0	v_{bc}	0	$-\frac{2}{\sqrt{3}}i_A$	$-v_{bc}$	$-\frac{1}{\sqrt{3}}v_{bc}$
I_5V_2	S_3, S_6, S_8	0	i_C	$-i_C$	0	$-v_{bc}$	v_{bc}	0	$\frac{2}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{bc}$
I_5V_3	S_2, S_6, S_8	0	$-i_B$	i_B	v_{bc}	$-v_{bc}$	0	0	$-\frac{2}{\sqrt{3}}i_B$	v_{bc}	$-\frac{1}{\sqrt{3}}v_{bc}$
I_5V_4	S_2, S_6, S_9	0	i_A	$-i_A$	v_{bc}	0	$-v_{bc}$	0	$\frac{2}{\sqrt{3}}i_A$	v_{bc}	$\frac{1}{\sqrt{3}}v_{bc}$
I_5V_5	S_2, S_5, S_9	0	$-i_C$	i_C	0	v_{bc}	$-v_{bc}$	0	$-\frac{2}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{bc}$
I_5V_6	S_3, S_5, S_9	0	i_B	$-i_B$	$-v_{bc}$	v_{bc}	0	0	$\frac{2}{\sqrt{3}}i_B$	$-v_{bc}$	$\frac{1}{\sqrt{3}}v_{bc}$
I_6V_1	S_1, S_5, S_8	i_A	$-i_A$	0	v_{ab}	0	$-v_{ab}$	i_A	$-\frac{1}{\sqrt{3}}i_A$	v_{ab}	$\frac{1}{\sqrt{3}}v_{ab}$
I_6V_2	S_1, S_4, S_8	$-i_C$	i_C	0	0	v_{ab}	$-v_{ab}$	$-i_C$	$\frac{1}{\sqrt{3}}i_C$	0	$\frac{2}{\sqrt{3}}v_{ab}$
I_6V_3	S_2, S_4, S_8	i_B	$-i_B$	0	$-v_{ab}$	v_{ab}	0	i_B	$-\frac{1}{\sqrt{3}}i_B$	$-v_{ab}$	$\frac{1}{\sqrt{3}}v_{ab}$
I_6V_4	S_2, S_4, S_7	$-i_A$	i_A	0	$-v_{ab}$	0	v_{ab}	$-i_A$	$\frac{1}{\sqrt{3}}i_A$	$-v_{ab}$	$-\frac{1}{\sqrt{3}}v_{ab}$
I_6V_5	S_2, S_5, S_7	i_C	$-i_C$	0	0	$-v_{ab}$	v_{ab}	i_C	$-\frac{1}{\sqrt{3}}i_C$	0	$-\frac{2}{\sqrt{3}}v_{ab}$
I_6V_6	S_1, S_5, S_7	$-i_B$	i_B	0	v_{ab}	$-v_{ab}$	0	$-i_B$	$\frac{1}{\sqrt{3}}i_B$	v_{ab}	$-\frac{1}{\sqrt{3}}v_{ab}$
I_0V_0	S_1, S_4, S_7	0	0	0	0	0	0	0	0	0	0
I_0V_0	S_2, S_5, S_8	0	0	0	0	0	0	0	0	0	0
I_0V_0	S_3, S_6, S_9	0	0	0	0	0	0	0	0	0	0

Table B.1: SSV Switching Combinations

B.4 Dead-Time Commutation

Dead-Time commutation is achieved by breaking the supply from one input phase for a short period and then reapplying it to the next phase completing the commutation. This violates rule one of the constraints of MC operation. While the disconnected time is only short, it is not ideal as snubber networks must be installed to prevent destruction of the switching devices. This in turn means that one of the MC advantages, which is its relatively small size as compared to other AC-AC converters, is compromised by the snubber circuits.

B.5 Overlap Commutation

Overlap commutation is achieved by connecting the output phase to two input phases for a short time before disconnecting the outgoing phase. This commutation method also violates one of the MC constraints as two of the input phases are shorted. This is unfavourable as large reactors must be inserted on the input phases to prevent line currents from rising above the safe operation point of the switching devices. Once again this is not favourable as the size advantage of the MC is compromised.

B.6 Four Step Semi-Soft Commutation

The principle of four step semi-soft commutation is that in steady state both IGBTs in the commutation cell are gated. When commutation is required the following four steps are performed:

1. The current direction is determined and the non-conducting device is switched off.

2. The device in the incoming cell that will carry the current is gated. This allows

current to flow from both phases. However, it does not create a shoot-through condition because the direction of current flow is controlled and can only flow in the same direction for both phases.

3. Outgoing cell is switched off.

4. The non-conducting device on the incoming cell is then switched on.

This method of commutation relies on the system being able to determine the direction of current flow. This is usually achieved using current transducers. This can be a problem at low current levels as the transducer can return erroneous measurements (especially in high current applications). This may result in the system making incorrect decisions which can also result in shoot-through conditions.

Appendix C

Two Step SSCC Model

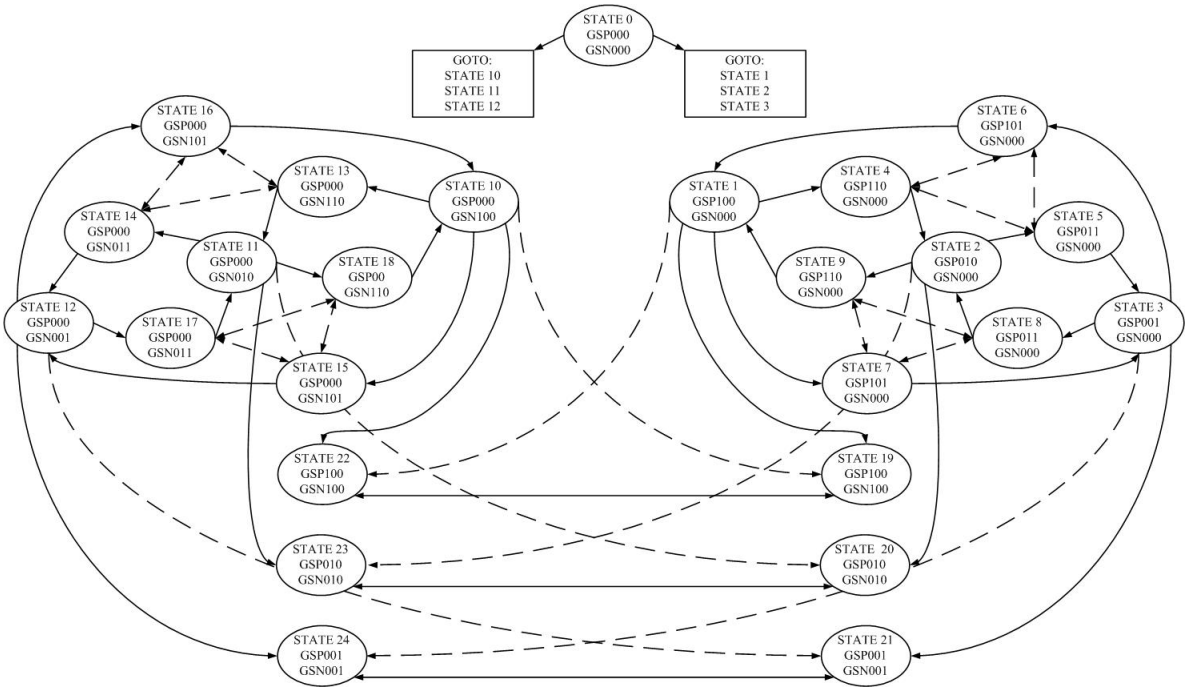


Figure C.1: SSCC Model Flow Diagram

```
SUBROUTINE COMCONT( MS, I, IdirIN, GSP, GSN, CHI, CHO )
    INCLUDE 'nd.h'              ! contains dimensioning information for EMTDC FORTRAN 77 Version
    INCLUDE 's0.h'              ! defines CCIN(*,*) and VDC(*,*)
    INCLUDE 's1.h'              ! defines TIME, DELT and TIMEZERO
    INCLUDE 'fnames.h'          ! defines IUNIT
    INCLUDE 'emtstor.h'         ! defines STORF(*) and NSTORF
    INCLUDE 'emtconst.h'
    INTEGER MS(3), IdirIN, GSP(3), GSN(3), CHI, CHO
    REAL I
    REAL TOCONST, Tcsconst, Tmkconst, Tbkconst, ITH
    INTEGER STATE, Idir, ERROR
    REAL T0, Tmk, Tbk, Tcs
    INTEGER LSTORI, LSTORF, NoSTORI, NoSTORF
    LOGICAL ComHold
    NoSTORI = 3
    NoSTORF = 10
    LSTORI = NSTORI
    NSTORI = NSTORI + NoSTORI
    LSTORF = NSTORF
    NSTORF = NSTORF + NoSTORF
    IF ( TIMEZERO ) THEN
        CALL RDCMNT
        READ (IUNIT,*) TOCONST, Tcsconst, Tmkconst, Tbkconst, ITH
        STORI(LSTORI) = 0   ! STATE
        STORI(LSTORI+1) = 0 ! Idir
        STORI(LSTORI+2) = 0 ! ERROR
        STORF(LSTORF) = 0   ! T0
        STORF(LSTORF+1) = 0 ! Tmk
        STORF(LSTORF+2) = 0 ! Tcs
        STORF(LSTORF+3) = 0 ! Tbk
        STORF(LSTORF+4) = TOCONST ! TOCONST
        STORF(LSTORF+5) = Tmkconst ! Tmkconst
        STORF(LSTORF+6) = Tcsconst ! Tcsconst
        STORF(LSTORF+7) = ITH     ! ITH
        STORF(LSTORF+8) = Tbkconst ! Tbkconst
        STORF(LSTORF+9) = 0       ! T4
    ENDIF
    STATE = STORI(LSTORI)
    Idir = STORI(LSTORI+1)
    T0 = STORF(LSTORF)
    Tmk = STORF(LSTORF+1)
    Tcs = STORF(LSTORF+2)
    Tbk = STORF(LSTORF+3)
    TOCONST = STORF(LSTORF+4)
    Tmkconst = STORF(LSTORF+5)
    Tcsconst = STORF(LSTORF+6)
    ITH = STORF(LSTORF+7)
    Tbkconst = STORF(LSTORF+8)
    ERROR = STORI(LSTORI+2)
    IF ( CHI .GE. 1 ) THEN
        ComHold = 1
    ELSE
        ComHold = 0
    ENDIF
    CHO = 0
    GSP = 0
    GSN = 0
    SELECT CASE ( STATE )
    ! POSITIVE COMMUTATION
    CASE(1)
        IF ( ComHold ) THEN
            IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                STATE = 19
            ENDIF
        ELSE
            IF ( MS(1) .GE. 1 ) THEN
                IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                    T0 = TIME + TOCONST ! Set Dead Time (not used in this code)
                    Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                    Idir = -1
                    IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                        IF ( ABS(I) .LT. ITH ) THEN
                            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                            STATE = 22
                        ELSE ! Not a likely condition (theoretically impossible with ideal components)
                            STATE = 10
                        ENDIF
                    ENDIF
                ELSE
                    Idir = 1
                    IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 19
                    ENDIF
                ENDIF
            ELSE
                IF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
                    CHO = 1
                    Tmk = TIME + Tmkconst  ! Set for Commutation "make"
```

```
                STATE = 4
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 7
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 19
                ENDIF
            ENDIF
            Idir = 1
        ENDIF
    ENDIF
    GSP(1) = 1
CASE(2)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
            STATE = 20
        ENDIF
    ELSE
        IF ( MS(2) .GE. 1 ) THEN
            IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = -1
                IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                    IF ( ABS(I) .LT. ITH ) THEN
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 23
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        STATE = 11
                    ENDIF
                ENDIF
            ELSE
                Idir = 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 20
                ENDIF
            ENDIF
        ELSE
            IF ( ( MS(1) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 9
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 5
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 20
                ENDIF
            ENDIF
            Idir = 1
        ENDIF
    ENDIF
    GSP(2) = 1
CASE(3)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
            STATE = 21
        ENDIF
    ELSE
        IF ( MS(3) .GE. 1 ) THEN
            IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = -1
                IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                    IF ( ABS(I) .LT. ITH ) THEN
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 24
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        STATE = 12
                    ENDIF
                ENDIF
            ELSE
                Idir = 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 21
                ENDIF
            ENDIF
        ELSE
            IF ( ( MS(1) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 6
            ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
```

```
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 8
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 21
                ENDIF
            ENDIF
            Idir = 1
        ENDIF
    ENDIF
    GSP(3) = 1
CASE(4)
! 1-2
    CHO = 1
    GSP(1) = 1
    Idir = 1
    IF ( MS(2) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 2
        ENDIF
        Idir = 1
        GSP(2) = 1
    ELSEIF ( MS(3) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 7
    ELSE
        STATE = 1
    ENDIF
CASE(5)
! 2-3
    CHO = 1
    GSP(2) = 1
    Idir = 1
    IF ( MS(3) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 3
        ENDIF
        GSP(3) = 1
    ELSEIF ( MS(1) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 9
    ELSE
        STATE = 2
    ENDIF
CASE(6)
! 3-1
    CHO = 1
    GSP(3) = 1
    Idir = 1
    IF ( MS(1) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 1
        ENDIF
        GSP(1) = 1
    ELSEIF ( MS(2) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 8
    ELSE
        STATE = 3
    ENDIF
CASE(7)
! 1-3
    CHO = 1
    GSP(1) = 1
    Idir = 1
    IF ( MS(3) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 3
        ENDIF
        GSP(3) = 1
    ELSEIF ( MS(2) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 4
    ELSE
        STATE = 1
    ENDIF
CASE(8)
! 3-2
    CHO = 1
    GSP(3) = 1
    Idir = 1
    IF ( MS(2) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 2
        ENDIF
        GSP(2) = 1
    ELSEIF ( MS(1) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 6
    ELSE
```

```
        STATE = 3
    ENDIF
CASE(9)
! 2-1
    CHO = 1
    GSP(2) = 1
    Idir = 1
    IF ( MS(1) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 1
        ENDIF
        GSP(1) = 1
    ELSEIF ( MS(3) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 5
    ELSE
        STATE = 2
    ENDIF
! NEGATIVE COMMUTATION
CASE(10)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
            STATE = 22
        ENDIF
    ELSE
        IF ( MS(1) .GE. 1 ) THEN
            IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + TOCONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = 1
                IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                    IF ( ABS(I) .LT. ITH ) THEN
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 19
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        STATE = 1
                    ENDIF
                ENDIF
            ELSE
                Idir = -1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 22
                ENDIF
            ENDIF
        ELSE
            IF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 13
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 16
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 22
                ENDIF
            ENDIF
            Idir = -1
        ENDIF
    ENDIF
    GSN(1) = 1
CASE(11)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
            STATE = 23
        ENDIF
    ELSE
        IF ( MS(2) .GE. 1 ) THEN
            IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + TOCONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = 1
                IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                    IF ( ABS(I) .LT. ITH ) THEN
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 20
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        STATE = 2
                    ENDIF
                ENDIF
            ELSE
                Idir = -1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 23
                ENDIF
            ENDIF
        ELSE
            IF ( ( MS(1) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
```

```
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 18
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 14
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 23
                ENDIF
            ENDIF
            Idir = -1
        ENDIF
    ENDIF
    GSN(2) = 1
CASE(12)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
            Tmk = TIME + Tmkconst ! Set for multi direction current "make"
            STATE = 24
        ENDIF
    ELSE
        IF ( MS(3) .GE. 1 ) THEN
            IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + TOCONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = 1
                IF ( TIME .GT. Tbk ) THEN ! Check for Commutation "break"
                    IF ( ABS(I) .LT. ITH ) THEN
                        Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                        STATE = 21
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        STATE = 3
                    ENDIF
                ENDIF
            ELSE
                Idir = -1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 24
                ENDIF
            ENDIF
        ELSE
            IF ( ( MS(1) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for multi direction current "break"
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 15
            ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 17
            ELSE
                ERROR = ERROR + 1
                IF ( ( ABS(I) .LT. ITH ) .AND. ( TIME .GT. Tbk ) ) THEN ! Check for Commutation "break"
                    Tmk = TIME + Tmkconst ! Set for multi direction current "make"
                    STATE = 24
                ENDIF
            ENDIF
            Idir = -1
        ENDIF
    ENDIF
    GSN(3) = 1
CASE(13)
! 10-11
    CHO = 1
    GSN(1) = 1
    Idir = -1
    IF ( MS(2) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 11
        ENDIF
        GSN(2) = 1
    ELSEIF ( MS(3) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 16
    ELSE
        STATE = 10
    ENDIF
CASE(14)
! 11-12
    CHO = 1
    GSN(2) = 1
    Idir = -1
    IF ( MS(3) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 12
        ENDIF
        GSN(3) = 1
    ELSEIF ( MS(1) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 18
    ELSE
        STATE = 11
```

```
    ENDIF
CASE(15)
! 12-10
    CHO = 1
    GSN(3) = 1
    Idir = -1
    IF ( MS(1) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 10
        ENDIF
        GSN(1) = 1
    ELSEIF ( MS(2) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 17
    ELSE
        STATE = 12
    ENDIF
CASE(16)
! 10-12
    CHO = 1
    GSN(1) = 1
    Idir = -1
    IF ( MS(3) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 12
        ENDIF
        GSN(3) = 1
    ELSEIF ( MS(2) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 13
    ELSE
        STATE = 10
    ENDIF
CASE(17)
! 12-11
    CHO = 1
    GSN(3) = 1
    Idir = -1
    IF ( MS(2) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 11
        ENDIF
        GSN(2) = 1
    ELSEIF ( MS(1) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 15
    ELSE
        STATE = 12
    ENDIF
CASE(18)
! 11-10
    CHO = 1
    GSN(2) = 1
    Idir = -1
    IF ( MS(1) .GE. 1 ) THEN
        IF ( TIME .GT. Tmk ) THEN
            Tbk = TIME + Tbkconst ! Set for Commutation "break"
            STATE = 10
        ENDIF
        GSN(1) = 1
    ELSEIF ( MS(3) .GE. 1 ) THEN
        Tmk = TIME + Tmkconst
        STATE = 14
    ELSE
        STATE = 11
    ENDIF
! Positive multi-directional commutation (current direction change)
CASE(19)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
            Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
            STATE = 1
        ENDIF
    ELSE
        IF ( MS(2) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst  ! Set for Commutation "make"
            STATE = 4
        ELSEIF ( MS(3) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst ! Set for Commutation "make"
            STATE = 7
        ELSE
            IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = -1
                IF ( ABS(I) .LT. ITH ) THEN
                    STATE = 22
                ELSE ! Not a likely condition (theoretically impossible with ideal components)
                    IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
                        Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                        STATE = 10
                    ELSE
```

```
                        STATE = 22
                    ENDIF
                ENDIF
            ELSE
                IF ( ABS(I) .GT. ITH ) THEN
                    STATE = 1
                ENDIF
            ENDIF
            GSN(1) = 1
        ENDIF
    ENDIF
    GSP(1) = 1
CASE(20)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
            Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
            STATE = 2
        ENDIF
    ELSE
        IF ( MS(1) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst  ! Set for Commutation "make"
            STATE = 9
        ELSEIF ( MS(3) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst ! Set for Commutation "make"
            STATE = 5
        ELSE
            IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = -1
                IF ( ABS(I) .LT. ITH ) THEN
                    STATE = 23
                ELSE ! Not a likely condition (theoretically impossible with ideal components)
                    IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
                        Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                        STATE = 11
                    ELSE
                        STATE = 23
                    ENDIF
                ENDIF
            ELSE
                IF ( ABS(I) .GT. ITH ) THEN
                    STATE = 2
                ENDIF
            ENDIF
            GSN(2) = 1
        ENDIF
    ENDIF
    GSP(2) = 1
CASE(21)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
            Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
            STATE = 3
        ENDIF
    ELSE
        IF ( MS(1) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst  ! Set for Commutation "make"
            STATE = 6
        ELSEIF ( MS(2) .GE. 1 ) THEN
            CHO = 1
            Tmk = TIME + Tmkconst ! Set for Commutation "make"
            STATE = 8
        ELSE
            IF ( ( I .LT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                Idir = -1
                IF ( ABS(I) .LT. ITH ) THEN
                    STATE = 24
                ELSE ! Not a likely condition (theoretically impossible with ideal components)
                    IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
                        Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                        STATE = 12
                    ENDIF
                ENDIF
            ELSE
                IF ( ABS(I) .GT. ITH ) THEN
                    STATE = 3
                ENDIF
            ENDIF
            GSN(3) = 1
        ENDIF
    ENDIF
    GSP(3) = 1
! Negative multi-directional commutation (current direction change)
CASE(22)
    IF ( ComHold ) THEN
        IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
            Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
            STATE = 10
        ENDIF
    ELSE
```

```
            IF ( MS(2) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 13
            ELSEIF ( MS(3) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 16
            ELSE
                IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                    T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                    Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                    Idir = 1
                    IF ( ABS(I) .LT. ITH ) THEN
                        STATE = 19
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
                            Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                            STATE = 1
                        ELSE
                            STATE = 19
                        ENDIF
                    ENDIF
                ELSE
                    IF ( ABS(I) .GT. ITH ) THEN
                        STATE = 10
                    ENDIF
                ENDIF
                GSP(1) = 1
            ENDIF
        ENDIF
        GSN(1) = 1
    CASE(23)
        IF ( ComHold ) THEN
            IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
                Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
                STATE = 11
            ENDIF
        ELSE
            IF ( MS(1) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 18
            ELSEIF ( MS(3) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 14
            ELSE
                IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                    T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                    Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                    Idir = 1
                    IF ( ABS(I) .LT. ITH ) THEN
                        STATE = 20
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
                            Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                            STATE = 2
                        ELSE
                            STATE = 20
                        ENDIF
                    ENDIF
                ELSE
                    IF ( ABS(I) .GT. ITH ) THEN
                        STATE = 11
                    ENDIF
                ENDIF
                GSP(2) = 1
            ENDIF
        ENDIF
        GSN(2) = 1
    CASE(24)
        IF ( ComHold ) THEN
            IF ( ( ABS(I) .GT. ITH ) .AND. ( TIME .GT. Tmk ) ) THEN ! Check for multi-direction current "make"
                Tbk = TIME + Tbkconst ! Set for multi-direction current "break"
                STATE = 12
            ENDIF
        ELSE
            IF ( MS(1) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst  ! Set for Commutation "make"
                STATE = 15
            ELSEIF ( MS(2) .GE. 1 ) THEN
                CHO = 1
                Tmk = TIME + Tmkconst ! Set for Commutation "make"
                STATE = 17
            ELSE
                IF ( ( I .GT. 0 ) .AND. ( TIME .GT. Tcs ) ) THEN ! Check for current direction change settle time
                    T0 = TIME + T0CONST ! Set Dead Time (not used in this code)
                    Tcs = TIME + Tcsconst ! Set Current Direction Change Settle Time
                    Idir = 1
                    IF ( ABS(I) .LT. ITH ) THEN
                        STATE = 21
                    ELSE ! Not a likely condition (theoretically impossible with ideal components)
                        IF ( TIME .GT. Tmk ) THEN ! Check for multi-direction current "make"
```

```
                        Tbk = TIME + Tbkconst ! Set for multi-direction "break"
                        STATE = 3
                    ELSE
                        STATE = 21
                    ENDIF
                ENDIF
            ELSE
                IF ( ABS(I) .GT. ITH ) THEN
                    STATE = 12
                ENDIF
            ENDIF
            GSP(3) = 1
        ENDIF
    ENDIF
    GSN(3) = 1
! ERROR COMMUTATION
CASE DEFAULT
    IF ( TIME .LE. T0 ) THEN
        STATE = 0
    ELSE
        IF ( IdirIN .GE. 1 ) THEN
            IF ( MS(1) .GE. 1 ) THEN
                STATE = 1
            ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                STATE = 2
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                STATE = 3
            ELSE
                STATE = 0
                ERROR = ERROR + 1
            ENDIF
        ELSEIF ( IdirIN .LE. -1 ) THEN
            IF ( MS(1) .GE. 1 ) THEN
                STATE = 10
            ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                STATE = 11
            ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                STATE = 12
            ELSE
                STATE = 0
                ERROR = ERROR + 1
            ENDIF
        ELSE
            IF ( Idir .GE. 0 ) THEN
                IF ( MS(1) .GE. 1 ) THEN
                    STATE = 1
                ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                    STATE = 2
                ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                    STATE = 3
                ELSE
                    STATE = 0
                    ERROR = ERROR + 1
                ENDIF
            ELSE
                IF ( MS(1) .GE. 1 ) THEN
                    STATE = 10
                ELSEIF ( ( MS(2) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                    STATE = 11
                ELSEIF ( ( MS(3) .GE. 1 ) .AND. ( TIME .GT. Tbk ) ) THEN
                    STATE = 12
                ELSE
                    STATE = 0
                    ERROR = ERROR + 1
                ENDIF
            ENDIF
        ENDIF
    ENDIF
END SELECT
STORI(LSTORI) = STATE
STORI(LSTORI+1) = Idir
STORI(LSTORI+2) = ERROR
STORF(LSTORF) = T0
STORF(LSTORF+1) = Tmk
STORF(LSTORF+2) = Tcs
STORF(LSTORF+3) = Tbk
END
```

Appendix D

Space Vector Modulation Controller Model Code

```
SUBROUTINE DUTYTIME( Q, THETA_GRID, THETA_ROT, d1, d2, d3, d4, d0 )
INCLUDE "emtconst.h"
REAL Q, THETA_GRID, THETA_ROT, d1, d2, d3, d4, d0
!dau -> 00 -> d1
!dav -> 01 -> d2
!dbu -> 10 -> d3
!dbv -> 11 -> d4
d1 = Q * SIN( (PI_/3.0) - THETA_ROT )*SIN( (PI_/3.0) - THETA_GRID ) !dau
d2 = Q * SIN( (PI_/3.0) - THETA_ROT) * SIN(THETA_GRID) !dav
d3 = Q * SIN( THETA_ROT ) * SIN( (PI_/3.0) - THETA_GRID) !dbu
d4 = Q * SIN( THETA_ROT ) * SIN(THETA_GRID) !dbv
d0 = 1.0 - ( d1 + d2 + d3 + d4 )
END
SUBROUTINE ADJSSV( THETA_IN, THETA_OUT, LVECT, UVECT )
INCLUDE "emtconst.h"
REAL THETA_IN, THETA_OUT
INTEGER LVECT, UVECT
REAL L_THETA
REAL T_THETA
INTEGER SECTOR
IF ( THETA_IN .GE. 0 ) THEN
    L_THETA = AMOD( THETA_IN, ( 2.0 * PI_ ) )
    T_THETA = AMOD( ( THETA_IN + ( PI_ / 6.0 ) ), ( 2.0 * PI_ ) )
ELSEIF ( THETA_IN .GE. ( 2.0 * PI_ ) ) THEN
    L_THETA = AMOD( ( THETA_IN - ( 2.0 * PI_ ) ), ( 2.0 * PI_ ) )
    T_THETA = AMOD( ( THETA_IN - ( 2.0 * PI_ ) + ( PI_ / 6.0 ) ), ( 2.0 * PI_ ) )
ELSE
    L_THETA = AMOD( ( THETA_IN + ( 2.0 * PI_ ) ), ( 2.0 * PI_ ) )
    T_THETA = AMOD( ( THETA_IN + ( 2.0 * PI_ ) + ( PI_ / 6.0 ) ), ( 2.0 * PI_ ) )
ENDIF
IF ( ( T_THETA .GE. 0 ) .AND. ( T_THETA .LT. ( PI_ / 3.0 ) ) ) THEN
    SECTOR = 1
    LVECT = 6
    THETA_OUT = T_THETA
ELSEIF ( ( T_THETA .GE. ( PI_ / 3.0 ) ) .AND. ( T_THETA .LT. ( ( 2.0 * PI_ ) / 3.0 ) ) ) THEN
    SECTOR = 2
    LVECT = 1
    THETA_OUT = L_THETA - ( PI_ / 6.0 )
ELSEIF ( ( T_THETA .GE. ( ( 2.0 * PI_ ) / 3.0 ) ) .AND. ( T_THETA .LT. PI_ ) ) THEN
    SECTOR = 3
    LVECT = 2
    THETA_OUT = L_THETA - ( PI_ / 2.0 )
ELSEIF ( ( T_THETA .GE. PI_ ) .AND. ( T_THETA .LT. ( 4.0 * PI_ / 3.0 ) ) ) THEN
    SECTOR = 4
    LVECT = 3
    THETA_OUT = L_THETA - ( 5.0 * PI_ / 6.0 )
ELSEIF ( ( T_THETA .GE. ( 4.0 * PI_ / 3.0 ) ) .AND. ( T_THETA .LT. ( 5.0 * PI_ / 3.0 ) ) ) THEN
    SECTOR = 5
    LVECT = 4
    THETA_OUT = L_THETA - ( 7.0 * PI_ / 6.0 )
```

```
ELSEIF ( ( T_THETA .GE. ( 5.0 * PI_ / 3.0 ) ) .AND. ( T_THETA .LT. ( 2.0 * PI_ ) ) ) THEN
    SECTOR = 6
    LVECT = 5
    THETA_OUT = L_THETA - ( 3.0 * PI_ / 2.0 )
ELSE
    SECTOR = 6
    LVECT = 5
    IF ( T_THETA .LT. 0 ) THEN
        THETA_OUT = L_THETA + ( PI_ / 2.0 )
    ELSE
        THETA_OUT = L_THETA - ( 3.0 * PI_ / 2.0 )
    ENDIF
ENDIF
UVECT = SECTOR
END
SUBROUTINE SVMODULATOR( SMAT, Vrotor, Vgrid, Igtheta, Q)
INCLUDE 'nd.h'              ! contains dimensioning information for EMTDC FORTRAN 77 Version
  INCLUDE 's0.h'                ! defines CCIN(*,*) and VDC(*,*)
  INCLUDE 's1.h'                ! defines TIME, DELT and TIMEZERO
  INCLUDE 'fnames.h'            ! defines IUNIT
  INCLUDE 'emtstor.h'           ! defines STORF(*) and NSTORF
INCLUDE 'emtconst.h'
INTEGER SMAT(9)
REAL Vrotor(3), Vgrid(3), Igtheta, Q
REAL Fs, Vin
INTEGER Swc
INTEGER CLK, VECT(4), ZERO, SELECT
REAL Ts, THETA_G_OUT, THETA_R_OUT
REAL VrotorPOL(2), VgridPOL(2)
REAL VrotorDQZ(3), VgridDQZ(3)
REAL DT(5)
REAL Tr, Ta(5)
INTEGER LSTORF ! Locally defined NSTORF array index
INTEGER NoSTORF ! Number of locally defined STORF elements
INTEGER LSTORI ! Locally defined NSTORI array index
INTEGER NoSTORI ! Number of locally defined STORI elements
! System variables declarations
NoSTORF = 15
NoSTORI = 12
LSTORF = NSTORF
LSTORI = NSTORI
NSTORF = NSTORF + NoSTORF
NSTORI = NSTORI + NoSTORI
IF ( TIMEZERO ) THEN
    CALL RDCMNT
    READ (IUNIT,*) Fs, Swc, Tr
    STORF(LSTORF) = Fs
    STORF(LSTORF+14) = Tr
    STORI(LSTORI) = Swc
    STORF(LSTORF+1) = 0
    STORF(LSTORF+2) = 0
    STORF(LSTORF+3) = 0
    STORF(LSTORF+4) = 0
    STORF(LSTORF+5) = 0
    STORF(LSTORF+6) = 0
    STORF(LSTORF+7) = 0
    STORI(LSTORI+1) = 0
    STORI(LSTORI+2) = 1
    STORI(LSTORI+3) = 2
    STORI(LSTORI+4) = 1
    STORI(LSTORI+5) = 2
    STORI(LSTORI+6) = 1
    STORI(LSTORI+7) = 1
ENDIF
Fs = STORF(LSTORF)
Tr = STORF(LSTORF+14)
Swc = STORI(LSTORI)
Ts = 1.0 / Fs
CALL PKTRANS( Vrotor(1), Vrotor(2), Vrotor(3), VrotorDQZ(1), VrotorDQZ(2), VrotorDQZ(3), PI_BY2, 1 )
CALL RECT2POL( VrotorDQZ(1), VrotorDQZ(2), VrotorPOL(1), VrotorPOL(2), 1 )
CALL PKTRANS( Vgrid(1), Vgrid(2), Vgrid(3), VgridDQZ(1), VgridDQZ(2), VgridDQZ(3), PI_BY2, 1 )
CALL RECT2POL( VgridDQZ(1), VgridDQZ(2), VgridPOL(1), VgridPOL(2), 1 )
IF ( STORF(LSTORF+1) .GT. AMOD(TIME, Ts) ) THEN
    CLK = 1
ELSE
    CLK = 0
ENDIF
STORF(LSTORF+1) = AMOD(TIME, Ts)
IF ( CLK .EQ. 1 ) THEN
    CALL ADJSSV( ( VgridPOL(2) + Igtheta ), THETA_G_OUT, VECT(1), VECT(2) )
    CALL ADJSSV( VrotorPOL(2), THETA_R_OUT, VECT(3), VECT(4) )
    CALL DUTYTIME( Q, THETA_G_OUT, THETA_R_OUT, DT(1), DT(2), DT(3), DT(4), DT(5) )
    STORF(LSTORF+2) = TIME
    STORI(LSTORI+2) = VECT(1)
    STORI(LSTORI+3) = VECT(2)
```

```
    STORI(LSTORI+4) = VECT(3)
    STORI(LSTORI+5) = VECT(4)
    IF ( STORI(LSTORI+1) .EQ. 0 ) THEN
        Ta(1) = ( ( DT(5) * Ts ) / 2.0 )
        Ta(2) = ( DT(1) * Ts )
        Ta(3) = ( DT(3) * Ts )
        Ta(4) = ( DT(4) * Ts )
        Ta(5) = ( DT(2) * Ts )
    ELSE
        Ta(1) = ( ( DT(5) * Ts ) / 2.0 )
        Ta(2) = ( DT(2) * Ts )
        Ta(3) = ( DT(4) * Ts )
        Ta(4) = ( DT(3) * Ts )
        Ta(5) = ( DT(1) * Ts )
    ENDIF
    IF ( Tr .GT. 0 ) THEN
        IF ( Ta(1) .LT. Tr ) THEN
            Ta(1) = 0
        ENDIF
        IF ( Ta(2) .LT. Tr ) THEN
            Ta(2) = 0
        ENDIF
        IF ( Ta(3) .LT. Tr ) THEN
            Ta(3) = 0
        ENDIF
        IF ( Ta(4) .LT. Tr ) THEN
            Ta(4) = 0
        ENDIF
        IF ( Ta(5) .LT. Tr ) THEN
            Ta(5) = 0
        ENDIF
    ENDIF
    STORF(LSTORF+3) = Ta(1) + STORF(LSTORF+2)
    STORF(LSTORF+4) = Ta(2) + STORF(LSTORF+3)
    STORF(LSTORF+5) = Ta(3) + STORF(LSTORF+4)
    STORF(LSTORF+6) = Ta(4) + STORF(LSTORF+5)
    STORF(LSTORF+7) = Ta(5) + STORF(LSTORF+6)
    STORF(LSTORF+8) = Ta(1)
    STORF(LSTORF+9) = Ta(2)
    STORF(LSTORF+10) = Ta(3)
    STORF(LSTORF+11) = Ta(4)
    STORF(LSTORF+12) = Ta(5)
ENDIF
IF ( STORI(LSTORI+1) .EQ. 0 ) THEN
    IF ( ( TIME .GE. STORF(LSTORF+2) ) .AND. ( TIME .LT. STORF(LSTORF+3) ) ) THEN
        CALL ASSERTZERO( SMAT, STORI(LSTORI+11), ZERO ) ! 00
        SELECT = 0
    ELSEIF ( ( TIME .GE. STORF(LSTORF+3) ) .AND. ( TIME .LT. STORF(LSTORF+4) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+2), STORI(LSTORI+4) ) ! 00
        STORI(LSTORI+6) = STORI(LSTORI+4)
        STORI(LSTORI+7) = STORI(LSTORI+2)
        SELECT = 1
    ELSEIF ( ( TIME .GE. STORF(LSTORF+4) ) .AND. ( TIME .LT. STORF(LSTORF+5) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+2), STORI(LSTORI+5) ) ! 10
        STORI(LSTORI+6) = STORI(LSTORI+5)
        STORI(LSTORI+7) = STORI(LSTORI+2)
        SELECT = 2
    ELSEIF ( ( TIME .GE. STORF(LSTORF+5) ) .AND. ( TIME .LT. STORF(LSTORF+6) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+3), STORI(LSTORI+5) ) ! 11
        STORI(LSTORI+6) = STORI(LSTORI+5)
        STORI(LSTORI+7) = STORI(LSTORI+3)
        SELECT = 3
    ELSEIF ( ( TIME .GE. STORF(LSTORF+6) ) .AND. ( TIME .LT. STORF(LSTORF+7) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+3), STORI(LSTORI+4) ) ! 01
        STORI(LSTORI+6) = STORI(LSTORI+4)
        STORI(LSTORI+7) = STORI(LSTORI+3)
        SELECT = 4
    ELSEIF ( TIME .GE. STORF(LSTORF+7) ) THEN
        CALL ASSERTZERO( SMAT, STORI(LSTORI+11), ZERO )
        SELECT = 0
    ELSE
        SMAT = 0
    ENDIF
ELSE
    IF ( ( TIME .GE. STORF(LSTORF+2) ) .AND. ( TIME .LT. STORF(LSTORF+3) ) ) THEN
        CALL ASSERTZERO( SMAT, STORI(LSTORI+11), ZERO )
        SELECT = 0
    ELSEIF ( ( TIME .GE. STORF(LSTORF+3) ) .AND. ( TIME .LT. STORF(LSTORF+4) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+3), STORI(LSTORI+4) ) ! 01
        STORI(LSTORI+6) = STORI(LSTORI+4)
        STORI(LSTORI+7) = STORI(LSTORI+3)
        SELECT = -1
    ELSEIF ( ( TIME .GE. STORF(LSTORF+4) ) .AND. ( TIME .LT. STORF(LSTORF+5) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+3), STORI(LSTORI+5) ) ! 11
        STORI(LSTORI+6) = STORI(LSTORI+5)
        STORI(LSTORI+7) = STORI(LSTORI+3)
        SELECT = -2
    ELSEIF ( ( TIME .GE. STORF(LSTORF+5) ) .AND. ( TIME .LT. STORF(LSTORF+6) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+2), STORI(LSTORI+5) ) ! 10
```

```
        STORI(LSTORI+6) = STORI(LSTORI+5)
        STORI(LSTORI+7) = STORI(LSTORI+2)
        SELECT = -3
    ELSEIF ( ( TIME .GE. STORF(LSTORF+6) ) .AND. ( TIME .LT. STORF(LSTORF+7) ) ) THEN
        CALL ASSERTVECT( SMAT, STORI(LSTORI+2), STORI(LSTORI+4) ) ! 00
        STORI(LSTORI+6) = STORI(LSTORI+4)
        STORI(LSTORI+7) = STORI(LSTORI+2)
        SELECT = -4
    ELSEIF ( TIME .GE. STORF(LSTORF+7) ) THEN
        CALL ASSERTZERO( SMAT, STORI(LSTORI+11), ZERO )
        SELECT = 0
    ELSE
        SMAT = 0
    ENDIF
ENDIF
CALL MAJORPHASE( SMAT, STORI(LSTORI+11) )
IF ( CLK .EQ. 1 ) THEN
    IF ( STORI(LSTORI+1) .EQ. 0 ) THEN
        IF ( Swc .EQ. 1 ) THEN
            STORI(LSTORI+1) = 1
        ELSE
            STORI(LSTORI+1) = 0
        ENDIF
    ELSE
        STORI(LSTORI+1) = 0
    ENDIF
ENDIF
! Internal Variable Output
    STORF(LSTORF+13) = Q
    STORI(LSTORI+8) = CLK
    STORI(LSTORI+9) = ZERO
    STORI(LSTORI+10) = SELECT
END
SUBROUTINE PKTRANS( A, B, C, D, Q, Z, THETA, DIR )
INCLUDE "emtconst.h"
REAL A, B, C, D, Q, Z, THETA
INTEGER DIR
IF ( DIR .EQ. 1 ) THEN
    ! Convert ABC three phase to DQ0 two phase
    D = ( 2.0 / 3.0 ) * ( ( A * SIN(THETA) ) +~
    ( B * SIN(THETA - ( TWO_PI / 3.0 ) ) ) + ( C * SIN(THETA + ( TWO_PI / 3.0 ) ) ) )
    Q = ( 2.0 / 3.0 ) * ( ( A * COS(THETA) ) +~
    ( B * COS(THETA - ( TWO_PI / 3.0 ) ) ) + ( C * COS(THETA + ( TWO_PI / 3.0 ) ) ) )
    Z = ( 1.0 / 3.0 ) * ( A + B + C )
ELSE
    ! Convert DQ0 two phase to ABC three phase
    A = ( ( Q * COS(THETA) ) + ( D * SIN(THETA) ) + Z )
    B = ( ( Q * COS(THETA - ( TWO_PI / 3.0 )) ) + ( D * SIN(THETA - ( TWO_PI / 3.0 )) ) + Z )
    C = ( ( Q * COS(THETA + ( TWO_PI / 3.0 )) ) + ( D * SIN(THETA + ( TWO_PI / 3.0 )) ) + Z )
ENDIF
END
SUBROUTINE RECT2POL( RL, IMG, MAG, THETA, DIR )
INCLUDE "emtconst.h"
REAL RL, IMG, MAG, THETA
INTEGER DIR
IF ( DIR .EQ. 1 ) THEN
    ! Convert Rectangular to Polar
    IF ( RL .NE. 0 ) THEN
        THETA = ATAN( IMG / RL )
        IF ( RL .LT. 0 ) THEN
            THETA = THETA + PI_
        ENDIF
        IF ( THETA .LT. 0 ) THEN
            THETA = THETA + TWO_PI
        ENDIF
        MAG = SQRT( ( IMG ** 2.0 ) + ( RL ** 2.0 ) )
    ELSE
        IF ( IMG .GT. 0 ) THEN
            THETA = PI_ / 2.0
        ELSEIF ( IMG .LT. 0 ) THEN
            THETA = 3.0 * PI_ / 2.0
        ELSE
            THETA = 0
        ENDIF
        MAG = IMG
    ENDIF
ELSE
    RL = MAG * COS(THETA)
    IMG = MAG * SIN(THETA)
ENDIF
END
SUBROUTINE MAJORPHASE( SWM, PH )
INTEGER SWM(9), PH
INTEGER PHC(3), I
PHC = 0
DO I = 1,9,1
    IF ( SWM(I) .GE. 1 ) THEN
        IF ( ( I .EQ. 1 ) .OR. ( I .EQ. 4 ) .OR. ( I .EQ. 7 ) ) THEN
```

```
            PHC(1) = PHC(1) + 1
        ELSEIF ( ( I .EQ. 2 ) .OR. ( I .EQ. 5 ) .OR. ( I .EQ. 8 ) ) THEN
            PHC(2) = PHC(2) + 1
        ELSEIF ( ( I .EQ. 3 ) .OR. ( I .EQ. 6 ) .OR. ( I .EQ. 9 ) ) THEN
            PHC(3) = PHC(3) + 1
        ENDIF
    ENDIF
ENDDO
IF ( PHC(1) .GE. 2 ) THEN
    PH = 1
ELSEIF ( PHC(2) .GE. 2 ) THEN
    PH = 2
ELSEIF ( PHC(3) .GE. 2 ) THEN
    PH = 3
ELSE
    PH = 0
ENDIF
END
SUBROUTINE ASSERTZERO( SWM, PH, SELECT )
INTEGER SWM(9), PH, SELECT
SWM = 0
IF ( PH .EQ. 1 ) THEN
    SWM(1) = 1
    SWM(4) = 1
    SWM(7) = 1
    SELECT = 1
ELSEIF ( PH .EQ. 2 ) THEN
    SWM(2) = 1
    SWM(5) = 1
    SWM(8) = 1
    SELECT = 2
ELSEIF ( PH .EQ. 3 ) THEN
    SWM(3) = 1
    SWM(6) = 1
    SWM(9) = 1
    SELECT = 3
ELSE
    SWM(1) = 1
    SWM(4) = 1
    SWM(7) = 1
    SELECT = 1
ENDIF
END
SUBROUTINE ASSERTVECT( SWM, VECT_GRID, VECT_ROTOR )
INTEGER SWM(9), VECT_ROTOR, VECT_GRID
SWM = 0
SELECT CASE ( VECT_GRID )
CASE(1)
    SELECT CASE ( VECT_ROTOR )
    CASE(1)
        ! acc
        SWM(1) = 1
        SWM(6) = 1
        SWM(9) = 1
    CASE(2)
        ! aac
        SWM(1) = 1
        SWM(4) = 1
        SWM(9) = 1
    CASE(3)
        ! cac
        SWM(3) = 1
        SWM(4) = 1
        SWM(9) = 1
    CASE(4)
        ! caa
        SWM(3) = 1
        SWM(4) = 1
        SWM(7) = 1
    CASE(5)
        ! cca
        SWM(3) = 1
        SWM(6) = 1
        SWM(7) = 1
    CASE(6)
        ! aca
        SWM(1) = 1
        SWM(6) = 1
        SWM(7) = 1
    CASE DEFAULT
        ! acc
        SWM(1) = 1
        SWM(6) = 1
        SWM(9) = 1
    END SELECT
CASE(2)
    SELECT CASE( VECT_ROTOR )
    CASE(1)
        ! bcc
        SWM(2) = 1
```

```
            SWM(6) = 1
            SWM(9) = 1
        CASE(2)
            ! bbc
            SWM(2) = 1
            SWM(5) = 1
            SWM(9) = 1
        CASE(3)
            ! cbc
            SWM(3) = 1
            SWM(5) = 1
            SWM(9) = 1
        CASE(4)
            ! cbb
            SWM(3) = 1
            SWM(5) = 1
            SWM(8) = 1
        CASE(5)
            ! ccb
            SWM(3) = 1
            SWM(6) = 1
            SWM(8) = 1
        CASE(6)
            ! bcb
            SWM(2) = 1
            SWM(6) = 1
            SWM(8) = 1
        CASE DEFAULT
            ! bcc
            SWM(2) = 1
            SWM(6) = 1
            SWM(9) = 1
        END SELECT
    CASE(3)
        SELECT CASE ( VECT_ROTOR )
        CASE(1)
            ! baa
            SWM(2) = 1
            SWM(4) = 1
            SWM(7) = 1
        CASE(2)
            ! bba
            SWM(2) = 1
            SWM(5) = 1
            SWM(7) = 1
        CASE(3)
            ! aba
            SWM(1) = 1
            SWM(5) = 1
            SWM(7) = 1
        CASE(4)
            ! abb
            SWM(1) = 1
            SWM(5) = 1
            SWM(8) = 1
        CASE(5)
            ! aab
            SWM(1) = 1
            SWM(4) = 1
            SWM(8) = 1
        CASE(6)
            ! bab
            SWM(2) = 1
            SWM(4) = 1
            SWM(8) = 1
        CASE DEFAULT
            ! baa
            SWM(2) = 1
            SWM(4) = 1
            SWM(7) = 1
        END SELECT
    CASE(4)
        SELECT CASE ( VECT_ROTOR )
        CASE(1)
            ! caa
            SWM(3) = 1
            SWM(4) = 1
            SWM(7) = 1
        CASE(2)
            ! cca
            SWM(3) = 1
            SWM(6) = 1
            SWM(7) = 1
        CASE(3)
            ! aca
            SWM(1) = 1
            SWM(6) = 1
            SWM(7) = 1
        CASE(4)
            ! acc
```

```
            SWM(1) = 1
            SWM(6) = 1
            SWM(9) = 1
        CASE(5)
            ! aac
            SWM(1) = 1
            SWM(4) = 1
            SWM(9) = 1
        CASE(6)
            ! cac
            SWM(3) = 1
            SWM(4) = 1
            SWM(9) = 1
        CASE DEFAULT
            ! caa
            SWM(3) = 1
            SWM(4) = 1
            SWM(7) = 1
        END SELECT
    CASE(5)
        SELECT CASE ( VECT_ROTOR )
        CASE(1)
            ! cbb
            SWM(3) = 1
            SWM(5) = 1
            SWM(8) = 1
        CASE(2)
            ! ccb
            SWM(3) = 1
            SWM(6) = 1
            SWM(8) = 1
        CASE(3)
            ! bcb
            SWM(2) = 1
            SWM(6) = 1
            SWM(8) = 1
        CASE(4)
            ! bcc
            SWM(2) = 1
            SWM(6) = 1
            SWM(9) = 1
        CASE(5)
            ! bbc
            SWM(2) = 1
            SWM(5) = 1
            SWM(9) = 1
        CASE(6)
            ! cbc
            SWM(3) = 1
            SWM(5) = 1
            SWM(9) = 1
        CASE DEFAULT
            ! cbb
            SWM(3) = 1
            SWM(5) = 1
            SWM(8) = 1
        END SELECT
    CASE(6)
        SELECT CASE ( VECT_ROTOR )
        CASE(1)
            ! abb
            SWM(1) = 1
            SWM(5) = 1
            SWM(8) = 1
        CASE(2)
            ! aab
            SWM(1) = 1
            SWM(4) = 1
            SWM(8) = 1
        CASE(3)
            ! bab
            SWM(2) = 1
            SWM(4) = 1
            SWM(8) = 1
        CASE(4)
            ! baa
            SWM(2) = 1
            SWM(4) = 1
            SWM(7) = 1
        CASE(5)
            ! bba
            SWM(2) = 1
            SWM(5) = 1
            SWM(7) = 1
        CASE(6)
            ! aba
            SWM(1) = 1
            SWM(5) = 1
            SWM(7) = 1
        CASE DEFAULT
```

```
        ! abb
        SWM(1) = 1
        SWM(5) = 1
        SWM(8) = 1
    END SELECT
CASE DEFAULT
    SELECT CASE ( VECT_ROTOR )
    CASE(1)
        ! acc
        SWM(1) = 1
        SWM(6) = 1
        SWM(9) = 1
    CASE(2)
        ! aac
        SWM(1) = 1
        SWM(4) = 1
        SWM(9) = 1
    CASE(3)
        ! cac
        SWM(3) = 1
        SWM(4) = 1
        SWM(9) = 1
    CASE(4)
        ! caa
        SWM(3) = 1
        SWM(4) = 1
        SWM(7) = 1
    CASE(5)
        ! cca
        SWM(3) = 1
        SWM(6) = 1
        SWM(7) = 1
    CASE(6)
        ! aca
        SWM(1) = 1
        SWM(6) = 1
        SWM(7) = 1
    CASE DEFAULT
        ! acc
        SWM(1) = 1
        SWM(6) = 1
        SWM(9) = 1
    END SELECT
END SELECT
END
SUBROUTINE MODINDEX( Vrotor, Vin, Igtheta, Q )
INCLUDE 'nd.h'              ! contains dimensioning information for EMTDC FORTRAN 77 Version
  INCLUDE 's0.h'                ! defines CCIN(*,*) and VDC(*,*)
  INCLUDE 's1.h'                ! defines TIME, DELT and TIMEZERO
  INCLUDE 'fnames.h'            ! defines IUNIT
  INCLUDE 'emtstor.h'           ! defines STORF(*) and NSTORF
INCLUDE 'emtconst.h'
REAL Vrotor(3), Vin, Igtheta, Q
REAL VrotorPOL(2), VrotorDQZ(3)
REAL VinT
VinT = Vin * ( SQRT_2 / SQRT_3 )
CALL PKTRANS( Vrotor(1), Vrotor(2), Vrotor(3), VrotorDQZ(1), VrotorDQZ(2), VrotorDQZ(3), PI_BY2, 1 )
CALL RECT2POL( VrotorDQZ(1), VrotorDQZ(2), VrotorPOL(1), VrotorPOL(2), 1 )
Q = ( ( VrotorPOL(1) / VinT ) / ( COS(Igtheta) * ( 3.0 / 2.0 ) ) )
IF ( Q .GT. 1.0 ) THEN
    Q = 1.0
ELSEIF ( Q .LT. 0 ) THEN
    Q = 0.0
ENDIF
END
```

Appendix E

Krause Wound Rotor Induction Machine Model

```
SUBROUTINE WRIMDFIG( NA, NB, NC, RA, RB, RC, iTm, iwr, SMODE, SS )
    INCLUDE 'nd.h'              ! contains dimensioning information for EMTDC FORTRAN 77 Version
      INCLUDE 's0.h'               ! defines CCIN(*,*) and VDC(*,*)
      INCLUDE 's1.h'               ! defines TIME, DELT and TIMEZERO
      INCLUDE 'fnames.h'           ! defines IUNIT
      INCLUDE 'emtstor.h'          ! defines STORF(*) and NSTORF
    INCLUDE 'emtconst.h'
      INTEGER NA, NB, NC, RA, RB, RC, SMODE, SS
    REAL iTm, iwr
    INTEGER LSTORF ! Locally defined NSTORF array index
    INTEGER NoSTORF ! Number of locally defined STORF elements
    ! System variables declarations
    REAL Sbase, Ebase, fbase, PP, N, Typewz
    REAL Rs, Rr, Xls, Xlr, Xm, B, TypeJH, JH
    REAL wbase, Ibase, Zbase, Tbase, wrbase, H
    REAL Ias, Ibs, Ics, Iar, Ibr, Icr
    REAL Vqs, Vds, Vzs, Vqr, Vdr, Vzr
    REAL Vqs1, Vds1, Vqr1, Vdr1
    REAL Iqs, Ids, Izs, Iqr, Idr, Izr
    REAL Lqs, Lds, Lzs, Lqr, Ldr, Lzr, Lmq, Lmd
    REAL pLqs, pLds, pLzs, pLqr, pLdr, pLzr, pwr
    REAL Ys, Yr
    REAL wr, THETA
    REAL wz, THETAZ
    REAL Tm, Te
    REAL Xaqd
    REAL K
    REAL Vrads, Vthetas
    NoSTORF = 46
    LSTORF = NSTORF
    NSTORF = NSTORF + NoSTORF
    IF ( TIMEZERO ) THEN
        CALL RDCMNT
        READ (IUNIT,*) Sbase, Ebase, fbase
        CALL RDCMNT
        READ (IUNIT,*) PP, N, Typewz
        CALL RDCMNT
        READ (IUNIT,*) Rs, Rr, Xls, Xlr, Xm, B, TypeJH, JH
        STORF(LSTORF) = Sbase
        STORF(LSTORF+1) = Ebase
        STORF(LSTORF+2) = fbase
        STORF(LSTORF+3) = PP
        STORF(LSTORF+4) = N
        STORF(LSTORF+5) = Typewz
        STORF(LSTORF+6) = Rs
        STORF(LSTORF+7) = Rr
        STORF(LSTORF+8) = Xls
        STORF(LSTORF+9) = Xlr
```

```
    STORF(LSTORF+10) = Xm
    STORF(LSTORF+11) = B
    STORF(LSTORF+12) = TypeJH
    STORF(LSTORF+13) = JH
    wbase = ( 2.0 * PI_ * fbase )
    Ibase = Sbase / ( SQRT_3 * Ebase )
    Zbase = Ebase / ( SQRT_3 * Ibase )
    Tbase = ( Sbase * PP ) / wbase
    wrbase = ( wbase / PP )
    IF ( TypeJH .EQ. 0 ) THEN
        H = (1.0 / 2.0) * ( ( 1.0 / PP ) ** 2.0 ) * ( ( JH * ( wbase ** 2.0 ) ) / ( Sbase * ( 10 ** 6.0 ) ) )
    ELSE
        H = JH
    ENDIF
    STORF(LSTORF+36) = wbase
    STORF(LSTORF+37) = Ibase
    STORF(LSTORF+38) = Zbase
    STORF(LSTORF+39) = Tbase
    STORF(LSTORF+40) = wrbase
    STORF(LSTORF+41) = H
! Differential Equation Initial Conditions
    STORF(LSTORF+14) = 0    ! Lqs
    STORF(LSTORF+15) = 0    ! Lds
    STORF(LSTORF+16) = 0    ! Lzs
    STORF(LSTORF+17) = 0    ! Lqr
    STORF(LSTORF+18) = 0    ! Ldr
    STORF(LSTORF+19) = 0    ! Lzr
    IF ( SMODE ) THEN
        STORF(LSTORF+20) = iwr  ! wr
    ELSE
        STORF(LSTORF+20) = 0
    ENDIF
    STORF(LSTORF+28) = 0    ! THETA
    STORF(LSTORF+29) = 0    ! THETAZ
    pwr = 0
    IF (NA.NE.0) ENABCCIN(NA,SS) = .TRUE.
    IF (NB.NE.0) ENABCCIN(NB,SS) = .TRUE.
    IF (NC.NE.0) ENABCCIN(NC,SS) = .TRUE.
    IF (RA.NE.0) ENABCCIN(RA,SS) = .TRUE.
    IF (RB.NE.0) ENABCCIN(RB,SS) = .TRUE.
    IF (RC.NE.0) ENABCCIN(RC,SS) = .TRUE.
  ENDIF
Sbase = STORF(LSTORF)
Ebase = STORF(LSTORF+1)
fbase = STORF(LSTORF+2)
PP = STORF(LSTORF+3)
N = STORF(LSTORF+4)
Typewz = STORF(LSTORF+5)
Rs = STORF(LSTORF+6)
Rr = STORF(LSTORF+7)
Xls = STORF(LSTORF+8)
Xlr = STORF(LSTORF+9)
Xm = STORF(LSTORF+10)
B = STORF(LSTORF+11)
TypeJH = STORF(LSTORF+12)
JH =  STORF(LSTORF+13)
wbase =  STORF(LSTORF+36)
Ibase =  STORF(LSTORF+37)
Zbase =  STORF(LSTORF+38)
Tbase =  STORF(LSTORF+39)
wrbase =  STORF(LSTORF+40)
H =  STORF(LSTORF+41)
! Last time step calculations
Lqs = STORF(LSTORF+14)
Lds = STORF(LSTORF+15)
Lzs = STORF(LSTORF+16)
Lqr = STORF(LSTORF+17)
Ldr = STORF(LSTORF+18)
Lzr = STORF(LSTORF+19)
THETA = STORF(LSTORF+28)
! Calculate Terminal Voltage
K = SQRT_3 / ( SQRT_2 * Ebase )
IF ( ( NA .NE. 0 ) .AND. ( NB .NE. 0 ) .AND. ( NC .NE. 0 ) ) THEN
    CALL PKTRANS2( (K*VDC(NA,SS)), (K*VDC(NB,SS)), (K*VDC(NC,SS)), Vqs1, Vds1, Vzs, 0, 1 )
ENDIF
IF ( ( RA .NE. 0 ) .AND. ( RB .NE. 0 ) .AND. ( RC .NE. 0 ) ) THEN
    CALL PKTRANS2( (N*K*VDC(RA,SS)), (N*K*VDC(RB,SS)), (N*K*VDC(RC,SS)), Vqr1, Vdr1, Vzr, 0, 1 )
ENDIF
CALL CART2POL( Vqs1, -1*Vds1, Vrads, Vthetas )
IF ( Typewz .EQ. 0 ) THEN
    wz = 0
    THETAZ = STORF(LSTORF+29)
ELSEIF ( Typewz .EQ. 1 ) THEN
```

```
    wz = 1.0
    STORF(LSTORF+29) = Vthetas
    IF ( STORF(LSTORF+29) .GE. TWO_PI ) THEN
        STORF(LSTORF+29) = STORF(LSTORF+29) - TWO_PI
    ELSEIF ( STORF(LSTORF+29) .LT. 0 ) THEN
        STORF(LSTORF+29) = STORF(LSTORF+29) + TWO_PI
    ENDIF
    THETAZ = STORF(LSTORF+29)
ELSE
    wz = STORF(LSTORF+20)
    STORF(LSTORF+29) = THETA
    IF ( STORF(LSTORF+29) .GE. TWO_PI ) THEN
        STORF(LSTORF+29) = STORF(LSTORF+29) - TWO_PI
    ELSEIF ( STORF(LSTORF+29) .LT. 0 ) THEN
        STORF(LSTORF+29) = STORF(LSTORF+29) + TWO_PI
    ENDIF
    THETAZ = STORF(LSTORF+29)
ENDIF
CALL CLRKTRANS2( Vqs1, Vds1, Vqs, Vds, THETAZ)
CALL CLRKTRANS2( Vqr1, Vdr1, Vqr, Vdr, (THETAZ - THETA))
IF ( SMODE ) THEN
    wr = iwr
    pwr = ( iwr - STORF(LSTORF+20) ) / DELT
ELSE
    wr = STORF(LSTORF+20)
    Tm = iTm
ENDIF
!***************************
!
!*  Calculate system start  *
!
!***************************
Xaqd = ( 1.0 / ( (1.0/Xm) + (1.0/Xls) + (1.0/Xlr) ) )
Lmq = Xaqd * ((Lqs/Xls) + (Lqr/Xlr))
Lmd = Xaqd * ((Lds/Xls) + (Ldr/Xlr))
pLqs = wbase * ( Vqs - (wz * Lds) + (Rs/Xls)*(Lmq - Lqs) )
pLds = wbase * ( Vds + (wz * Lqs) + (Rs/Xls)*(Lmd - Lds) )
pLzs = wbase * ( Vzs - (Rs/Xls)*Lzs)
pLqr = wbase * ( Vqr - ((wz - wr)*Ldr) + (Rr/Xlr)*(Lmq - Lqr) )
pLdr = wbase * ( Vdr + ((wz - wr)*Lqr) + (Rr/Xlr)*(Lmd - Ldr) )
pLzr = wbase * ( Vzr - (Rr/Xlr)*Lzr)
Iqs = (1/Xls) * (Lqs - Lmq)
Ids = (1/Xls) * (Lds - Lmd)
Izs = (1/Xls) * Lzs
Iqr = (1/Xlr) * (Lqr - Lmq)
Idr = (1/Xlr) * (Ldr - Lmd)
Izr = (1/Xlr) * Lzr
Te = Lds*Iqs - Lqs*Ids
! Trapezoidal Integration
Lqs = Lqs + (DELT/2.0) * ( STORF(LSTORF+21) + pLqs )
Lds = Lds + (DELT/2.0) * ( STORF(LSTORF+22) + pLds )
Lzs = Lzs + (DELT/2.0) * ( STORF(LSTORF+23) + pLzs )
Lqr = Lqr + (DELT/2.0) * ( STORF(LSTORF+24) + pLqr )
Ldr = Ldr + (DELT/2.0) * ( STORF(LSTORF+25) + pLdr )
Lzr = Lzr + (DELT/2.0) * ( STORF(LSTORF+26) + pLzr )
IF ( SMODE ) THEN
    ! Speed Control Mode
    Tm = Te - ( 2.0 * H * pwr ) - ( B * wr )
ELSE
    ! Torque Control Mode
    pwr = (1.0/(2.0 * H)) * (Te - Tm - ( B * wr ) )
    ! Trapezoidal Integration
    wr = wr + (DELT/2.0) * ( STORF(LSTORF+27) + pwr )
ENDIF
!*************************
!
!*  Calculate system end  *
!
!*************************
! Convert currents pack to three phase stationary reference frame
IF ( ( NA .NE. 0 ) .AND. ( NB .NE. 0 ) .AND. ( NC .NE. 0 ) ) THEN
    CALL PKTRANS2( Ias, Ibs, Ics, Iqs, Ids, Izs, THETAZ, 0 )
    Ias = Ias * Ibase
    Ibs = Ibs * Ibase
    Ics = Ics * Ibase
ELSE
    Ias = 0
    Ibs = 0
    Ics = 0
ENDIF
IF ( ( RA .NE. 0 ) .AND. ( RB .NE. 0 ) .AND. ( RC .NE. 0 ) ) THEN
    CALL PKTRANS2( Iar, Ibr, Icr, Iqr, Idr, Izr, (THETAZ - THETA), 0 )
    Iar = Iar * Ibase * N
    Ibr = Ibr * Ibase * N
    Icr = Icr * Ibase * N
```

```
    ELSE
        Iar = 0
        Ibr = 0
        Icr = 0
    ENDIF
    !Ys = ( 1 / Rs ) + ( DELT / ( 2.0 * ( Xls + Xm ) ) ) * Zbase
    !Yr = ( 1 / Rr ) + ( DELT / ( 2.0 * ( Xlr + Xm ) ) ) * Zbase * ( N ** 2.0 )
    Ys = 0
    Yr = 0
    ! Rotor angle calculation
    THETA = THETA + ( wr * DELT * wbase )
    IF ( THETA .GE. TWO_PI ) THEN
        THETA = THETA - TWO_PI
    ELSEIF ( THETA .LT. 0 ) THEN
        THETA = THETA + TWO_PI
    ENDIF
    ! Current injection on external connections
    IF (NA.NE.0) CCIN(NA,SS) = CCIN(NA,SS) + Ias
    IF (NB.NE.0) CCIN(NB,SS) = CCIN(NB,SS) + Ibs
    IF (NC.NE.0) CCIN(NC,SS) = CCIN(NC,SS) + Ics
    IF (RA.NE.0) CCIN(RA,SS) = CCIN(RA,SS) + Iar
    IF (RB.NE.0) CCIN(RB,SS) = CCIN(RB,SS) + Ibr
    IF (RC.NE.0) CCIN(RC,SS) = CCIN(RC,SS) + Icr
    GGIN(NA,SS) = GGIN(NA,SS) + Ys
    GGIN(NB,SS) = GGIN(NB,SS) + Ys
    GGIN(NC,SS) = GGIN(NC,SS) + Ys
    GGIN(RA,SS) = GGIN(RA,SS) + Yr
    GGIN(RB,SS) = GGIN(RB,SS) + Yr
    GGIN(RC,SS) = GGIN(RC,SS) + Yr
    ! Store Values for next time step
    STORF(LSTORF+14) = Lqs      ! Lqs
    STORF(LSTORF+15) = Lds      ! Lds
    STORF(LSTORF+16) = Lzs      ! Lzs
    STORF(LSTORF+17) = Lqr      ! Lqr
    STORF(LSTORF+18) = Ldr      ! Ldr
    STORF(LSTORF+19) = Lzr      ! Lzr
    STORF(LSTORF+20) = wr       ! wr
    STORF(LSTORF+21) = pLqs     ! pLqs
    STORF(LSTORF+22) = pLds     ! pLds
    STORF(LSTORF+23) = pLzs     ! pLzs
    STORF(LSTORF+24) = pLqr     ! pLqr
    STORF(LSTORF+25) = pLdr     ! pLdr
    STORF(LSTORF+26) = pLzr     ! pLzr
    STORF(LSTORF+27) = pwr      ! pwr
    STORF(LSTORF+28) = THETA    ! THETA
    ! Output internal variables
    STORF(LSTORF+30) = Te
    STORF(LSTORF+31) = Tm
    STORF(LSTORF+32) = Vqs
    STORF(LSTORF+33) = Vds
    STORF(LSTORF+34) = Vqr
    STORF(LSTORF+35) = Vdr
    STORF(LSTORF+42) = Iqs
    STORF(LSTORF+43) = Ids
    STORF(LSTORF+44) = Iqr
    STORF(LSTORF+45) = Idr
RETURN
END
SUBROUTINE PKTRANS2( A, B, C, Q, D, Z, THETA, DIR )
    INCLUDE 'emtconst.h'
    REAL A, B, C, Q, D, Z, THETA
    INTEGER DIR
    IF ( DIR .EQ. 1 ) THEN
        ! Convert ABC three phase to DQ0 two phase
        Q = ( 2.0 / 3.0 ) * ( ( A * COS(THETA) ) + ( B * COS(THETA - ( TWO_PI / 3.0 ) ) ) + ( C * COS(THETA + ( TWO_PI
        D = ( 2.0 / 3.0 ) * ( ( A * SIN(THETA) ) + ( B * SIN(THETA - ( TWO_PI / 3.0 ) ) ) + ( C * SIN(THETA + ( TWO_PI
        Z = ( 1.0 / 3.0 ) * ( A + B + C )
    ELSE
        ! Convert DQ0 two phase to ABC three phase
        A = ( ( Q * COS(THETA) ) + ( D * SIN(THETA) ) + Z )
        B = ( ( Q * COS(THETA - ( TWO_PI / 3.0 )) ) + ( D * SIN(THETA - ( TWO_PI / 3.0 )) ) + Z )
        C = ( ( Q * COS(THETA + ( TWO_PI / 3.0 )) ) + ( D * SIN(THETA + ( TWO_PI / 3.0 )) ) + Z )
    ENDIF
END
SUBROUTINE CLRKTRANS2( A, B, Q, D, THETA)
    INCLUDE "emtconst.h"
    REAL A, B, C, D, Q, THETA
    Q = ( A * COS(THETA) ) - ( B * SIN(THETA) )
    D = ( A * SIN(THETA) ) + ( B * COS(THETA) )
END
SUBROUTINE CART2POL( X, Y, RAD, THETA )
    INCLUDE 'emtconst.h'
    REAL X, Y, RAD, THETA
```

```
    REAL TEMP
    IF ( X .NE. 0 ) THEN
        TEMP = ATAN( Y / X )
        IF ( X .GT. 0 ) THEN
            IF ( Y .GT. 0 ) THEN
                THETA = TEMP
            ELSE
                THETA = TEMP + TWO_PI
            ENDIF
        ELSE
            THETA = TEMP + PI_
        ENDIF
    ELSE
        IF ( Y .GT. 0 ) THEN
            THETA = PI_ / 2.0
        ELSE
            THETA = 3 * PI_ / 2.0
        ENDIF
    ENDIF
    RAD = SQRT( X**2.0 + Y**2.0 )
END
```

Appendix F

Maximum Power Point Tracking Controller

```
SUBROUTINE MPPT( wr, Pm, wr_init, wr_out, mode )
INCLUDE 'nd.h'              ! contains dimensioning information for EMTDC FORTRAN 77 Version
  INCLUDE 's0.h'                ! defines CCIN(*,*) and VDC(*,*)
  INCLUDE 's1.h'                ! defines TIME, DELT and TIMEZERO
  INCLUDE 'fnames.h'            ! defines IUNIT
  INCLUDE 'emtstor.h'           ! defines STORF(*) and NSTORF
INCLUDE 'emtconst.h'
REAL wr, Pm, wr_init, wr_out
LOGICAL mode
REAL wr_step, dPmdt_ct, Tres
INTEGER LSTORF ! Locally defined NSTORF array index
INTEGER NoSTORF ! Number of locally defined STORF elements
INTEGER LSTORI ! Locally defined NSTORI array index
INTEGER NoSTORI ! Number of locally defined STORI elements
REAL dPm, dwr
INTEGER STATE
! System variables declarations
NoSTORF = 10
NoSTORI = 0
LSTORF = NSTORF
LSTORI = NSTORI
NSTORF = NSTORF + NoSTORF
NSTORI = NSTORI + NoSTORI
IF ( TIMEZERO ) THEN
    CALL RDCMNT
    READ (IUNIT,*) wr_step, dPmdt_ct, Tres
    STORF( LSTORF + 0 ) = wr_step ! Step up / down of wr_out
    STORF( LSTORF + 1 ) = dPmdt_ct ! Pm criterion for change analysis
    STORF( LSTORF + 2 ) = 0       ! Sum of wr at Tsample. Stop at Tave
    STORF( LSTORF + 3 ) = 0       ! Sum of Pm at Tsample. Stop at Tave
    STORF( LSTORF + 4 ) = 0.0     ! Last Pm measurement
    STORF( LSTORF + 5 ) = 1.0     ! Target wr_out
    STORF( LSTORF + 6 ) = Tres    ! Minimum response time
    STORF( LSTORF + 7 ) = 0.0     ! Next measurement time
    STORF( LSTORF + 8 ) = 0       ! dPm
    STORF( LSTORF + 9 ) = 0       ! dPm /dwr
ENDIF
wr_step = STORF( LSTORF + 0 )
dPmdt_ct = STORF( LSTORF + 1 )
Tres = STORF( LSTORF + 6 )
IF ( Pm .NE. 0.0 ) THEN
    STORF( LSTORF + 8 ) = ABS((Pm - STORF( LSTORF + 4 )) / DELT)
ELSE
    STORF( LSTORF + 8 ) = 0.0
ENDIF
IF ( mode ) THEN
    STORF( LSTORF + 5 ) = wr_init
    STORF( LSTORF + 2 ) = wr
```

```
    STORF( LSTORF + 3 ) = Pm
    STORF( LSTORF + 7 ) = TIME + Tres
ELSE
    IF ( ( STORF( LSTORF + 8 ) .LE. dPmdt_ct ) .AND. ( TIME .GE. STORF( LSTORF + 7 ) ) ) THEN
        dwr = wr - STORF( LSTORF + 2 )
        dPm = Pm - STORF( LSTORF + 3 )
        STORF( LSTORF + 9 ) = dPm / dwr
        IF ( STORF( LSTORF + 9 ) .LE. 0.0 ) THEN
            STORF( LSTORF + 5 ) = STORF( LSTORF + 5 ) - wr_step
        ELSE
            STORF( LSTORF + 5 ) = STORF( LSTORF + 5 ) + wr_step
        ENDIF
        STORF( LSTORF + 2 ) = wr
        STORF( LSTORF + 3 ) = Pm
        STORF( LSTORF + 7 ) = TIME + Tres
    ENDIF
ENDIF
STORF( LSTORF + 4 ) = Pm
wr_out = STORF( LSTORF + 5 )
END
```

Appendix G

Model Data

$\mathbf{S_{base}}$	$\mathbf{V_{base}}$	$\omega_{\mathbf{base}}$	$\omega_{\mathbf{rated}}$	**PL**	$\mathbf{T_{base}}$	$\mathbf{I_{base}}$
1.68 MVA 1.68MVA	2.3kV	376.99rad/s	1786RPM	4 Poles	8.905kNm	421.34A

J	**H**	$\mathbf{r_s}$	$\mathbf{r_r}$
$63.87kgm^2$	0.676s	9.2×10^{-3}pu	6.98×10^{-3}pu

$\mathbf{X_M}$	$\mathbf{X_{ls}}$	$\mathbf{X_{lr}}$	$\mathbf{X_s}$	$\mathbf{X_r}$
4.137pu	0.0717pu	0.0717pu	4.2087pu	4.2087pu

Table G.1: Machine Parameters for a 1.68 MVA WRIM [19]

$\mathbf{S_{base}}$	$\mathbf{V_{base}}$	$\omega_{\mathbf{base}}$	**PL**	$\mathbf{T_{base}}$
7.5kW	415kV	314.16rad/s	6 Poles	71.6196Nm

J	**H**	$\mathbf{r_s}$	$\mathbf{r_r}$
$0.573kgm^2$	0.4189s	0.1284pu	0.10194pu

$\mathbf{X_M}$	$\mathbf{X_{ls}}$	$\mathbf{X_{lr}}$	$\mathbf{X_s}$	$\mathbf{X_r}$
2.6255pu	1.0515pu	0.5773pu	3.677pu	3.2028pu

Table G.2: Machine Parameters for a 7.5kW WRIM [1]

$\mathbf{S_{base}}$	$\omega_{\mathbf{base}}$	R	ρ
1.5MW	1.3963rad/s	40m	$1.224kg/m^3$

Table G.3: Wind Turbine Model Parameters

References

[1] C. Watthanasarn, *Optimal Conrol and Application of AC-AC Matrix Converters*, PhD, University of Bradford, 1997.

[2] S. Muller, M. Deicke, and M. De Doncker, *Adjustable Speed Generators for Wind Turbines Based on Doubly-Fed Induction Machines and 4-Quadrant IGBT Converters Linked to the Rotor*, Industry Applications Conference, 2000. Conference Record of the 2000 IEEE **4**, 2249 (2000).

[3] S. Muller, M. Deicke, and R. De Doncker, *Doubly fed induction generator systems for wind turbine*, Industry Applications Magazine, IEEE **8**, 26 (2002).

[4] P. Cartwright, L. Holdsworth, J. Ekanayake, and N. Jenkins, *Co-ordinated voltage control strategy for a doubly-fed induction generator (DFIG)-based wind farm*, Generation, Transmission and Distribution, IEE Proceedings- **151**, 495 (2004).

[5] L. Zhang and C. Watthanasarn, *A Matrix Converter Excited Doubly-Fed Induction Machine as a Wind Power Generator*, Power Electronics and Variable Speed Drives, 1998. Seventh International Conference on , 532 (1998).

[6] R. Pena, J. Clare, and G. Asher, *Doubly fed induction generator using back-to-back PWM converters and its application to variable speed wind-energy generation*, Electric Power Applications, IEE Proceedings- **143**, 231 (1996).

[7] R. Pena, J. Clare, and G. Asher, *A doubly fed induction generator using back-to-back PWM converters supplying an isolated load from a variable speed wind turbine*, Electric Power Applications, IEE Proceedings- **143**, 380 (1996).

[8] L. Huber, D. Borojevic, D. Zhuang, and F. Lee, *Design and Implementation of a Three-Phase to Three-Phase Matrix Converter with Input Factor Correction*, Applied Power Electronics Conference and Exposition, 1993. APEC '93.Conference Proceedings 1993., Eighth Annual , 860 (1993).

[9] L. Huber and D. Borojevic, *Space Vector Modulated Three-Phase to Three-Phase Matrix Converter with Input Power Factor Correction*, Industry Applications, IEEE Transactions on **31**, 1234 (1995).

[10] L. Zhang, C. Wattahanasarn, and W. Shepard, *Application of a Matrix Converter for the Power Control of a Variable-Speed Wind-Turbine Driving a Doubly-Fed Induction Generator*, Industrial Electronics, Control and Instrumentation, 1997. IECON 97. 23rd International Conference on **2**, 906 (1997).

[11] L. Empringham, P. Wheeler, and J. Clare, *Intelligent Commutation of Matrix Converter Bi-directional Switch Cells using Novel Gate Drive Techniques*, Power Electronics Specialists Conference, 1998. PESC 98 Record. 29th Annual IEEE **1**, 707 (1998).

[12] P. Wheeler, J. Clare, and L. Empringham, *Matrix Converters*, Power Engineering Journal **16**, 173 (2002).

[13] P. Wheeler, J. Clare, and L. Empringham, *A Matrix Converter Induction Motor Drive using Intelligent Gate Drive Level Current Commutation Techniques*, Industry Applications Conference **3**, 1936 (2000).

[14] S. Ferreira Pinto and J. Fernando Silva, *Input filter design for sliding mode controlled matrix converters*, Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual **2**, 648 (2001).

[15] K. Zhou and D. Wang, *Relationship between Space-Vector Modulation and Three-Phase Carrier-Based PWM: A Comprehensive Analysis*, Industrial Electronics, IEEE Transactions on **49**, 186 (2002).

[16] S. Muller, U. Ammann, and S. Rees, *New modulation strategy for a matrix converter with a very small mains filter*, Power Electronics Specialist, 2003. PESC '03. IEEE 34th Annual Conference on **3**, 1275 (2003).

[17] P. Wheeler and D. A. Grant, *Reducing the Semiconductor Losses in a Matrix Converter*, IEE Colloquium on Variable Speed Drives and Motion Control , 14/1 (1992).

[18] N. Burany, *Safe control of four-quadrant switches*, Industry Applications Society Annual Meeting, Conference Record of IEEE **1**, 1190 (1989).

[19] P. Krause, O. Wasynczuk, and S. Sudoff, *Analysis of Electric Machinery* (New York, IEEE Press, 1986).

[20] K. Ogata, *Modern Control Engineering*, 4th ed. (Englewood Cliffs, N.J. : Prentice Hall, 2002).

[21] Q. Wang and L. Chang, *An Independent Maximum Power Extraction Strategy for Wind Energy Conversion Systems*, IEEE Proceedings on Electrical and Computer Engineering **1**, 1142 (1999).

[22] Q. Wang and L. Chang, *An Independent Maximum Power Extraction Algorithm for Iverter-Based Variable Speed Wind Turbine Systems*, IEEE Transactions on Power Electronics **19**, 1242 (2004).

[23] Mathworks, *Simulink SimPowerSystems Reference*.

[24] P. Q. Joint Technical Committee EL-034, *Limits Assessment of emission limits for fluctuating loads in MV and HV power systems*, Australian/New Zealand Standard Electromagnetic compatibility (EMC) **AS/NZS 61000.3.7:2001** (2001).

[25] B. Bose, *Modern power electronics and AC drives* (Upper Saddle River, NJ : Prentice Hall, 2002).