Universal Utility Interface for Plug-in Hybrid Electric Vehicles with Vehicle-to-Grid Functionality

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nathan David Weise

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advisor: Ned Mohan

August, 2011

Acknowledgements

It is my honor and distinct pleasure to thank the many people who have guided me through my graduate schooling. I would like to express my sincere gratitude to my advisor, Professor Ned Mohan, for giving me the opportunity to work in the area of power electronics, for guiding me through the research topics, and finally, for believing in me. I feel honored to be a student of Professor Mohan's.

I would like thank my committe members, Professor Bruce Wollenberg, Professor William Robbins, and Professor Willard Miller.

I gratefully acknowledge the financial support from Institute of Renewable Energy and Environment, University of Minnesota.

Special thanks to Dr. Hariharan Krishnaswami for all of advice, ideas, and guidance, when I was just getting started. Thanks to Dr. Krushna K. Mohapatra for all the guidance during my initial stages of research. Thanks to Apurva, Kaushik, Gysler, Rashmi, Ranjan for the various discussions, debates, collaboration, ideas, and support.

I would like to thank my parents who have believed in me from the start and provided unfailing love and support.

Dedication

This thesis is dedicated to my parents, Nancy and Jim, my love Katelin, my stepmother Ruth, my sisters, Caitlyn and Ashley, my Grandparents Grandma Weise, Grandpa Weise, and Grandma Keil, my aunts, uncles, cousins, and the rest of my family and friends. I would have never accomplished what I have without the huge support system of family and friends. I am blessed to have them and for that I am eternally grateful.

Abstract

Utility transmission lines are under-utilized during the night. Charging Plug-in Hybrid Electric Vehicles during the night and redistributing the power stored in the batteries utilizes the transmission lines more efficiently and is the motivation for this thesis. In an example scenario, a vehicle is fully charged at home during the night, and is driven to work in the morning. The vehicle is plugged in at work and continues to charge. Under peak power demand the grid can command some power from these vehicles, relieving stress on the utility. This thesis develops an in-vehicle interface to achieve this functionality.

A novel topology is developed for an all-in-one system for electric vehicles. The system is a bi-directional battery charger when plugged into single or three-phase, and doubles as the motor inverter when the vehicle is driven. The novel topology allows to source or sink power from the grid. A control scheme was developed for single-phase and three-phase inputs and simulation results are presented. A prototype of the topology is developed. Results from the prototype are compared against the simulation results and theory. Different modulation strategies are presented and compared.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Renewable energy sources like wind, solar, and wave are becoming more predominant because of the rising demand of power, the dependence on foreign oil, and the detrimental effects of converting fossil fuels to power. Due to rising oil and gas prices and the drive to become more green, vehicle companies like Toyota , General Motors, Nissan, are developing more Hybrid Electric Vehicles (HEV), Plug-in Hybrid Electric Vehicles (PHEV), and Electric Vehicles (EV). With the introduction PHEV's and EV's into the market and the increasing popularity, demand, and availability of these vehicles, the grid is becoming more stressed. While more PHEV's and EV's hit the street each year, the availability to implement Vehicle-to-grid (V2G) becomes more feasible and practical.

Parking lots full of PHEV's and EVs can be used for ancillary services [1]. Also, V2G can become distributed generation and supply power to the grid during peak loads [2] [3] [4]. V2G can be used to supply reactive power to the grid [5]. V2G can be used for home back up power. It can also be used to set up a micro grid for military type applications.

1.1 State of the Art

There are many topologies investigated in the literature for Vehicle-to-Grid applications [6, 7, 8, 9, 10, 11]. The topologies are either isolated or not isolated. The most common topology is the grid tied inverter shown in Fig. 1.1 [12] [13] [14]. A three-phase grid-tied inverter is shown, but in household applications it will be a single-phase grid-tied inverter.

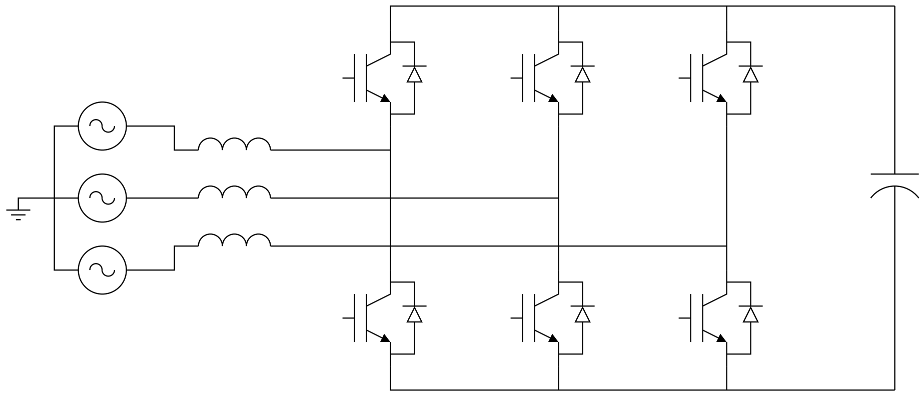


Figure 1.1: Grid Tied Inverter

The grid tied inverter is not an isolated topology but is well researched and documented [13] [14]. The advantage of this topology, besides its widespread use, is that the current can be controlled because of the input inductances in the line. Typically the switching frequency is large, compared to the input voltage frequency, leading to very small switching ripple on the input current. This leads to input currents with very low Total Harmonic Distortion. Another advantage of this topology is it is able to be reused. The PHEV or EV motor's inverter can have contactors that disconnect from the motor and connect with inductors in series with the input line voltages [7]. This alleviates car manufactures from the cost of two separate inverters but makes the system more complex from a systems engineer point of view. Lastly, there is only one switch or diode in the path of the current for each phase, leading to very efficient operation compared to topologies with more stages of conversion.

Another topology researched in EV applications is the multilevel inverter [15]. The advantage of this topology is there are more levels and the current ripple can be reduced, but the disadvantage is more switches and more complex control.

One topology reuses the motor windings as the input inductances for the grid tied inverter. These type of topologies are termed integrated because they reuse parts already there for traction, ie the motor windings in this case [16, 17, 18].

The Z-source inverter is a bi-directional topology [19]. The draw back with this topology is the additional passive components , two inductors, two capacitors, and one

IGBT. These additional components add cost and increase reliability concerns.

There are also isolated topologies for PHEV and EV charging. One area of isolated topologies are the inductive charging topologies [20] [21] [22]. These topologies have no electrical connection therefore they have the advantage of isolation for safety. However inductive topologies are less efficient than there conductive counterparts.

Another isolated topology includes reusing the motor as an isolation transformer at 60Hz [23]. This topology has the benefit of reusing 2 sets of 3 phase windings as an isolation transformer. The drawback is the design of motor because it has two separate conflicting goals. On one hand the motor design is optimized for traction applications and on the other hand the motor is designed for 60Hz isolation applications. There also can be high amounts of magnetizing current depending on the design of the motor. Another drawback is the possibility of creating a sinusoidal varying flux in air gap, leading to rotation of the motor shaft which is undesirable when stationary and charging.

The multi port converter is another isolated topology that has the ability to draw power from multiple sources [24, 25, 26, 27, 28, 28, 29]. The multi port converter is better suited for multiple DC sources, for example a fuel cell and a battery pack.

Typically isolated topologies for AC to DC are created by using two converters back to back [30] [31]. An example of this type of topology is shown in Fig. 1.2.

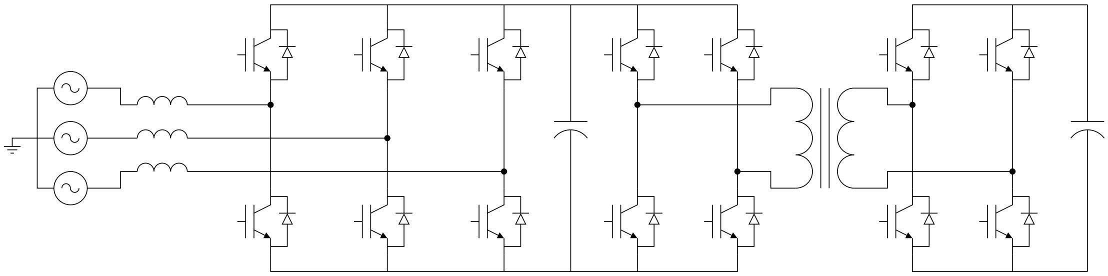


Figure 1.2: Grid Tied Inverter back to back Dual Active Bridge

The first converter is a bidirectional AC to DC grid-tied inverter. The second converter is a DC to DC isolated dual active bridge [32, 33, 34, 35, 36, 37, 38]. The disadvantage of this topology is the DC link in between the two converters. The capacitor raises the cost of the topology and also reliability concerns. The other disadvantage of this converter, compared to the grid tied inverter, is the efficiency. The efficiency is the product of the efficiency of the inverter and efficiency of the dual active bridge. The advantage here is the DC bus is isolated from the AC source.

The back to back current link isolated topology is shown in Fig. 1.3 [39].

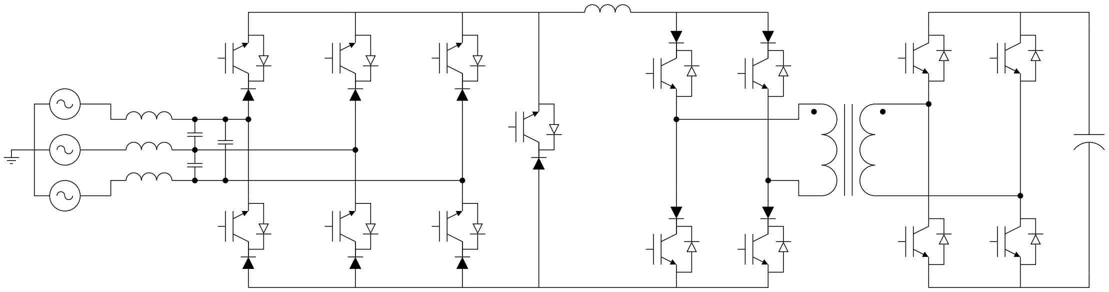


Figure 1.3: Isolated Current Link AC to DC Converter

This topology creates a current link with the inductor. The disadvantage of this topology is the extra inductor needed. Also, the size of the inductor is large adding cost and reducing power density. The efficiency also is lower than the DC link structure because the current flows through additional diodes.

Another topology is the resonant high frequency link converter 1.4 [40].

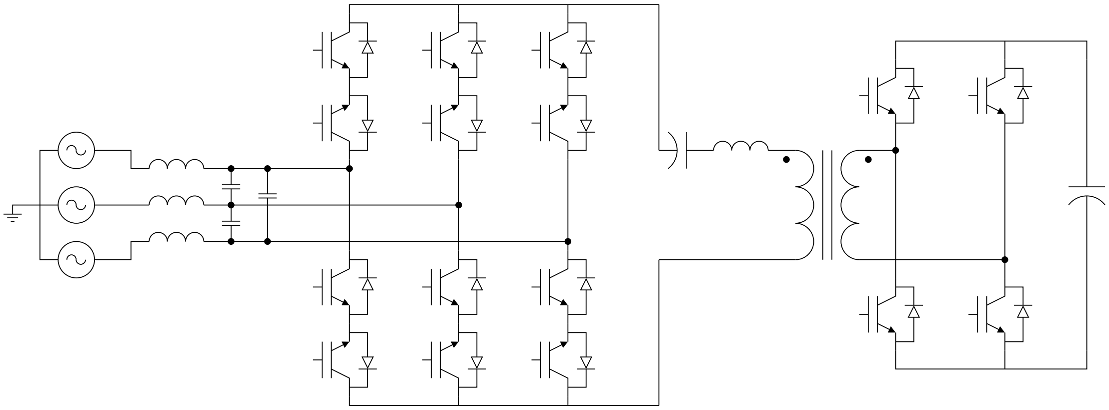


Figure 1.4: Isolated Resonant Link Converter

An advantage of this topology is that it does not create an intermediate current or voltage link. A disadvantage of this converter is the additional resonant tank elements, capacitor and inductor.

1.2 Proposed Converter

The proposed converter, in which this thesis and research is based, is shown in Fig. 1.5. During traction operation, meaning the vehicle is moving from place to place, the transformer is disconnected from the inverter by means of contactors. The DC-DC converter maintains a constant DC bus voltage and the inverter employs some type of modulation to drive the motor. During charging or discharging operation, the vehicle is stationary, the motor is disconnected from the inverter by means of contactors. The motor inverter is reused as a H-Bridge for the bidirectional isolated AC to DC converter. The inductance, L, of the transformer is the lumped leakage inductance referred to the secondary. The input converter is a three-phase to single-phase matrix converter. The switches are four quadrant switches and are realized with a pair of emitter tied IGBT's. The input can accept single-phase for household type applications or it can accept three-phase for high power fast charging installations. This converter is bidirectional in nature and can source or sink power. This thesis will consider this topology only for charging and discharging operations. The bidirectional DC-DC converter between the DC bus and the battery pack will be modelled as a load resistance R when charging and as a

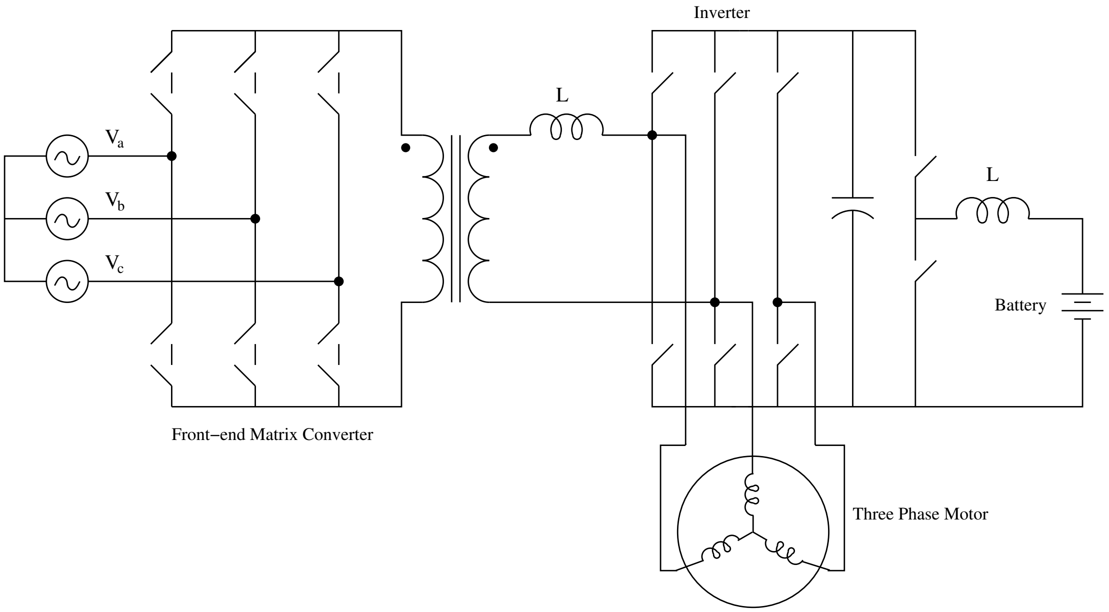


Figure 1.5: Proposed Topology

DC source when discharging. This topology has the following advantages over isolated back to back AC to DC converters: no intermediate DC link capacitor or current link inductor, higher reliability, higher power density, and smaller volume.

1.3 Motivation and Scope

Grid-tied inverters are predominant in EV chargers. There is no AC to DC isolated bidirectional charger in literature without an intermediate current or voltage link. The motivation of this thesis is to present a new topology applicable to EV's , Solar installations, home backup, and military mirco-grid applications. Another objective of this thesis is to provide modulation and control strategies for single-phase and three-phase input. This thesis analyzes the proposed modulation and control, and provides simulation and hardware results.

1.4 Contributions

The contributions of this thesis are:

- A novel Single-Phase/Three-Phase AC to DC isolated bidirectional converter
- Detailed analysis of the converter for single-phase and three-phase control
- Detailed analysis of DAB modulation strategy for single-phase and three-phase input
- Design procedure for DAB modulation strategy of the converter
- Hardware results confirming the theoretical analysis of DAB modulation strategy for single-phase and three-phase input

1.5 Organization

Chapter 1 introduces the current state of the art research in electric vehicle bidirectional chargers. It also introduces a novel converter which is the subject of this thesis. In Chapter 2 the converter is considered with single-phase input. A model of the average power flow per cycle, based on the first harmonic, is derived and used as the basis of control. Controllers are designed for the power and voltage loops. Results are given for the input currents, harmonics, and bus voltages. Chapter 3 discusses control of the converter based on three-phase input. The control uses a duty ratio based method to create a quasi square wave. A power and voltage controller are designed. Input currents, current harmonics, and output voltages results are given and discussed. Chapter 4 introduces a new modulation method for single-phase AC to DC dual active bridge. The input current, current harmonics, and bus voltage simulation results are given and match theoretical analysis. Chapter 5 introduces a new modulation strategy for three-phase AC to DC dual active bridge. The input currents, power, utilization factor, rms ripple current are all analysed. Simulation results are given to confirm the analytical results. Lastly in Chapter 6 and 7, the hardware is presented, and results for the DAB modulation method are given in single-phase and three-phase input.

1.6 Conclusion

In Chapter 1 the context and layout of the thesis is established. EV bidirectional chargers are introduced and a novel isolated bidirectional topology is introduced.

Chapter 2

Single Phase Power and Voltage Control

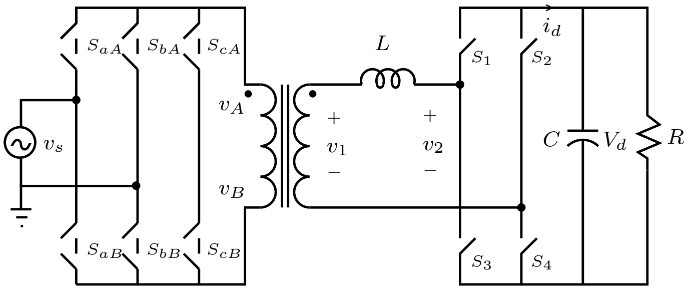
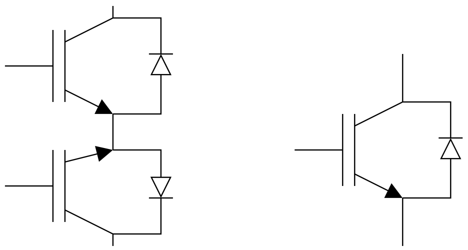


Figure 2.1: Single Phase Topology

The topology is investigated for single-phase input and is shown in Fig. 2.1. The switches S_{aA}, S_{bA}, S_{cA}, S_{aB}, S_{bB}, and S_{cB} are four quadrant switches. The four quadrant switches are realized as shown in Fig. 2.2a. The transformer magnetizing inductance is neglected, the turns ratio is assumed to be one, and the total leakage inductance is refereed to the secondary and represented by L. The switches S_1, S_2, S_3, and S_4 are reused from the motor inverter and are two quadrant switches as shown in Fig. 2.2b. The capacitor, C, is the DC Bus capacitor and the load is modelled as R during charging. When feeding power back to the grid, R will be removed and the DC bus will be fed by a voltage source.



(a) Four Quadrant

(b) Two Quadrant

Figure 2.2: Switches

2.1 Analysis

The input matrix converter is comprised of four-quadrant switches S_{aA}, S_{bA}, S_{cA}, S_{aB}, S_{bB}, and S_{cB}. In the single-phase configuration switches S_{cA} and S_cB are not connected to a phase therefore will be considered in the off state during the analysis. The input voltage, $v_s(t)$, is sinusoidal with frequency ω_i, as shown in (2.1). During this analysis I have selected ω_i as $2\pi 60$ or 60Hz and $\hat{V}_s$ as $120\sqrt{2}$ because it is the most common in the U.S.

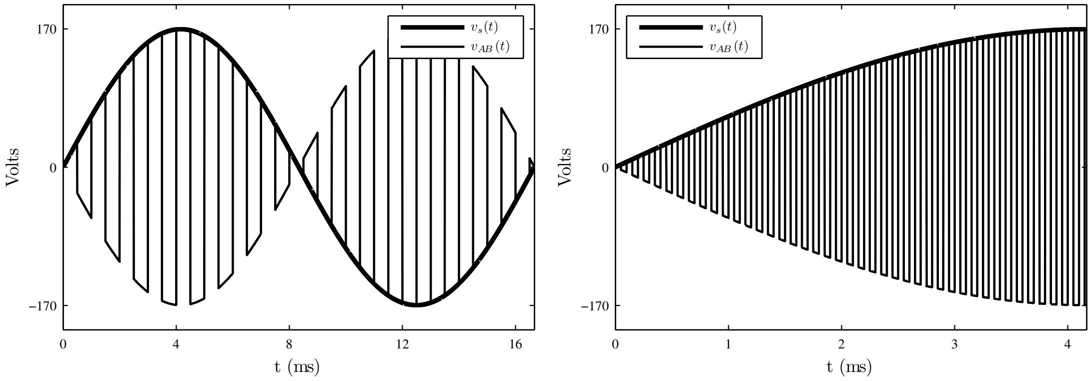
$$v_s(t) = \hat{V}_s \sin(\omega_i t) \tag{2.1}$$

The input matrix converter has two states. The first state is when switches S_{aA} and S_{bB} are on and switches SbA and S_{aB} are off. During this state the voltage $+v_s(t)$ is applied to v_{AB}. In the second state, switches S_{bA} and S_{aB} are on and switches SaA and S_{bB} are off. During this state the voltage $-v_s(t)$ is applied to v_{AB}. The states are on for 50% of the time at a frequency of f_s. The voltage applied to v_{AB} is shown in (2.2).

$$v_{AB}(t) = \widehat{V}_s \sin(2\pi f_i t)\,\mathrm{sgn}(\sin(2\pi f_s t)) \tag{2.2}$$

Fig. 2.3a shows the output of the matrix converter, for a lower f_s of $1kHz$, for illustrative purposes and Fig. 2.3a shows the output of the matrix converter for a f_s of $10kHz$.

For this analysis, I have picked the turns ratio of the transformer to be one, $n = 1$, therefore $v_1(t)$ is equal to $v_{AB}(t)$. The output of the half-bridge ,$v_2(t)$, is a square wave of magnitude V_d. The output square wave of the half bridge is at frequency f_s. The output of the half bridge is represented mathematically in (2.3). The duty ratio of the square



(a) 1kHz

(b) 10kHz

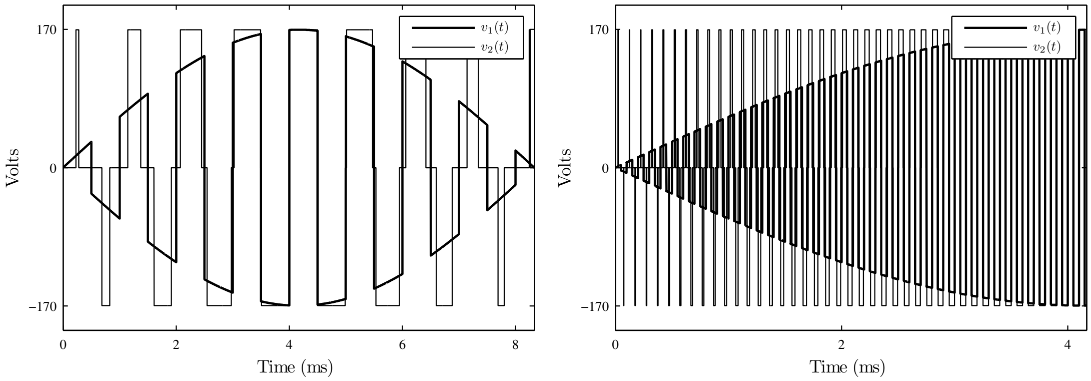
Figure 2.3: Matrix Converter Output

wave is adjustable from zero to one by adjusting $\phi(t)$. For example if $d = 1$, then $\phi = 0$, therefore both squares are in phase in (2.3) and add to a square wave of magnitude V_d. On the contrary, if $d = 0$, then $\phi = \frac{\pi}{2}$, therefore both square waves are out of phase by 180 degrees and sum to zero. The duty ratio is given in (2.4). This selection of the duty ratio ensures that for any half period of f_s, the average voltage applied to $v_1(t)$ is equal to the average voltage applied to $v_2(t)$. For visualization purposes, (2.3) has been plotted in Fig. 2.4a at $f_s = 1kHz$ and in Fig. 2.4b at $f_s = 10kHz$.

$$\begin{aligned} v_2(t) =& \frac{V_d}{2}[\text{sgn}\left[\sin(2\pi f_s t - \phi(t) - \theta(t))\right] \\ &+ \text{sgn}\left[\sin(2\pi f_s t + \phi(t) - \theta(t))\right]] \end{aligned} \tag{2.3}$$

$$d(t) = 1 - \phi(t)\frac{2}{\pi} \tag{2.4}$$

The phase shift $\theta(t)$ is zero in both Fig. 2.4a and Fig. 2.4b. In order to do power flow there must be a finite phase shift. In the next section a power controller will be evaluated, designed, and simulated.



(a) 1kHz

(b) 10kHz

Figure 2.4: Bridge Outputs

2.1.1 Power Controller

The power controller is designed to control the power in and out of the converter. The plant is assumed to be similar to the power systems analogy as shown in (2.5) [41, 42]. The power in (2.6) is the average power over a cycle of f_s at time t. Note that in power systems the two waveforms are sine waves but in this scheme they are square waves, therefore in this analysis the model is valid only for the first harmonic power flow. The fundamental of v_1 is $\frac{4}{\pi}$ times the voltage at that time. The fundamental of v_2, considering a duty ratio of 1,is $\frac{4}{\pi}$ times the bus voltage V_d. The control diagram of the systems is shown in Fig. 2.5. In single-phase the power command is multiplied by a sine squared reference to insure sinusoidal input current with respect to the input voltage. The actual power transferred is measured by multiplying i_d by V_d, integrating that product during a cycle of f_s and dividing by T_s to get the average power transferred in that cycle.

$$P = \frac{V_1 V_2}{\omega L} \sin\theta \tag{2.5}$$

$$P(t) = \frac{16}{\pi^2} \frac{v_1(t) v_2(t)}{\omega_s L} \sin\theta \tag{2.6}$$

A simple Proportional Integral controller is selected for control. The equation in (2.6) is given a small perturbation about θ. The equation is then solved in (2.7), (2.8), (2.9), and the result is (2.10). In (2.10), $v_1(t)$ and $v_2(t)$ vary with time, but their product vary in a sine squared fashion. The reference is also multiplied by a sine squared reference therefore $v_1(t)$ is replaced by the constant peak input voltage $\hat{V}_s$, and $v_2(t)$ is replaced by the constant bus voltage V_c.

$$P + \widetilde{P} = \frac{16}{\pi^2} \frac{v_1(t)v_2(t)}{\omega_s L} \sin(\theta + \Delta\theta) \tag{2.7}$$

$$P + \widetilde{P} = \frac{16}{\pi^2} \frac{v_1(t)v_2(t)}{\omega_s L} [\cos\theta \sin\Delta\theta + \cos\Delta\theta \sin\theta] \tag{2.8}$$

$$P + \widetilde{P} = \frac{16}{\pi^2} \frac{v_1(t)v_2(t)}{\omega_s L} [\Delta\theta \cos\theta + \sin\theta] \tag{2.9}$$

$$\frac{\widetilde{P}}{\Delta\theta} = \frac{16}{\pi^2} \frac{v_1(t)v_2(t)}{\omega_s L} \cos\theta \tag{2.10}$$

The PI controller is designed around a linearized plant as shown in Fig. 2.6. The plant is a constant gain around the operating point θ. The PI controller is selected because even though the plant we design the control loop around is a constant, the actual plant is not linear, and the PI controller will take care of the errors associated with the model inaccuracies.

The gain of the power stage is defined in (2.11). In order to design the controller gains a bandwidth and phase margin are chosen to suit the application. Next equations (2.12) and (2.13) are solved to determine the controller gains k_i and k_p. The values for solving the system of equations, (2.12) and (2.13), are shown in Table 2.1.

$$K_{pow} = \frac{\hat{V}_s \hat{V}_d}{\omega_s L} \cos\theta \tag{2.11}$$

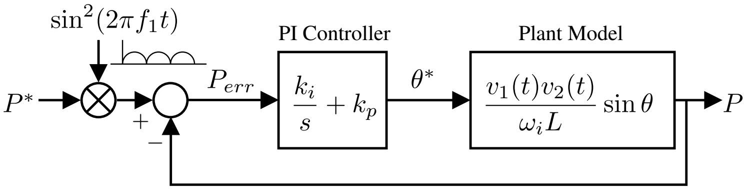


Figure 2.5: Actual System

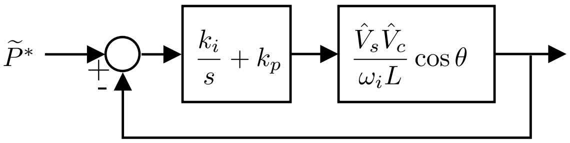


Figure 2.6: Linearized Control Design

$$|G_{OL}(s)|_{s=j2\pi f_{cp}} = \left| \frac{k_p s + k_i}{s} K_{pow} \right|_{s=j2\pi f_{cp}} = 1 \tag{2.12}$$

$$\angle G_{OL}(s)|_{s=j2\pi f_{cp}} + 180 = \phi_{PM} \tag{2.13}$$

Parameter	**Value**	**Unit**
$\hat{V}_s$	170	V
$\hat{V}_d$	300	V
f_s	10	kHz
L	50	μH
θ	45	degrees
ϕ_{PM}	60	degrees

Table 2.1: Control Design Variables

The results for the design of the controller gains are shown in Table 2.2. The complete system in Fig. 2.1 is simulated using the designed PI controller gains. The load R is assumed to be 45Ω and C is 1000μF.

Parameter	**Value**
k_p	4.3e-5
k_i	0.5

Table 2.2: Controller Gains

A power command of 4kW is given to the controller in Fig. 2.5. The results of the actual power flow plotted with the commanded power are shown in Fig. 2.7. The results of the power command show the actual power tracking the sine squared reference power command. The results of the average input current and input voltage are shown in Fig.

2.8a. The average input current is sinusoidal and in phase with the input voltage thus obtaining unity power factor. A fourier analysis is done on the switched current $i_s(t)$, in Fig. 2.8b, to determine the harmonics present in the input current. The results of the fourier analysis are shown in Fig. 2.9a and Fig. 2.9b. The spectrum in Fig. 2.9a shows the fundamental at 60Hz, some low level line frequency harmonics, and the switching harmonics at 10kHz. The switching harmonics can be removed from the input current by properly designing a LC input filter. In Fig. 2.9b the fourier analysis is zoomed in to capture the magnitude levels of the line frequency harmonics. The 3rd harmonic is present at 3% and the 5th harmonic is present at 0.8%.

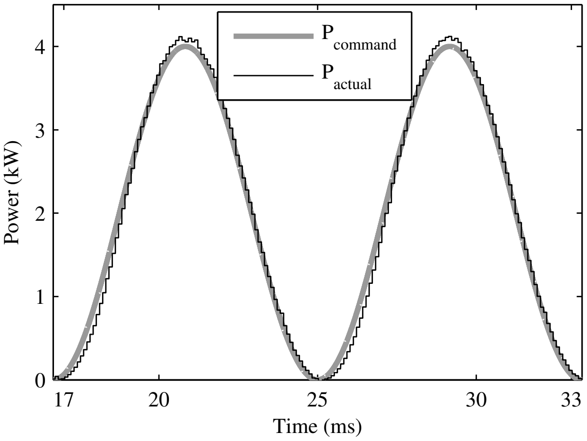
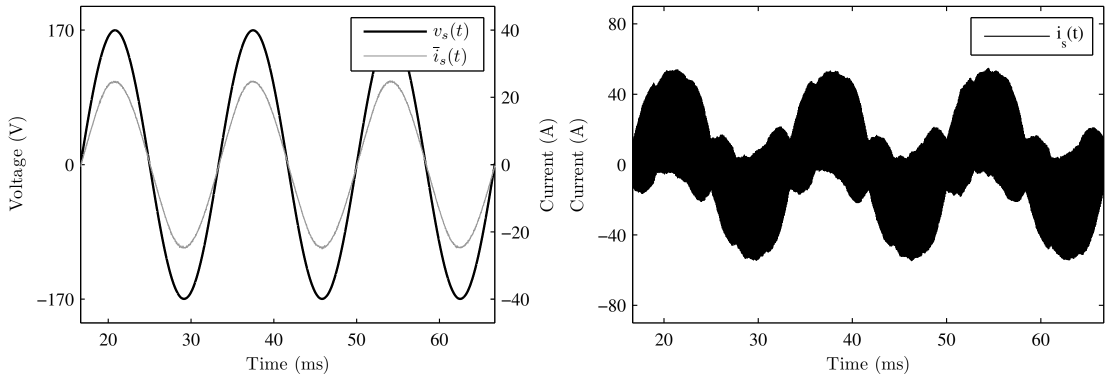


Figure 2.7: Power Command

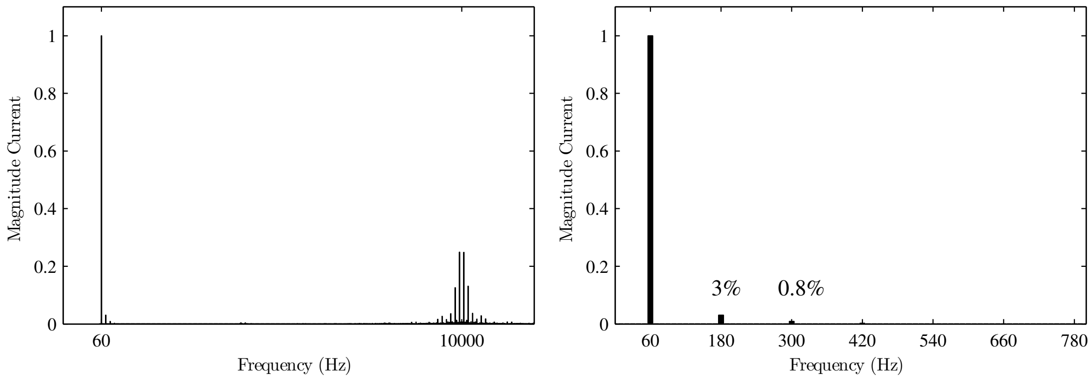
The present values of the 3rd and 5th harmonic meet the requirements for maximum current distortion in IEEE spec but further effort can be done to reduce these harmonics. The power controller is given step command from 4kW down to 2kW in order to investigate the dynamics of the power controller design. The step function, power command, and actual power are plotted in Fig. 2.10. The actual power settles within 5ms which is sufficient because in battery charging applications, the command won't change often.



(a) Average Input Current

(b) Input Switch Current

Figure 2.8: Input Currents



(a) Fourier Analysis

(b) Line Frequency Harmonics

Figure 2.9: Harmonic Analysis

2.1.2 Power Controller Redesign

The power controller design provided acceptable current harmonics in the input current but will be redesigned in order to reduce the 3rd harmonic. The third harmonic is present because attenuation at 120Hz is not low enough which is similar to a PFC circuit. This time Eq. (2.12) and (2.14) are used to design the gains of the power controller. Eq. allows us to set the gain of the third harmonic of the closed loop system.

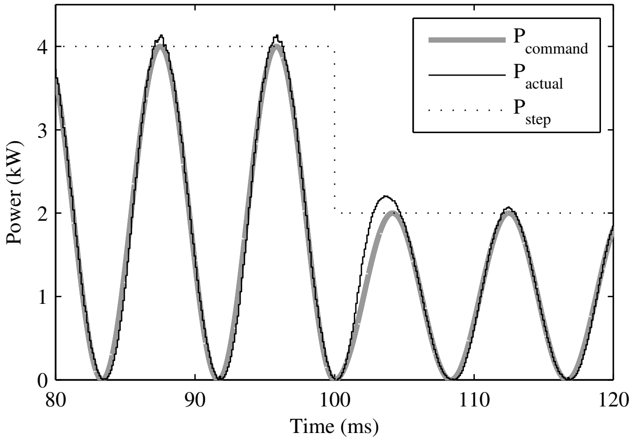


Figure 2.10: Power Step Command

$$|G_{OL}(s)|_{s=j2\pi 180} = \left| \frac{k_p s + k_i}{s} K_{pow} \right|_{s=j2\pi 180} = x \tag{2.14}$$

The new gains are shown in Table 2.3. The system is simulated with the controller gains and the response in a step power command from 4kW to 2kW are shown in Fig. 2.11. The bandwidth of the controller lowered due to the lowering of the gain at the third harmonic. Again, the bandwidth is acceptable because during the course of charging a battery the power command does not change often. The fourier spectrum of the input current is shown in Fig. 2.12a and 2.12b.

Parameter	**Value**
k_p	4.3e-5
k_i	0.1

Table 2.3: Controller Gains

Similar to the previous control design, there is current at 60Hz, and the switching frequency of 10kHz. The third and fifth harmonic are now both below 1% of the fundamental. The conclusion of the new power controller design is that the 3rd and 5th harmonics can both be reduced below 1% at the expense of the control loop bandwidth.

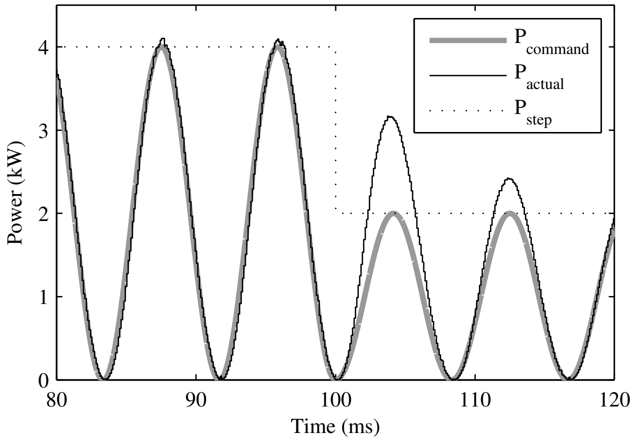
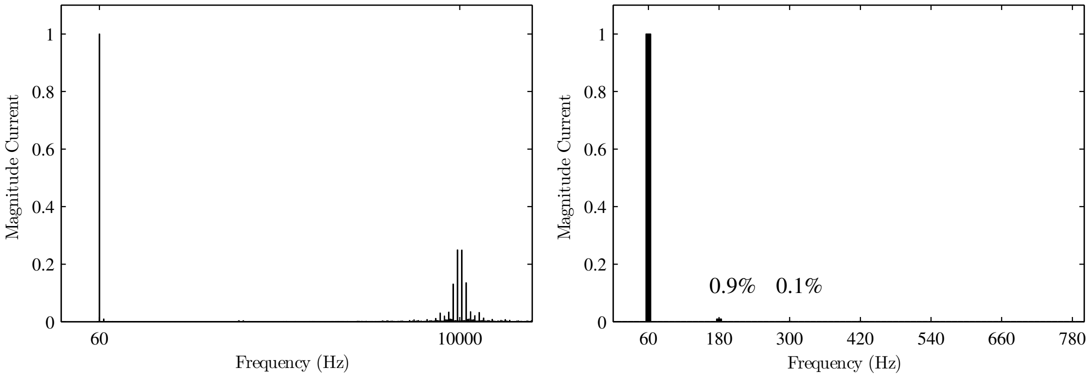


Figure 2.11: Power Step Command



(a) Fourier Analysis

(b) Line Frequency Harmonics

Figure 2.12: Harmonic Analysis

2.1.3 Steady State Analysis

In this section the steady state for $v_d(t)$ will be analyzed and solved for. The circuit shown in Fig. 2.13 represents the average system. The average current $\bar{i}_d(t)$ (2.15) flows into the RC circuit. The current $i_s(t)$ and duty cycle $d(t)$ are defined in (2.16) and (2.17). The current $i_d(t)$ is derived in (2.18) and is of sine squared form. The steady state voltage is solved in Laplace domain because it is simple. The voltage $V(s)$ is solved for in (2.19) and then reverse transformed back to time domain. The current $I(s)$ is transformed to Laplace domain in (2.20). The impedance of the RC circuit is defined in (2.21).

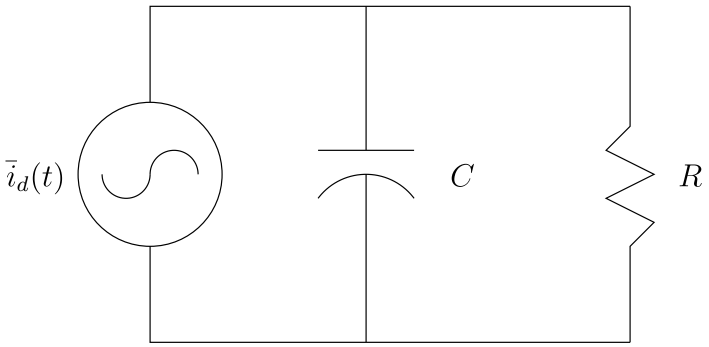


Figure 2.13: Simplified Voltage Circuit

$$\bar{i}_d(t) = \bar{i}_s(t)d(t) \tag{2.15}$$

$$\bar{i}_s(t) = \hat{I}_s \left|\sin \omega_i t\right| \tag{2.16}$$

$$d(t) = \frac{\hat{V}_s}{V_d} \left|\sin \omega_i t\right| \tag{2.17}$$

$$\bar{i}_d(t) = \hat{I}_s \frac{\hat{V}_s}{V_d} \sin^2 \omega_i t \tag{2.18}$$

$$V(s) = I(s)Z(s) \tag{2.19}$$

$$\mathcal{L}(\sin^2 \omega_i t) = \frac{1}{2}\mathcal{L}(1 - \cos 2\omega_i t) = \frac{1}{2}\left[\frac{1}{s} + \frac{s}{s^2 + (2\omega_i)^2}\right] = I(s) \tag{2.20}$$

$$Z(s) = \frac{\frac{1}{sC}R}{\frac{1}{sC} + R} = R\frac{1}{1 + \frac{s}{1/RC}} \tag{2.21}$$

Eq. (2.22) defines the magnitude k of the sine squared average current flowing into the RC circuit. Eq. (2.20) and (2.21) are multiplied together then partial fraction decomposition is used to get the result (2.23). The first fraction in (2.23) is the DC

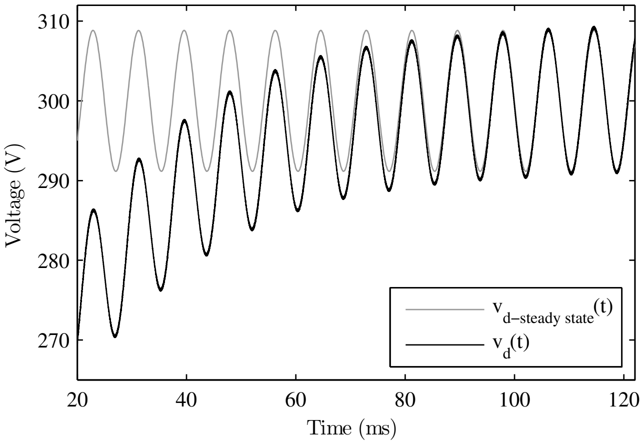


Figure 2.14: Steady State Voltage

component of the voltage $v_d(t)$. The second fraction is a decay component so for steady state it is disregarded. The third term in (2.23) contains the second harmonic component of $v_d(t)$. Taking the inverse Laplace transform of the first and third fraction in (2.23) leads to the time domain equation of $v_d(t)$ (2.24) (2.25). The peak to peak ripple voltage is defined in (2.26).

$$k = \hat{I}_s \frac{\hat{V}_s}{V_d} \tag{2.22}$$

$$V(s) = \frac{kR}{2s} - \frac{2C^3kR^4\omega_i^2}{(1+CRs)\left(1+4C^2R^2\omega_i^2\right)} + \frac{-kRs - 4CkR^2\omega_i^2}{2\left(s^2+4\omega_i^2\right)\left(1+4C^2R^2\omega_i^2\right)} \tag{2.23}$$

$$z = \sin^{-1}\left[\frac{1}{\sqrt{1+(2w_iRC)^2}}\right] \tag{2.24}$$

$$v_d(t) = \frac{kR}{2} + \frac{kR}{2\sqrt{1+(2w_iRC)^2}}\sin(z - 2\omega_i t) \tag{2.25}$$

$$v_{d,pp-ripple} = \frac{R\hat{I}_s\hat{V}_s}{2V_c\sqrt{1+(2w_iRC)^2}} \tag{2.26}$$

The system is simulated for a 2kW load where $R = 45$ and $C = 1000\mu F$. The results of the simulation for $v_d(t)$ are shown in Fig. 2.14. The steady state equation of $v_d(t)$ (2.25) is plotted in grey in Fig. 2.14 and the actual system is plotted in black. The simulation shows the system powering up and converges to the steady state solution by time 80ms. Fig. 2.14 validates the steady state analysis.

2.2 Power Plant Error

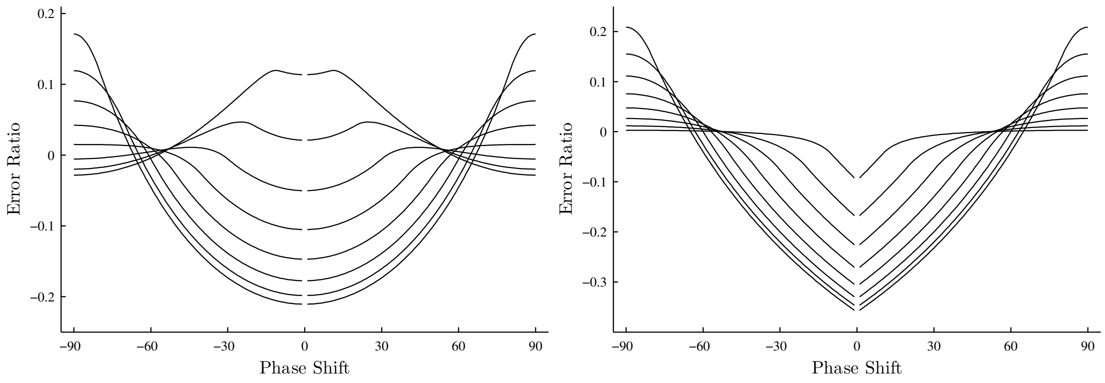
The power controller is designed around a plant that is an model of the actual system and because of this point, there will be a difference between the power that flows in the actual system and the power that flows in the model of the system. The error between the two will be explored in order to determine the amount of error. The three power flows will be investigated as follows: 1st harmonic power flow, square wave power flow, and the actual power flow. The average power flow of the first harmonic of the square waves is calculated in (2.27). The first harmonic of v_1 is a sine wave of magnitude $\frac{4}{\pi}120\sqrt{2}$. The first harmonic of v_2 is a sine wave of magnitude $\frac{4}{\pi}120\sqrt{2}\cos\phi$. The power is calculated across ϕ, ranging from 10 to 80 degrees in steps of 10 degrees, and across θ, ranging from -90 degrees to 90 degrees. Next the average power is calculated for two square waves in (2.28) [43, 44, 45, 46]. v_1 is a square wave of magnitude $120\sqrt{2}\cos\phi$ and v_2 is a square wave governed by (2.3). For each ϕ, in steps of 10 degrees from 10 to 80 degrees, the power is calculated for all θ from -90 to 90 degrees. Lastly, the power of the actual systems is calculated by simulation for ϕ, in steps of 10 degrees from 10 to 80 degrees, and all θ from -90 degrees to 90 degrees.

$$P_{1st-harmonic} = \frac{V_1 V_2}{\omega_s L}\sin\theta \tag{2.27}$$

$$P_{square} = \frac{V_1 V_2}{\omega_s L}\theta\left(1 - \frac{|\theta|}{\pi}\right) \tag{2.28}$$

The difference between the actual power flow and the 1st harmonic and square wave power flows will give insight into the accuracy of the model. The error between the 1st harmonic power flow and the actual power flow is calculated in (2.29) and shown in Fig. 2.15a. The error between the square wave power flow and the acutal power is calculated in (2.30) and shown in Fig. 2.15b.

$$E_{1st-harmonic} = \frac{P_{actual} - P_{1st-harmonic}}{P_{1st-harmonic}} \tag{2.29}$$



(a) 1st Harmonic Error

(b) Square Wave Error

Figure 2.15: Power Error

$$E_{square} = \frac{P_{actual} - P_{square}}{P_{square}} \tag{2.30}$$

The error between the 1st harmonic power flow and the actual power flow ranges from .18pu to -.2pu for all θ and given set points of ϕ. The error between the square wave power flow and the actual power flow ranges from .21pu to -.35pu for all θ and given set points of ϕ. The error is not minimal by any means but is small enough that if a robust controller is properly designed it can account for the error present in this analysis.

2.3 Voltage Controller Design

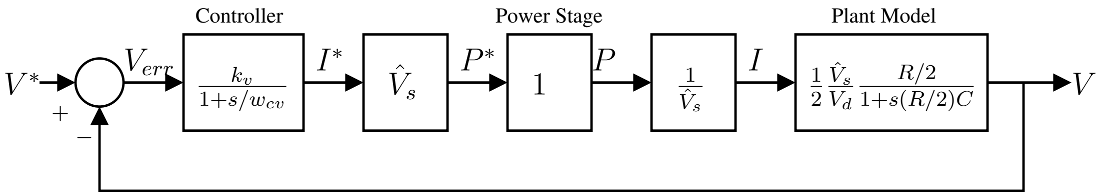


Figure 2.16: Voltage Control Loop

The voltage control loop is depicted in Fig. 2.16. This circuit is similar to a single-phase PFC in that the current into the RC circuit is of the form of a sinusoid squared.

The power stage transfer function is shown in (2.31) and is derived in [47]

$$\frac{v_d(s)}{i_d(t)} = \frac{1}{2}\frac{\hat{V}_s}{V_d}\frac{R/2}{1+s(R/2)C} \tag{2.31}$$

One approach to the voltage loop would be to select the controller as a low pass, as shown in (2.32). There will always be steady state error when this type of controller is used because the absence of a pole at the origin. Eq. (2.33) is defined to select the loop bandwidth and Eq. is defined to ensure 1.5% or less 2nd harmonic ripple in the output current command. These equations are solved as a system of equations to determine the gain k_v and pole w_{cv}.

$$G_v(s) = \frac{k_v}{1+s/w_{cv}} \tag{2.32}$$

$$\left|\frac{k_v}{1+s/w_{cv}}\frac{1}{2}\frac{\hat{V}_s}{V_c}\frac{R/2}{1+s(R/2)C}\right|_{s=j2\pi f_{cv}} = 1 \tag{2.33}$$

$$\left|\frac{k_v}{1+s/w_{cv}}\right|_{s=j2\pi 120} = \frac{.015\hat{I}_s}{V_{d,pp-ripple}} \tag{2.34}$$

The parameters used to solve for the pole and gain are shown in Table 2.4. The gain and pole are shown in Table 2.5. The whole system is simulated with aforementioned voltage controller and the results are investigated.

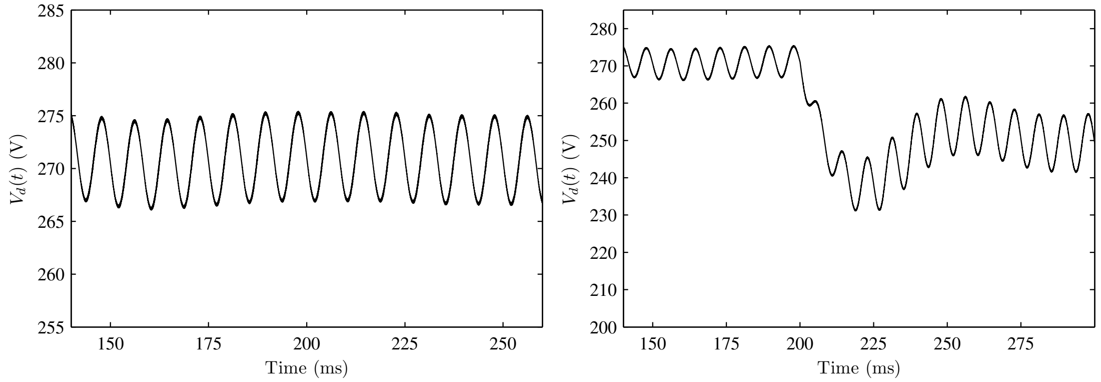
Parameter	**Value**	**Unit**
$\hat{V}_s$	170	V
$\hat{V}_d$	300	V
R	45	Ω
C	1000	μF
f_{cv}	10	Hz
$\hat{I}_s$	23.5294	A
$\hat{V}_{d,pp-ripple}$	8.84	V

Table 2.4: Voltage Control Design Variables

The voltage $v_d(t)$ is plotted in Fig. 2.17. The voltage contains an expected 120Hz component and also has a steady state error of about 30 volts.

Parameter	Value
w_{cv}	94.7329
k_v	0.3259

Table 2.5: Voltage Control Loop Parameters



(a) Steady State Voltage

(b) Voltage Load Step

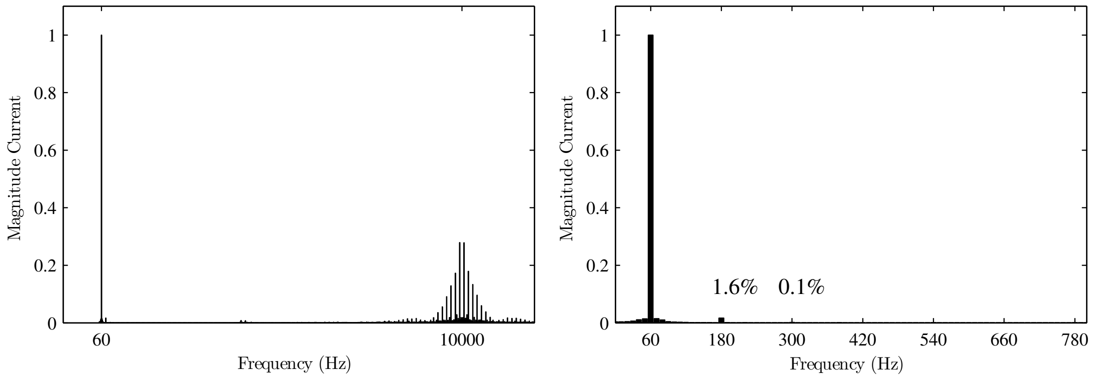
Figure 2.17: Output Voltage

The input current spectrum of $i_s(t)$ under steady state is shown in Fig. 2.18a and Fig. 2.18b. The input current contains the fundamental at 60Hz, some line frequency harmonics and the switching component at 10kHz. From the simulation data the magnitude of the third harmonic is 1.6% which matches closely with the designed value of 1.5%. If the application can work without closely tracking the 300 volts then the simple low pass controller will suffice.

2.3.1 Voltage Controller Redesigned

In order to track the voltage command and not have steady state error one must introduce a pole at the origin. A proportional integral or PI controller is selected to obtain zero steady state error. The voltage control loop is shown in Fig. 2.19 with a PI controller instead of the low pass controller.

The controller gains k_i and k_p were selected by using the ratio of k_i and k_p to create a zero to cancel the pole of the plant at full load. Then k_i was adjusted to create of crossover frequency of 10Hz. The control parameters determined from the analysis are



(a) 10kHz Spectrum

(b) Line Frequency Harmonics

Figure 2.18: Input Current Spectrum

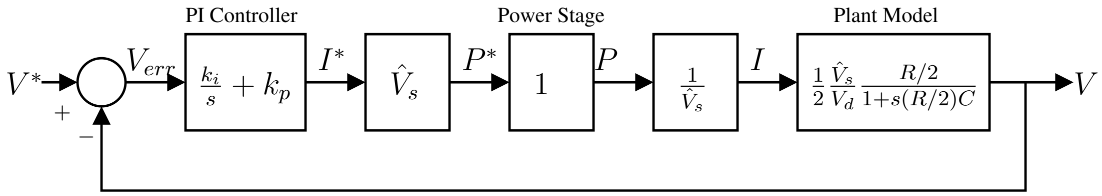


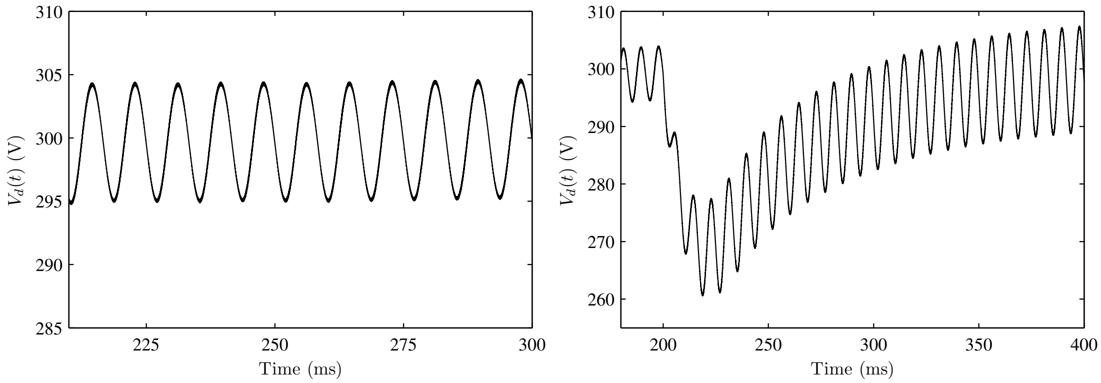
Figure 2.19: Voltage Control Loop

shown in Table 2.6. The entire system is simulated using the controller in Fig. 2.19 and gains in Table 2.6. The output voltage is shown in Fig. 2.20a. The output voltage now tracks the 300 Volt command due to the presence of a pole at the origin.

Parameter	**Value**
k_i	5
k_p	0.2250

Table 2.6: Voltage Control Loop Parameters

The system is given a step load at 200ms in which R changes from 90Ω to 45Ω which is equal to a step load from 1kW to 2kW. The results of the voltage due to the step change are plotted in Fig. 2.20b. The voltage recovers to 300V in 200ms. The voltage ripple also is larger with a larger load. The input current is investigated for the PI controller. The input current spectrum is shown in Fig. 2.21a and 2.21b. The input



(a) Voltage Steady State

(b) Voltage Step Load

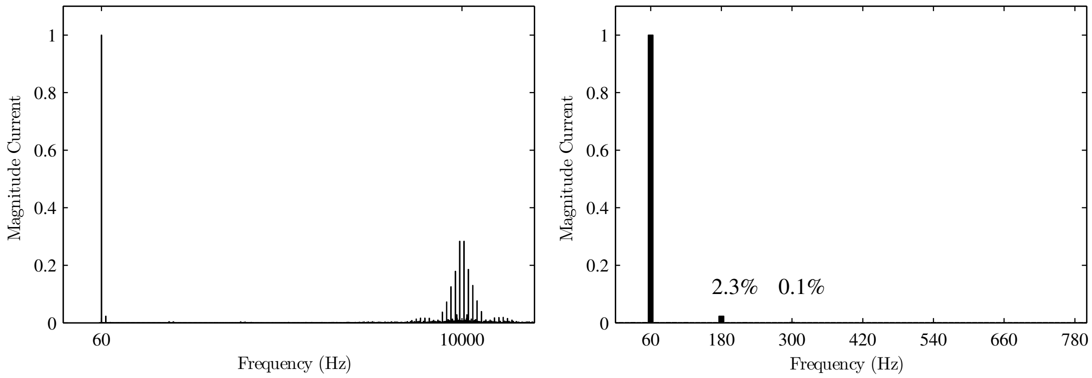
Figure 2.20: Output Voltage

current contains line 2.6% at the third harmonic, 0.1% at the fifth harmonic and the harmonics above the fifth are negligible. The total THD of the input current, neglecting the switching frequency harmonics, is 2.6% The third and fifth harmonics are higher than the low pass controller because the gain at those frequencies are higher and the design did not focus on reducing these. A PI controller with an additional pole past the cut-off frequency but before 120Hz would help reduce the third and fifth harmonics if it is desired to reduce the harmonics further but the harmonics present using the PI controller still conform to the IEEE 519 standard thus no further control will be investigated.

2.3.2 Bidirectional Power Flow

The topology in Fig. 2.1 is changed to Fig. 2.22 for the purpose of simulating bidirectional power flow.

The DC source represents a battery pack or a bidirectional DC to DC converter boosting the battery pack voltage [48] [49] [50] [51]. The topology in Fig. 2.22 is simulated with DC bus of 300 Volts. The voltage loop is removed and only the power loop is used. The parameters in Table 2.1 and 2.2 are used for simulation. The power controller is given a command of -4kW which is equivalent to supplying 4kW to the AC source from the DC source. The results of the average current and input voltage are shown in Fig. 2.23.



(a) 10kHz Spectrum

(b) Line Frequency Harmonics

Figure 2.21: Input Current Frequency Spectrum

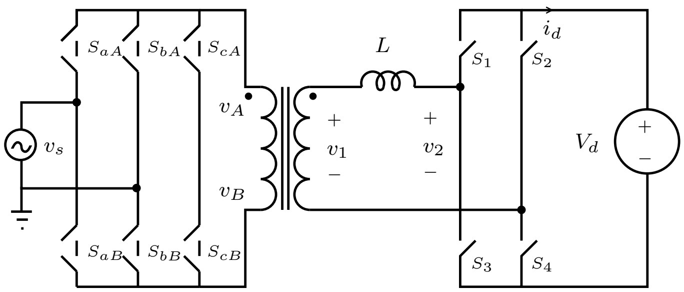


Figure 2.22: Bidirectional Simulation Topology

The current is sinusoidal and 180 degrees out of phase with the input voltage proving that bidirectional power flow is possible with the given topology and control. The input current harmonics, normalized to the fundamental, are shown in Fig. 2.24.

The input current contains 2.2% at the third harmonic and 0.4% at the fifth harmonic. The rest of the harmonics are negligible. The total harmonic distortion of the input current is 2.23% which is with in IEEE-519 specifications [52]. The percentage of current at the third and fifth is also within IEEE-519 specification.

2.4 Conclusion

In this chapter the single phase bidirectional charger is proposed. First a power controller was analyzed and designed. The error between the plant model and the actual

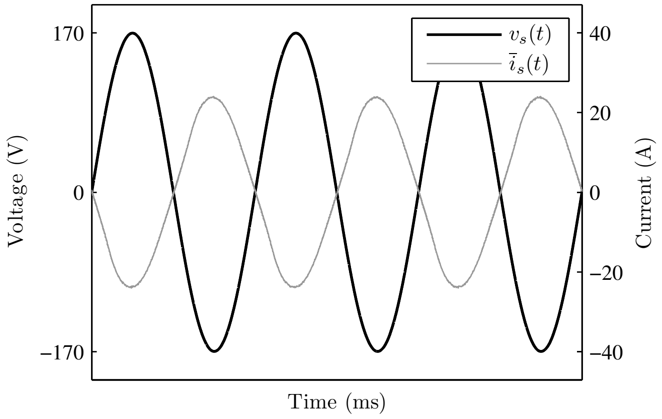


Figure 2.23: Bidirectional Power Flow

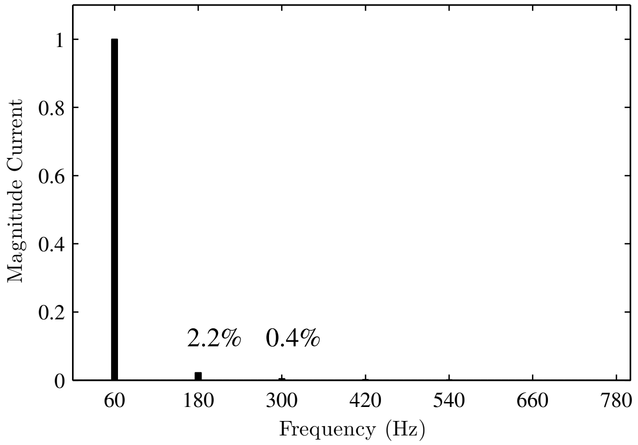


Figure 2.24: Current Harmonics

system was investigated to determine the accuracy of the model. It was determined the model to be accurate enough to use for design of the power control loop. Next, steady state analysis was done to determine the output voltage equations. Lastly, two different voltage loop controllers we discussed, analyzed, and presented. The input current was analyzed during each controller design to maintain low line frequency harmonics and low THD.

Chapter 3

Power and Voltage Control with Three Phase Input

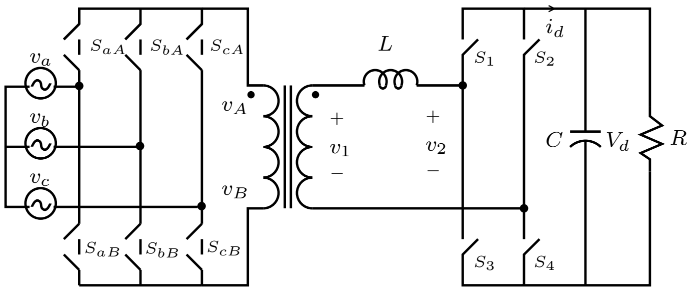


Figure 3.1: Three Phase Topology

The topology in Fig. 3.1 is explored for three-phase input. The sinusoidal input sources are defined in (3.1) and have a magnitude of V_i at a frequency of ω_i.

$$\begin{aligned}
v_a(t) &= V_i \sin(\omega_i t) \\
v_b(t) &= V_i \sin(\omega_i t - \frac{2\pi}{3}) \\
v_c(t) &= V_i \sin(\omega_i t - \frac{4\pi}{3})
\end{aligned} \tag{3.1}$$

The switches S_{aA}, S_{bA}, S_{cA}, S_{aB}, S_{bB}, and S_{cB} are four quadrant switches comprised of two emitter tied insulated gate bipolar transistors. In this analysis the magnetizing inductance is neglected and the leakage inductance of the transformer is lumped to the secondary (3.2). The turns ratio of the transformer is assumed to be unity. The switches S_1, S_2, S_3, and S_4 are reused from the motor inverter and are two quadrant switches realized as IGBT's. The DC bus capacitor for the inverter is reused as a DC source and the load resistance R is simulating charging a battery. The load resistance, R, and capacitor, C, will be replaced with a DC source when simulating supplying power to the grid.

$$L = n^2 L_p + L_s \tag{3.2}$$

3.1 Analysis

The three-phase to single-phase matrix converter consists of switches S_{aA}, S_{bA}, S_{cA}, S_{aB}, S_{bB}, and S_{cB}. Each of the three switches S_{aA}, S_{bA}, and S_{cA} are given a duty ratio defined in (3.3), (3.4), and (3.5) [53].

$$\begin{aligned} d_{aA}(t) =& K_i \sin(\omega t) + \frac{1}{2}|\sin(\omega t)| \\ &+ \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.3}$$

$$\begin{aligned} d_{bA}(t) =& K_i \sin(\omega t - \frac{2\pi}{3}) + \frac{1}{2}|\sin(\omega t - \frac{2\pi}{3})| \\ &+ \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.4}$$

$$\begin{aligned} d_{cA}(t) =& K_i \sin(\omega t - \frac{4\pi}{3}) + \frac{1}{2}|\sin(\omega t - \frac{4\pi}{3})| \\ &+ \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.5}$$

Similarly, the three switches S_{aB}, S_{bB}, and S_{cB} are given a duty ratio defined in (3.6), (3.7), and (3.8). The only difference in duty ratios between the top switches and the bottoms switches is the sign of K_i. All six duty ratios are shown in Fig. 3.2a and Fig. 3.2b.

$$\begin{aligned} d_{aB}(t) =& - K_i \sin(\omega t) + \frac{1}{2}|\sin(\omega t)| \\ & + \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.6}$$

$$\begin{aligned} d_{bB}(t) =& - K_i \sin(\omega t - \frac{2\pi}{3}) + \frac{1}{2}|\sin(\omega t - \frac{2\pi}{3})| \\ & + \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.7}$$

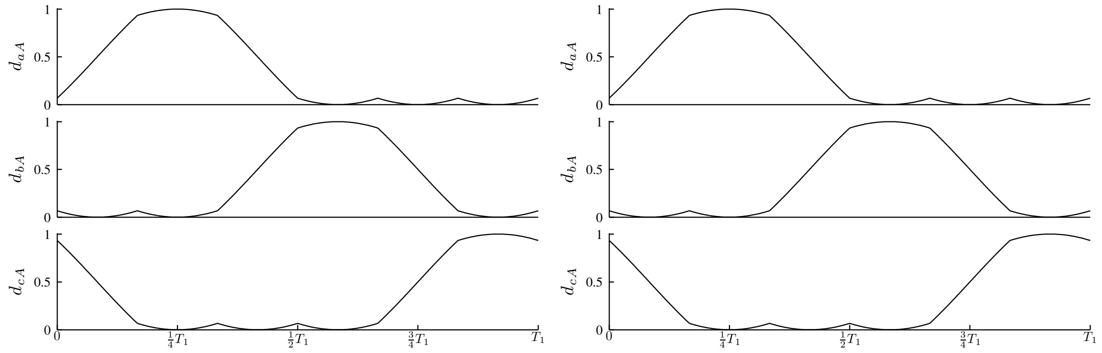
$$\begin{aligned} d_{cB}(t) =& - K_i \sin(\omega t - \frac{4\pi}{3}) + \frac{1}{2}|\sin(\omega t - \frac{4\pi}{3})| \\ & + \frac{1}{6}\left[2 - |\sin(\omega t)| - |\sin(\omega t - \frac{2\pi}{3})| - |\sin(\omega t - \frac{4\pi}{3})|\right] \end{aligned} \tag{3.8}$$

Now that each switch has been given a duty ratio, the actual switching signals must be generated. The duty ratios are compared to an inverter triangle ramp shown in Fig. 3.3. The switch S_{aA} is on when d_{aA} is greater than the ramp and is off otherwise (3.9). The switch S_{cA} is on when $d_{aA} + d_{bA}$ is less than the ramp and is off otherwise (3.10). The switch S_{bA} is on when the ramp is greater than d_{aA} but less than $d_{aA} + d_{bA}$ and is off otherwise (3.11).

$$S_{aA}(t) = \begin{cases} 0 & 0 < t < \frac{T_s}{2}\left[1 - d_{aA}(t)\right] \\ 1 & \frac{T_s}{2}\left[1 - d_{aA}(t)\right] < t < \frac{T_s}{2}\left[1 + d_{aA}(t)\right] \\ 0 & \frac{T_s}{2}\left[1 + d_{aA}(t)\right] < t < T_s \end{cases} \tag{3.9}$$

$$S_{bA}(t) = \begin{cases} 0 & 0 < t < \frac{T_s}{2}\left[1 - d_{aA}(t) - d_{bA}(t)\right] \\ 1 & \frac{T_s}{2}\left[1 - d_{aA}(t) - d_{bA}(t)\right] < t < \frac{T_s}{2}\left[1 - d_{aA}(t)\right] \\ 0 & \frac{T_s}{2}\left[1 - d_{aA}(t)\right] < t < \frac{T_s}{2}\left[1 + d_{aA}(t)\right] \\ 1 & \frac{T_s}{2}\left[1 + d_{aA}(t)\right] < t < \frac{T_s}{2}\left[1 + d_{aA}(t) + d_{bA}(t)\right] \\ 0 & \frac{T_s}{2}\left[1 + d_{aA}(t) + d_{bA}(t)\right] < t < T_s \end{cases} \tag{3.10}$$

$$S_{cA}(t) = \begin{cases} 1 & 0 < t < \frac{T_s}{2}\left[1 - d_{aA}(t) - d_{bA}(t)\right] \\ 0 & \frac{T_s}{2}\left[1 - d_{aA}(t) - d_{bA}(t)\right] < t < \frac{T_s}{2}\left[1 + d_{aA}(t) + d_{bA}(t)\right] \\ 1 & \frac{T_s}{2}\left[1 + d_{aA}(t) + d_{bA}(t)\right] < t < T_s \end{cases} \tag{3.11}$$



(a) Top Switch Duty Ratios

(b) Bottom Switch Duty Ratios

Figure 3.2: Duty Ratios

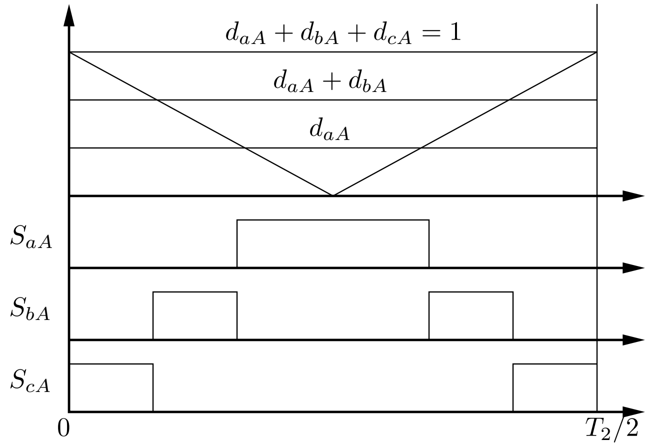


Figure 3.3: PWM Generation

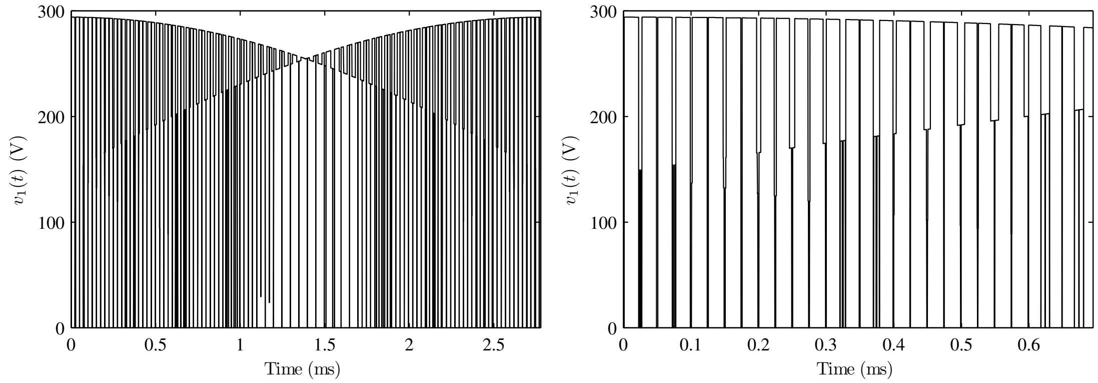
The switching signals for the bottom three switches S_{aB}, S_{bB}, and S_{cB} are generated the same exact way as previously discussed using there respective duty ratios.

The modulation is simulated to determine the output at v_{AB}, using the values in Table 3.1. The output of the matrix converter at v_{AB} is shown in Fig. 3.4a and 3.4b. Calculating the average voltage at v_{AB} during any T_s time period would result in 255 volts (3.12).

$$\bar{v}_{AB} = V_i \frac{3\sqrt{2}}{2} \tag{3.12}$$

V_i	170	Volts
f_i	60	Hz
f_s	10	kHz

Table 3.1: Modulation Values



(a) Virtual DC Link

(b) Virtual DC Link

Figure 3.4: Virtual DC Link

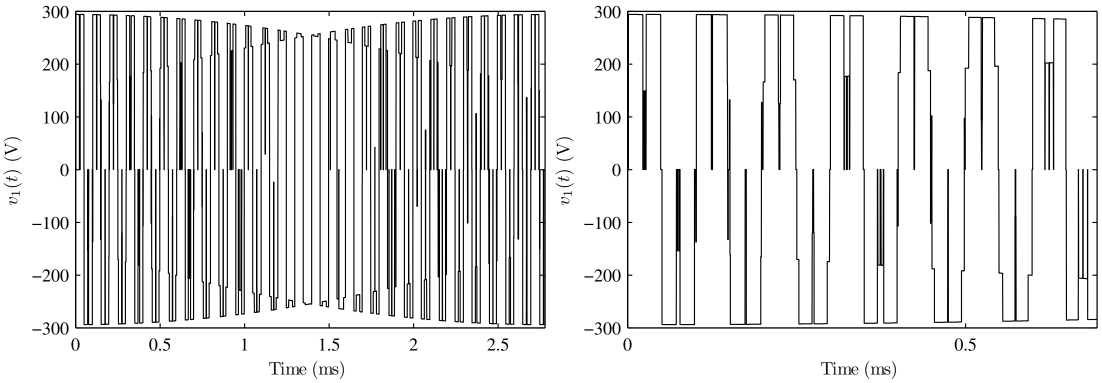
The present modulation scheme creates a virtual DC bus but the goal is to create a square wave. The modulation index K_i will be modified in (3.13) to create a square wave. The modulation index K_i now switches signs every half cycle of f_s.

$$K_i = \begin{cases} +\frac{1}{2} & 0 < t < T_s \\ -\frac{1}{2} & T_s < t < 2T_s \end{cases} \tag{3.13}$$

The simulation is run with the alternating modulation index and the results are shown in Fig. 3.5a and 3.5b.

The voltage v_{AB} is now a quasi square wave with a magnitude on the average of $3K_iV_i$. Now that the square wave is generated at v_{AB} hence v_1, the next step is to generate a square wave at v_2 with an equal average voltage during any $T_s/2$ time period.

The average voltage at v_1 will be constant $3K_iV_i$. The voltage at v_2 will by synthesized as follows, if $3K_iV_i > V_d$ then v_2 will follow (3.14), otherwise if $3K_iV_i < V_d$ the voltage v_2 will follow (3.16). Theta is the phase shift of the waveform at v_2 with respect to the waveform at v_1 and phi controls the width of the square wave at v_2. The two



(a) Quasi Square Wave

(b) Zoom Quasi Sqaure Wave

Figure 3.5: Quasi Square Wave

waveforms v_1 and v_2 are shown in Fig. 3.6. Note that the average voltage of v_1 and v_2 during any $T_s/2$ time period are equal. Power is controlled by adjusting the phase shift of v_2.

$$v_2(t) = V_d \operatorname{sgn}\left[\sin(2\pi f_s t - \theta(t))\right] \tag{3.14}$$

$$\phi(t) = \frac{\pi}{2}\left[1 - \frac{3V_i}{2V_d}\right] \tag{3.15}$$

$$\begin{aligned} v_2(t) =& \frac{V_d}{2}[\operatorname{sgn}\left[\sin(2\pi f_s t - \phi(t) - \theta(t))\right] \\ &+ \operatorname{sgn}\left[\sin(2\pi f_s t + \phi(t) - \theta(t))\right]] \end{aligned} \tag{3.16}$$

A controller to command the power into the RC network will be designed. With three-phase input, the average voltage at v_1 doesn't change unlike the single phase case. This has a distinct advantage because now the gain of the power stage doesn't vary sinusoidally with the input voltage. The power stage will be modelled like a power flow in a power system (3.17). The equation in power systems for power flow is with two sinusoidal sources but these sources are square waves so the model is an approximation.

$$P = \frac{V_1 V_2}{\omega L} \sin\theta \tag{3.17}$$

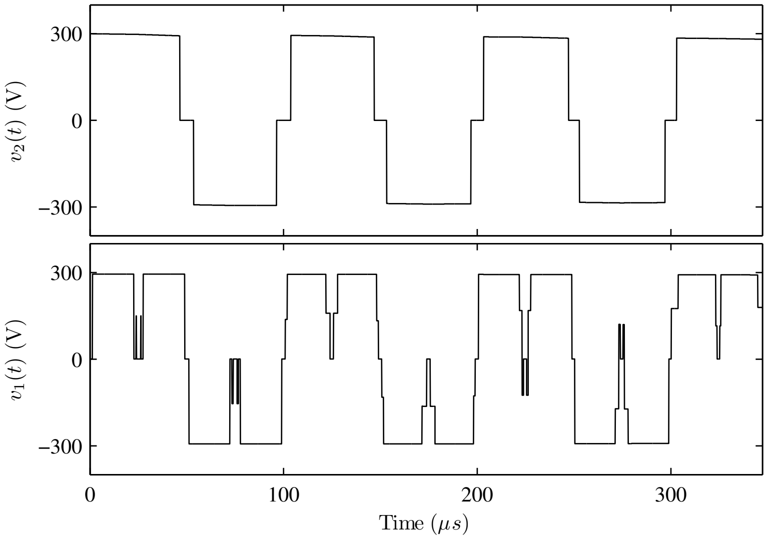


Figure 3.6: Phase Shift Voltage Waveforms

3.1.1 Modelling Error

The power systems model is not an exact model of the current system but the difference between the two is investigated to determine the amount of error. The model power is calculated using (3.17) and the actual power is calculated through simulation. The value of ϕ which determines the duty ratio of v_1 is swept from 0 to $\pi/2$ and θ is swept from $-\pi/2$ to $\pi/2$. The error ratio is calculated by subtracting the actual power from the first harmonic power and dividing by the first harmonic power. This error ratio only depends on θ and ϕ and is plotted in Fig. 3.7.

The black line envelops the extremes of the error at that specific θ and the grey area represents possible error for all θ and ϕ. The graph shows that the error remains between +10% and -30% for all operating conditions and shows that the model is reasonable accurate. The power can be controlled and the error can be reduced to zero by using a PI controller with a robust design.

3.1.2 Power Controller Design

With the model decided upon a power control loop must be designed. The power system equation (3.17) is given a perturbation about θ and gives Eq. (3.18).

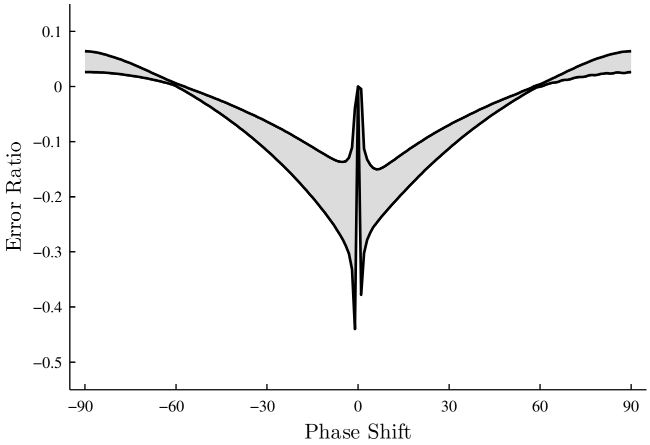


Figure 3.7: Modelling Error

$$\frac{\widetilde{P}}{\Delta\theta} = \frac{16}{\pi^2}\frac{3K_iV_iV_d}{\omega_s L}\cos\theta \tag{3.18}$$

The $\frac{4}{\pi}$ factors come from the first harmonic of the square waves at v_1 and v_2, the factor $3K_iV_i$ determines the magnitude of v_1 and V_d determines the magnitude of v_2. The equation is a constant therefore designing a PI controller is relatively simple. The power control loop design is shown in Fig. 3.8.

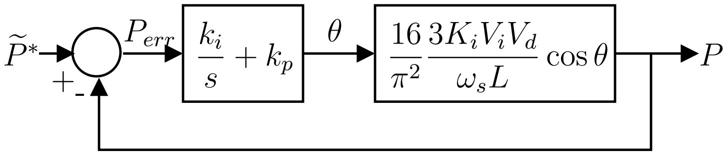


Figure 3.8: Three Phase Power Controller

The controller values k_i and k_p are calculated by solving two equations, the first one sets the open loop gain to 1 and the cross over frequency and the second sets the phase margin at the cross over frequency. The values in Table 3.2 are used to calculated the two controller gains.

The calculated controller gains come out to $k_i = 0.2293$ and $k_p = 2.1e - 5$. The controller is implemented on the real system and simulated. The load is 30Ω and the capacitor is $200\mu F$. A power command of 3kW is given to the controller at system

Parameter	Value	Unit
V_i	170	Volts
K_i	0.5	
V_d	300	Volts
f_s	10	kHz
L	50	μH
θ	45	degrees
PM	60	degrees

Table 3.2: Power Control Loop Design Variables

start up. The actual power and the command are plotted in Fig. 3.9. The actual power builds up to 3kW and tracks the command.

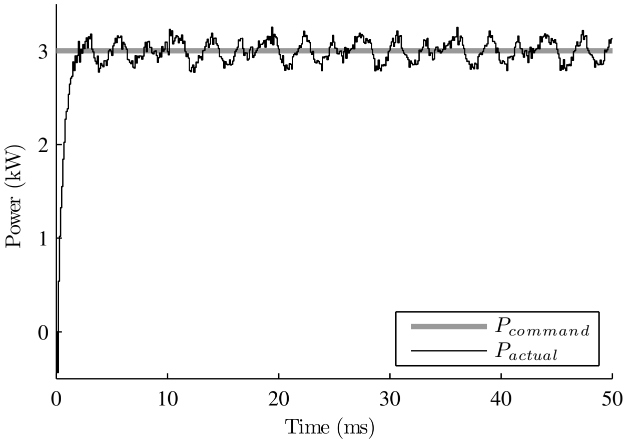


Figure 3.9: 3kW Power Command

The average current from the voltage sources v_a, v_b, and v_c are shown in Fig. 3.10. The currents are sinusoidal and in phase with the input voltage but one can determine that there are line frequency harmonics present. Ideally the currents would contain zero harmonics.

In order to give some insight, the switch currents for i_a, i_b, and i_c are shown in Fig. 3.11a, 3.11b, and 3.11c. The currents have the high frequency component at 10kHz on top of the fundamental at 60Hz. The first observation is that the currents are unbalanced

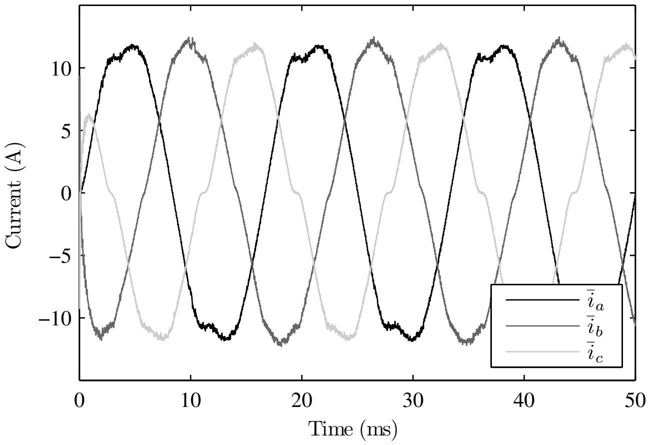
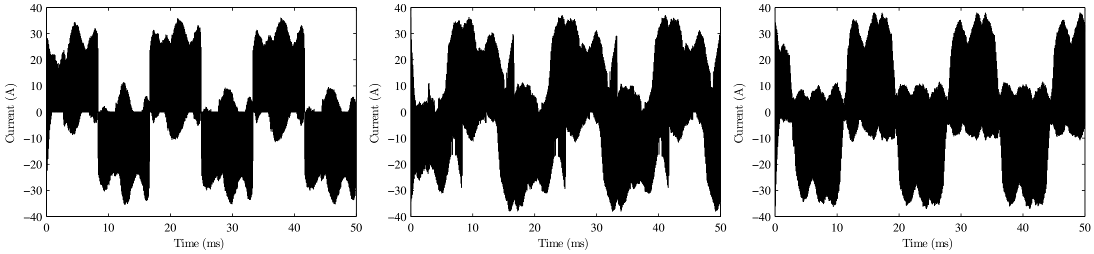


Figure 3.10: Average Currents

and each have their own different harmonics.



(a) Phase a Switch Current (b) Phase b Switch Current (c) Phase c Switch Current

Figure 3.11: Switch Currents

The Fourier spectrum is calculated on each of the three switching currents to determine the harmonic content of each. The spectra of all three currents is shown in Fig. 3.12. The harmonics are only shown up to 15th harmonic because the higher up harmonics can be attenuated with an input LC filter. The currents have a 180Hz component which is not normally possible in a balanced three phase system but is present because the converter is providing an unbalanced load. The 3rd and 5th harmonic are unbalanced and have a significant magnitude of 6%. The 7th up to 15th harmonic are also unbalanced and have a significant magnitude of 1-2%. These harmonics are unacceptable for IEEE-519 [52] specifications and are detrimental to the stability of the grid.

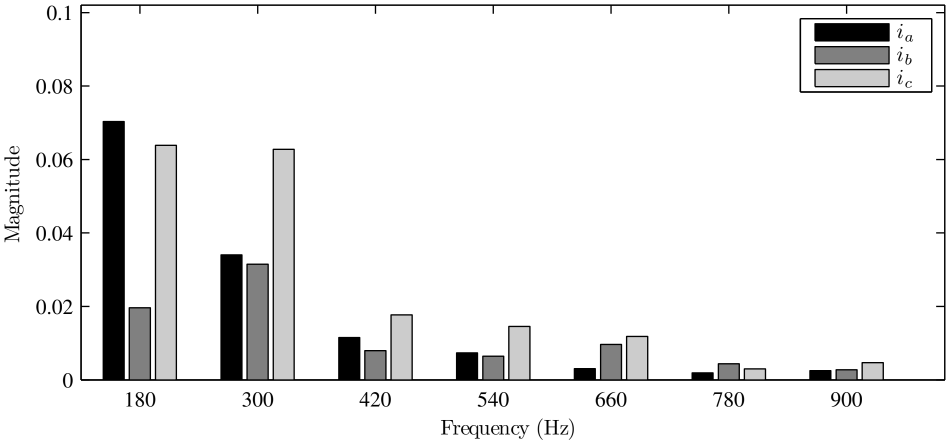


Figure 3.12: Current Harmonics

A solution to the harmonics and balancing of the load is needed before before hardware implementation.

3.1.3 Balancing Currents and Reducing Harmonics

In three phase there are 6 distinct sectors. These sectors change when one of the voltage sources changes its voltage level respective to the other two sources. These six distinct sectors are shown in Fig. 3.13. In the first sector the voltage v_a is max, v_b is min, and v_c is mid. When v_b becomes greater than v_c the sector switches to sector two and v_b becomes mid, v_c becomes min, and v_a remains max. The duty ratios follow the input voltages and maintain this behaviour. The input currents during each of these six sectors will be investigated for the three phase currents.

The switching input currents under steady state are plotted for all three phases, for one switching cycle, in each of the six sectors, in Fig. 3.14. The current i_a has a notch missing in sector two and also in sector five. This notch is due to the phase shift of the two voltage waveforms v_1 and v_2 and the current through the leakage inductance L.

The current i_b has a notch missing in sector one, three, four, and six. The current i_c has a notch missing in sector one, two, three, four, five, and six. None of the three currents have notches in a determinable pattern and the notches are defiantly not balanced between the three phases.

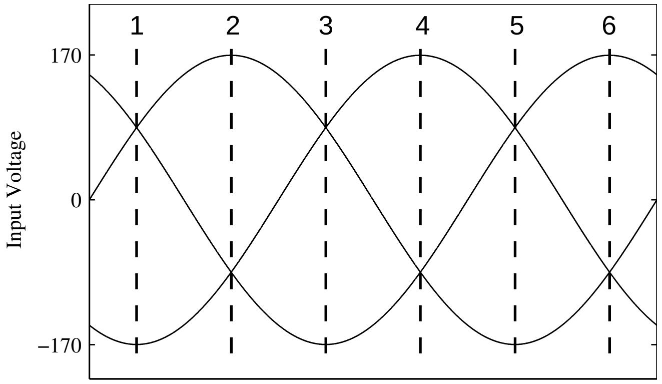


Figure 3.13: Voltage Six Sectors

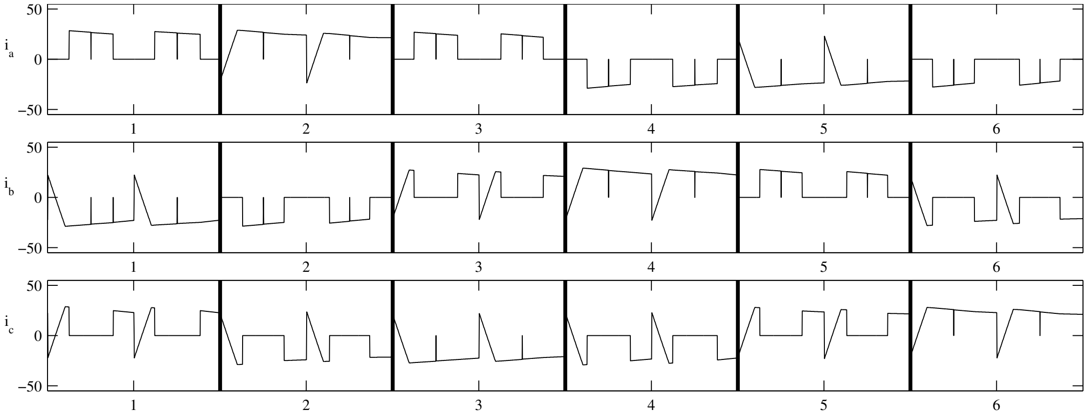


Figure 3.14: Input Current Sectors

One way to reduce the notches is to reduce the phase shift, but that is in contrast to power flow. If the allowable phase shift is reduced, then the available power flow reduces as well. Reducing the allowable phase shift is not a viable solution.

One solution is to switch the order in which the duty ratios are compared to the ramp, that way the notch comes out from each current the same amount of times irrespective of the sector. In the first switching cycle the switching signals would be generated as follows, when the ramp is less than d_{aA} then S_{aA} is on, otherwise, its off, when the ramp is greater than d_{aA} but less than $d_{aA}+d_{bA}$ then S_{bA} is on, otherwise, its off, finally when the ramp is greater than $d_{aA}+d_{bA}$ the switch S_{cA} is on other wise it is off. In the next switching cycle the switching signals would be generated as follows,

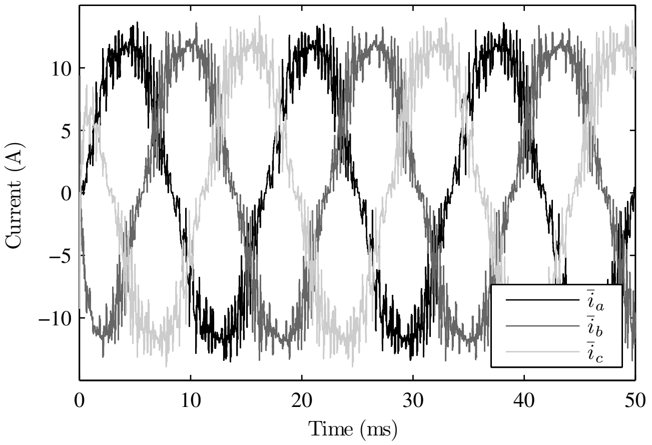
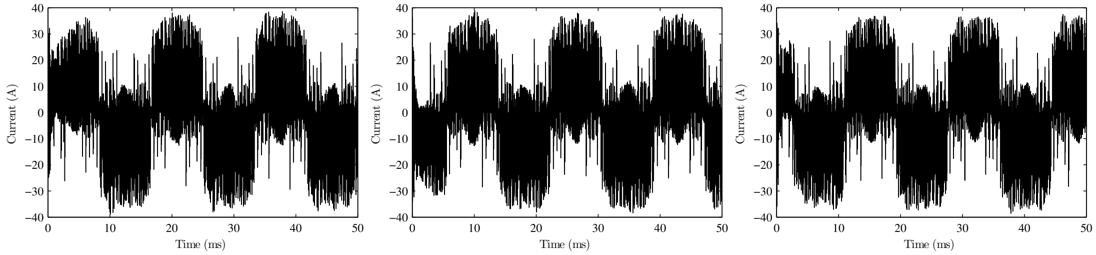


Figure 3.15: Average Three Phase Currents

when the ramp is less than d_{bA} then S_{bA} is on otherwise its off, when the ramp is greater than d_{bA} but less than $d_{bA} + d_{aA}$ then S_{aA} is on otherwise its off, finally when the ramp is greater than $d_{bA} + d_{aA}$ the switch S_{cA} is on other wise it is off. The first cycle is named abc and the second cycle is named bac because of the order in which they are compared to the ramp. The simulation is rerun with the same values but this time the duty ratios are compared in a different order each cycle. The duty ratios will be compared in a repeating six cycles. The six cycles in order are as follows, abc, bac, acb, cab, bca, and cba. The switching input current, average current, and input current harmonic content will be investigated for this new duty ratio switching scheme.

The average current for all three phases is presented in Fig. 3.15 using the duty ratio order swap. The average currents are now clearly balanced and sinusoidal. The average currents do contain some line frequency harmonics but they are perceived to be lower. The switching currents are plotted for all three phases in Fig. 3.16a, 3.16b, and 3.16c

The switching currents for all three phases are balanced as well. The outline of the fundamental of 60Hz is easily observable in all three switching currents. The Fourier analysis is done on all three switching currents to determine the harmonic content of the input currents. The harmonic content of the input switching currents are normalized to the fundamental at 60Hz. The harmonic content of the three currents are shown in Fig. 3.17.



(a) Phase a Switch Current (b) Phase b Switch Current (c) Phase c Switch Current

Figure 3.16: Switch Currents

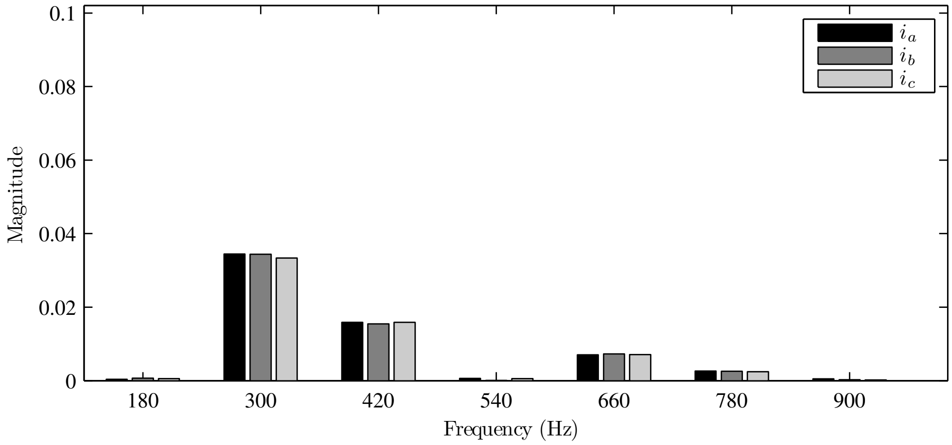


Figure 3.17: Input Current Harmonics

Clearly the third harmonic is gone from all three input switching currents contrary to the previous simulation where the harmonics were unbalanced and as high as 7% of the fundamental. The 5th harmonic is balanced and is less than 4% of the fundamental. The 7th harmonic is also balanced and less than 2% of the fundamental. The 9th harmonic is negligible compared to previously where it was 1-2% of the fundamental and unbalanced. The 11th harmonic is balanced and less that 1% of the fundamental. The 13th and 15th harmonic are negligible compared to 1% and unbalanced in case where the duty ratio comparison was done in the same order. The changing of the order in which the duty ratios are compared to the ramp resulted in a great reduction of the harmonics in the input currents. The input currents are balanced, the third, ninth, and fifteenth harmonic have been eliminated, and the harmonics in the fifth, seventh, eleventh, and thirteenth have been reduced.

During simulation it was observed that the voltage level of the bus affects the magnitude of the harmonics. As the voltage of the bus increases and is farther from the average voltage applied at v_1, $3K_iV_i$, the more the harmonics increase. The optimal value of the bus voltage is $3K_iV_i$.

3.1.4 Voltage Control

A voltage controller is designed to regulate the voltage at the DC bus. The RC network impedance is modelled in Eq. (3.19). The impedance gives the current to voltage transfer function. The controller will be designed to output a current reference which will be multiplied by the DC bus voltage to get a power command.

$$\frac{V(s)}{I(s)} = \frac{R}{sRC+1} \tag{3.19}$$

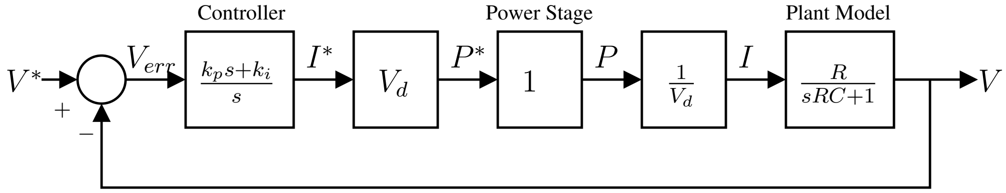


Figure 3.18: Three Phase Voltage Control Loop

The controller design is depicted in Fig. 3.18. A PI controller is selected as the controller and the plant is modelled as the impedance of the RC network (3.19). The power stage is considered to have a gain of one because the bandwidth of the controller will be smaller than the bandwidth of the power controller. The gains V_c and $\frac{1}{V_c}$ are inserted to converter current to power command and power back to current. The controller gains k_i and k_p are found by selecting the bandwidth and phase margin, and solving the two equations (3.20) and (3.21).

$$\left| \frac{k_p s + k_i}{s} \frac{R}{sRC+1} \right|_{s=j2\pi f_{co}} = 1 \tag{3.20}$$

$$\angle \left[\frac{k_p s + k_i}{s} \frac{R}{sRC + 1} \right]_{s=j2\pi f_{cp}} + 180 = \phi_{PM} \tag{3.21}$$

The values used to calculate the controller gains are shown in Table 3.3. The bandwidth is selected as 200Hz, and the phase margin is selected as 70 degrees.

Parameter	Value	Unit
R	1	Ohms
C	200	μF
f_{co}	200	Hz
ϕ_{PM}	70	degrees

Table 3.3: Voltage Control Loop Values

The calculated values of the controller gains are shown in Table 3.4.

Parameter	Value
k_i	241.31
k_p	0.2353

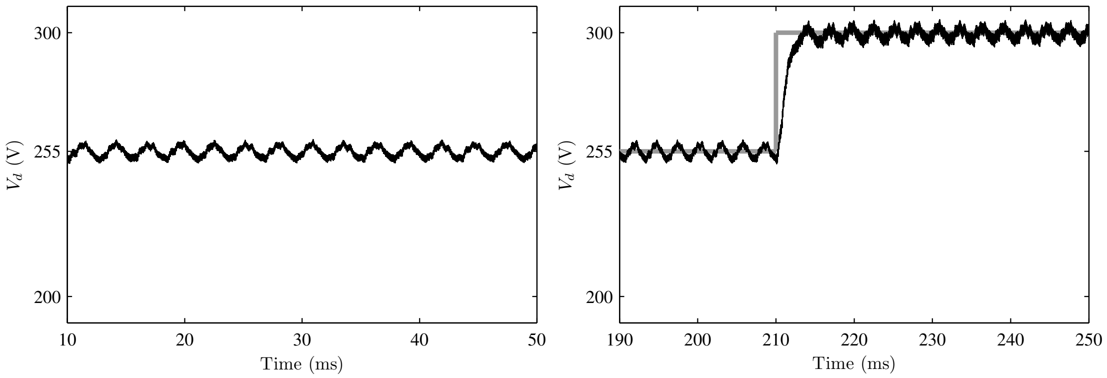
Table 3.4: Voltage Controller Gains

The controller is simulated on the actual system using MATLAB Simulink. The controller is given a voltage of command of 255 volts and the load is 6.5Ω. The steady state voltage is plotted in Fig. 3.19a. The DC bus voltage closely tracks the given voltage command.

At time 210ms the voltage controller is given a step command from 255 volts to 300 volts. The voltage command and actual voltage are plotted in Fig. 3.19b.

In most electric vehicles there is a bidirectional DC to DC converter between the DC bus and the battery. A more accurate simulation of actual conditions would be to change the load resistance at some instant of time and observe the voltage. The objective is to maintain the DC bus voltage constant irrespective of load.

The input currents for a 10kW load, after the LC filter, are plotted in Fig. 3.20. The currents are sinusoidal, in phase with the input voltage, and contain harmonics but the magnitudes are with in IEEE-519 specifications. The current harmonics present in the



(a) Steady State Voltage

(b) Voltage Command

Figure 3.19: Capacitor Voltage

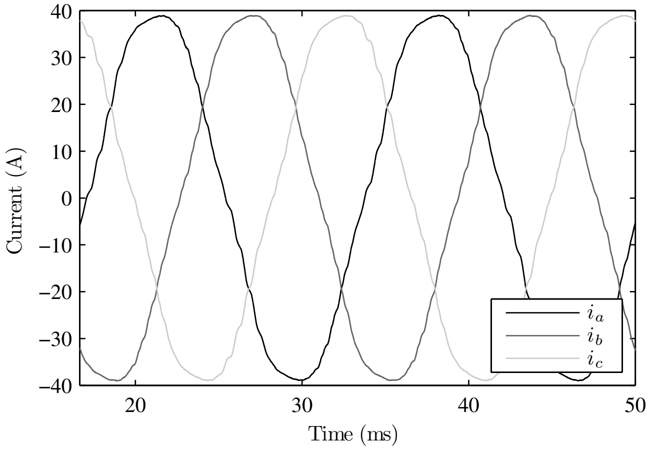


Figure 3.20: Filtered Currents for 10kW Load

inputs currents are shown in Fig. 3.21. The current contains 4% in the 5th harmonic and less than 1% in the 7th harmonic. The other line frequency harmonics are negligible. The total harmonic distortion in the current is 4.17%.

3.1.5 Bidirectional Power Flow

The RC network in Fig. 3.1 is replaced by a DC source in order to simulate bidirectional power flow, as shown in Fig. 3.22.

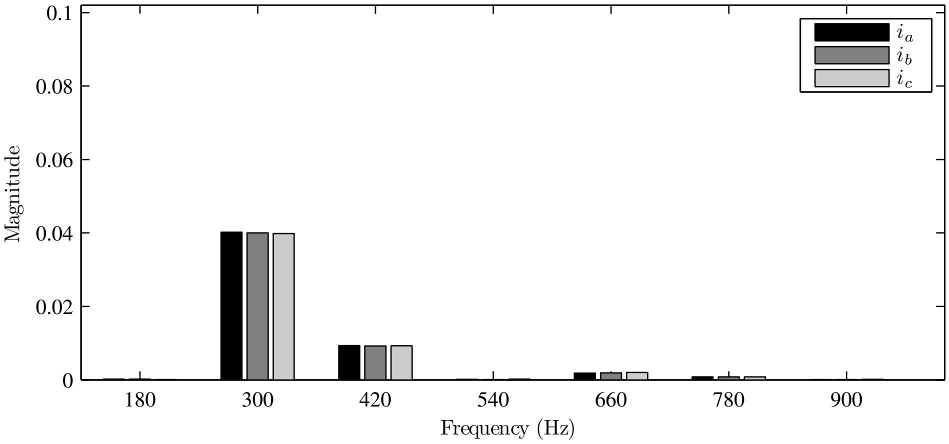


Figure 3.21: Current Spectrum for 10kW Load

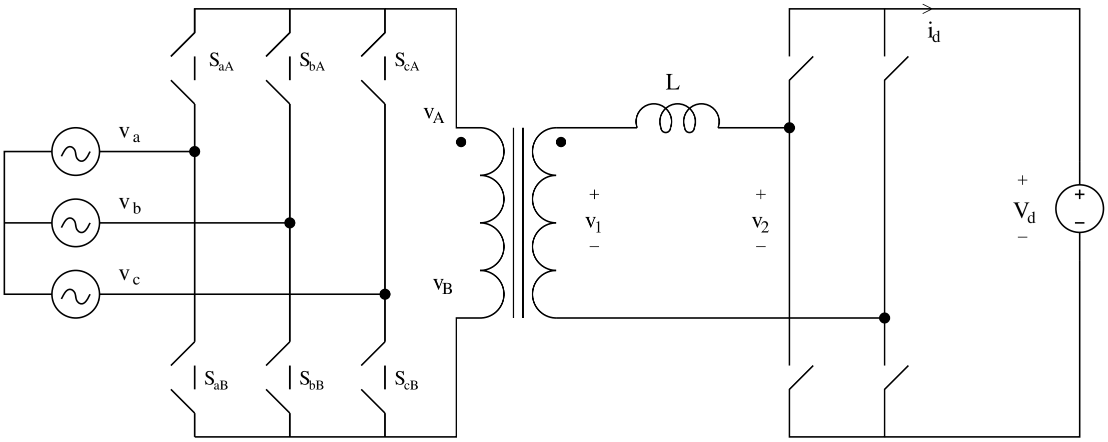


Figure 3.22: Bidirectional System Simulation

The RC network cannot be used when putting power back into the AC sources because the energy will be consumed by the resistor and putting power back into the grid, eventually decaying the capacitor to zero energy.

The DC source represents either a battery pack or a bidirectional DC to DC converter, between the battery pack and the DC bus, maintaining the voltage at a desired set point.

The system in Fig. 3.22 is simulated using only the power control loop with the values in Table 3.2. The DC bus is set to 255 Volts and the power controller is given a

-10kW command. The input voltages and currents after an LC filter are shown in Fig. 3.23.

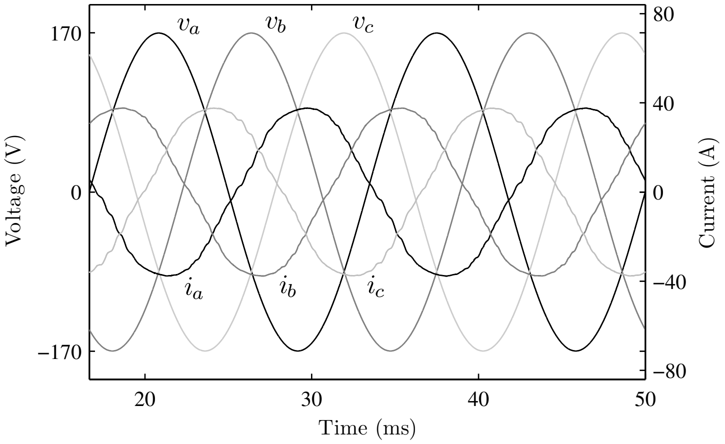


Figure 3.23: Input Voltage and Filtered Current

The input currents are 180 degrees out of phase compared to the respective input voltages. The converter is providing 10kW of power from the DC to the AC sources. The input currents do contain some line frequency harmonics. The harmonic spectrum of the input current is shown for all three phases in Fig. 3.24.

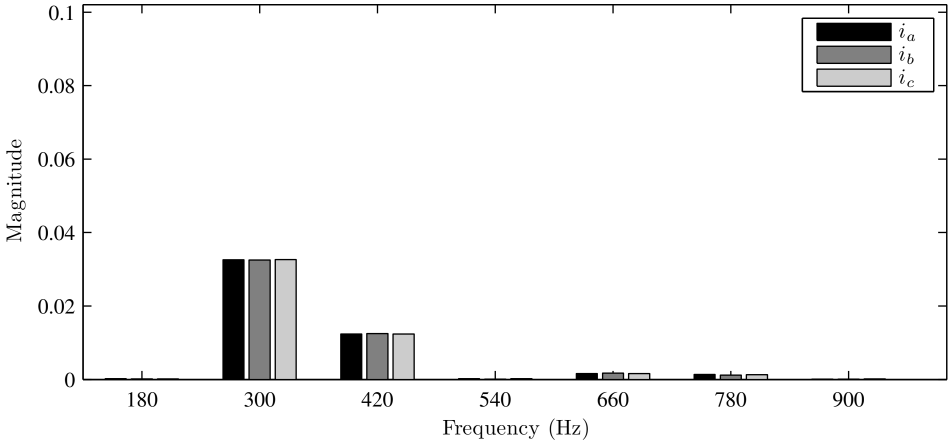


Figure 3.24: Harmonic Content

The results are very similar to the charging 10kW simulation. The input currents

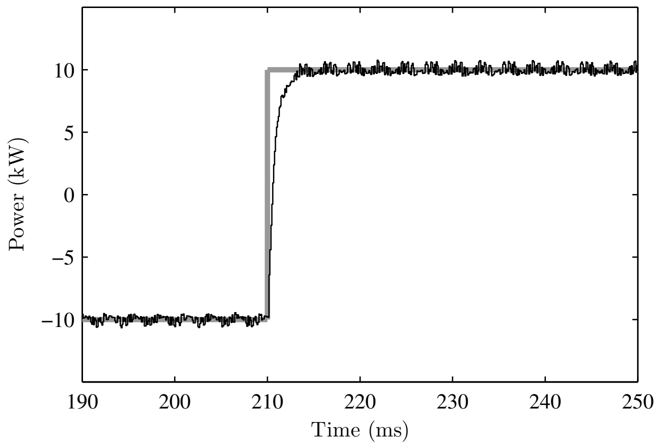


Figure 3.25: Power Step Command from -10kW to 10kW

contain zero third harmonic, 3.2% in the fifth harmonic, and 1.2% in the seventh harmonic. The higher harmonics are below 0.2% and negligible.

Finally a step power command from -10kW to 10kW is given to the controller in order to demonstrate the change from supplying power to the grid from the battery to charging the battery from the grid. The results of the step in power are shown in Fig. 3.25.

The actual power tracks the command and changes from supplying power to the grid to supplying power to the DC source from the grid. The converter can change modes from source to sink by a simple change of the power command.

3.2 Conclusion

In this chapter a control design was discussed and implemented for three phase input. A power controller was designed around the power systems model. It was shown that considering the model contained error with respect to the actual system, a properly designed PI controller is suitable for power control. The input current was analyzed and found to contain high line frequency harmonics and also contained unbalanced current harmonics. The source of the harmonics was discussed and a simple change to the strategy was implemented. After the modification to the strategy, the third and ninth harmonic are zero, the fifth, seventh, eleventh, and thirteenth are now balanced. The input currents still contain some harmonics but they are greatly reduced, below IEEE-519 specifications for harmonic content, and below total IEEE-519 THD requirements. The RC network was removed to demonstrate that the converter with the presented control strategy was capable of bidirectional power flow.

Chapter 4

Single-Phase Modulation using Dual Active Bridge Analysis

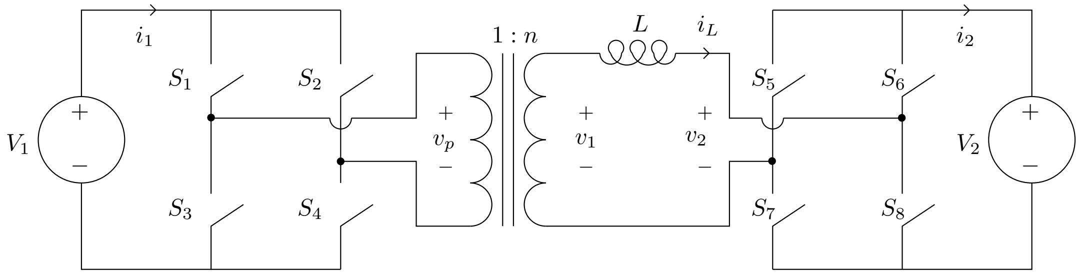


Figure 4.1: DC-DC Dual Active Bridge

This Chapter is put together with the collaboration of my colleagues Gysler and Kaushik. A paper, including this work, will be be submitted to a future conference.

4.1 Analysis

A DC to DC dual active bridge is depicted in Fig. 4.1. The magnetizing inductance of the transformer is neglected and the turns ratio is assumed to be $1 : n$. The leakage inductance of the transformer is lumped to the secondary side, $L = L_p n^2 + L_s$, for analysis. V_1 and V_2 and DC sources. In order to flow power between sources in a Dual Active Bridge, a square wave is applied to v_1 and a phase shifted square wave is applied to v_2.

4.1.1 DC - DC Dual Active Bridge Analysis

During the first $T_s/2$ time period, S_1 and S_4 are closed, V_1 is applied to to v_p and nV_1 is applied to v_1. During the second $T_s/2$ time period, S_2 and S_3 are closed, $-V_1$ is applied to v_p and $-nV_1$ is applied to v_1. The two switching states are applied at a frequency of f_s and the waveform at v_1 is shown in Fig. 4.2. The average voltage of v_2 is synthesized to be equal to v_1 during the time period of $\frac{T_s}{2}$. The width of the pulse at v_2 is assigned the duty ratio in (4.1) and is depicted in the second waveform in Fig. 4.2. The leg with switches S_5 and S_7 are phase shifted from switches S_6 and S_8 in order to generate the duty ratio.

$$d = \frac{nV_1}{V_2} \tag{4.1}$$

In order to transfer power, the voltage waveform at v_2 must be phase shifted from v_1. The phase shift in Fig. 4.1 is shown as Δt in the third waveform. Equation (4.2) imposes a condition on the maximum phase shift allowed to maintain the width of the v_2 pulse within its respective $T_s/2$ time period. Graphically it means that the pulse at v_2 must stay inside the pulse at v_1 as shown in Fig. 4.1. Equations (4.3) and (4.4) set up and impose the condition. Equation (4.3) defines the ratio δ which is confined to -1 to 1 and controls the phase shift from -90 degrees to 90 degrees.

$$\frac{T_s}{4} + \Delta t + \frac{dT_s}{4} < \frac{T_s}{2} \tag{4.2}$$

$$\delta = \frac{\Delta t}{T_s/4} \tag{4.3}$$

$$|\delta| < 1 - d \tag{4.4}$$

In Fig. 4.2, v_L and the current in the inductor i_L are plotted for a phase shift of Δt. Assuming an initial condition of zero current through the inductor, the currents I_1, I_2, and I_3 are calculated in (4.5), (4.6), and (4.7).

$$I_1 = \frac{nV_1}{L}\left(\frac{T_s}{4} + \Delta t - \frac{dT_s}{4}\right) \tag{4.5}$$

$$I_2 = I_1 + \frac{nV_1 - V_2}{L}\frac{dT_s}{2} \tag{4.6}$$

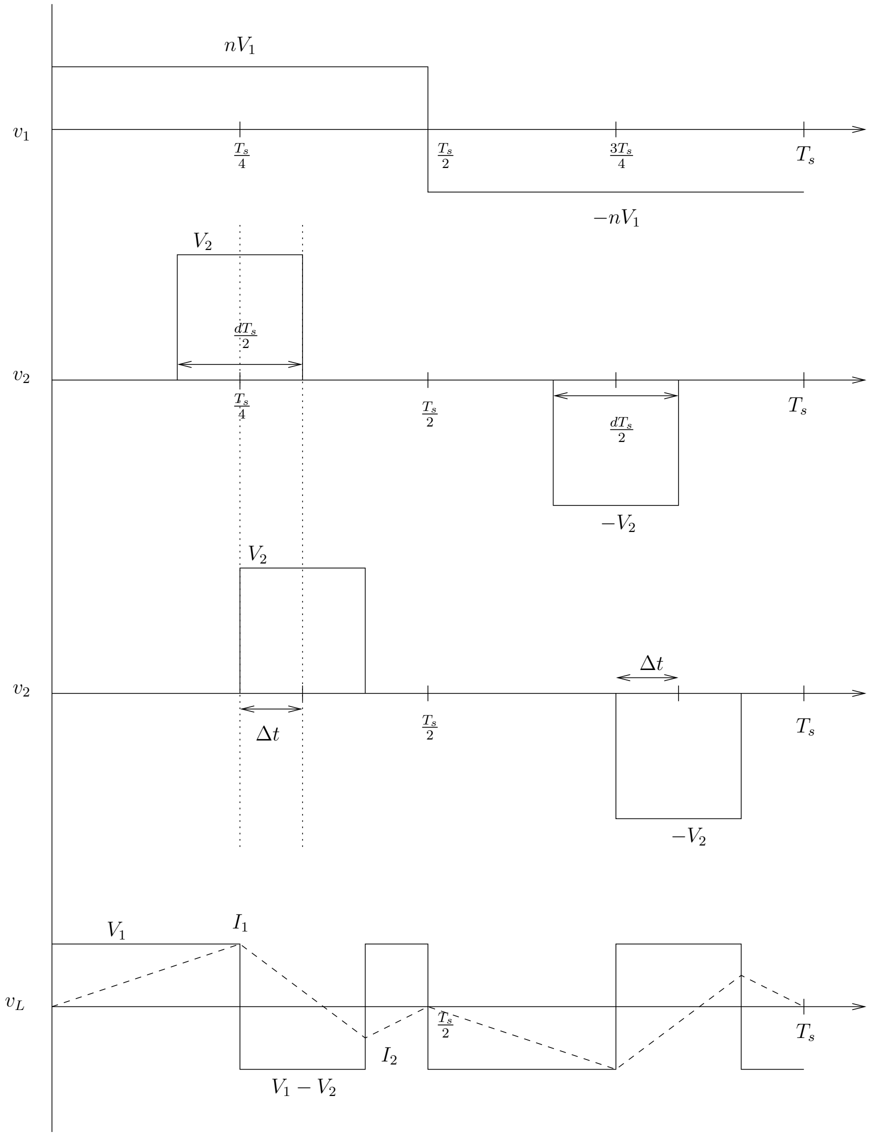


Figure 4.2: DC-DC DAB Plots

$$I_3 = I_2 + \frac{nV_1}{L}\left(\frac{T_s}{4} - \Delta t - \frac{dT_s}{4}\right) \tag{4.7}$$

Solving for the current I_3, using (4.5), (4.6), (4.7), and (4.1), leads to $I_3 = 0$. This results shows that at each instance the primary switches change the current is zero which means the primary bridge is switched at zero current.

The average power transferred in to v_2 during a $\frac{T_s}{2}$ time period is calculated in equation (4.8) and (4.9).

$$P_{avg} = V_2 \frac{I_1 + I_2}{2} \frac{dT_s}{2} \frac{2}{T_s} \tag{4.8}$$

$$P_{avg} = \frac{nV_1V_2\delta}{4Lf_s} d \tag{4.9}$$

Assuming a lossless circuit, the average power during a $\frac{T_s}{2}$ time period delivered from V_1 is set equal to the average power into v_2 in equation (4.10). The average current of i_1 over a $\frac{T_s}{2}$ time period is calculated in (4.11). Given a fixed phase shift of δ, equation (4.11), the average current during a $\frac{T_s}{2}$ time period becomes proportional to the duty cycle d. The average current i_2 is calculated in (4.12) and varies with the duty ratio squared.

$$V_1 \bar{i_1} = \frac{nV_1V_2\delta}{2Lf_s} d \tag{4.10}$$

$$\bar{i_1} = \frac{nV_2\delta}{4Lf_s} d \tag{4.11}$$

$$\bar{i_2} = \frac{nV_2\delta}{4Lf_s} d^2 \tag{4.12}$$

The values in Table 4.1 are used in (4.9), (4.11), and (4.12) and the calculated values are in Table 4.2. The system in Fig. 4.1 is simulated with the parameters in Table 4.1 in order to verify (4.9), (4.11), and (4.12).

The plot of average power is shown in Fig. 4.3a and shows the calculated value in dashed grey, 1kW, coincides with the simulation value, in black, after the initial start up. The average current in the first source is calculated to be 10 Amps and coincides with the simulation results shown in Fig. 4.3b. Finally the average current into source two is calculated to be 5 Amps and agrees with the simulation results shown in black in Fig. 4.3c.

Parameter	**Value**	**Unit**
V_1	100	V
V_2	200	V
L	312.5	μH
f_s	1	kHz
δ	.125	
d	0.5	
n	1	

Table 4.1: Simulation Parameters

Parameter	**Value**	**Unit**
P_{avg}	1	kW
$\bar{i}_1$	10	A
$\bar{i}_2$	5	A

Table 4.2: Calculated Values

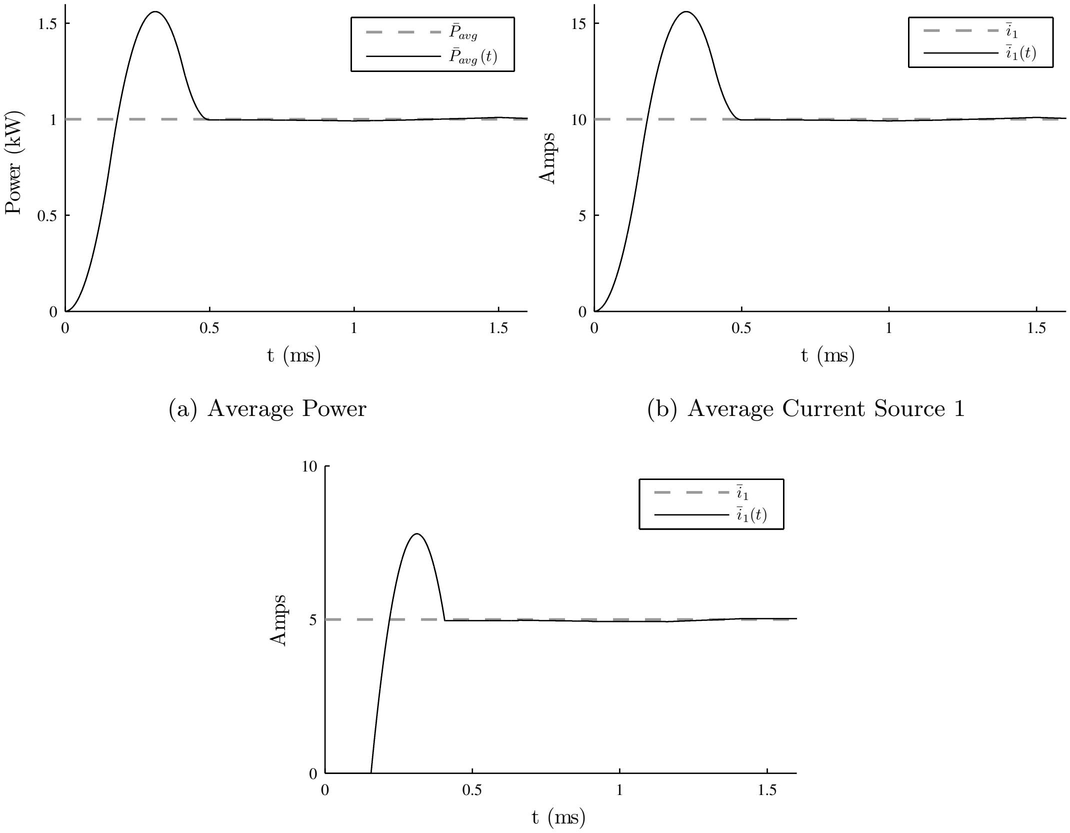
The voltages $v_1(t)$, $v_2(t)$, and current $i_L(t)$ are plotted for 4ms in Fig. 4.4. At every instance the voltage $v_1(t)$ changes the current i_L is zero. This simulation result confirms the theory and proves the primary switches are soft switched (ZCS).

In conclusion the theory resulting in equations (4.9), (4.11), and (4.12) is verified by simulation results. In the next section (4.11) will be exploited, by letting the DC source V_1 become sinusoidal and utilize the duty ratio to control the average current from the source.

4.1.2 AC - DC Dual Active Bridge Analysis

In this section the first source has been changed from a DC source to the AC source as shown in Fig. 4.5. The transformer magnetizing inductance is neglected and the leakage inductance is lumped to the secondary. The switches on the primary side now must be four quadrant because of the AC source voltage swings both positive and negative.

The AC source is defined in (4.13) with a magnitude of V_i and frequency of ω_i. The duty ratio is defined to vary with the input voltage shown in (4.14).



(a) Average Power

(b) Average Current Source 1

(c) Average Current Source 2

Figure 4.3: Steady State Simulation Results

$$v_s(t) = V_i \sin \omega_i t \tag{4.13}$$

$$d(t) = \frac{n\left|v_s(t)\right|}{V_d} = \frac{nV_i\left|\sin \omega_i t\right|}{V_d} \tag{4.14}$$

The duty ratio must be between zero and one so the absolute value of sine is taken to respect that condition. Letting the duty ratio in (4.11) vary by (4.14) obtains the result for the average current shown in (4.15).

$$\bar{i}_s(t) = \frac{nV_d\delta}{4Lf_s}d(t) = \frac{n^2V_s\delta}{4Lf_s}\left|\sin \omega_i t\right| \tag{4.15}$$

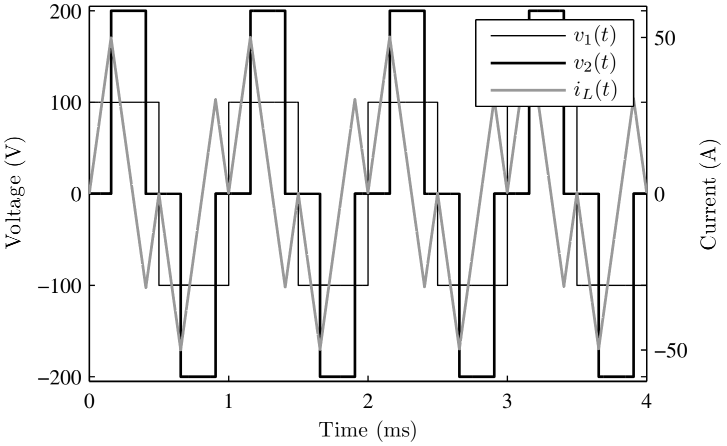


Figure 4.4: Zero Current Switching

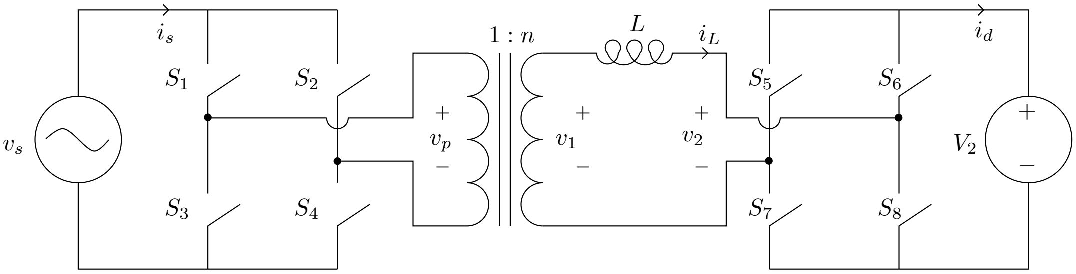


Figure 4.5: AC-DC Dual Active Bridge

The average current from the AC source, given a fixed phase shift δ, varies by the absolute value the input voltage which has the form of the absolute value of sine. The goal is to have the average input current to be sinusoidal and not absolute value of sine. The peculiarity here is to notice that at each $f_i/2$ cycle of the input voltage, the square wave applied at $v_1(t)$ changes phase by 180 degrees. The phase shift δ at v_2 is changed to offset the phase shift of v_1. The phase shift, δ, is a ratio from -1 to 1 and is changed 180 degrees by multiplying delta by negative one.

$$\bar{i}_s(t) = \begin{cases} \dfrac{n^2 V_i \delta}{4Lf_s} \left|\sin \omega_i t\right| & \text{if } 0 < t \le \frac{T_i}{2} \\[2em] -\dfrac{n^2 V_i \delta}{4Lf_s} \left|\sin \omega_i t\right| & \text{if } \frac{T_i}{2} < t \le T_i \end{cases} \tag{4.16}$$

The average current from the input source during the first half and second half period of the input voltage is shown in Eq. (4.16). By inspection (4.16) can be simplified to (4.17).

$$\bar{i}_s(t) = \frac{n^2 V_i \delta}{4 L f_s} \sin(\omega_i t) \tag{4.17}$$

The average current, $\bar{i}_s(t)$, now varies sinusoidally with the input voltage given a fixed phase shift δ. Also the input current is power factor corrected and in phase with the input voltage.

The peak average current $\hat{\bar{i}}_s$ is multiplied by the peak input voltage and one half to obtain the average power, (4.18), from the AC source.

$$\begin{aligned} P_{avg} &= \frac{n^2 V_i \delta}{4 L f_s} V_i \frac{1}{2} \\ &= \frac{n^2 V_i^2}{8 L f_s} \delta \end{aligned} \tag{4.18}$$

A simulation is carried out to verify the analytical and theoretical results. The simulations parameters used are shown in Table 4.3. The results of the input voltage and input current are shown in Fig. 4.6a. The input voltage is shown in black with a frequency of 60Hz and magnitude of 100 Volts. The average input current is shown in grey and is in phase with respect to the input voltage, sinusoidal, and has a magnitude of 100 Amps. The average current and power is compared between the theoretical equations and simulation results in Table 4.4. The average current and average power are in agreement providing verification of the analysis.

The switching current $i_s(t)$ is depicted in Fig. 4.6b. Besides the 10kHz switching component, a vivid 60Hz outline can be observed in Fig. 4.6b. A complete Fourier analysis is carried out on the switching currents in order to investigate the harmonic content in the input current.

Besides the 10kHz switching component, a vivid 60Hz outline can be observed in the waveform. A complete Fourier analysis is carried out on the switching currents in order to investigate the harmonic content in the input current. The results of the fourier analysis normalized to the magnitude at 60Hz are shown in Fig. 4.7a and 4.7b.

Parameter	**Value**	**Unit**
V_i	100	V
V_d	400	V
L	50	μH
f_s	10	kHz
δ	0.5	
d	0.5	
n	2	

Table 4.3: Simulation Parameters

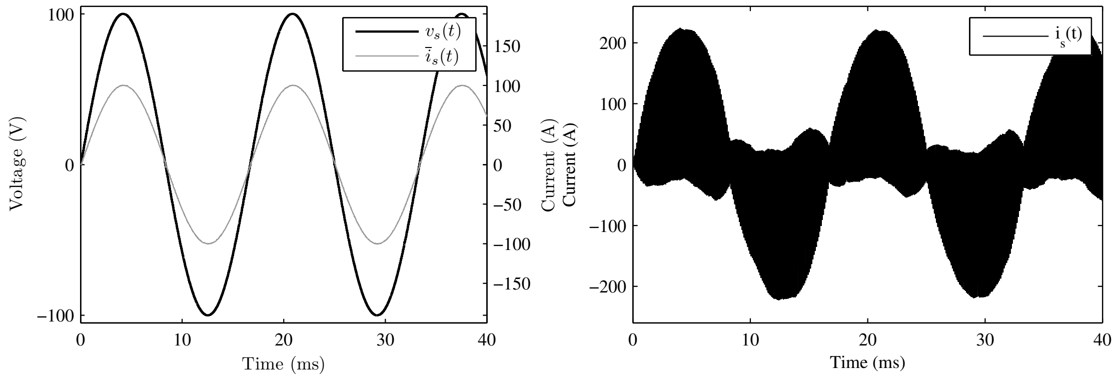
Parameter	**Calculation**	**Simulation**	**Unit**
$\hat{\hat{i}}_s$	100	100	Amps
P_{avg}	5000	5000	Watts

Table 4.4: Simulation Results

Fig. 4.7a shows the complete harmonic profile with the fundamental appearing at 60Hz and the switching components appearing at 10kHz. The switching component at 10kHz can be removed from the input current by a properly designed LC filter placed at the input. The filter should be designed with a cut off frequency much lower than 10kHz but not low enough to introduce significant phase delay at 60Hz. In Fig. 4.7b the first 13 harmonics of 60Hz are shown. In this modulation scheme there are negligible line frequency harmonics from the 3rd to 13th in the figure and higher until you reach the switching frequency components. One metric of the modulation scheme is the primary switches peak current rating compared to the average current from the source. This modulations scheme is similar to the peak current control method in PFC. The switch rating in this modulation scheme has to be double of the peak average current from the source. The primary switches peak current rating is derived in (4.19).

$$\hat{i}_{p,switch} = 2 * \hat{\hat{i}}_s = \frac{n^2 V_i}{2 L f_s}\delta \tag{4.19}$$

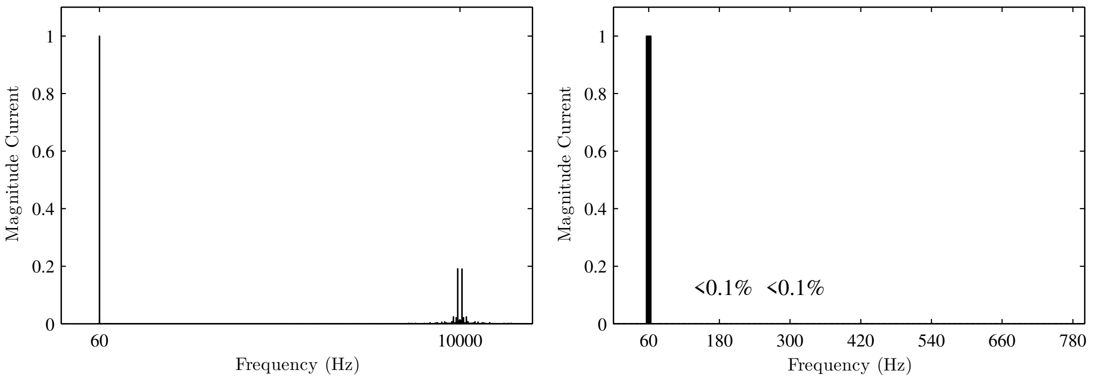
In the DC-DC Dual Active Bridge case it was shown that the modulation scheme has the advantage of zero current switching on the primary side of the circuit. In this analysis the input source was changed to an AC source. In order to draw sinusoidal



(a) Voltage and Average Current

(b) Switch Current

Figure 4.6: Simulation Results



(a) 10kHz Spectrum

(b) Line Frequency Spectrum

Figure 4.7: Frequency Spectrum

current the duty ratio is set to vary sinusoidally with the input voltage. In Fig. 4.8 the voltages $v_1(t)$, $v_2(t)$, and the current $i_L(t)$ are plotted at some instant of time. The inductor current i_L is plotted in grey. At each instant when the primary switches, meaning the voltage $v_1(t)$ changes from positive to negative, the current i_L is zero. The simulation results confirm zero current switching for the presented modulation scheme.

The average current into the DC source will be investigated. The average current for the AC-DC dual active bridge is defined in (4.20) by substituting the duty ratio with a time varying duty ratio. The average current into the DC source is derived in

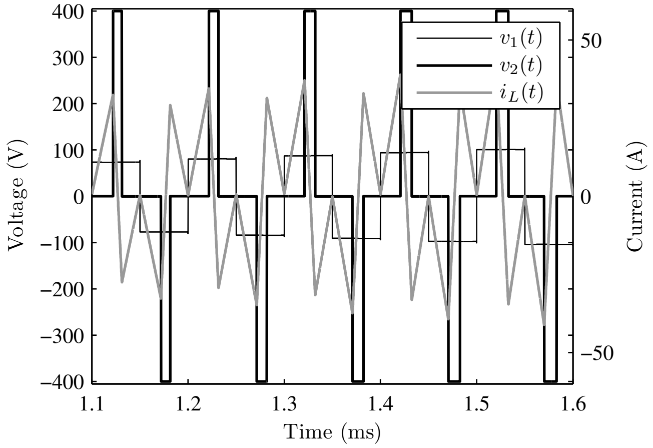


Figure 4.8: Zero Current Switching

(4.21) where $\hat{d} = \frac{nV_i}{V_d}$. The average current is of the form of a sine squared. This average current is similar to the current into the RC circuit from the previous chapter. If the DC source is replaced with an RC circuit the analysis for designing a voltage control loop would be similar to a single phase power factor correction circuit. The average current into the DC source results from simulation are shown in Fig. 4.9.

$$\begin{aligned} \bar{i}_d(t) &= \frac{nV_d\delta}{4Lf_s} d(t)^2 && (4.20) \\ &= \frac{n^2V_i\hat{d}}{4Lf_s}\delta(\sin^2 \omega_i t) && (4.21) \end{aligned}$$

By observation, the average current from simulation has the form of sine squared and a magnitude of 25 Amps. Using the simulations values from Table 4.3 in Eq. (4.21) returns a current magnitude of 25 Amps which coincides with the simulation result.

Another important metric in dual active bridge type of converters is the utilization factor of the transformer. This metric is a ratio of actual power transferred through the transformer to the load by how much reactive power flows back and fourth through the transformer. Before this calculation can be done some important quantities must be found first. The first two quantities that need to be defined are the rms voltage applied to the primary of the transformer v_p and the rms current into the primary i_p. Those quantities are defined and calculated in (4.22) and (4.23).

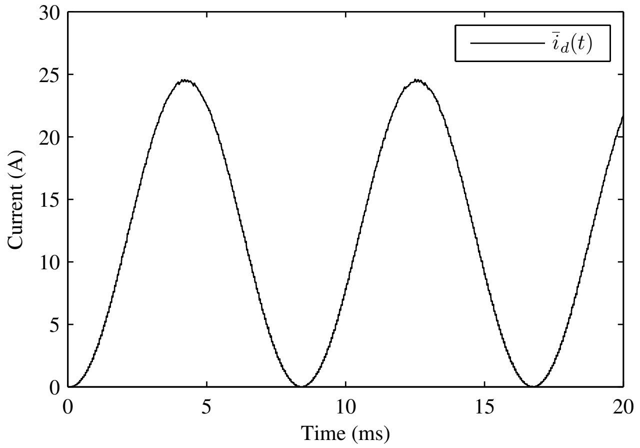


Figure 4.9: Average DC Current

$$\begin{aligned} v_{p,rms} &= \sqrt{60\int_0^{1/60} v_p^2(t)dt} \\ &= \frac{V_i}{\sqrt{2}} \end{aligned} \tag{4.22}$$

$$\begin{aligned} i_{p,rms} &= \sqrt{60\int_0^{1/60} i_p^2(t)dt} \\ &= \frac{n^2V_i\sqrt{6+18\delta^2-\frac{32nV_i}{\pi V_d}+\frac{9n^2V_i^2}{2V_d^2}}}{24\mathrm{Lf}_s} \end{aligned} \tag{4.23}$$

The equations are evaluated using the values in Table 4.3 and compared against simulation results in Table 4.5. The rms primary voltage and rms primary current are in close agreement.

The next two important quantities to be defined are the rms voltage applied to the secondary v_2 and the rms current through the secondary winding. Those quantities are defined and calculated in (4.24) and (4.25).

$$\begin{aligned} v_{2,rms} &= \sqrt{60\int_0^{1/60} v_2^2(t)dt} \\ &= \sqrt{\frac{2}{\pi}}\sqrt{\hat{d}}\mathrm{V}_d \end{aligned} \tag{4.24}$$

$$\begin{aligned} i_{L,rms} &= \sqrt{60\int_0^{1/60} i_L^2(t)dt} \\ &= \frac{nV_i\sqrt{6+18\delta^2-\frac{32nV_i}{\pi V_d}+\frac{9n^2V_i^2}{2V_d^2}}}{24\mathrm{Lf}_s} \end{aligned} \tag{4.25}$$

These rms equations are checked for validity by using the values in Table 4.3 and comparing those results with simulations values. The rms secondary voltage and rms secondary current results are shown in Table 4.5 and again are in close agreement. These four quantities will be used in determining the reactive power flow in the transformer.

The reactive power flow through the transformer is defined in (4.26). The reactive power flow is calculated using (4.26) and compared against the simulation result in Table 4.5. The calculated and simulated values are in close agreement and one thing to note is the value of the reactive power. The average power flow is 5kW while the reactive power flow is 1.8kW. Ideally, reactive power would be zero because that power will flow back and forth in the transformer losing power through winding resistances, core resistances, etc, but any time, the duty ratio is not one, there will be some reactive power flow. The reactive power flow is a excellent metric and should be anaylzed closely when designing the converter.

$$P_T = \frac{1}{2}(v_{p,rms}i_{p,rms} - v_{2,rms}i_{L,rms}) \tag{4.26}$$

$$P_T = \frac{d\left(2\sqrt{d}+d\sqrt{\pi}\right)V_d^2\sqrt{12\left(\pi+3\pi\delta^2\right)-\frac{64nV_i}{V_d}+\frac{9n^2\pi V_i^2}{V_d^2}}}{96L\pi f_s} \tag{4.27}$$

The reactive power is plotted in Fig. 4.10 for various operating points d, and phase shift δ from 0 to 1. The operating point of the simulation, $d = 0.5$, is shown in bold. One

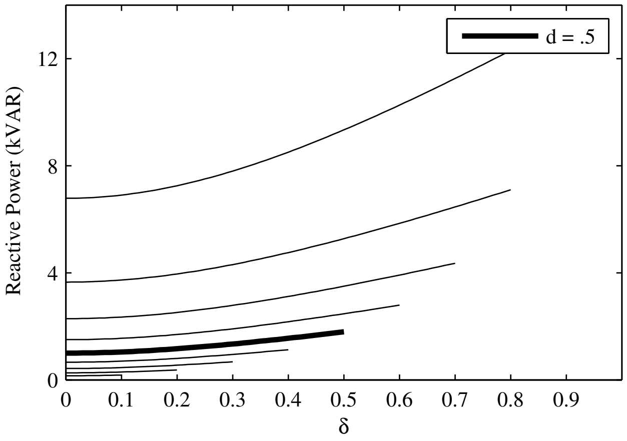


Figure 4.10: Reactive Power Flow

observation is that when δ is zero there is no actual power flow but about 1kW of reactive power flows. Another keen observation is that as duty ratio decreases more and more reactive power flows. This observation is intuitive because as the duty ratio decreases, a lesser amount of power is allowed into the DC source and more power circulates in the transformer. This keen observation should be kept in mind when designing a converter with this modulation strategy.

The next metric that lends some insight into the design of this converter is the utilization factor of the transformer. This factor is a ratio of reactive power to average power flow and is defined in (4.28).

$$U_f = \frac{P_{avg}}{P_T} \tag{4.28}$$

Ideally, this factor would be infinity, meaning that all the power is being delivered to the load and none is flowing back and fourth between the transformer. The utilization factor is plotted for various operating points of d, from 0.1 to 0.9 in steps of 0.1, and over all δ, from 0 to 1, in Fig. 4.11. The operating point of the simulation is shown in bold and shows that as δ increases the utilization factor increases and thus the reactive power flow increases, irrespective of the operation point d. The utilization factor is another useful metric in the design a converter with this modulation scheme.

Lastly, the rms quantity of $i_d(t)$ is defined in (4.29). This quantity is very useful

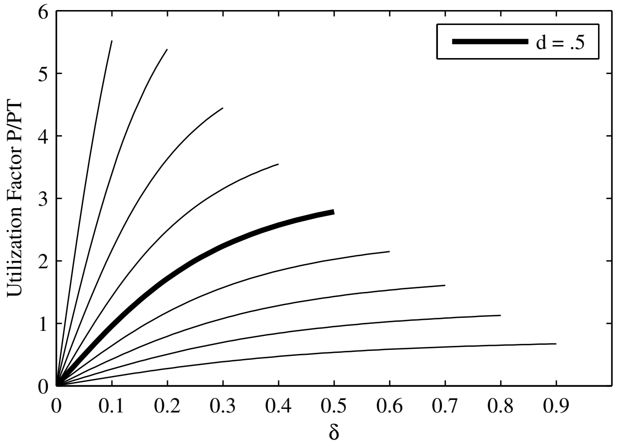


Figure 4.11: Utilization Factor

in determining system cost due to capacitor sizing and also it determines the capacitor current ripple rating.

$$\begin{aligned} i_{d,rms} &= \sqrt{60\int_0^{1/60} i_d^2(t)dt} \\ &= \frac{nT_sV_i\sqrt{d\left(80+240\delta^2-\frac{45n\pi V_i}{V_d}+\frac{64n^2V_i^2}{V_d^2}\right)}}{24L\sqrt{5\pi}} \end{aligned} \tag{4.29}$$

The rms ripple current is plotted for various operating points, d from 0.1 to 0.9 in steps of 0.1, and all δ, from 0 to 1, in Fig. 4.12. The operating point of the simulation is shown in bold in the figure. As delta increases at each operating point the rms ripple current increases which is to be expected but over the various operating points d there are is a local maximum and local minimum.

Starting with $d = 0.1$ and increasing d the rms ripple current increases until about $d = 0.45$ where is begins to come back down. Considering just rms ripple current and minimizing it, the obvious selection would be $d = 0$ but we know at that operating point there would be zero average power flow and large amounts of reactive power flow. This metric should be given careful attention when designing a converter with the proposed modulation scheme because it is directly related to the capacitor sizing and thus cost of the system.

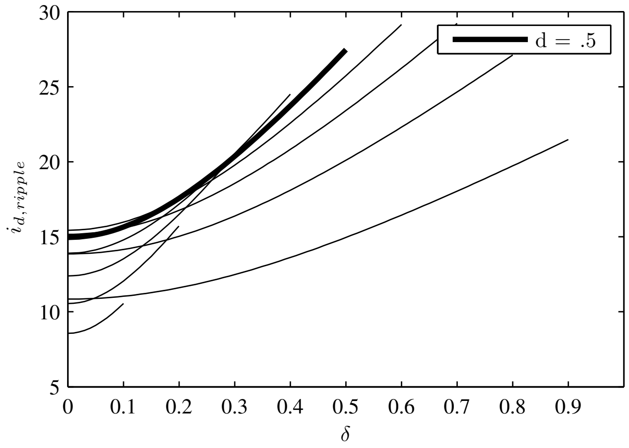


Figure 4.12: Capacitor RMS Ripple Current

Parameter	**Calculated**	**Simulation**	**Unit**
$v_{p,rms}$	70.71	70.71	V
$v_{1,rms}$	141.42	141.4	V
$v_{2,rms}$	225.67	225.9	V
$i_{p,rms}$	85.2	85.92	A
$i_{L,rms}$	42.6	42.96	A
$i_{dc,rms}$	27.47	27.32	A
$P_{dc,rms}$	10986	10950	W
P_T	1794	1814	W

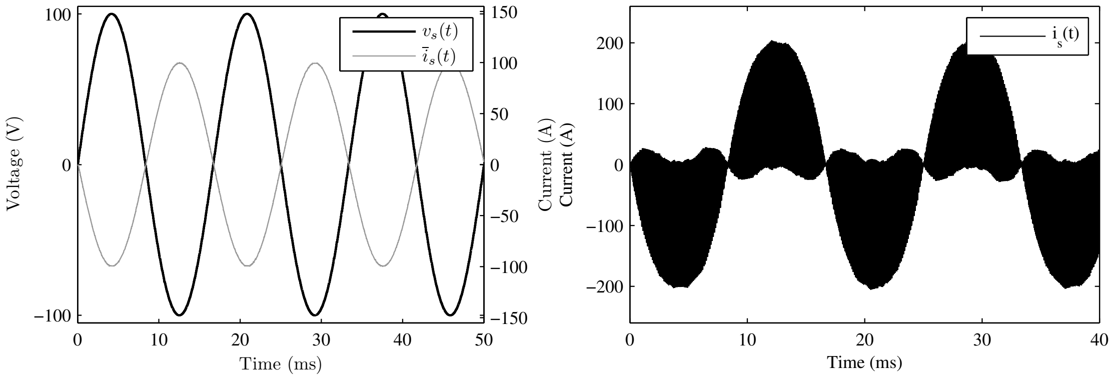
Table 4.5: Simulation Results

4.2 Bidirectional Power Flow

The converter in Fig. 4.1 is simulated with the values in Table 4.3 except δ is set to -0.5. The theory states that the current should be 180 degrees out of phase and pull power from the DC source and supply it to the AC source.

The average current is plotted along with the input voltage in Fig. 4.13a. The input current is sinusoidal and 180 degrees out of phase with the input voltage therefore power is flowing into the AC source. There are no harmonics present in the average current.

The switching current is plotted Fig. 4.13b. It contains the switching harmonic at



(a) Average Current

(b) Switch Current

Figure 4.13: Simulation Currents

10kHz and also the outline of the 60Hz component is observable in the waveform.

The votlages v_1 and v_2 and the current i_L are plotted in Fig. 4.14.

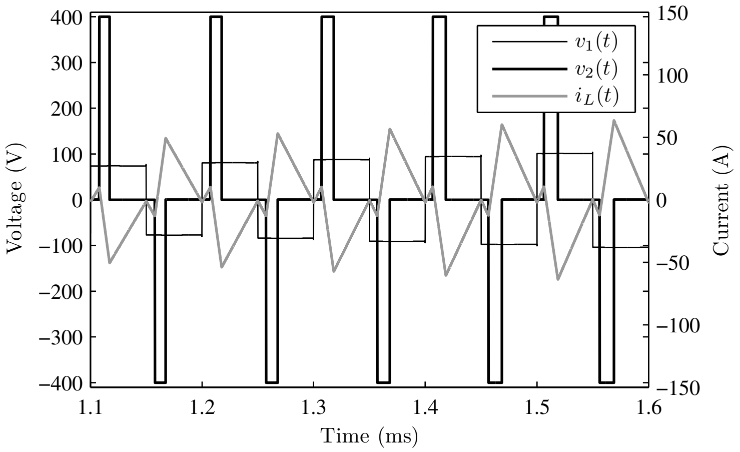


Figure 4.14: Modulation Cycle

The voltage waveform v_2 is shifted to the left with respect to v_1. The current is zero at each switching of v_1 proving that the input converter is soft switched irrespective of the phase shift δ.

The modulation scheme on the proposed converter is shown to work with bidirectional power flow.

4.3 Conclusion

The first section analyzed a DC-DC dual active bridge. A new modulation scheme was proposed. The equations for average current and average power were derived. In the next section the duty ratio was exploited from the DC-DC case. The modulation scheme was adapted to allow the duty ratio to vary sinusoidally with the input voltage. This resulted in sinusoidal current being drawn from the source. Important quantities like average current and power were derived for the AC-DC dual active bridge. The input current was analyzed for harmonics and contained zero line frequency harmonics. The average current into the DC source was found to vary by sine squared similarly to a single phase power factor correction circuit. The reactive power was analyzed, discussed, and noted as a considerable design factor. The transformer utilization factor was also analyzed and mentioned as another mindful design factor. The rms capacitor current was derived and mentioned as a factor in cost and size. Simulation results were given for positive and negative phase shifts confirming bidirectional power flow. Lastly simulation results were given and matched theoretical quantities.

Chapter 5

Three-Phase Modulation using Dual Active Bridge Analysis

This section presents an analysis of novel modulation scheme for a three-phase AC to DC dual active bridge converter. The converter is shown in Fig. 5.1. The input converter (A) is a three-phase to single-phase matrix converter and is comprised of the six switches S_{a1}, S_{b1}, S_{c1}, S_{a2}, S_{b2}, and S_{c2}. The six switches are four quadrant switches and each switch is realized with two emitter tied IGBTS. The input voltages supplied to the matrix converter are defined in (5.1). The output converter (B) is a H-bridge and consists of two quadrant switches, S_1, S_2, S_3, S_4. The switches in the H-bridge are

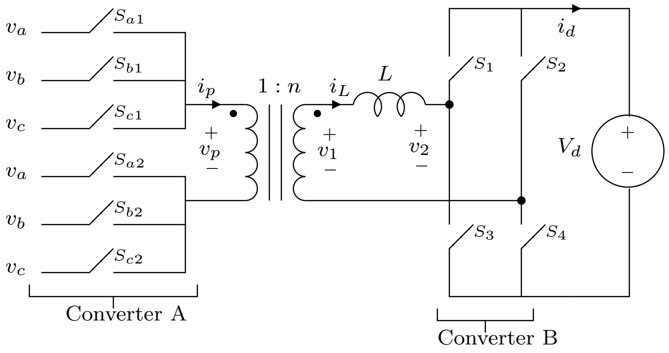


Figure 5.1: AC-DC Dual Active Bridge

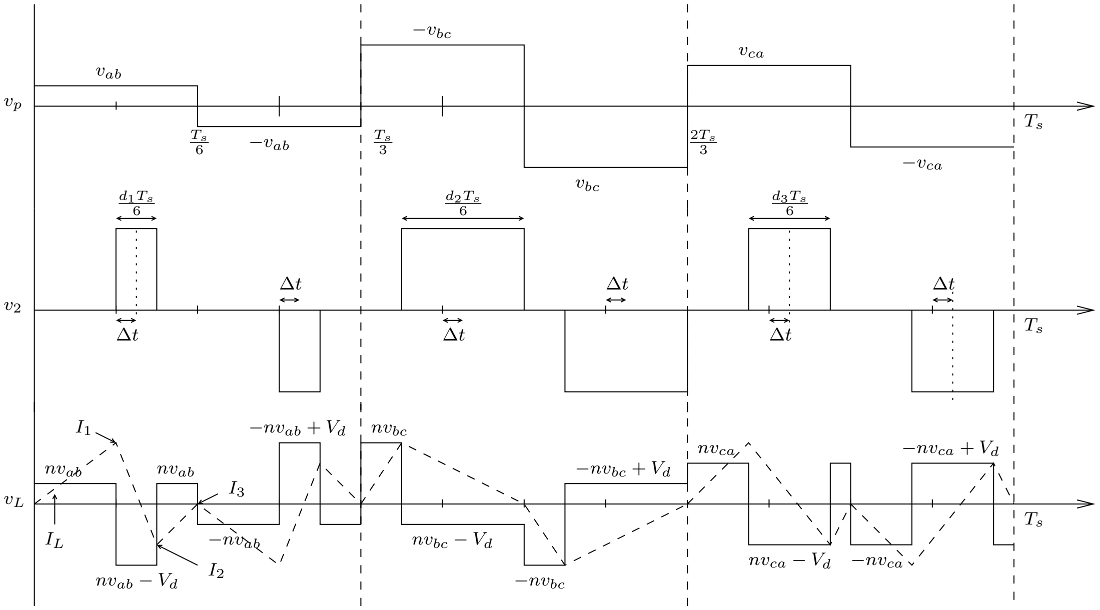


Figure 5.2: Modulation Cycle

realized with single IGBT's.

$$\begin{aligned} v_a(t) &= V_i \cos(\omega_i t) \\ v_b(t) &= V_i \cos\left(\omega_i t - \frac{2\pi}{3}\right) \\ v_c(t) &= V_i \cos\left(\omega_i t - \frac{4\pi}{3}\right) \end{aligned} \tag{5.1}$$

In this analysis the magnetizing inductance of the transformer is assumed to be very large therefore its effect is neglected. The entire leakage inductance of the transformer is lumped into a single inductance $L = L_p n^2 + L_s$ and is refereed to the secondary. The primary winding resistance, the secondary winding resistance, and the core loss resistance is ignored. The H-bridge is connected to a DC voltage source V_d. The transformer is modelled as an ideal transformer with turns ratio $1 : n$ followed by the lumped leakage inductance L in the secondary, shown in Fig. 5.1.

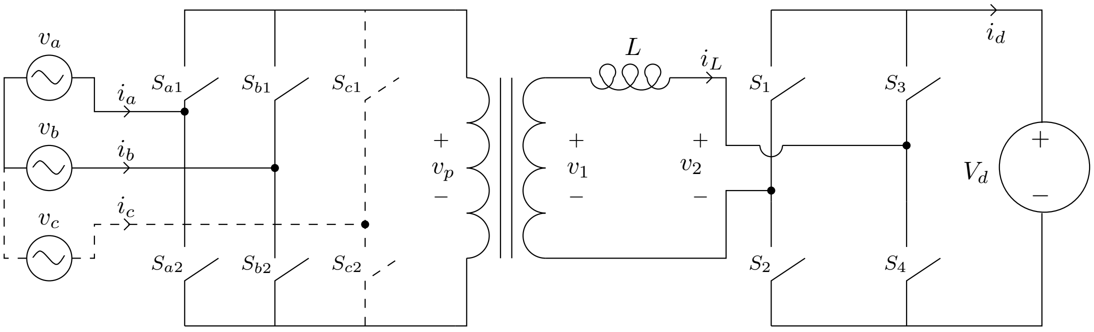


Figure 5.3: First Third Cycle

5.1 Analysis

One complete cycle of the proposed modulation scheme is shown in Fig. 5.2. Each of the three cycles will be discussed separately.

5.1.1 First Third Cycle Analysis

In the first third cycle only the voltage sources v_a and v_b are used. Source v_c is unused during the first third cycle, as shown in Fig. 5.3. Note that when v_c is not used, it looks similar to single phase AC to DC dual active bridge. During the first sixth of the cycle, switches S_{a1} and S_{b2} are closed, applying the voltage v_{ab} to the primary. During the second sixth of the cycle, S_{b1} and S_{a2} are closed, applying $-v_{ab}$ to the primary. The duty ratio d_1 is defined as (5.2), and is the ratio of time v_d is applied to v_2 by converter B.

The duty ratio is chosen such that during the first and second sixth of the cycle, the average voltage at v_1 is equal to the average voltage at v_2. This can be seen pictorially as v_p and v_2 in Fig. 5.2. The waveform at v_2 has a magnitude of V_d and a duration of $\frac{d_1 T_s}{6}$.

$$d_1(t) = \frac{|v_{ab}(t)|n}{V_d} \tag{5.2}$$

The phase shift of the waveform at v_2 with respect to v_1 is depicted as Δt in Fig. 5.2. Δt is restricted so that the pulse of V_d at v_2 lies within its respective $\frac{T_s}{6}$ time period. That is to say in the first sixth, the pulse of V_d must lie within the time period $\left[0 \quad \frac{T_s}{6}\right]$.

The inequality (5.3) is defined to impose the condition.

The phase shift δ is defined as a ratio from -1 to 1, which is equivalent to -90 to 90 degrees, shown in (5.4). Using the definition (5.4), (5.3) can be simplified to (5.5).

$$\frac{T_s}{12} + \Delta t + \frac{dT_s}{12} \leq \frac{T_s}{6} \tag{5.3}$$

$$\delta = \frac{\Delta t}{T_s/12} \tag{5.4}$$

$$|\delta| \leq 1 - d \tag{5.5}$$

The voltage across the inductor for the first third of the cycle is given in (5.6). It is assumed that at the beginning of the cycle, the inductor current is initially zero. The current rises to I_1, and decreases to I_2 at the two consecutive switchings of converter B. The current reaches I_3 at the first switching of converter A. The choice of d_1 guarantees that I_3 is equal to zero. The currents I_1, I_2, and I_3 are given in (5.7), (5.8), and (5.9). To prove that I_3 is zero, the equations (5.2), (5.7), (5.8), and (5.9) are used to solve for I_3. I_3 does indeed simplify to zero and implies that converter A is switched at zero current (ZCS).

In the second sixth of T_s, $-v_{ab}$ is applied to the primary. Converter B applies a $-V_d$ pulse of duty ratio d_1, phase shifted by the same amount δ, to v_2. Again the average voltage applied across the inductor is zero therefore the current at $\frac{T_s}{3}$ is zero, proving that converter A is switched ZCS during the first third of the cycle.

$$v_L(t) = nv_p(t) - v_2(t) = L\frac{di_L}{dt} \tag{5.6}$$

$$I_1 = \frac{nv_{ab}}{L}\left(\frac{T_s}{12} + \Delta t - \frac{d_1 T_s}{12}\right) \tag{5.7}$$

$$I_2 = I_1 + \frac{nv_{ab} - V_d}{L}\frac{d_1 T_s}{6} \tag{5.8}$$

$$I_3 = I_2 + \frac{nv_{ab}}{L}\left(\frac{T_s}{12} - \Delta t - \frac{d_1 T_s}{12}\right) \tag{5.9}$$

The average current i_{ab} is calculated for the first third of the cycle. During the first third of T_s, the current i_{ab} is same as i_L. The average current i_{ab} in the first sixth of T_s is the same as the average current in the second sixth of T_s because of the symmetry of the current waveform and voltage waveform. The average current is calculated for the first sixth of T_s in (5.10).

$$\begin{aligned} \langle \bar{i}_{ab} \rangle_{[0-\frac{T_s}{3}]} &= \frac{6}{T_s}\int_0^{\frac{T_s}{6}} i_L(t) \qquad (5.10) \\ &= \frac{n\delta V_d}{4Lf_s} d_1(t) \qquad (5.11) \end{aligned}$$

The average current $\bar{i}_{ab}$ for the first third of the cycle is given in (5.11). Given a constant phase shift δ, the average current $\bar{i}_{ab}$ is proportional to the duty ratio $d_1(t)$. The duty ratio $d_1(t)$ has the form the of the absolute value of $v_{ab}(t)$.

The virtual current i_{ab} only flows when either switches S_{a1} and S_{b2} are closed or when switches S_{b1} and S_{a2} are closed. This sequence only happens during the first third of T_s. The current i_{ab} is zero during the second third and third third of T_s because the sources v_a and v_b are not used. This average current $\bar{i}_{ab}$ over a full period of T_s results in being equal to $\frac{1}{3}$ of the average computed in (5.11). The average current $\bar{i}_{ab}$, over a full period of T_s, is given in (5.12). The average current $\bar{i}_{ab}$ has the form of the absolute value of a sinusoid. A maneuver is employed to make the current vary sinusoidally. For the first half cycle of ω_i, δ is fixed as a positive value and during the second half cycle of ω_i, δ is fixed as the negative of δ as shown in (5.13).

$$\bar{i}_{ab}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s}\left|\cos\left(\omega_i t + \frac{\pi}{6}\right)\right| \tag{5.12}$$

$$\bar{i}_{ab}(t) = \begin{cases} \dfrac{\sqrt{3}n^2 V_i\delta}{12Lf_s}\left|\cos\omega_i t + \frac{\pi}{6}\right| & \text{if } 0 < t \leq \frac{T_i}{2} \\ \\ -\dfrac{\sqrt{3}n^2 V_i\delta}{12Lf_s}\left|\cos\omega_i t + \frac{\pi}{6}\right| & \text{if } \frac{T_i}{2} < t \leq T_i \end{cases} \tag{5.13}$$

$$\bar{i}_{ab}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s}\cos\left(\omega_i t + \frac{\pi}{6}\right) \tag{5.14}$$

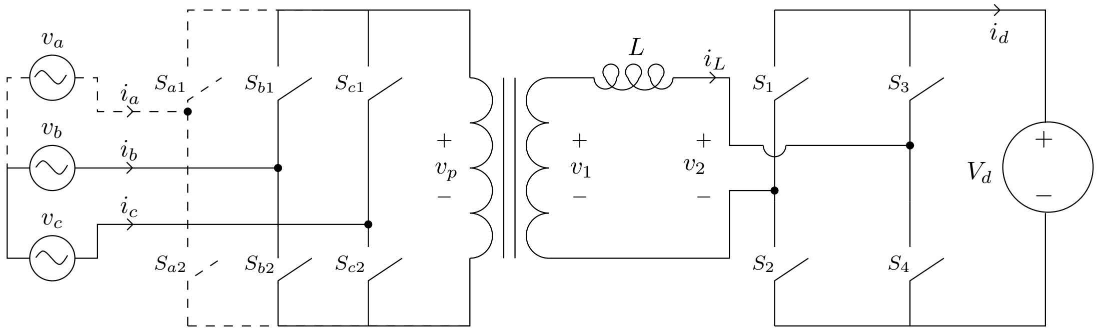


Figure 5.4: Second Third Cycle

Putting together the two half cycles of $\bar{i}_{ab}$ results in (5.14). The average current now varies sinusoidally with $v_{ab}(t)$. The analysis for the average current $\bar{i}_{ab}$ for the first third of T_s is complete.

5.1.2 Second Third of Period

Only the AC sources v_b and v_c are used in the second third of the cycle, as shown in Fig. 5.4. Again notice how the topology is similar to the single phase AC to DC dual active bridge. The average current during the second third of T_s is investigated. During the third sixth of T_s, switches S_{b1} and S_{c2} are closed, applying the voltage $-v_{bc}$ to the primary. During the fourth sixth of T_s switches S_{c1} and S_{b2} are closed applying the voltage v_{bc} to the primary shown in Fig. 5.2. The duty ratio for the second third of the cycle is defined in (5.15). The pulse of V_d is applied to v_2 for a duration of $\frac{d_2 T_s}{6}$ during the third sixth of T_s. Notice how again the pulse of V_d remains in its respective sixth of T_s time period. $-V_d$ is applied to v_2 in the fourth sixth of T_s as shown in Fig. 5.2.

$$d_2(t) = \frac{|v_{bc}(t)|n}{V_d} \tag{5.15}$$

The currents I_4 and I_5 are defined in (5.16) and (5.17).

$$I_4 = \frac{-nv_{bc}}{L}\left(\frac{T_s}{12} + \Delta t - \frac{d_2 T_s}{12}\right) \tag{5.16}$$

$$I_5 = I_4 + \frac{-nv_{bc} - V_d}{L}\frac{d_2 T_s}{6} \tag{5.17}$$

Solving for I_5 results in zero. Again the converter A switches when the current is zero. Similarly the current reaches zero at time $\frac{2T_s}{3}$. The converter A is completely soft switched at zero current during the second third of the cycle.

The current i_{bc} is equal to the inductor current during the second third time period. The average current $\bar{i}_{bc}$ is calculated in (5.18). The average current is the same during the third sixth of T_s and fourth sixth of T_s, therefore the average current is only calculated during the third sixth of T_s.

$$\langle \bar{i}_{bc} \rangle_{[\frac{Ts}{3} - \frac{2Ts}{3}]} = \frac{6}{T_s} \int_{\frac{Ts}{3}}^{\frac{T_s}{2}} i_L(t) \tag{5.18}$$

$$= \frac{n\delta V_d}{4Lf_s} d_2(t) \tag{5.19}$$

The calculated average current $\bar{i}_{bc}$ is shown in (5.19). It varies with the duty ratio $d_2(t)$. The average current $\bar{i}_{bc}$ is zero during the first third time period and third third time period therefore the average current during the time period T_s is $\frac{1}{3}$ of the calculated average current in (5.19). The average current over T_s time period is given in (5.20).

$$\bar{i}_{bc}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s} \left| \cos\left(\omega_i t - \frac{\pi}{2}\right) \right| \tag{5.20}$$

The phase shift is changed by 180 degrees every half cycle of the input voltage, similarly to the previous section, in order to remove the absolute value sign from the average current. During the positive half cycle of v_{bc} delta is positive and during the negative half cycle of $v_b c$, the sign of delta is negative. The resulting average current $\bar{i}_{bc}$ is (5.21).

$$\bar{i}_{bc}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s} \cos\left(\omega_i t - \frac{\pi}{2}\right) \tag{5.21}$$

5.1.3 Third Third of Cycle

Only the AC sources v_c and v_a are used in the last third of the cycle, as shown in Fig. 5.5. Once again note how the topology is similar to the single phase AC to DC dual active bridge.

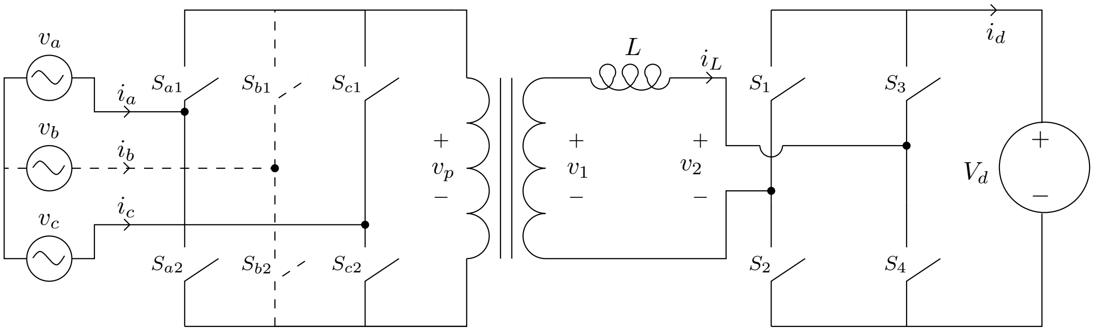


Figure 5.5: Third Third Cycle

During the fifth sixth of T_s ,switches S_{c1} and S_{a2} are closed, and v_{ca} is applied to the primary. Finally during the last sixth of T_s, switches S_{a1} and S_{c2} are closed, and $-v_{ca}$ is applied to the primary, depicted in Fig. 5.2.

The duty ratio for the last third time period of T_s is defined in (5.22). The pulse of V_d is applied to v_2 for a duration of $\frac{d_3 T_s}{6}$ during the fifth sixth of T_s. $-V_d$ is applied to v_2 in the last sixth of T_s as shown in Fig. 5.2.

$$d_3(t) = \frac{|v_{ca}(t)|n}{V_d} \tag{5.22}$$

The currents during last third of T_s I_6, I_7, and I_8 are calculated in (5.23), (5.24), and (5.25). Solving (5.25) using (5.23) and (5.24) reduces to zero current. The converter A is switched at zero current at time $\frac{5T_s}{6}$. In the last sixth of T_s the current again reaches zero concluding that the converter A is soft switched during the last third of T_s.

$$I_6 = \frac{nv_{ca}}{L}\left(\frac{T_s}{12} + \Delta t - \frac{d_3 T_s}{12}\right) \tag{5.23}$$

$$I_7 = I_6 + \frac{nv_{ca} - V_d}{L}\frac{d_3 T_s}{6} \tag{5.24}$$

$$I_8 = I_7 + \frac{nv_{ca}}{L}\left(\frac{T_s}{12} - \Delta t - \frac{d_3 T_s}{12}\right) \tag{5.25}$$

The current i_{ca} is equal to the inductor current during the last third of time period T_s. The average current $\bar{i}_{ca}$ is calculated in (5.26). The average current is the same during the fifth sixth of T_s and the last sixth of T_s, therefore the average current is only calculated during the fifth sixth of T_s.

$$\langle \bar{i}_{ca} \rangle_{[\frac{2T_s}{3} - Ts]} = \frac{6}{T_s} \int_{\frac{2T_s}{3}}^{\frac{5T_s}{6}} i_L(t) \tag{5.26}$$

$$= \frac{n\delta V_d}{4Lf_s} d_3(t) \tag{5.27}$$

The average current $\bar{i}_{ca}$ during the last third of T_s is shown in (5.27). The average current, given a fixed phase shift, varies with the duty ratio $d_3(t)$. The average current $\bar{i}_{ca}$ is zero during the first and second thirds of T_s therefore the average current $\bar{i}_{ca}$ over the time period T_s is $\frac{1}{3}$ of the average current calculated in (5.27). The average current $\bar{i}_{ca}$ over the time period T_s is given in (5.28).

$$\bar{i}_{ca}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s} \left| \cos\left(\omega_i t - \frac{7\pi}{6}\right) \right| \tag{5.28}$$

The absolute value can be removed by revisiting the change of delta in the two previous sections. The value of delta is fixed and positive for the positive half cycle of $v_{ca}(t)$, and delta is negative of the fixed valued for the negative half cycle of $v_{ca}(t)$. The end result is the average current defined in (5.29). The average current varies sinusoidally with the voltage $v_{ca}(t)$.

$$\bar{i}_{ca}(t) = \frac{\sqrt{3}n^2\delta V_i}{12Lf_s} \cos\left(\omega_i t - \frac{7\pi}{6}\right) \tag{5.29}$$

5.2 Virtual to Real Currents

The average currents $\bar{i}_{ab}$, $\bar{i}_{bc}$, and $\bar{i}_{ca}$ are virtual currents and only flow when their respective voltage sources are connected to the primary. The average virtual currents need to be converted to the average current per voltage source. In Fig. 5.6 the virtual currents are depicted.

Mesh analysis is used to convert the virtual currents to average currents per source.

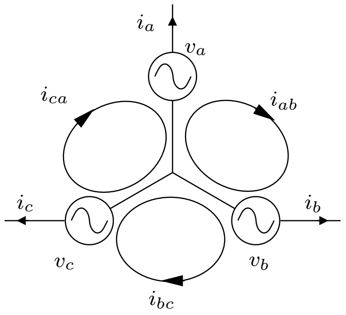


Figure 5.6: Virtual Currents

The average current $\bar{i}_a$ is equal to the difference between $\bar{i}_{ab}$ and $\bar{i}_{ca}$ as shown in (5.30).

$$\begin{aligned}\bar{i}_a &= \bar{i}_{ab} - \bar{i}_{ca}\\ \bar{i}_b &= \bar{i}_{bc} - \bar{i}_{ab}\\ \bar{i}_c &= \bar{i}_{ca} - \bar{i}_{bc}\end{aligned} \tag{5.30}$$

Similarly the average currents $\bar{i}_b$ and $\bar{i}_c$ are the difference of their mesh current counterparts (5.30).

The average currents $\bar{i}_a$, $\bar{i}_b$, and $\bar{i}_c$ are calculated using (5.14) (5.21) (5.29) and (5.30) and results in (5.31).

$$\begin{aligned}\bar{i}_a &= \frac{n^2\delta V_i}{4Lf_s}\cos\left(\omega_i t\right)\\ \bar{i}_b &= \frac{n^2\delta V_i}{4Lf_s}\cos\left(\omega_i t - \frac{2\pi}{3}\right)\\ \bar{i}_c &= \frac{n^2\delta V_i}{4Lf_s}\cos\left(\omega_i t - \frac{4\pi}{3}\right)\end{aligned} \tag{5.31}$$

The space vector approach is another method of calculating the average currents $\bar{i}_a$, $\bar{i}_b$, and $\bar{i}_c$. The voltages v_a, v_b, and v_c and average currents $\bar{i}_{ab}$, $\bar{i}_{bc}$, and $\bar{i}_{ca}$ are shown as space vectors in Fig. 5.7. Vector math is used to calculated the average currents $\bar{i}_a$, $\bar{i}_b$, and $\bar{i}_c$ and results are shown in Fig. 5.7.

Clearly, from the calculated average currents (5.31) and the space vector results in Fig. 5.7, the average currents are sinusoidal and in-phase with the input voltages. This results in unity power factor correction. Also note that there are no line frequency harmonics present.

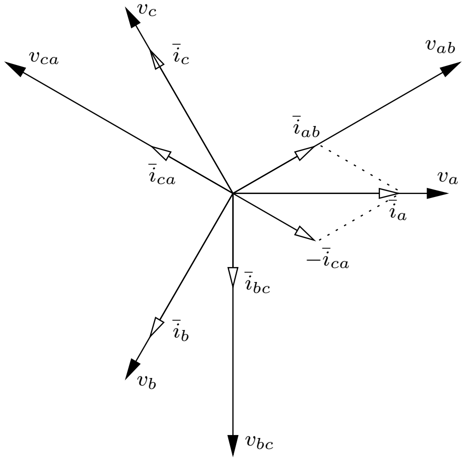


Figure 5.7: Space Vectors

5.2.1 Average Power

The duty cycles in (5.2), (5.15), and (5.22) are time varying and they reach their maximum when the corresponding line to line voltages peak. The maximum value of the duty ratios is defined as $\hat{d}$, and is given in (5.32).

Given that δ remains constant during a complete cycle of ω_i, by (5.5), the maximum value of $|\delta|$ must be less than or equal to $(1 - \hat{d})$.

$$\hat{d} = \frac{n\sqrt{3}V_i}{V_d} \tag{5.32}$$

The average power for this converter can be computed over a cycle of T_s using the equations (5.31) and (5.33). The average power, $\bar{P}$, is constant over T_s and is the same power that flows through this converter over a complete cycle of the input voltages.

$$\langle\bar{P}\rangle_{T_s} = v_a\bar{i}_a + v_b\bar{i}_b + v_c\bar{i}_c \tag{5.33}$$

$$\bar{P} = \frac{1}{8}\frac{\hat{d}^2 V_d^2}{Lf_s}\delta \tag{5.34}$$

The average power, calculated in (5.34), is changed to per unit using the base (5.35) and plotted for fixed values of $\hat{d}$ from 0.1 to 0.9, over the allowable range of δ in Fig. 5.8.

$$P_{base} = \frac{V_d^2}{2\pi f_s L} \tag{5.35}$$

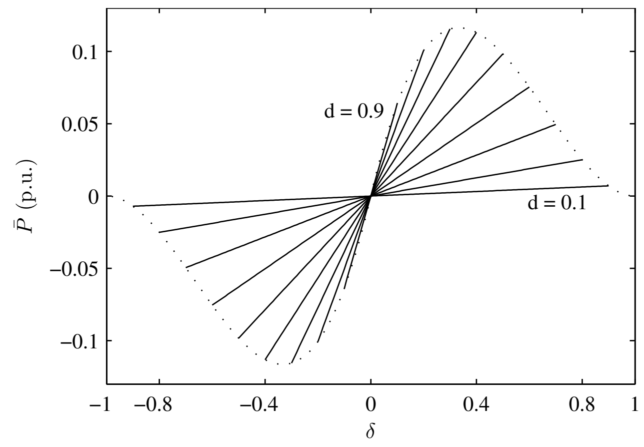


Figure 5.8: Average Power Per Unit

Irrespective of the maximum duty ratio $\hat{d}$, the power is a linear function of the phase shift. Controlling the power using this modulation scheme is relatively simple compared to the control scheme present in previous chapters. The maximum power, $\bar{P}_{pu}$, per unit feasible in this modulation strategy is $(\frac{\pi}{27})$ and is realized at $\hat{d}$ equal to $\frac{2}{3}$ and δ equal to $\frac{1}{3}$.

5.2.2 Utilization Factor

An important metric in analyzing dual active bridge type of converters is the ratio of average power to reactive power in the transformer. The reactive power in the transformer is defined in (5.36).

$$P_T = \frac{1}{2}(v_{p_{rms}} i_{p_{rms}} + v_{2rms} i_{Lrms}) \tag{5.36}$$

The reactive power defined in (5.36) is equivalent to the volt-ampere rating of the transformer. The reactive power, in per unit, is plotted in Fig. 5.9, for fixed $\hat{d}$ from 0.1 to 0.9 in steps of 0.1 and over all δ from -1 to 1.

The results of the reactive power show that for low duty ratios there is a lot of reactive power compared to high duty ratios where there is little reactive power. Intuitively this

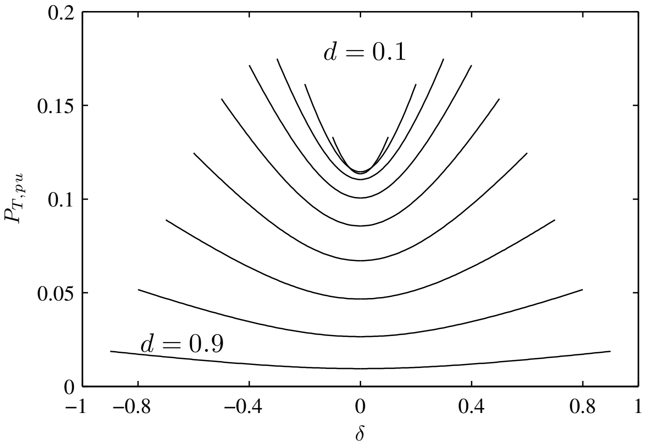


Figure 5.9: Reactive Power Per Unit

makes sense because the duty ratio is essentially the amount of time that the DC source is connected to the transformer. The longer the DC source is connected to the transformer, the more power is transferred into the source, conversely the shorter time the DC source is connected to the transformer the more time the power goes back and forth in the transformer.

The utilization factor is defined as the reactive power divided by average power (5.37). The utilization factor gives insight into how much over power is being transferred compared to the power that is being transferred back and forth in the transformer. For example, even when δ is equal to zero, the net average power transferred is zero, but a non trivial amount flows back and forth between the two sources.

$$U_f = \frac{P_T}{\bar{P}} \tag{5.37}$$

The utilization factor has been plotted in Fig. 5.10 as a function of δ for various fixed values of $\hat{d}$ from 0.1 to 0.9 in steps of 0.1.

The maximum utilization factor of the converter occurs at $\hat{d}$ equal to $\frac{2}{3}$. It is the same $\hat{d}$ that provides the maximum amount of power transfer. These two metrics give important insight into potential converter losses in the transformer and transformer ratings. They can and should be used when designing a converter with the presented modulation scheme.

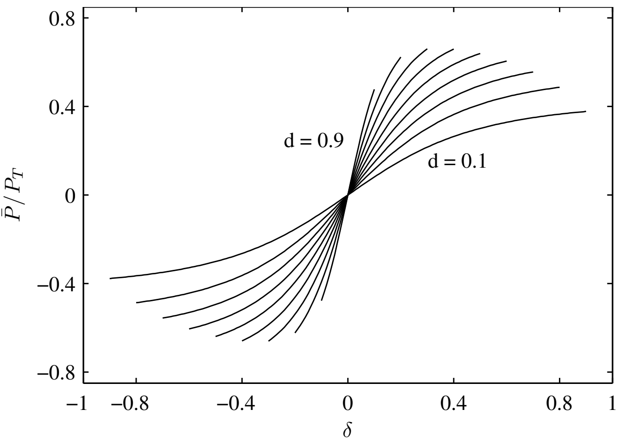


Figure 5.10: Transformer Utilization

5.2.3 RMS Ripple Current

Another important metric in AC to DC dual active bridges is the RMS ripple current into the DC source. The RMS ripple current gives an insight into the sizing of the capacitor which also effects the cost. The RMS ripple current is given in (5.38).

$$i_{d,ripple,rms} = \frac{V_d\hat{d}\left(-15\sqrt{\pi}\delta\hat{d} + \sqrt{5}\sqrt{\hat{d}\left(80 - 45\hat{d}\pi + 240\delta^2 + 64\hat{d}^2\right)}\right)}{120L\sqrt{\pi}f_s} \tag{5.38}$$

The per unit RMS ripple current, is the RMS ripple current divided by the current base($i_{base} = \frac{V_d}{2\pi f_s L}$) and results in (5.39). It is plotted for various fixed $\hat{d}$ from 0.1 to 0.9 in steps of 0.1, as a function of δ in Fig. 5.11.

$$i_{d,ripple,rms,PU} = \frac{1}{60}\hat{d}\left(-15\pi\delta\hat{d} + \sqrt{5\pi}\sqrt{\hat{d}\left(80 - 45\hat{d}\pi + 240\delta^2 + 64\hat{d}^2\right)}\right) \tag{5.39}$$

Considering a design just based on the minimum ripple current would lead to a duty ratio of zero or as close to zero as possible which is infeasible. One note in the plots in Fig. 5.11, is that there is maximum near $\hat{d}$ equal to $\frac{2}{3}$. The trade off while operating at $\hat{d}$ equal to $\frac{2}{3}$, is while the converter operates at the maximum average power and utilization, the ripple current is also maximum.

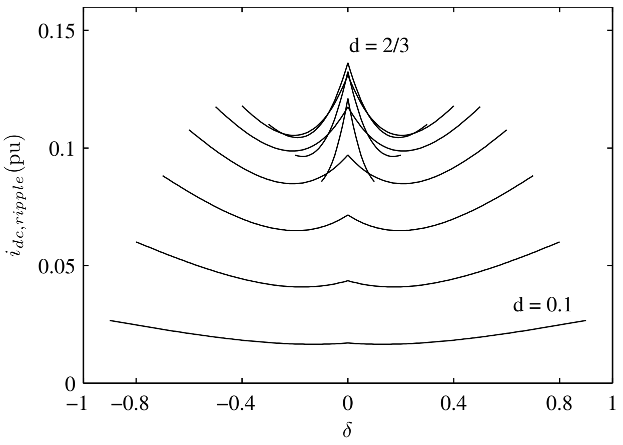


Figure 5.11: Output RMS Ripple Current

Parameter	**Value**	**Unit**
L	50	μH
V_i	100	V
V_d	400	V
f_i	60	Hz
f_s	3.33	kHz
n	1	
δ	0.3	

Table 5.1: Simulation Values

It is the opinion of the author that the optimum choice of $\hat{d}$ is $\frac{2}{3}$. The correct rms rating of the capacitor based on that choice will have to be selected above the calculated rms ripple current.

5.3 Simulation

The converter in Fig. 5.1 is simulated in MATLAB Simulink, with the parameters listed in Table 5.1, using the presented modulation scheme. In this simulation the phase shift is positive therefore power is put into the DC source.

Fig. 5.12 shows the simulation results of v_1, v_2, and i_L for a complete period of

T_s. Note the frequency at which the voltage is applied to the transformer is 3 times the modulation scheme frequency f_s. At each switching of the voltage v_1 in Fig. 5.12, the current i_L is zero which confirms the theoretical results of zero current switching in converter A.

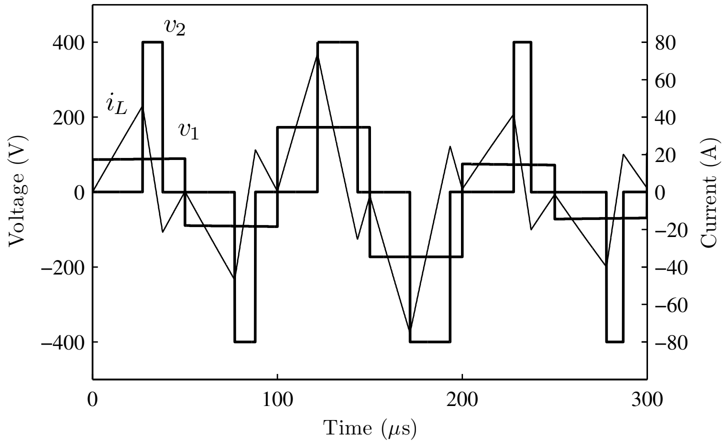


Figure 5.12: Simulation Result : Zero Current Switching (ZCS)

The simulation results of the average input currents $\bar{i}_a$ $\bar{i}_b$, and $\bar{i}_c$ are shown in Fig. 5.13.

They are observed to be sinusoidal, in phase with their respective input voltages, and contain no line frequency harmonics. This result confirms unity input power factor correction. Using the values in Table 5.1 combined with the equations in (5.31), the peak of the average currents is 15A which matches with the simulation results.

The switched currents for $i_a(t)$, $i_b(t)$, and $i_c(t)$ are shown in Fig. 5.14.

The switched currents show the 3 times f_s, or 10kHz, harmonic as well as the outline of the 60Hz component. The calculated FFT on the input switch currents is shown in Fig. 5.15 and is normalized to the fundamental 60Hz.

The input current contains the fundamental at 60Hz and some more switching components around the 3.3kHz range. The switching components around 3.3kHz can be removed from the input current by inserting and properly designing a LC filter.

The Fourier spectrum of the input current is closely investigated around 60 Hz in Fig. 5.16. There are zero line frequency harmonics and therefore 0% THD in the input

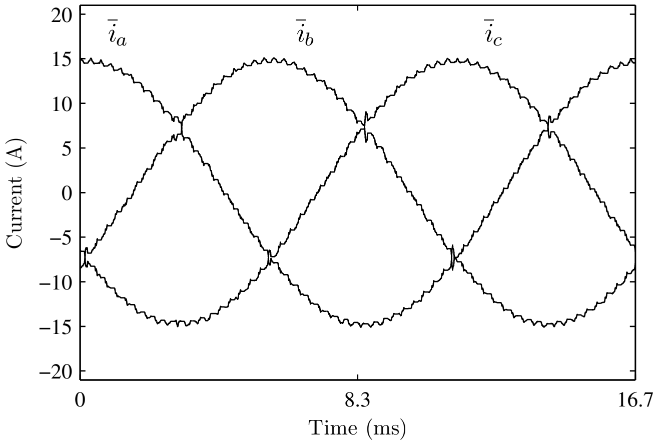
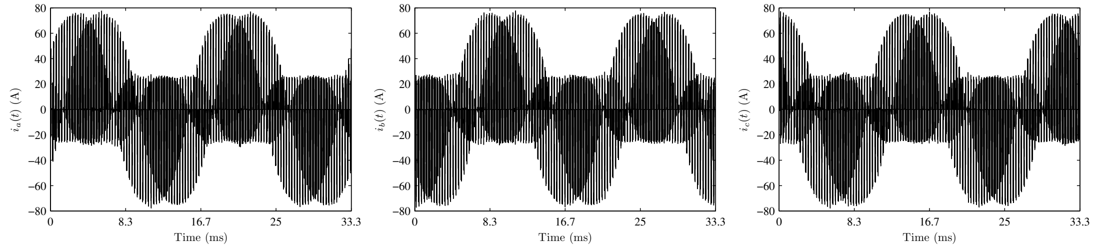


Figure 5.13: Average Three Phase Currents



(a) Phase A Switch Current (b) Phase B Switch Current (c) Phase C Switch Current

Figure 5.14: Switch Currents

current. The harmonic content of the other phases b and c are the exact same as phase a and contain zero line frequency harmonics. This modulation scheme provides a distinct advantage of the previously present control strategies in that it provides no line frequency harmonics.

5.4 Bidirectional Power Flow

The converter in Fig. 5.1 is simulated in MATLAB Simulink, using the parameters listed in Table 5.1, except this time the phase shift δ is negative, -0.3 instead of 0.3. In this simulation the phase shift is negative therefore power is put into the three-phase AC source.

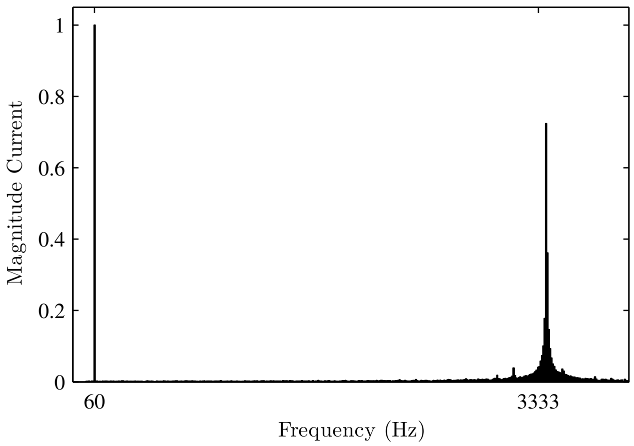


Figure 5.15: Frequency Spectrum of i_a

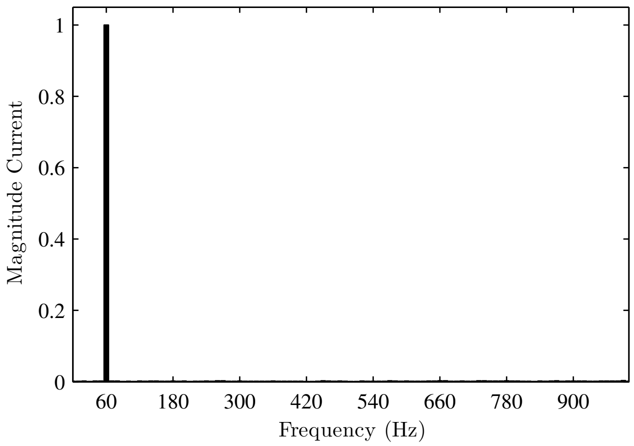


Figure 5.16: Frequency Spectrum of i_a

The two voltages, v_1 and v_2, and the current i_L is plotted for one complete cycle of the modulation scheme in Fig. 5.17. One thing to note in the figure is that the voltage v_2 is shifted to the left with respect to v_1 which reflects the change of δ from 0.3 to -0.3. The current i_L is zero at each switching of v_1, therefore, converter A is soft switched. Zero current switching doesn't depend on the direction of power flow.

The filtered current from AC source $v_a(t)$ is shown in Fig. 5.18. The current is sinusoidal and 180 degrees out of phase with the input voltage. This result shows power

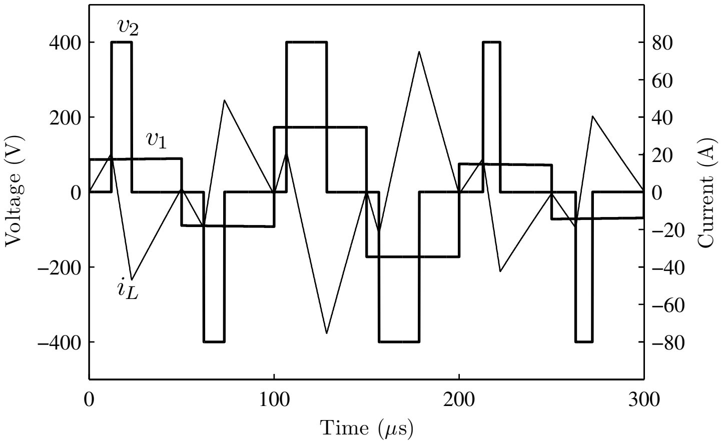


Figure 5.17: One Cycle of Modulation Scheme

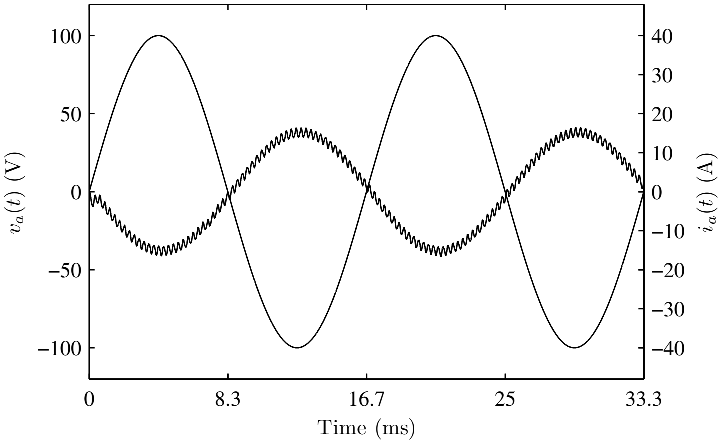


Figure 5.18: Filtered Phase A Current

is being put into the AC sources.

5.5 Conclusion

In this section a three-phase AC to DC dual active bridge modulation scheme was presented. The three-phase scheme built upon the previously discussed single-phase modulation scheme. The average currents from the sources were derived and found to track the incoming voltages. The magnitude of the average currents can be controlled by the phase shift δ and is a linear control variable. Important metrics including reactive power, utilization factor, and RMS ripple current were discussed and derived. Simulation results were presented for proposed modulation scheme. The input currents are sinusoidal and in phase with the input voltages. The input currents contain no line frequency harmonics and the input converter A is soft switched under all load conditions. The converter is simulated with a negative phase shift providing power back to the three-phase AC sources. This result proves the modulation scheme and converter can provide bidirectional power flow.

Chapter 6

Hardware Results for Single Phase Input

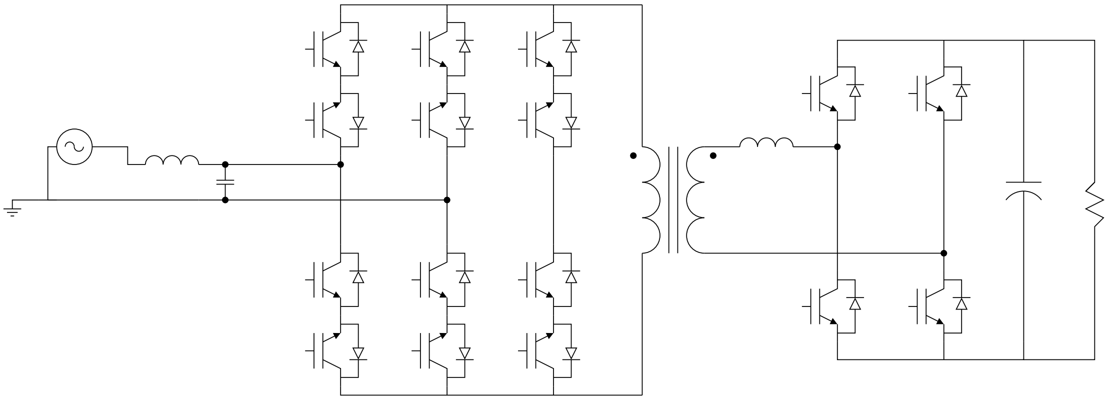


Figure 6.1: Single Phase Hardware Topology

6.1 Power Board

The IGBT module used in this design is 1200V Microsemi APTGT75TDU120PG. They are rated for 75 Amps and can switch up to 20kHz. One module is used per power board. Each power board is comprised of one module or six IGBTS, with three pairs of emitter tied IGBT's. Each module then has 3 bidirectional switches. Each board is a three-phase to one output building block. Three boards creates a regular three-phase to three-phase matrix converter. Two boards creates a three-phase to single-phase matrix converter which is used in this project.

6.2 Single Phase Experimental Results

The single-phase topology is built and shown in Fig. 6.2. The parts in the figure are as follows: one is the input filter inductance, two is input filter capacitance, three is the single phase transformer, four is the DC bus capacitor, five is the h-bridge inverter, six is the leakage inductance, seven is the input matrix converter, and eight is the FPGA control board.

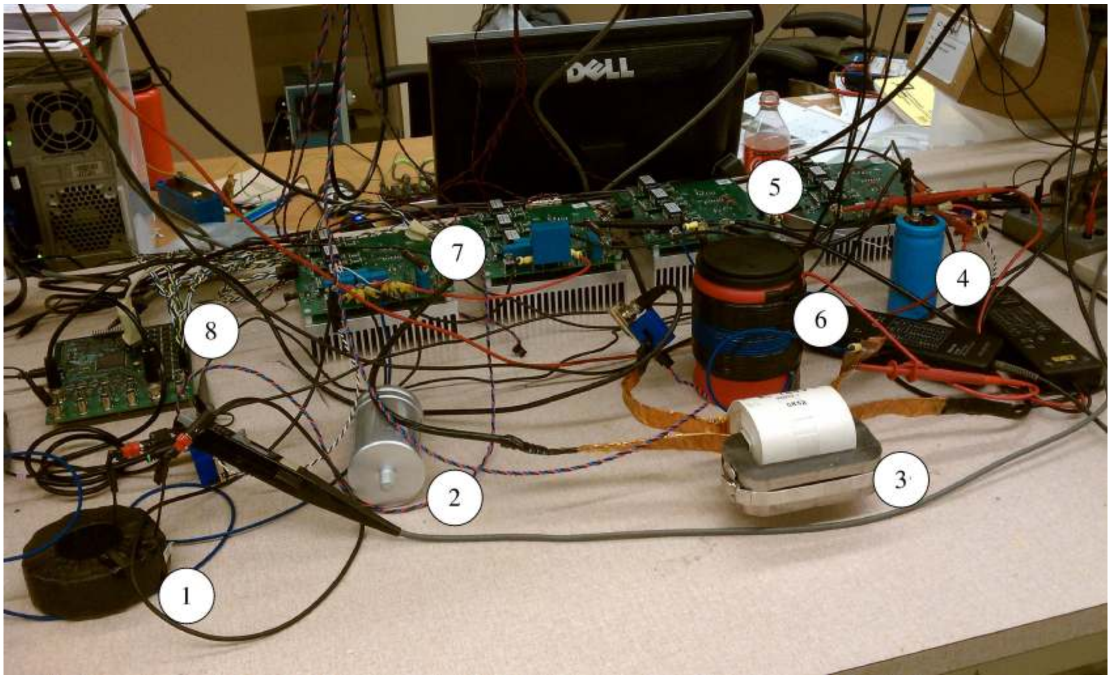


Figure 6.2: Single Phase Input Experimental Setup

6.2.1 Experiment

The FPGA code was written to implement the duty ratios, matrix converter switching, h-bridge switching, over current protection, and over voltage protection. The duty

Table 6.1: Experimental Values

Parameter	**Value**	**Unit**
L_f	5	mH
C_f	40	μF
L	50	uH
V_i	100	V
V_d	244	V
f_i	60	Hz
f_s	10	kHz
n	1	
δ	0.3	

ratios are calculated using the input voltage and bus voltage, which is measured using a LEM sensor and fed back to the FPGA. The code accepts a fixed phase shift. The experiment is run using the values in Table 6.1. The transformer was wound to minimize the leakage inductance and an external inductance is added in series to drown out the leakage inductance. This is done in order to be able to change the leakage inductance quickly, without having to redesign and rewind a new transformer.

The primary and h-bridge voltage waveforms are plotted in Fig. 6.3. The primary

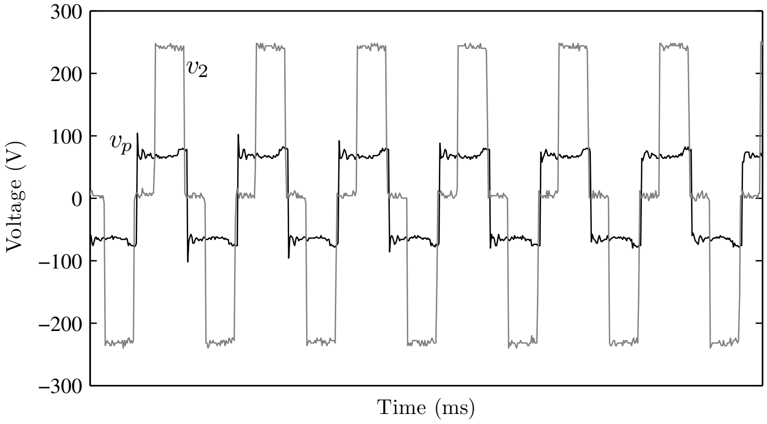


Figure 6.3: Primary and H-Bridge Voltage

voltage is shown in black and the h-bridge voltage is shown in grey. Note how the average voltage over a half cycle of the primary and h-bridge are equal. The h-bridge voltage is shifted positive with respect to the primary.

The primary voltage and primary current are shown in Fig. 6.4.

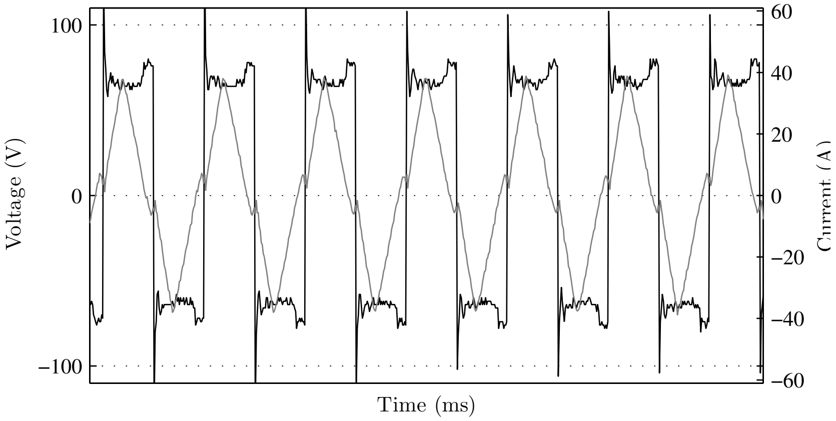


Figure 6.4: Primary Voltage and Current

The current is at or very close to zero at each switching of the primary voltage. This results shows the input matrix converter is soft switched, zero current switching. Four step commutation is used in the FPGA but when the current is very close to zero, due to timing and measurement inaccuracies, the fpga cannot determine the direction of the current. During this time all the switches are turned off and the clamp circuit provides a path for the primary current. The high voltage spike present on the primary voltage is the instance in time where the clamp circuit is conducting.

The input current is shown in Fig. 6.5. The input current is sinusoidal. The harmonics present in the input current are shown in Fig. 6.6.

The input current contains 3.5% in the third, 4% in the fifth, 1.5% in the seventh, 1% in ninth, and 1% in the eleventh. These harmonics are above IEEE-519. The harmonics present in the input current are coming from two sources. The first source is the LC filter. The input voltage from the grid contains some harmonics at the fifth, seventh, eleventh, and thirteenth. The impedance of the LC filter at these harmonics is quite low and leads to some significant currents at the harmonics. The second source is the

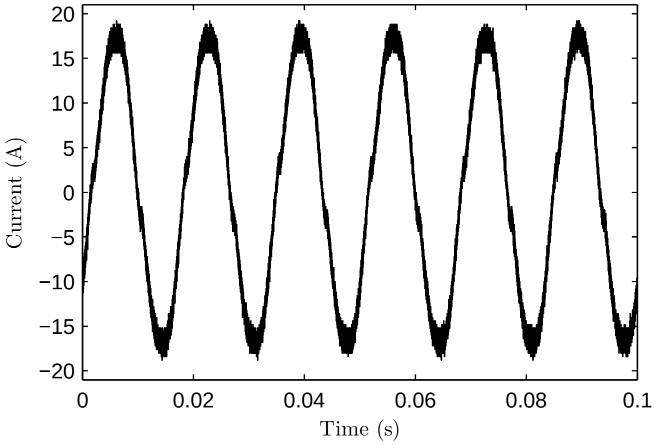


Figure 6.5: Input Current

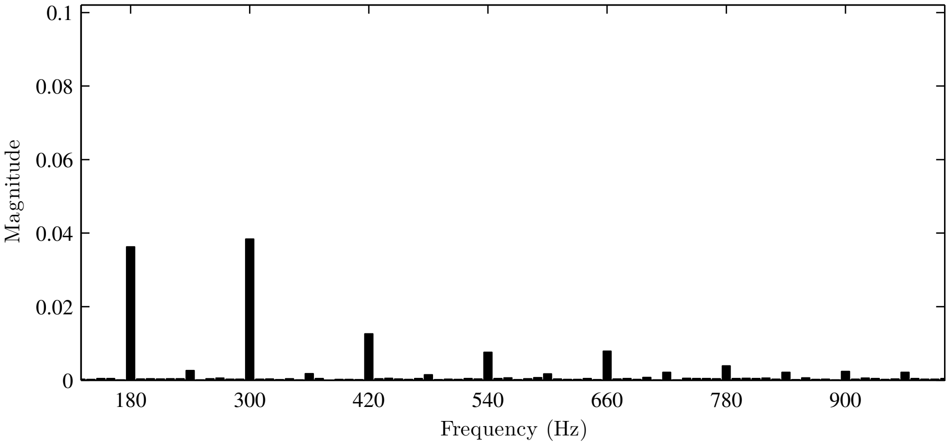


Figure 6.6: Input Current Harmonics

duty ratio calculation. The duty ratio is calculated from the measured input voltage and because the input voltage contains harmonics, then the average current drawn will contain those harmonics.

Chapter 7

Hardware Results for Three Phase Input

7.1 Three Phase Experimental Results

The three-phase topology in Fig. 7.1 is built and shown in Fig. 7.2

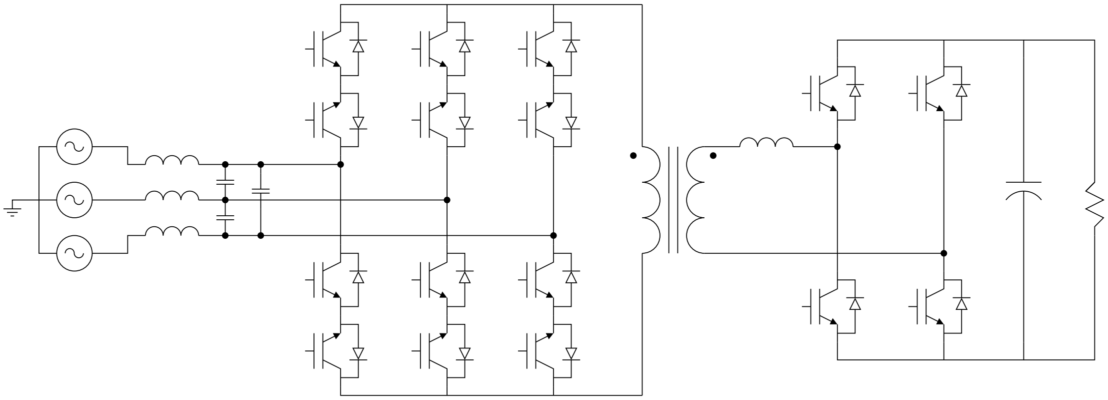


Figure 7.1: Hardware Topology

The components of the hardware setup in Fig. 7.2 are as follows: one is the FPGA controller board which measures the input line voltages, the DC bus voltage, and the primary current, and it outputs all the switching signals for the matrix converter and h-bridge. The FPGA is also running four step commutation for the matrix converter. Two

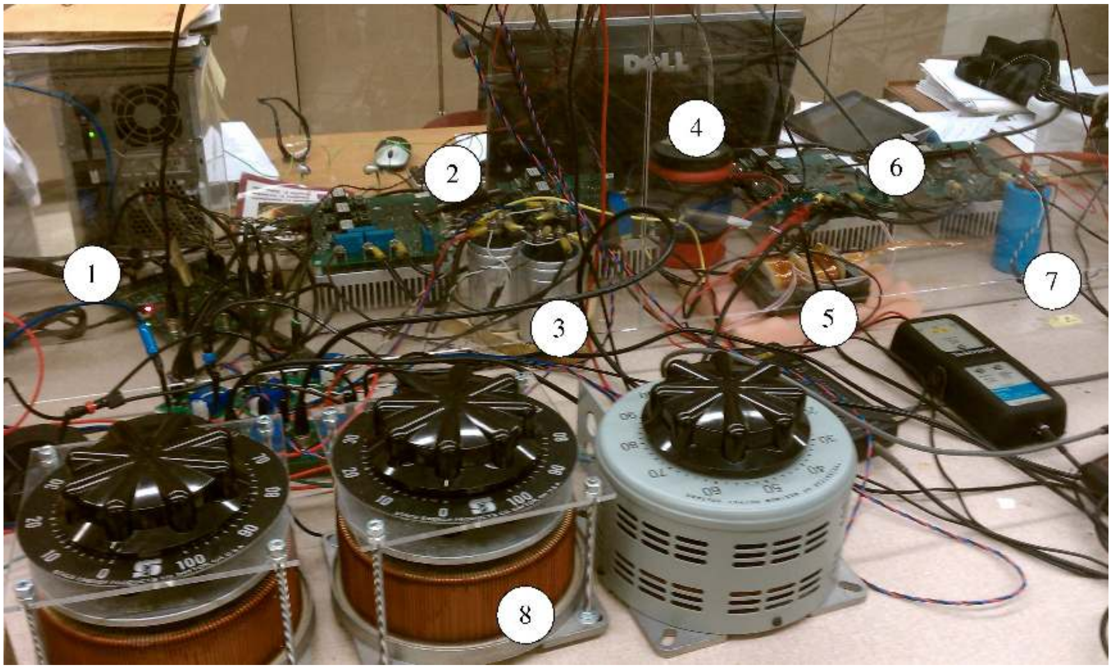


Figure 7.2: Three Phase Input Experimental Setup

is the three-phase to single-phase matrix converter. Three is the input filter capacitor bank. Four is the leakage inductance of the transformer. Five is the transformer. Six is the h-bridge inverter. Seven is the DC bus capacitor. Eight is three variacs and they are used to control the input voltages.

7.2 Experiment

The experiment is run using the values in Table 7.1.

The voltage across the primary of the transformer and the voltage output from the h-bridge are shown in Fig. 7.3.

The phase shift of v_2 with respect to the primary is positive meaning power is coming from the grid and being consumed by the resistive load. Note that the average voltage of v_2 for any half cycle is equal to the average voltage v_p. This results confirms the implementation of the duty ratios. Also note how the voltage pulse v_2 stays with in its respective half time period which confirms the analytical result.

Parameter	**Value**	**Unit**
L_f	5	mH
C_f	40	μF
L	35	uH
V_i	100	V
V_d	300	V
f_i	60	Hz
f_s	5	kHz
n	1	

Table 7.1: Experimental Values

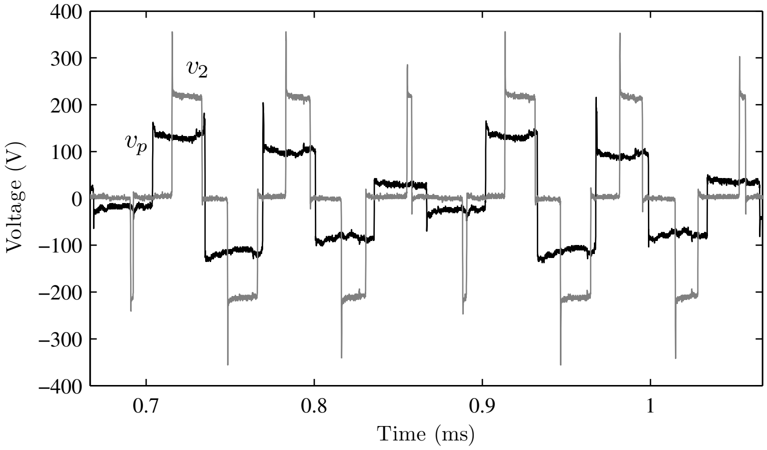


Figure 7.3: Primary and H-Bridge Voltage

The primary voltage and primary current are shown in Fig. 7.4.

The current is at or almost zero for every switching of the primary voltage. This result confirms zero current switching of the matrix converter. The current is not exactly zero because the analysis did not consider the effects of the magnetizing inductance of the transformer. A controller that drives the current to zero each half cycle is an area for future research.

The input voltage and input current are shown in Fig. 7.5. The input voltage contains ripple from the matrix converter switching. This ripple can be removed with

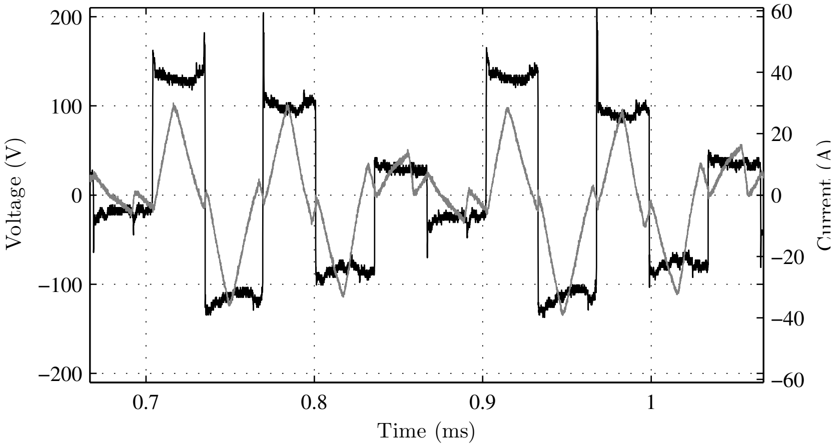


Figure 7.4: Primary Voltage and Current

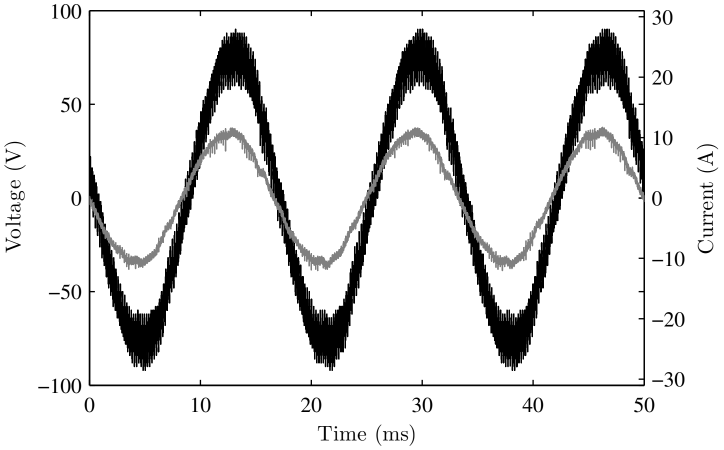


Figure 7.5: Phase A Input Voltage and Current

a properly designed LC filter and is another area considered for future research. The input current is sinusoidal and in phase with the input voltage.

The three phase input currents are shown in Fig. 7.6. The three-phase input currents are sinusoidal and in phase with their respective phase voltages.

The input current contains some line frequency harmonics and are shown in Fig. 7.7

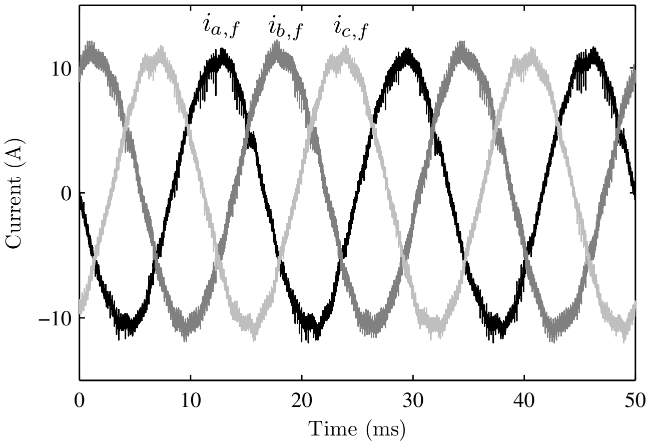


Figure 7.6: Input Currents

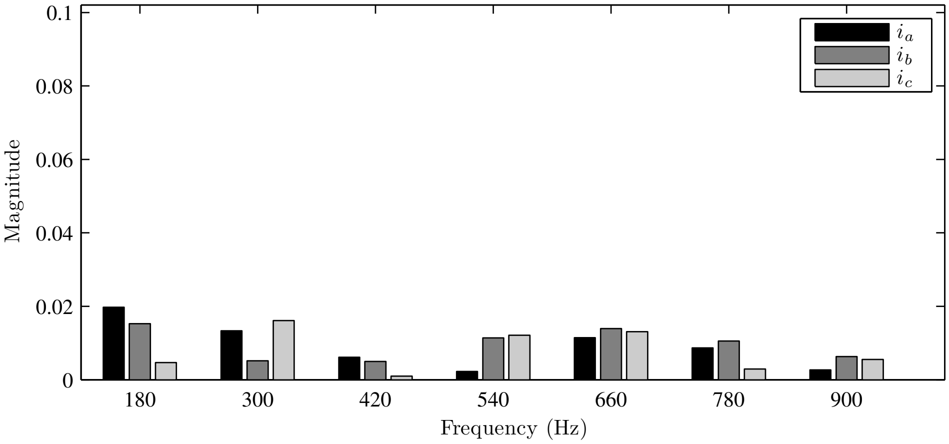


Figure 7.7: Input Current Harmonics

The input currents contain less than 2% in harmonics from third to fifteen. The harmonics in the input current are coming from the grid voltage harmonics. The grid contains harmonics at the fifth seventh eleventh and thirteenth. The LC filters impedance at these frequencies is small and therefore the LC filter draws undesired currents at each respective harmonic. The experimental results are not under input current control. The harmonic currents in the input can be removed by implementing input current control, and is the subject of future research work.

7.3 Bidirectional Power Flow

The resistive load R is removed and replaced with a DC source. The experiment is run again but with a negative phase shift in order to put power back onto the grid.

The primary voltage and primary current are shown in Fig. 7.8. The primary current is very close to zero during all transitions of the matrix converter, leading to zero current switching of the input matrix converter.

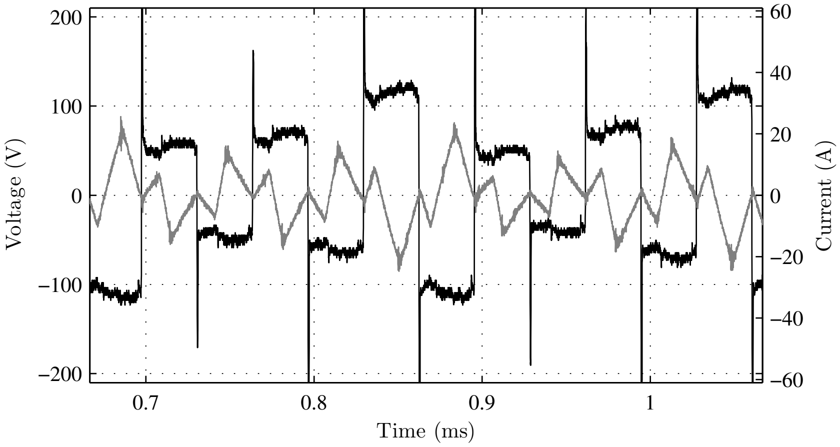


Figure 7.8: Zero Current Switching

The primary voltage and h-bridge voltage are shown in Fig. 7.9. The primary voltage is plotted in black and the h-bridge voltage is plotted in grey. The h-bridge voltage has negative phase shift with respect to the primary.

The input currents for all three phases are shown in Fig. 7.10. The currents contain a significant amount of harmonics.

The magnitude of the current harmonics are shown in Fig. 7.11. The third harmonic is present, the fifth has 4 to 6%, the seventh has almost 10%, the ninth has 3%, the eleventh has 5%, and the thirteenth has 2%. These harmonic levels are not acceptable. The source of the harmonics is two fold. The first source of the harmonics is from the LC filter. The grid contains some fifth, seventh, eleventh and thirteenth harmonics on the voltages. The LC filter impedance at these harmonics is low therefore drawing a significant amount of current at those respective currents. Secondly the duty ratio reference is derived from the voltage at the input filter capacitors. The voltage at the

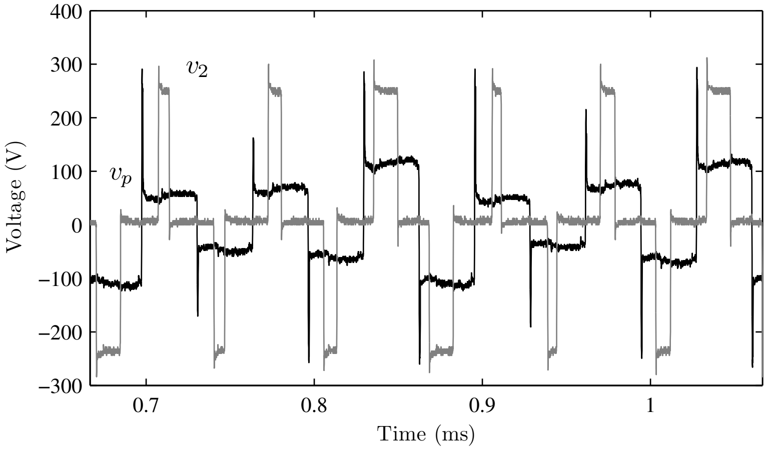


Figure 7.9: Primary and H-Bridge Voltage

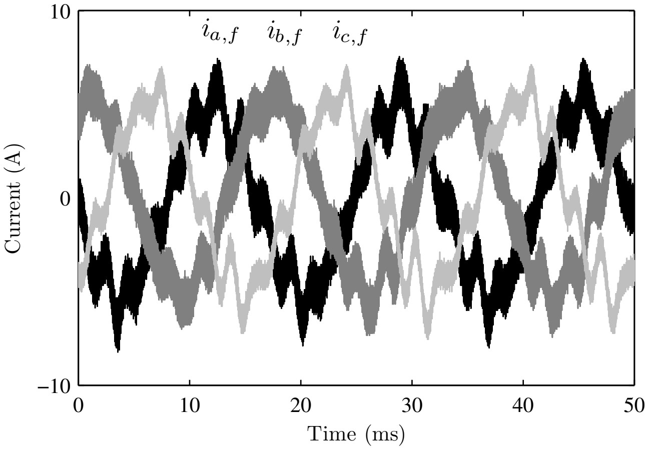


Figure 7.10: Input Currents

capacitors contained large ripple and transients. These transients and ripple injected were measured and lead to injection in to the duty ratio calculation on-board the FPGA.

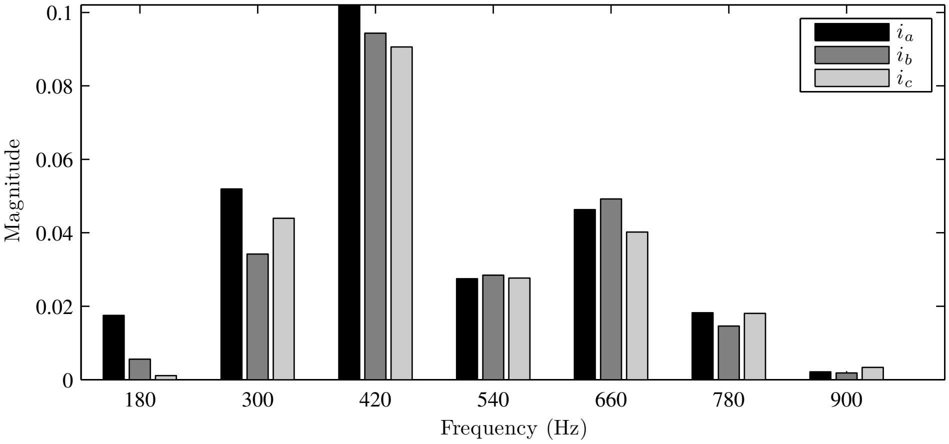


Figure 7.11: Input Current Harmonics

Chapter 8

Conclusion

EV's and PHEV's continue to penetrate into the consumer market. With more EV's and PHEV's on the streets, implementing V2G becomes more feasible and advantageous. This thesis introduces a novel topology for a isolated bidirectional vehicle charger that accepts single-phase or three-phase input. This topology has a distinct advantage over other isolated AC to DC topologies because it doesn't have an intermediate current or voltage link.

8.1 Single Phase Control

In Chapter 2 a model is derived for the average power that flows per cycle. Based on this model a power controller is designed to follow the power reference. The power reference in single-phase is exactly like a power factor correction circuit. The controller is designed and results for the input current, and power are given. The input current contains some harmonics so the controller is redesigned to reduce the harmonics. Given a fixed power flowing, steady state analysis is done to derive equations for the voltage and voltage ripple. Simulations results confirm the steady state analysis. A voltage controller is designed around the power loop. The voltage control designed is very similar to a power factor correction circuit. Results are given for the steady state voltage, the input current, and the input current harmonics. The total harmonic distortion is found to be with in IEEE-519 specifications.

8.2 Three Phase Control

In Chapter 3 a model is derived for the average power that flows per cycle. The difference between this model and the model in the single phase case, is that in the single phase case the average power per cycle changes and in the three phase case the average power per cycle is very close to constant. A power controller is designed around the model. Results for input current show a considerable amount of unbalanced harmonics. The duty ratio based method is adjusted by switching the order in which the duty ratios are compared to ramp. This adjustment leads to balanced currents and reduced harmonics. A voltage controller is designed and results of the voltage, input current, and input current harmonics are given. The RC network is replaced by a DC source to show the converter working in both modes, charging and discharging.

8.3 Single Phase DAB Modulation Strategy

In Chapter 4, a new modulation strategy is presented. The modulation strategy has zero line frequency harmonics. The control variable is also a linear relationship with power, therefore a voltage control design would be simple. Important metrics, like current ripple, and utilization factor are derived to aide in system design. simulations results are given, for charging and discharging, and match analytical results.

8.4 Three Phase DAB Modulation Strategy

Chapter 5 presents a new modulation strategy for three-phase AC to DC dual active bridge converter. The input currents contain zero line frequency harmonics. The switching currents contain the fundamental and components at one third of the switching frequency applied to the transformer. Conceptually, this point is the result of interweaving three single-phase ac to dc dual active bridges. The average power has a linear relationship with the control variable. Important metrics like average current, average power, utilization factor, and rms ripple current are derived. Simulation results are compared and match with analytical results.

8.5 Hardware

Chapter 6 and Chapter 7 presents experimental results for the modulation strategy presented in Chapter 4 and Chapter 5. The input currents are sinusoidal but contain harmonics. The harmonics come from two sources. The first source of the harmonics is the utility voltage contains harmonics and the LC filter's impedance at those harmonics is low therefore some harmonic current is drawn. The LC filter impedance is the predominant factor of the input current harmonics. The second source of the input harmonics is the utility voltage is used to determine the duty ratio. The duty ratio is proportional to the average current therefore since the duty ratio is derived from the input voltage, the average currents will contain harmonics. These current harmonics can be removed by using dq control for the input current which is subject to future research.

8.6 Future Work

Future problem to investigate in this work include:

- The gird voltage often contains harmonics and because of the impedance of the input LC filter at these harmonics, some undesired harmonic current flows. Input current control using dq transform and current source rectifier control, simulation and hardware results are needed [54, 55, 56, 57, 58, 59].

- This thesis included hardware results but one difficulty often encountered was flux balance of the transformer. Design and implement a flux controller inside the FPGA based off of the primary current.

- Implement cycle by cycle zero current controller. The analysis of the DAB modulation method excluded the magnetizing inductance. The controller would be designed maintain zero current at each switching of the matrix converter.

- Compare the presented topology with indirect matrix converter at the input. Analyse the trade-offs between the two topologies.

- Implement closed loop control of the DAB modulation strategy.

- Optimize the design of the input filter based on the sizing of the converter, switching frequency, and dq current control.

References

[1] M. El Chehaly, O. Saadeh, C. Martinez, and G. Joos. Advantages and applications of vehicle to grid mode of operation in plug-in hybrid electric vehicles. In *Electrical Power Energy Conference (EPEC), 2009 IEEE*, pages 1 –6, 2009.

[2] Jasna Tomic and Willett Kempton. Using fleets of electric-drive vehicles for grid support. *Journal of Power Sources*, 168(2):459 – 468, 2007.

[3] B. Kramer, S. Chakraborty, and B. Kroposki. A review of plug-in vehicles and vehicle-to-grid capability. In *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, pages 2278 –2283, 2008.

[4] Doo yong Jung, Young-Hyok Ji, Chung yuen Won, Taeck-Kie Lee, Su-Won Lee, and Kwang duck Seo. Grid-connected electric vehicles charger station based on lithium polymer battery energy storage system. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –5, sept. 2010.

[5] M.C. Kisacikoglu, B. Ozpineci, and L.M. Tolbert. Examination of a phev bidirectional charger system for v2g reactive power compensation. In *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, pages 458 –465, 2010.

[6] D.C. Erb, O.C. Onar, and A. Khaligh. Bi-directional charging topologies for plug-in hybrid electric vehicles. In *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, pages 2066 –2072, 2010.

[7] S. Haghbin, K. Khan, S. Lundmark, M. Alaku andla, O. Carlson, M. Leksell, and O. Wallmark. Integrated chargers for ev's and phev's: examples and new solutions.

In *Electrical Machines (ICEM), 2010 XIX International Conference on*, pages 1 –6, 2010.

[8] Lisheng Shi, A. Meintz, and M. Ferdowsi. Single-phase bidirectional ac-dc converters for plug-in hybrid electric vehicle applications. In *Vehicle Power and Propulsion Conference, 2008. VPPC '08. IEEE*, pages 1 –5, 2008.

[9] Young-Joo Lee, A. Khaligh, and A. Emadi. Advanced integrated bidirectional ac/dc and dc/dc converter for plug-in hybrid electric vehicles. *Vehicular Technology, IEEE Transactions on*, 58(8):3970 –3980, 2009.

[10] X. Zhou, S. Lukic, S. Bhattacharya, and A. Huang. Design and control of grid-connected converter in bi-directional battery charger for plug-in hybrid electric vehicle application. In *Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE*, pages 1716 –1721, 2009.

[11] Xiaohuz Zhou, Gangyao Wang, S. Lukic, S. Bhattacharya, and A. Huang. Multi-function bi-directional battery charger for plug-in hybrid electric vehicle application. In *Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE*, pages 3930 –3936, 2009.

[12] N.D. Weise, K.K. Mohapatra, and N. Mohan. Universal utility interface for plug-in hybrid electric vehicles with vehicle-to-grid functionality. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1 –8, 2010.

[13] S.V. Araujo, P. Zacharias, and R. Mallwitz. Highly efficient single-phase transformerless inverters for grid-connected photovoltaic systems. *Industrial Electronics, IEEE Transactions on*, 57(9):3118 –3128, sept. 2010.

[14] I.H. Hwang, K.S. Ahn, H.C. Lim, and S.S. Kim. Design, development and performance of a 50 kw grid connected pv system with three phase current-controlled inverter. In *Photovoltaic Specialists Conference, 2000. Conference Record of the Twenty-Eighth IEEE*, pages 1664 –1667, 2000.

[15] S. Jaganathan and Wenzhong Gao. Battery charging power electronics converter and control for plug-in hybrid electric vehicle. In *Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE*, pages 440 –447, 2009.

[16] A. Bruye? andre, L. De Sousa, B. Bouchez, P. Sandulescu, X. Kestelyn, and E. Semail. A multiphase traction/fast-battery-charger drive for electric or plug-in hybrid vehicles: Solutions for control in traction mode. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –7, sept. 2010.

[17] S. Lacroix, E. Laboure, and M. Hilairet. An integrated fast battery charger for electric vehicle. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –6, sept. 2010.

[18] L. De Sousa, B. Silvestre, and B. Bouchez. A combined multiphase electric drive and fast battery charger for electric vehicles. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –6, sept. 2010.

[19] O. Ellabban, J.V. Mierlo, and P. Lataire. Control of a bidirectional z-source inverter for hybrid electric vehicles in motoring, regenerative braking and grid interface operations. In *Electric Power and Energy Conference (EPEC), 2010 IEEE*, pages 1 –6, 2010.

[20] M.G. Egan, D.L. O'Sullivan, J.G. Hayes, M.J. Willers, and C.P. Henze. Power-factor-corrected single-stage inductive charger for electric vehicle batteries. *Industrial Electronics, IEEE Transactions on*, 54(2):1217 –1226, 2007.

[21] D.J. Thrimawithana, U.K. Madawala, and Yu Shi. Design of a bi-directional inverter for a wireless v2g system. In *Sustainable Energy Technologies (ICSET), 2010 IEEE International Conference on*, pages 1 –5, 2010.

[22] U. K. Madawala and D. J. Thrimawithana. A bi-directional inductive power interface for electric vehicles in v2g systems. *Industrial Electronics, IEEE Transactions on*, PP(99):1, 2011.

[23] S. Haghbin, M. Alakula, K. Khan, S. Lundmark, M. Leksell, O. Wallmark, and O. Carlson. An integrated charger for plug-in hybrid electric vehicles based on a special interior permanent magnet motor. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –6, sept. 2010.

[24] Sung Young Kim, Ilsu Jeong, Kwanghee Nam, and Hong-Seok Song. Three-port full bridge converter application as a combined charger for phevs. In *Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE*, pages 461 –465, 2009.

[25] H. Krishnaswami and N. Mohan. A current-fed three-port bi-directional dc-dc converter. In *Telecommunications Energy Conference, 2007. INTELEC 2007. 29th International*, pages 523 –526, 30 2007-oct. 4 2007.

[26] H. Krishnaswami and N. Mohan. Constant switching frequency series resonant three-port bi-directional dc-dc converter. In *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE*, pages 1640 –1645, june 2008.

[27] H. Tao, A. Kotsopoulos, J.L. Duarte, and M.A.M. Hendrix. Family of multiport bidirectional dc-dc converters. *Electric Power Applications, IEE Proceedings -*, 153(3):451 – 458, may 2006.

[28] H. Krishnaswami and N. Mohan. Three-port series-resonant dc-dc converter to interface renewable energy sources with bidirectional load and energy storage ports. *Power Electronics, IEEE Transactions on*, 24(10):2289 –2297, oct. 2009.

[29] H. Tao, A. Kotsopoulos, J.L. Duarte, and M.A.M. Hendrix. A soft-switched three-port bidirectional converter for fuel cell and supercapacitor applications. In *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, pages 2487 –2493, 2005.

[30] H.K. Krishnamurthy and R. Ayyanar. Building block converter module for universal (ac-dc, dc-ac, dc-dc) fully modular power conversion architecture. In *Power Electronics Specialists Conference, 2007. PESC 2007. IEEE*, pages 483 –489, june 2007.

[31] H. Krishnamurthy and R. Ayyanar. Stability analysis of cascaded converters for bidirectional power flow applications. In *Telecommunications Energy Conference, 2008. INTELEC 2008. IEEE 30th International*, pages 1 –8, sept. 2008.

[32] F. Krismer, J. Biela, and J.W. Kolar. A comparative evaluation of isolated bi-directional dc/dc converters with wide input and output voltage range. In *Industry*

Applications Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005, volume 1, pages 599 – 606 Vol. 1, oct. 2005.

[33] A.K. Jain and R. Ayyanar. Pwm control of dual active bridge: comprehensive analysis and experimental verification. In *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, pages 909 –915, nov. 2008.

[34] M.H. Kheraluwala and R.W. De Doncker. Single phase unity power factor control for dual active bridge converter. In *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, pages 909 –916 vol.2, October 1993.

[35] H. Krishnaswami and V. Ramanarayanan. Control of high-frequency ac link electronic transformer. *Electric Power Applications, IEE Proceedings -*, 152(3):509 – 516, may 2005.

[36] R.W.A.A. De Doncker, D.M. Divan, and M.H. Kheraluwala. A three-phase soft-switched high-power-density dc/dc converter for high-power applications. *Industry Applications, IEEE Transactions on*, 27(1):63 –73, 1991.

[37] E.R. Ronan, S.D. Sudhoff, S.F. Glover, and D.L. Galloway. A power electronic-based distribution transformer. *Power Delivery, IEEE Transactions on*, 17(2):537 –543, April 2002.

[38] Tiefu Zhao, Liyu Yang, Jun Wang, and A.Q. Huang. 270 kva solid state transformer based on 10 kv sic power devices. In *Electric Ship Technologies Symposium, 2007. ESTS '07. IEEE*, pages 145 –149, May 2007.

[39] J.F. Zhao J.G. Jiang X.W. Yang. Acdcdc isolated converter with bidirectional power flow capability. *IET Power Electronics*, 2009.

[40] Shesh Narayan Vaishnav and H. Krishnaswami. Single-stage isolated bi-directional converter topology using high frequency ac link for charging and v2g applications of phev. In *VPPC 2011*, 2011.

[41] Ned Mohan. *First Course on Power Systems*. MNPERE, 2006.

[42] Thomas J. Overbye Mulukutla S. Sarma and J. Duncan Glover. *Power Systems Analysis and Design*. CL-Engineering, 2011.

[43] M.N. Kheraluwala, R.W. Gascoigne, D.M. Divan, and E.D. Baumann. Performance characterization of a high-power dual active bridge dc-to-dc converter. *Industry Applications, IEEE Transactions on*, 28(6):1294 –1301, 1992.

[44] G. Guidi, M. Pavlovsky, A. Kawamura, T. Imakubo, and Y. Sasaki. Efficiency optimization of high power density dual active bridge dc-dc converter. In *Power Electronics Conference (IPEC), 2010 International*, pages 981 –986, june 2010.

[45] Chun kit Leung, S. Dutta, Seunghun Baek, and S. Bhattacharya. Design considerations of high voltage and high frequency three phase transformer for solid state transformer application. In *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, pages 1551 –1558, sept. 2010.

[46] Hengsi Qin and J.W. Kimball. Ac-ac dual active bridge converter for solid state transformer. In *Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE*, pages 3039 –3044, sept. 2009.

[47] Ned Mohan. *Power Electronics*. MNPERE, 2009 edition, 2009.

[48] O. Ellabban, O. Hegazy, J. Van Mierlo, and P. Lataire. A dsp digital controller design and implementation of a high power boost converter in hybrid electric vehicles. In *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pages 1 –6, sept. 2010.

[49] Jih-Sheng Lai and D.J. Nelson. Energy management power converters in hybrid electric and fuel cell vehicles. *Proceedings of the IEEE*, 95(4):766 –777, april 2007.

[50] M.B. Camara, H. Gualous, F. Gustin, and A. Berthon. Design and new control of dc/dc converters to share energy between supercapacitors and batteries in hybrid vehicles. *Vehicular Technology, IEEE Transactions on*, 57(5):2721 –2735, sept. 2008.

[51] S. Inoue and H. Akagi. A bidirectional dc-dc converter for an energy storage system with galvanic isolation. *Power Electronics, IEEE Transactions on*, 22(6):2299 –2306, nov. 2007.

[52] Ieee recommended practices and requirements for harmonic control in electrical power systems. *IEEE Std 519-1992*, 1993.

[53] K.K. Mohapatra, P. Jose, A. Drolia, G. Aggarwal, and S. Thuta. A novel carrier-based pwm scheme for matrix converters that is easy to implement. In *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, pages 2410–2414, June 2005.

[54] Jilei Gao, Hu Sun, Xiaojie You, and T.Q. Zheng. A novel control strategy for current-source rectifiers with space vector modulation. In *IEEE Power Electronics Specialists Conference (PESC), 2008.*, pages 3255 –3258, june 2008.

[55] M.P. Kazmierkowski and L. Malesani. Current control techniques for three-phase voltage-source pwm converters: a survey. *Industrial Electronics, IEEE Transactions on*, 45(5):691 –703, oct 1998.

[56] M. Tomasini, R. Feldman, P. Wheeler, J. Clare, C. Klumpner, and U. Mupambireyi. dq-control of high-power current source rectifiers utilizing selective harmonic elimination. In *Power Electronics and Applications, 2009. EPE '09. 13th European Conference on*, pages 1 –11, sept. 2009.

[57] A.M. Qiu, Yun Wei Li, Bin Wu, N. Zargari, and Yanfei Liu. High performance current source inverter fed induction motor drive with minimal harmonic distortion. In *Power Electronics Specialists Conference, 2007. PESC 2007. IEEE*, pages 79 –85, june 2007.

[58] J.R. Espinoza, G. Joos, J.I. Guzman, L.A. Moran, and R.P. Burgos. Selective harmonic elimination and current/voltage control in current/voltage-source topologies: a unified approach. *Industrial Electronics, IEEE Transactions on*, 48(1):71 –81, feb 2001.

[59] J.R. Espinoza and G. Joos. State variable decoupling and power flow control in pwm current-source rectifiers. *Industrial Electronics, IEEE Transactions on*, 45(1):78 –87, feb 1998.

Appendix A

Protection Module

This module is written in Verilog and shuts down the circuit if there is an over voltage at the DC bus, or an over current in the primary of the transformer.

```
module protector(
 input clk,
 input [11:0] voltage_sense,
 input signed [11:0] current_sense,
 output reg over_voltage,
 output reg over_current
 );
parameter OVERVOLT = 1900; // 1 count is 5mv on A2D, 250mV is 10V on bus, 1 count is .2 Volts on bus!
// 250 is 50 volts
// 350 is 68.32 volts
// 410 is 80 Volts
// 80Vm 80/10*.25/.00488
// 461 is 90 Volts
// 512 is 100 Volts
//614 is 120 Volts
// 768 is 150 Volts 150/10*.25/.00488
// 896 is 175 volts 175/10*.25/.00488
```

```
//1025 is 200 volts
//1153 is 225 volts
//1280 is 250 volts
//1409 is 275 volts
//1537 is 300 volts
//1665 is 325 volts
//1793 is 350 volts
//1953 is 375  volts
parameter OVERCUR = 1127;
// 1 amp = 1 mA
// 1 mA x 110 = 110 mV
// 110mV = 22.54 counts
// 20 amps , 20/1000 * 110/.00488 = 451
// 25 amps , 25/1000 * 110/.00488 = 563
// 30 amps , 30/1000 * 100/.00488 = 614
// 40 amps , 40/1000 * 100/.00488 = 819
// 45 amps , 45/1000 * 100/.00488 = 922
// 50 amps , 50/1000 * 100/.00488 = 1024
// 55 amps , 55/1000 * 100/.00488 = 1127
always @(posedge clk)
 begin
  if (voltage_sense[11] == 0 && voltage_sense > OVERVOLT)
   over_voltage <= 1;
  else
   over_voltage <= 0;

  if (current_sense > OVERCUR  || current_sense < -OVERCUR)
   over_current <= 1;
  else
   over_current <= 0;

 end
```

```
endmodule
```

Appendix B

Single Phase Matrix Converter Module

This code is written in Verilog. The code implements a single phase matrix converter with four step commutation. The input voltage is chopped at a frequency selected by OP FREQ.

```
module top(
 input clk,
 input AD1_Busy, AD6_Busy,
 output AD1_Cnvstb, AD6_Cnvstb,
 output AD1_CSb, AD6_CSb,
 output AD1_Sclk, AD6_Sclk,
 input AD1_DinA, AD6_DinA,
 input AD1_DinB, AD6_DinB,
 output [1:0] cur,
 output voltage_dir,
 output testout2,
 output reg [7:0] Q,
 output reg [3:0] Qlow,  // These outputs will always be low ↘
    →logic 0
 output [11:0] Q_Hbridge  // These outputs create the H-Bridge
 );


parameter state_A  = 0;
parameter state_B  = 1;
parameter state_0 = 2;
parameter state_AB_p_1 = 8;
parameter state_AB_p_2 = 9;
parameter state_AB_p_3 = 10;
parameter state_AB_n_1 = 11;
parameter state_AB_n_2 = 12;
parameter state_AB_n_3 = 13;

parameter OP_FREQ = 2500;// Sets the frequency of the matrix converter and h bridge
parameter HB_DELAY_COUNT = 50; //Sets amount of clock cycles to delay when switching the H bridge states
parameter BAND_MAX = 2047;
parameter PHASE_SHIFT_COUNT = 300;//500; // Sets the amount of clock cycles to delay!
parameter DUTY_RATIO_COUNT = 100; //Sets the amount of clock cycles until start of duty cycle
parameter VBUS_COUNTS = 100;


reg [4:0] main_state = 0; //Size state variable 1 msb greater than what is required.
reg [4:0] hb_state = 0;
reg A2B = 0; // 1 means going from A to B, 0 means going from B to A
reg [8:0] hb_delay_counter = 0;
reg reset = 0;
reg [11:0] band=25;// 25* 4.89mV = 125mV
reg [11:0] band2 = 4; // band for votlage direction detector
reg [8:0] delay_count=0;
reg [25:0] counter=0;
reg v_dir = 0;
always @(posedge clk or posedge reset) begin
Qlow = 4'b0000;
counter <= counter + 1;

if (reset)
 main_state <= state_A;
else
begin

/* State Machine for Single Phase Matrix Converter */
 case (main_state)
  state_A: begin //STATE A
  Q <= 8'b11000011;
  if (counter == OP_FREQ)
   begin
   counter <= 0;
   A2B <= 1;
   if (i_dir == 2'b00)
   main_state <= state_0;
   if (i_dir == 2'b10)  // POS CUR
   main_state <= state_AB_p_1;
   if (i_dir == 2'b11)  //NEG CUR
   main_state <= state_AB_n_1;
   end
  end

  state_B: begin  //STATE B
```

```
Q <= 8'b00111100;
if (counter == OP_FREQ)
 begin
 counter <= 0;
 A2B <= 0;
 if (i_dir == 2'b00)
 main_state <= state_0;
 if (i_dir == 2'b10)  // POS CUR
 main_state <= state_AB_p_3;
 if (i_dir == 2'b11)  //NEG CUR
 main_state <= state_AB_n_3;
 end
end

state_0: begin  // STATE ZERO
Q <= 8'b00000000;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_B;
 if (A2B == 0)
 main_state <= state_A;
 end
end

state_AB_p_1: begin  // STATE AB Positive 1
Q <= 8'b10000001;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_AB_p_2;
 if (A2B == 0)
 main_state <= state_A;
 end
end

state_AB_p_2: begin  // STATE AB Positive 2
Q <= 8'b10100101;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_AB_p_3;
 if (A2B == 0)
 main_state <= state_AB_p_1;
 end
end

state_AB_p_3: begin  // STATE AB Positive 3
Q <= 8'b00100100;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_B;
 if (A2B == 0)
 main_state <= state_AB_p_2;
 end
```

```
end

state_AB_n_1: begin  // STATE AB Negative 1
Q <= 8'b01000010;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_AB_n_2;
 if (A2B == 0)
 main_state <= state_A;
 end
end
state_AB_n_2: begin  // STATE AB Negative 2
Q <= 8'b01011010;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
 if (A2B == 1)
 main_state <= state_AB_n_3;
 if (A2B == 0)
 main_state <= state_AB_n_1;
 end
end
state_AB_n_3: begin  // STATE AB Negative 3
Q <= 8'b00011000;
delay_count <= delay_count + 1;
if (delay_count == 25)
 begin
 delay_count <= 0;
```

```
   if (A2B == 1)
   main_state <= state_B;
   if (A2B == 0)
   main_state <= state_AB_n_2;
   end
  end

    endcase

end
end

endmodule
```

Appendix C

H-Bridge Module

This code is written in Verilog. It implements a H-Bridge with variable phase shift and variable duty ratio or width of pulse.

```
module H_Bridge(
 input clk,
 input [25:0] counter,
 input v_dir,
 input [11:0] duty_ratio_count,
 input [10:0] phase_shift_count,
 output reg [11:0] Q
 );

/* H Bridge States */
parameter hb_alloff = 1;
parameter hb_Vplus = 2;
parameter hb_Vminus = 3;
parameter hb_Vdzero = 4;
parameter hb_Vnzero = 0;
parameter hb_AVd = 5;
parameter hb_BVd = 6;
parameter hb_AVn = 7;
parameter hb_BVn = 8;

/* Dead time */
/* 20clks * 20ns = 400ns */
parameter HB_DELAY_COUNT = 20;

/* registers */
reg [4:0] hb_state = 0;
reg [8:0] hb_delay_counter = 0;
reg [11:0] duty_count_sample = 1200;
reg v_dir_s = 1;

always @(posedge clk) begin

//if (counter == 2500)
// duty_count_sample <= duty_ratio_count;

/* State Machine for H bridge */
 case (hb_state)
 default:
  begin
  Q <= 12'b000000000000;
  hb_state <= hb_Vnzero;
  end

 hb_Vplus:
  begin
    Q <= 12'b000100000010;
   if (duty_count_sample > phase_shift_count && counter == ↘
      →2500−duty_count_sample+phase_shift_count)
    begin
    if (v_dir_s == 1)
     hb_state <= hb_AVd;
    else
     hb_state <= hb_BVn;
    end
   if (duty_count_sample <= phase_shift_count && counter == ↘
      →phase_shift_count-duty_count_sample)
    begin
    if (v_dir_s == 1)
     hb_state <= hb_AVd;
    else
     hb_state <= hb_BVn;
    end
  end
 hb_Vminus:
  begin
  Q <= 12'b000001001000;
  if (duty_count_sample > phase_shift_count && counter == ↘
     →2500-duty_count_sample+phase_shift_count)
   begin
   if (v_dir_s == 1)
     hb_state <= hb_AVn;
    else
     hb_state <= hb_BVd;
   end
  if (duty_count_sample <= phase_shift_count && counter == ↘
     →phase_shift_count-duty_count_sample)
   begin
   if (v_dir_s == 1)
     hb_state <= hb_AVn;
    else
     hb_state <= hb_BVd;
   end
```

```
end

hb_Vdzero:
begin
Q <= 12'b000101000000; //Vd zero
if (counter == duty_count_sample+phase_shift_count)
 begin
 if (v_dir_s == 1)
   hb_state <= hb_BVd;
  else
   hb_state <= hb_AVd;
 end
end

hb_Vnzero:
begin
Q <= 12'b000000001010; //Vn zero
if (counter == 1)
 begin
 v_dir_s <= v_dir;
 duty_count_sample <= duty_ratio_count;
 end

if (counter == duty_count_sample+phase_shift_count)
 begin
 if (v_dir_s == 1)
   hb_state <= hb_BVn;
  else
   hb_state <= hb_AVn;
 end
end
```

```
hb_AVd:
begin
Q <= 12'b000100000000; //A Vd
hb_delay_counter <= hb_delay_counter + 1;
if (hb_delay_counter == HB_DELAY_COUNT)
 begin
 hb_delay_counter <= 0;
 if (v_dir_s == 1)
   hb_state <= hb_Vdzero;
  else
   hb_state <= hb_Vplus;
 end
end

hb_BVd:
begin
Q <= 12'b000001000000;//B Vd
hb_delay_counter <= hb_delay_counter + 1;
if (hb_delay_counter == HB_DELAY_COUNT)
 begin
 hb_delay_counter <= 0;
 if (v_dir_s == 1)
   hb_state <= hb_Vminus;
  else
   hb_state <= hb_Vdzero;
 end
end

hb_AVn:
begin
Q <= 12'b000000001000;//A Vn
hb_delay_counter <= hb_delay_counter + 1;
```

```
    if (hb_delay_counter == HB_DELAY_COUNT)
     begin
     hb_delay_counter <= 0;
     if (v_dir_s == 1)
        hb_state <= hb_Vnzero;
      else
        hb_state <= hb_Vminus;
     end
    end

    hb_BVn:
    begin
    Q <= 12'b000000000010;//B Vn
    hb_delay_counter <= hb_delay_counter + 1;
    if (hb_delay_counter == HB_DELAY_COUNT)
     begin
     hb_delay_counter <= 0;
     if (v_dir_s == 1)
        hb_state <= hb_Vplus;
      else
        hb_state <= hb_Vnzero;
     //v_dir <= input_voltage_dir_clean;
     end
    end
  endcase
// END OF STATE MACHINE
  end
endmodule
```

Appendix D

Three Phase Matrix Converter Module

This code is written in Verilog. It implements a three-phase to single-phase matrix converter. The modulation presented in Chapter 5 and 6, is the modulation implemented in the code below. It also has implemented four-step commutation.

```
module matrix_converter(
 input clk,
 input [25:0] counter,
 input [1:0] current_direction,
 input vab_pol, vbc_pol, vca_pol,
 input fault,
 output reg [11:0] Q = 0
 );
/* Current Directions */
parameter POS_CUR = 2'b10;
parameter UNK_CUR = 2'b00;
parameter NEG_CUR = 2'b11;

/* Matrix Converter States */
parameter state_all_off  =  0;
parameter state_Vab_plus =  1;
parameter state_Vab_minus = 2;
parameter state_Vbc_plus =  3;
parameter state_Vbc_minus = 4;
parameter state_Vca_plus =  5;
parameter state_Vca_minus = 6;

parameter state_Vab_Vba_zero = 7;
parameter state_Vab_Vbc_zero = 8;
parameter state_Vab_Vcb_zero = 9;
parameter state_Vba_Vbc_zero = 10;
parameter state_Vba_Vcb_zero = 11;
parameter state_Vbc_Vcb_zero = 12;
parameter state_Vbc_Vca_zero = 13;
parameter state_Vbc_Vac_zero = 14;
parameter state_Vcb_Vca_zero = 15;
parameter state_Vcb_Vac_zero = 16;
parameter state_Vca_Vac_zero = 17;
parameter state_Vca_Vab_zero = 18;
parameter state_Vca_Vba_zero = 19;
parameter state_Vac_Vab_zero = 20;
parameter state_Vac_Vba_zero = 21;

/* V_ab */
parameter state_Vab_ba_p_1 = 22;
parameter state_Vab_ba_p_2 = 23;
parameter state_Vab_ba_p_3 = 24;
parameter state_Vab_ba_n_1 = 25;
parameter state_Vab_ba_n_2 = 26;
parameter state_Vab_ba_n_3 = 27;

parameter state_Vab_bc_p_1 =28;
parameter state_Vab_bc_p_2 =29;
parameter state_Vab_bc_p_3 =30;
parameter state_Vab_bc_n_1 =31;
parameter state_Vab_bc_n_2 =32;
parameter state_Vab_bc_n_3 =33;

parameter state_Vab_cb_p_1 =34;
parameter state_Vab_cb_p_2 =35;
parameter state_Vab_cb_p_3 =36;
parameter state_Vab_cb_n_1 =37;
parameter state_Vab_cb_n_2 =38;
parameter state_Vab_cb_n_3 =39;

parameter state_Vba_bc_p_1 = 40;
parameter state_Vba_bc_p_2 = 41;
parameter state_Vba_bc_p_3 = 42;
parameter state_Vba_bc_n_1 = 43;
parameter state_Vba_bc_n_2 = 44;
parameter state_Vba_bc_n_3 = 45;

parameter state_Vba_cb_p_1 = 46;
parameter state_Vba_cb_p_2 = 47;
parameter state_Vba_cb_p_3 = 48;
parameter state_Vba_cb_n_1 = 49;
parameter state_Vba_cb_n_2 = 50;
parameter state_Vba_cb_n_3 = 51;
/* V_bc */
parameter state_Vbc_cb_p_1 = 52;
parameter state_Vbc_cb_p_2 = 53;
parameter state_Vbc_cb_p_3 = 54;
parameter state_Vbc_cb_n_1 = 55;
parameter state_Vbc_cb_n_2 = 56;
parameter state_Vbc_cb_n_3 = 57;

parameter state_Vbc_ca_p_1 = 58;
parameter state_Vbc_ca_p_2 = 59;
parameter state_Vbc_ca_p_3 = 60;
parameter state_Vbc_ca_n_1 = 61;
parameter state_Vbc_ca_n_2 = 62;
parameter state_Vbc_ca_n_3 = 63;

parameter state_Vbc_ac_p_1 = 64;
parameter state_Vbc_ac_p_2 = 65;
parameter state_Vbc_ac_p_3 = 66;
parameter state_Vbc_ac_n_1 = 67;
parameter state_Vbc_ac_n_2 = 68;
parameter state_Vbc_ac_n_3 = 69;

parameter state_Vcb_ca_p_1 = 70;
parameter state_Vcb_ca_p_2 = 71;
parameter state_Vcb_ca_p_3 = 72;
parameter state_Vcb_ca_n_1 = 73;
parameter state_Vcb_ca_n_2 = 74;
parameter state_Vcb_ca_n_3 = 75;

parameter state_Vcb_ac_p_1 = 76;
parameter state_Vcb_ac_p_2 = 77;
parameter state_Vcb_ac_p_3 = 78;
parameter state_Vcb_ac_n_1 = 79;
parameter state_Vcb_ac_n_2 = 80;
parameter state_Vcb_ac_n_3 = 81;


/* V_ca */
```

```
parameter state_Vca_ac_p_1 = 82;
parameter state_Vca_ac_p_2 = 83;
parameter state_Vca_ac_p_3 = 84;
parameter state_Vca_ac_n_1 = 85;
parameter state_Vca_ac_n_2 = 86;
parameter state_Vca_ac_n_3 = 87;

parameter state_Vca_ab_p_1 = 88;
parameter state_Vca_ab_p_2 = 89;
parameter state_Vca_ab_p_3 = 90;
parameter state_Vca_ab_n_1 = 91;
parameter state_Vca_ab_n_2 = 92;
parameter state_Vca_ab_n_3 = 93;

parameter state_Vca_ba_p_1 = 94;
parameter state_Vca_ba_p_2 = 95;
parameter state_Vca_ba_p_3 = 96;
parameter state_Vca_ba_n_1 = 97;
parameter state_Vca_ba_n_2 = 98;
parameter state_Vca_ba_n_3 = 99;

parameter state_Vac_ab_p_1 = 100;
parameter state_Vac_ab_p_2 = 101;
parameter state_Vac_ab_p_3 = 102;
parameter state_Vac_ab_n_1 = 103;
parameter state_Vac_ab_n_2 = 104;
parameter state_Vac_ab_n_3 = 105;

parameter state_Vac_ba_p_1 = 106;
parameter state_Vac_ba_p_2 = 107;
parameter state_Vac_ba_p_3 = 108;
parameter state_Vac_ba_n_1 = 109;
parameter state_Vac_ba_n_2 = 110;
parameter state_Vac_ba_n_3 = 111;

parameter freq_c = 1649; // changed this because 0 adds ↘
   →another
parameter DELAY_COUNT = 10;
parameter DELAY_COUNT0 = 30;
parameter FLUX_ADJ = 60; //60

/* Registers */
reg [7:0] mc_state = 1;
reg [8:0] delay_counter = 0; /* 2^9 = 512 counts */
reg     tick = 0;
reg     d = 1;
reg section = 0;  //  This variable holds wether in first or ↘
   →second section of square wave.

always @(posedge clk)
 begin
  if (counter == freq_c)
  section <= ~section;
 end



always @(posedge clk) begin

if (fault == 1)
 begin
 Q <= 12'b000000000000;
 mc_state <= state_all_off;
 end
else
```

```
 begin

case (mc_state)
 default:
  begin
   Q <= 12'b000000000000;
  end
 state_all_off:
  begin
  Q <= 12'b000000000000;
  end

 state_Vab_plus: /* STATE V_ab */
  begin
   Q <= 12'b110000001100;
   if (counter == freq_c-FLUX_ADJ  && tick == 0 && section == ↘
   →0)//only allows neg adjustment
    begin
     tick <= 1;
     if (current_direction == POS_CUR)
      mc_state <= state_Vab_ba_p_1;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vab_ba_n_1;
     else
      mc_state <= state_Vab_Vba_zero;
    end
   else if (counter == freq_c  && tick == 1 && section == 1)
    begin
    d <= vbc_pol;
    tick <= 0;
    if ( vbc_pol == 1)
     begin
     if (current_direction == POS_CUR)
      mc_state <= state_Vab_bc_p_1;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vab_bc_n_1;
     else
      mc_state <= state_Vab_Vbc_zero;
     end
    else
     begin
     if (current_direction == POS_CUR)
      mc_state <= state_Vab_cb_p_1;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vab_cb_n_1;
     else
      mc_state <= state_Vab_Vcb_zero;
     end
    end
  end

 state_Vab_minus: /* STATE V_ba */
  begin
   Q <= 12'b001100110000;
   if (counter == freq_c-FLUX_ADJ && tick == 0 && section == ↘
   →0)
    begin
     tick <= 1;
     if (current_direction == POS_CUR)
      mc_state <= state_Vab_ba_p_3;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vab_ba_n_3;
     else
      mc_state <= state_Vab_Vba_zero;
    end
   else if (counter == freq_c  && tick == 1 && section == 1)
     begin
     d <= vbc_pol;
     tick <= 0;
     if ( vbc_pol == 1)
      begin
      if (current_direction == POS_CUR)
       mc_state <= state_Vba_bc_p_1;
      else if (current_direction == NEG_CUR)
       mc_state <= state_Vba_bc_n_1;
      else
       mc_state <= state_Vba_Vbc_zero;
      end
     else
      begin
      if (current_direction == POS_CUR)
       mc_state <= state_Vba_cb_p_1;
      else if (current_direction == NEG_CUR)
       mc_state <= state_Vba_cb_n_1;
      else
       mc_state <= state_Vba_Vcb_zero;
      end
     end
  end




 state_Vbc_plus: /* STATE V_bc */
  begin
```

```
Q <= 12'b001100000011;
if (counter == freq_c-FLUX_ADJ && tick == 0 && section == ↘
→0)
 begin
  tick <= 1;
  if (current_direction == POS_CUR)
   mc_state <= state_Vbc_cb_p_1;
  else if (current_direction == NEG_CUR)
   mc_state <= state_Vbc_cb_n_1;
  else
   mc_state <= state_Vbc_Vcb_zero;
 end
else if (counter == freq_c  && tick == 1 && section == 1)
  begin
  d <= vca_pol;
  tick <= 0;
  if ( vca_pol == 1)
   begin
   if (current_direction == POS_CUR)
    mc_state <= state_Vbc_ca_p_1;
   else if (current_direction == NEG_CUR)
    mc_state <= state_Vbc_ca_n_1;
   else
    mc_state <= state_Vbc_Vca_zero;
   end
  else
   begin
   if (current_direction == POS_CUR)
    mc_state <= state_Vbc_ac_p_1;
   else if (current_direction == NEG_CUR)
    mc_state <= state_Vbc_ac_n_1;
   else
```

```
      mc_state <= state_Vbc_Vac_zero;
     end
    end
 end
```

```
state_Vbc_minus: /* STATE V_cb */
 begin
  Q <= 12'b000011001100;
  if (counter == freq_c -FLUX_ADJ && tick == 0 && section == ↘
  →0)
   begin
    tick <= 1;
    if (current_direction == POS_CUR)
     mc_state <= state_Vbc_cb_p_3;
    else if (current_direction == NEG_CUR)
     mc_state <= state_Vbc_cb_n_3;
    else
     mc_state <= state_Vbc_Vcb_zero;
   end
  else if (counter == freq_c  && tick == 1 && section == 1)
    begin
    d <= vca_pol;
    tick <= 0;
    if ( vca_pol == 1)
     begin
     if (current_direction == POS_CUR)
      mc_state <= state_Vcb_ca_p_1;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vcb_ca_n_1;
     else
      mc_state <= state_Vcb_Vca_zero;
     end
    else
     begin
     if (current_direction == POS_CUR)
      mc_state <= state_Vcb_ac_p_1;
     else if (current_direction == NEG_CUR)
      mc_state <= state_Vcb_ac_n_1;
     else
      mc_state <= state_Vcb_Vac_zero;
     end
    end
 end



state_Vca_plus: /* STATE V_ca */
 begin
 Q <= 12'b000011110000;
 if (counter == freq_c -FLUX_ADJ && tick == 0 && section == 0)
  begin
  tick <= 1;
  if (current_direction == POS_CUR)
   mc_state <= state_Vca_ac_p_1;
  else if (current_direction == NEG_CUR)
   mc_state <= state_Vca_ac_n_1;
  else
```

```
   mc_state <= state_Vca_Vac_zero;
  end
 else if (counter == freq_c  && tick == 1 && section == 1)
  begin
  d <= vab_pol;
  tick <= 0;
  if ( vab_pol == 1)
   begin
   if (current_direction == POS_CUR)
    mc_state <= state_Vca_ab_p_1;
   else if (current_direction == NEG_CUR)
    mc_state <= state_Vca_ab_n_1;
   else
    mc_state <= state_Vca_Vab_zero;
   end
  else
   begin
   if (current_direction == POS_CUR)
    mc_state <= state_Vca_ba_p_1;
   else if (current_direction == NEG_CUR)
    mc_state <= state_Vca_ba_n_1;
   else
    mc_state <= state_Vca_Vba_zero;
   end
  end
 end




state_Vca_minus: /* STATE V_ac */
 begin
```

```
Q <= 12'b110000000011;
if (counter == freq_c-FLUX_ADJ && tick == 0 && section == 0)
 begin
  tick <= 1;
  if (current_direction == POS_CUR)
   mc_state <= state_Vca_ac_p_3;
  else if (current_direction == NEG_CUR)
   mc_state <= state_Vca_ac_n_3;
  else
   mc_state <= state_Vca_Vac_zero;
 end
else if (counter == freq_c  && tick == 1 && section == 1)
begin
d <= vab_pol;
tick <= 0;
if ( vab_pol == 1)
 begin
 if (current_direction == POS_CUR)
  mc_state <= state_Vac_ab_p_1;
 else if (current_direction == NEG_CUR)
  mc_state <= state_Vac_ab_n_1;
 else
  mc_state <= state_Vac_Vab_zero;
 end
else
 begin
 if (current_direction == POS_CUR)
  mc_state <= state_Vac_ba_p_1;
 else if (current_direction == NEG_CUR)
  mc_state <= state_Vac_ba_n_1;
 else
```

```
    mc_state <= state_Vac_Vba_zero;
   end
  end
 end




  /* ZERO STATES */
state_Vab_Vba_zero: /* STATE V_ab to V_ba all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vab_minus;
   else
    mc_state <= state_Vab_plus;
   end
 end
state_Vab_Vbc_zero: /* STATE V_ab to V_bc all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_plus;
   end
 end
state_Vab_Vcb_zero: /* STATE V_ab to V_cb all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_minus;
   end
 end
state_Vba_Vbc_zero: /* STATE V_ba to V_bc all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_plus;
   end
 end
state_Vba_Vcb_zero: /* STATE V_ba to V_cb all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_minus;
   end
 end
state_Vbc_Vcb_zero: /* STATE V_bc to V_cb all off */
```

```
 begin
  Q <= 12'b000000000000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT0)
    begin
    delay_counter <= 0;
    if ( d == 1 )
     mc_state <= state_Vbc_minus;
    else
     mc_state <= state_Vbc_plus;
    end
 end
state_Vbc_Vca_zero: /* STATE V_bc to V_ca all off */
 begin
  Q <= 12'b000000000000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT0)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_plus;
    end
 end
state_Vbc_Vac_zero: /* STATE V_bc to V_ac all off */
 begin
  Q <= 12'b000000000000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT0)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_minus;
    end
 end
```

```
state_Vcb_Vca_zero: /* STATE V_cb to V_ca all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_plus;
   end
 end
state_Vcb_Vac_zero: /* STATE V_cb to V_ac all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_minus;
   end
 end
state_Vca_Vac_zero: /* STATE V_ca to V_ac all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vca_minus;
   else
    mc_state <= state_Vca_plus;
   end
 end
state_Vca_Vab_zero: /* STATE V_ca to V_ab all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_plus;
   end
 end
state_Vca_Vba_zero: /* STATE V_ca to V_ba all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_minus;
   end
 end
state_Vac_Vab_zero: /* STATE V_ac to V_ab all off */
 begin
  Q <= 12'b000000000000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT0)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_plus;
   end
 end
state_Vac_Vba_zero: /* STATE V_ac to V_ba all off */
```

```
  begin
   Q <= 12'b000000000000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT0)
    begin
    delay_counter <= 0;
    mc_state <= state_Vab_minus;
    end
  end

/* ======================================== */
/* Vab state transitions */
/* ======================================== */


  /* Vab plus to Vab minus state transitions */

 state_Vab_ba_p_1: /* STATE V_ab to Vba Pos Cur Transition 1 ↘
   →*/
  begin
   Q <= 12'b100000000100;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
       if (d == 1)
         mc_state <= state_Vab_ba_p_2;
       else
         mc_state <= state_Vab_plus;
    end
  end
```

```
state_Vab_ba_p_2: /* STATE V_ab to Vba Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b101000010100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1)
      mc_state <= state_Vab_ba_p_3;
   else
      mc_state <= state_Vab_ba_p_1;
   end
 end
state_Vab_ba_p_3: /* STATE V_ab to Vba Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b001000010000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1)
      mc_state <= state_Vab_minus;
   else
     mc_state <= state_Vab_ba_p_2;
   end
 end
state_Vab_ba_n_1: /* STATE V_ab to Vba Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b010000001000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if (d == 1)
     mc_state <= state_Vab_ba_n_2;
   else
     mc_state <= state_Vab_plus;
   end
 end
state_Vab_ba_n_2: /* STATE V_ab to Vba Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b010100101000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1)
     mc_state <= state_Vab_ba_n_3;
   else
     mc_state <= state_Vab_ba_n_1;
   end
 end
state_Vab_ba_n_3: /* STATE V_ab to Vba Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000100100000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
```

```
    if ( d == 1 )
      mc_state <= state_Vab_minus;
    else
      mc_state <= state_Vab_ba_n_2;
    end
  end

/* NEW TRANSITIONS */
 state_Vab_bc_p_1:
  begin
  Q <= 12'b100000000100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_bc_p_2;
   end
  end
 state_Vab_bc_p_2:
  begin
  Q <= 12'b101000000101;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_bc_p_3;
   end
  end
 state_Vab_bc_p_3:
  begin
  Q <= 12'b001000000001;
  delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vbc_plus;
  end
 end
state_Vab_bc_n_1:
 begin
 Q <= 12'b010000001000;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_bc_n_2;
  end
 end
state_Vab_bc_n_2:
 begin
 Q <= 12'b010100001010;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_bc_n_3;
  end
 end
state_Vab_bc_n_3:
 begin
 Q <= 12'b000100000010;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vbc_plus;
  end
 end



state_Vab_cb_p_1:
 begin
 Q <= 12'b100000001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_cb_p_2;
  end
 end
state_Vab_cb_p_2:
 begin
 Q <= 12'b100010001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_cb_p_3;
  end
 end
state_Vab_cb_p_3:
 begin
 Q <= 12'b000001001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
```

```
  begin
  delay_counter <= 0;
  mc_state <= state_Vbc_minus;
  end
 end
state_Vab_cb_n_1:
 begin
 Q <= 12'b010000001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_cb_n_2;
  end
 end
state_Vab_cb_n_2:
 begin
 Q <= 12'b010001001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vab_cb_n_3;
  end
 end
state_Vab_cb_n_3:
 begin
 Q <= 12'b000001001100;
 delay_counter <= delay_counter + 1;
 if (delay_counter == DELAY_COUNT)
  begin
  delay_counter <= 0;
  mc_state <= state_Vbc_minus;
  end
 end


/* Vab minus to Vbc plus state transitions */

state_Vba_bc_p_1: /* STATE V_ba to Vbc Pos Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b001100010000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vba_bc_p_2;
   end
 end
state_Vba_bc_p_2: /* STATE V_ba to Vbc Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b001100010001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vba_bc_p_3;
   end
 end
state_Vba_bc_p_3: /* STATE V_ba to Vbc Pos Cur Transition 3 ↘
  →*/
 begin
```

```
  Q <= 12'b001100000001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_plus;
   end
 end
state_Vba_bc_n_1: /* STATE V_ba to Vbc Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b001100100000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vba_bc_n_2;
   end
 end
state_Vba_bc_n_2: /* STATE V_ba to Vbc Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b001100100010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vba_bc_n_3;
   end
 end
state_Vba_bc_n_3: /* STATE V_ba to Vbc Neg Cur Transition 3 ↘
  →*/
  begin
   Q <= 12'b001100000010;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_plus;
    end
  end


/* Vab minus to Vbc minus state transitions */

 state_Vba_cb_p_1: /* STATE V_ba to Vcb Pos Cur Transition 1 ↘
   →*/
  begin
   Q <= 12'b001000010000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vba_cb_p_2;
    end
  end
 state_Vba_cb_p_2: /* STATE V_ba to Vcb Pos Cur Transition 2 ↘
   →*/
  begin
   Q <= 12'b001010010100;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vba_cb_p_3;
    end
 end
state_Vba_cb_p_3: /* STATE V_ba to Vcb Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000010000100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_minus;
    end
 end
state_Vba_cb_n_1: /* STATE V_ba to Vcb Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000100100000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vba_cb_n_2;
    end
 end
state_Vba_cb_n_2: /* STATE V_ba to Vcb Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b000101101000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
    begin
```

```
   delay_counter <= 0;
   mc_state <= state_Vba_cb_n_3;
   end
 end
state_Vba_cb_n_3: /* STATE V_ba to Vcb Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000001001000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_minus;
   end
 end



/* Vbc plus to Vbc minus state transitions */

state_Vbc_cb_p_1: /* STATE V_bc to Vcb Pos Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b001000000001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vbc_cb_p_2;
   else
    mc_state <= state_Vbc_plus;
   end
 end
state_Vbc_cb_p_2: /* STATE V_bc to Vcb Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b001010000101;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vbc_cb_p_3;
   else
    mc_state <= state_Vbc_cb_p_1;
   end
 end
state_Vbc_cb_p_3: /* STATE V_bc to Vcb Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000010000100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vbc_minus;
   else
    mc_state <= state_Vbc_cb_p_2;
   end
 end
state_Vbc_cb_n_1: /* STATE V_bc to Vcb Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000100000010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1)
    mc_state <= state_Vbc_cb_n_2;
   else
    mc_state <= state_Vbc_plus;
   end
 end
state_Vbc_cb_n_2: /* STATE V_bc to Vcb Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b000101001010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vbc_cb_n_3;
   else
    mc_state <= state_Vbc_cb_n_1;
   end
 end
state_Vbc_cb_n_3: /* STATE V_bc to Vcb Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000001001000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
```

```
    begin
    delay_counter <= 0;
    if ( d == 1 )
     mc_state <= state_Vbc_minus;
    else
     mc_state <= state_Vbc_cb_n_2;
    end
 end

/* Vbc plus to Vca plus state transitions */

state_Vbc_ca_p_1: /* STATE V_bc to Vca Pos Cur Transition 1 ↘
   →*/
 begin
  Q <= 12'b001000000001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ca_p_2;
    end
 end
state_Vbc_ca_p_2: /* STATE V_bc to Vca Pos Cur Transition 2 ↘
   →*/
 begin
  Q <= 12'b001010010001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ca_p_3;
    end
 end
state_Vbc_ca_p_3: /* STATE V_bc to Vca Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000010000001;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_plus;
    end
 end
state_Vbc_ca_n_1: /* STATE V_bc to Vca Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000100000010;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ca_n_2;
    end
 end
state_Vbc_ca_n_2: /* STATE V_bc to Vca Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b000101100010;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ca_n_3;
```

```
     end
   end
  state_Vbc_ca_n_3: /* STATE V_bc to Vca Neg Cur Transition 3 ↘
    →*/
   begin
    Q <= 12'b000001100000;
    delay_counter <= delay_counter + 1;
    if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vca_plus;
     end
   end

 /* ======================================= */
 /* Vbc plus to Vca minus state transitions */
 /* ======================================= */
  state_Vbc_ac_p_1: /* STATE V_bc to Vac Pos Cur Transition 1 ↘
    →*/
   begin
    Q <= 12'b001000000011;
    delay_counter <= delay_counter + 1;
    if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vbc_ac_p_2;
     end
   end
  state_Vbc_ac_p_2: /* STATE V_bc to Vac Pos Cur Transition 2 ↘
    →*/
   begin
    Q <= 12'b101000000011;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ac_p_3;
    end
 end
state_Vbc_ac_p_3: /* STATE V_bc to Vac Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b100000000011;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_minus;
    end
 end
state_Vbc_ac_n_1: /* STATE V_bc to Vac Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000100000011;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vbc_ac_n_2;
    end
 end
state_Vbc_ac_n_2: /* STATE V_bc to Vac Neg Cur Transition 2 ↘
  →*/
 begin
```

```
  Q <= 12'b010100000011;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vbc_ac_n_3;
   end
 end
state_Vbc_ac_n_3: /* STATE V_bc to Vac Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b010000000011;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_minus;
   end
 end

/* Vbc minus to Vca plus state transitions */

state_Vcb_ca_p_1: /* STATE V_cb to Vca Pos Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000011000100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vcb_ca_p_2;
   end
 end
state_Vcb_ca_p_2: /* STATE V_cb to Vca Pos Cur Transition 2 ↘
  →*/
 begin
   Q <= 12'b000011010100;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vcb_ca_p_3;
     end
 end
state_Vcb_ca_p_3: /* STATE V_cb to Vca Pos Cur Transition 3 ↘
  →*/
 begin
   Q <= 12'b000011010000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vca_plus;
     end
 end
state_Vcb_ca_n_1: /* STATE V_cb to Vca Neg Cur Transition 1 ↘
  →*/
 begin
   Q <= 12'b000011001000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vcb_ca_n_2;
```

```
    end
  end
 state_Vcb_ca_n_2: /* STATE V_cb to Vca Neg Cur Transition 2 ↘
   →*/
  begin
   Q <= 12'b000011101000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vcb_ca_n_3;
    end
  end
 state_Vcb_ca_n_3: /* STATE V_cb to Vca Neg Cur Transition 3 ↘
   →*/
  begin
   Q <= 12'b000011100000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_plus;
    end
  end

/* Vbc minus to Vca minus state transitions */

 state_Vcb_ac_p_1: /* STATE V_cb to Vac Pos Cur Transition 1 ↘
   →*/
  begin
   Q <= 12'b000010000100;
   delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vcb_ac_p_2;
   end
 end
state_Vcb_ac_p_2: /* STATE V_cb to Vac Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b100010000101;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vcb_ac_p_3;
   end
 end
state_Vcb_ac_p_3: /* STATE V_cb to Vac Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b100000000001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_minus;
   end
 end
state_Vcb_ac_n_1: /* STATE V_cb to Vac Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000001001000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vcb_ac_n_2;
   end
 end
state_Vcb_ac_n_2: /* STATE V_cb to Vac Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b010001001010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vcb_ac_n_3;
   end
 end
state_Vcb_ac_n_3: /* STATE V_cb to Vac Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b010000000010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_minus;
   end
 end

//NEEDS DIRECTIONAL LOGIC
/* Vca plus to Vca minus state transitions */
```

```
state_Vca_ac_p_1: /* STATE V_ca to Vac Pos Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000010010000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vca_ac_p_2;
   else
    mc_state <= state_Vca_plus;
   end
 end
state_Vca_ac_p_2: /* STATE V_ca to Vac Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b100010010001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vca_ac_p_3;
   else
    mc_state <= state_Vca_ac_p_1;
   end
 end
state_Vca_ac_p_3: /* STATE V_ca to Vac Pos Cur Transition 3 ↘
  →*/
 begin
```

```
  Q <= 12'b100000000001;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vca_minus;
   else
    mc_state <= state_Vca_ac_p_2;
   end
 end
state_Vca_ac_n_1: /* STATE V_ca to Vac Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000001100000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   if ( d == 1 )
    mc_state <= state_Vca_ac_n_2;
   else
    mc_state <= state_Vca_plus;
   end
 end
state_Vca_ac_n_2: /* STATE V_ca to Vac Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b010001100010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
```

```
    delay_counter <= 0;
    if ( d == 1 )
     mc_state <= state_Vca_ac_n_3;
    else
     mc_state <= state_Vca_ac_n_1;
    end
  end
 state_Vca_ac_n_3: /* STATE V_ca to Vac Neg Cur Transition 3 ↘
   →*/
  begin
   Q <= 12'b010000000010;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    if ( d == 1 )
     mc_state <= state_Vca_minus;
    else
     mc_state <= state_Vca_ac_n_2;
    end
  end


/* Vca plus to Vab plus state transitions */

 state_Vca_ab_p_1: /* STATE V_ca to Vab Pos Cur Transition 1 ↘
   →*/
  begin
   Q <= 12'b000010010000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
   delay_counter <= 0;
   mc_state <= state_Vca_ab_p_2;
   end
 end
state_Vca_ab_p_2: /* STATE V_ca to Vab Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b100010010100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_ab_p_3;
   end
 end
state_Vca_ab_p_3: /* STATE V_ca to Vab Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b100000000100;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_plus;
   end
 end
state_Vca_ab_n_1: /* STATE V_ca to Vab Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000001100000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
```

```
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_ab_n_2;
    end
  end
 state_Vca_ab_n_2: /* STATE V_ca to Vab Neg Cur Transition 2 ↘
   →*/
  begin
   Q <= 12'b010001101000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vca_ab_n_3;
    end
  end
 state_Vca_ab_n_3: /* STATE V_ca to Vab Neg Cur Transition 3 ↘
   →*/
  begin
   Q <= 12'b010000001000;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vab_plus;
    end
  end

/* Vca plus to Vab minus state transitions */

 state_Vca_ba_p_1: /* STATE V_ca to Vba Pos Cur Transition 1 ↘
   →*/
 begin
  Q <= 12'b000010110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_ba_p_2;
   end
 end
state_Vca_ba_p_2: /* STATE V_ca to Vba Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b001010110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_ba_p_3;
   end
 end
state_Vca_ba_p_3: /* STATE V_ca to Vba Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b001000110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_minus;
   end
 end
```

```
state_Vca_ba_n_1: /* STATE V_ca to Vba Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b000001110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_ba_n_2;
   end
 end
state_Vca_ba_n_2: /* STATE V_ca to Vba Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b000101110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vca_ba_n_3;
   end
 end
state_Vca_ba_n_3: /* STATE V_ca to Vba Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b000100110000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_minus;
   end
 end


/* Vca minus to Vab plus state transitions */

state_Vac_ab_p_1: /* STATE V_ac to Vab Pos Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b110000000001;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vac_ab_p_2;
    end
 end
state_Vac_ab_p_2: /* STATE V_ac to Vab Pos Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b110000000101;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vac_ab_p_3;
    end
 end
state_Vac_ab_p_3: /* STATE V_ac to Vab Pos Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b110000000100;
   delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_plus;
   end
 end
state_Vac_ab_n_1: /* STATE V_ac to Vab Neg Cur Transition 1 ↘
  →*/
 begin
  Q <= 12'b110000000010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vac_ab_n_2;
   end
 end
state_Vac_ab_n_2: /* STATE V_ac to Vab Neg Cur Transition 2 ↘
  →*/
 begin
  Q <= 12'b110000001010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vac_ab_n_3;
   end
 end
state_Vac_ab_n_3: /* STATE V_ac to Vab Neg Cur Transition 3 ↘
  →*/
 begin
  Q <= 12'b110000001000;
```

```
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vab_plus;
    end
  end

/* Vca minus to Vab minus state transitions */
 state_Vac_ba_p_1: /* STATE V_ac to Vba Pos Cur Transition 1 ↘
   →*/
  begin
   Q <= 12'b100000000001;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vac_ba_p_2;
    end
  end
 state_Vac_ba_p_2: /* STATE V_ac to Vba Pos Cur Transition 2 ↘
   →*/
  begin
   Q <= 12'b101000010001;
   delay_counter <= delay_counter + 1;
   if (delay_counter == DELAY_COUNT)
    begin
    delay_counter <= 0;
    mc_state <= state_Vac_ba_p_3;
    end
  end
```

```
state_Vac_ba_p_3: /* STATE V_ac to Vba Pos Cur Transition 3 */
 begin
  Q <= 12'b001000010000;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vab_minus;
   end
 end
state_Vac_ba_n_1: /* STATE V_ac to Vba Neg Cur Transition 1 */
 begin
  Q <= 12'b010000000010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vac_ba_n_2;
   end
 end
state_Vac_ba_n_2: /* STATE V_ac to Vba Neg Cur Transition 2 */
 begin
  Q <= 12'b010100100010;
  delay_counter <= delay_counter + 1;
  if (delay_counter == DELAY_COUNT)
   begin
   delay_counter <= 0;
   mc_state <= state_Vac_ba_n_3;
   end
   end
  state_Vac_ba_n_3: /* STATE V_ac to Vba Neg Cur Transition 3 */
   begin
    Q <= 12'b000100100000;
    delay_counter <= delay_counter + 1;
    if (delay_counter == DELAY_COUNT)
     begin
     delay_counter <= 0;
     mc_state <= state_Vab_minus;
     end
   end
endcase

end // End of the if statement

end
endmodule
```

Appendix E

H-Bridge Module

This module is implemented in Verilog and creates the switching signals for the H-bridge, given the three duty ratios and the phase shift presented in Chapter 5.

```
module H_Bridge_v3(
 input clk,
 input signed [25:0] counter,
 input signed [11:0] vab_duty_ratio_count_in,
 input signed [11:0] vbc_duty_ratio_count_in,
 input signed [11:0] vca_duty_ratio_count_in,
 input signed [10:0] phase_shift_count,
 input fault,
 output reg [11:0] Q = 0
 );
/* H Bridge States */
parameter hb_alloff = 1;
parameter hb_Vplus = 2;
parameter hb_Vminus = 3;
parameter hb_Vdzero = 4;
parameter hb_Vnzero = 0;
parameter hb_AVd = 5;
parameter hb_BVd = 6;
parameter hb_AVn = 7;
parameter hb_BVn = 8;

/* Fixed Parameters */
parameter HB_DELAY_COUNT = 20; /* Dead time *//* 20clks * 20ns↘
   → = 400ns */
parameter OP_FREQ = 1650;

/* registers */
reg [4:0] hb_state = 0;
reg [8:0] hb_delay_counter = 0;
reg signed [15:0] trans1=−1,trans2=−1,trans3=−1,trans4=−1,↘
   →transl=−1,transr = −1, trans_nextframe =−1;
reg signed [11:0] vab_duty_ratio_count = 0,↘
   →vbc_duty_ratio_count = 0,vca_duty_ratio_count = 0;

reg [1:0] section = 1;
reg update= 1;
reg start_bit = 1;

/* These are the saturation blocks for the duty ratios, make ↘
   →sure the values are nice */
always @(vab_duty_ratio_count_in)
 begin
 if (vab_duty_ratio_count_in > 800)
  vab_duty_ratio_count <= 800;
 else if (vab_duty_ratio_count_in < 20)
  vab_duty_ratio_count <= 20;
 else
  vab_duty_ratio_count <= vab_duty_ratio_count_in;
 end
always @(vbc_duty_ratio_count_in)
```

```
 begin
 if (vbc_duty_ratio_count_in > 800)
  vbc_duty_ratio_count <= 800;
 else if (vbc_duty_ratio_count_in < 20)
  vbc_duty_ratio_count <= 20;
 else
  vbc_duty_ratio_count <= vbc_duty_ratio_count_in;
 end
always @(vca_duty_ratio_count_in)
 begin
 if (vca_duty_ratio_count_in > 800)
  vca_duty_ratio_count <= 800;
 else if (vca_duty_ratio_count_in < 20)
  vca_duty_ratio_count <= 20;
 else
  vca_duty_ratio_count <= vca_duty_ratio_count_in;
 end

/* Generates the frame transistions , updates section counter ↘
   →*/
always @(posedge clk)
 begin
 if (start_bit == 1)
   begin
   trans1 <=800;
   trans2 <=900;
   trans3 <=2450;
   trans4 <=2550;
   end
 else begin
 if (counter == 2*OP_FREQ)
  begin
```

```
transl <= trans_nextframe;

if (section == 3) /* All calculations done for the Vab frame↘
   → */
 begin
 /* Sets value for next frame transition, this only happens ↘
    →when we overlap to the right */
 if (phase_shift_count > vab_duty_ratio_count)
  trans_nextframe <= phase_shift_count - ↘
     →vab_duty_ratio_count;
 else
  trans_nextframe <= -1;
 /* Determines if there is a extra transition in this frame,↘
    → happens when overlap to the left */
 if (phase_shift_count + vbc_duty_ratio_count <= 0)
  transr <= 2*OP_FREQ + phase_shift_count + ↘
     →vbc_duty_ratio_count;
 else
  transr <= -1;
 /* Calculates all transistions for this frame */
 trans1 <= vab_duty_ratio_count + phase_shift_count;
 trans2 <= OP_FREQ - vab_duty_ratio_count + ↘
    →phase_shift_count;
 trans3 <= OP_FREQ + vab_duty_ratio_count + ↘
    →phase_shift_count;
 trans4 <= (OP_FREQ*2) - vab_duty_ratio_count + ↘
    →phase_shift_count;
 end
```

```
if (section == 1) /* All calculations done for the Vbc frame↘
  → */
 begin
 /* Sets value for next frame transition, this only happens ↘
    →when we overlap to the right */
 if (phase_shift_count > vbc_duty_ratio_count)
  trans_nextframe <= phase_shift_count - ↘
     →vbc_duty_ratio_count;
 else
  trans_nextframe <= -1;
 /* Determines if there is a extra transition in this frame,↘
    → happens when overlap to the left */
 if (phase_shift_count + vca_duty_ratio_count <= 0)
  transr <= 2*OP_FREQ + phase_shift_count + ↘
     →vca_duty_ratio_count;
 else
  transr <= -1;
 /* Calculates all transistions for this frame */
 trans1 <= vbc_duty_ratio_count + phase_shift_count;
 trans2 <= OP_FREQ - vbc_duty_ratio_count + ↘
    →phase_shift_count;
 trans3 <= OP_FREQ + vbc_duty_ratio_count + ↘
    →phase_shift_count;
 trans4 <= (OP_FREQ*2) - vbc_duty_ratio_count + ↘
    →phase_shift_count;
 end
if (section == 2) /* All calculations done for the Vca frame↘
  → */
 begin
 /* Sets value for next frame transition, this only happens ↘
    →when we overlap to the right */
 if (phase_shift_count > vca_duty_ratio_count)
```

```
  trans_nextframe <= phase_shift_count - ↘
     →vca_duty_ratio_count;
 else
  trans_nextframe <= -1;
 /* Determines if there is a extra transition in this frame,↘
    → happens when overlap to the left */
 if (phase_shift_count + vab_duty_ratio_count <= 0)
  transr <= 2*OP_FREQ + phase_shift_count + ↘
     →vab_duty_ratio_count;
 else
  transr <= -1;

 /* Calculates all transistions for this frame */
 trans1 <= vca_duty_ratio_count + phase_shift_count;
 trans2 <= OP_FREQ - vca_duty_ratio_count + ↘
    →phase_shift_count;
 trans3 <= OP_FREQ + vca_duty_ratio_count + ↘
    →phase_shift_count;
 trans4 <= (OP_FREQ*2) - vca_duty_ratio_count + ↘
    →phase_shift_count;
 end
end


/* This code just updates the frame/section number */
if (counter == OP_FREQ*2)
begin
if (section == 3)
 section <= 1;
else
 section <= section + 1;
end
```

```
  end

 end //end always block

/* State Logic */

always @(posedge clk)
 begin
 if (fault == 1)
  hb_state <= hb_alloff;
 else if (start_bit == 1)
  begin
  start_bit <= 0;
  hb_state <= hb_Vnzero;
  end
 else
  begin
  case (hb_state)
  default:
   begin
   hb_state <= hb_Vnzero;
   end

  hb_Vminus:
   begin
   if (counter == transl) /* This catches an transition that ↘
      →came from last frame */
    begin
    hb_state <=hb_AVn;
    end
   else if (counter == trans4)
  begin
  hb_state <=hb_AVn;
  end
 end

hb_Vnzero:
 begin
 if (counter == trans1)
  begin
  hb_state <=hb_BVn;
  end
 else if (counter == transr) /* This catches an transition ↘
    →that is from the next frame */
  begin
  hb_state <=hb_BVn;
  end
 else if (counter > trans1 && counter < 1650 ) /* This is ↘
    →for the duty decreasing glitch where transition ↘
    →happens in one frame moves in the next */
  begin
  hb_state <=hb_BVn;
  end
 end

hb_Vplus:
 begin
 if (counter == trans2)
  begin
  hb_state <=hb_AVd;
  end
 end
```

```
hb_Vdzero:
 begin
 if (counter == trans3)
  begin
  hb_state <=hb_BVd;
  end
 end

hb_AVd:
 begin
 hb_delay_counter <= hb_delay_counter + 1;
 if (hb_delay_counter == HB_DELAY_COUNT)
  begin
  hb_delay_counter <= 0;
  hb_state <= hb_Vdzero;
  end
 end

hb_BVd:
 begin
 hb_delay_counter <= hb_delay_counter + 1;
 if (hb_delay_counter == HB_DELAY_COUNT)
  begin
  hb_delay_counter <= 0;
  hb_state <= hb_Vminus;
  end
 end

 hb_AVn:
 begin
 hb_delay_counter <= hb_delay_counter + 1;
 if (hb_delay_counter == HB_DELAY_COUNT)
```

```
    begin
    hb_delay_counter <= 0;
    hb_state <= hb_Vnzero;
    end
   end

   hb_BVn:
   begin
   hb_delay_counter <= hb_delay_counter + 1;
   if (hb_delay_counter == HB_DELAY_COUNT)
    begin
    hb_delay_counter <= 0;
    hb_state <= hb_Vplus;
    end
   end

  endcase /*end case statement */

  end /*end if*/
 end /*always*/


/* State Machine for H bridge */
/* Output Logic */

always @(posedge clk) begin

if (fault == 1)
 begin
 Q <= 12'b000000000000;
 end
else
```

```
begin

case (hb_state)
default:
 begin
 Q <= 12'b000000000000;
 end

hb_alloff:
 begin
 Q <= 12'b000000000000;
 end

hb_Vplus:
 begin
 Q <= 12'b000100000010;
 end

hb_Vminus:
 begin
 Q <= 12'b000001001000;
 end

 hb_Vdzero:
 begin
 Q <= 12'b000101000000; //Vd zero
 end

 hb_Vnzero:
 begin
 Q <= 12'b000000001010; //Vn zero
 end
```

```
    hb_AVd:
    begin
    Q <= 12'b000100000000; //A Vd
    end

    hb_BVd:
    begin
    Q <= 12'b000001000000;//B Vd
    end

    hb_AVn:
    begin
    Q <= 12'b000000001000;//A Vn
    end

    hb_BVn:
    begin
    Q <= 12'b000000000010;//B Vn
    end

  endcase// END OF STATE MACHINE
  end // END of if for fault

end // End of always state machine

endmodule
```