Adaptive Control of a Step-Up Full-Bridge DC-DC Converter for Variable Low Input Voltage Applications

Elton Pepa

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Approved:

_____
Dr. Jason Lai (Chair)

_____ _____
Dr. Krishnan Ramu Dr. William T. Baumann

February 6, 2004
Blacksburg, Virginia

Keywords: adaptive control, phase shift modulation, full bridge, converter

Adaptive Control of a Step-Up Full-Bridge DC-DC Converter for Variable Low Input Voltage Applications

By
Elton Pepa
Electrical Engineering

Abstract

This thesis shows the implementation of a novel control scheme DC-DC converter. The converter is a phase-shifted full-bridge PWM converter that is designed to operate as a front stage of a power conversion system where the input is a variable low voltage high current source. The converter is designed to step-up the low voltage input to an acceptable level that can be inverted to a 120/240 V_{AC} 60Hz voltage for residential power. A DSP based adaptive control model is developed, taking into account line variations introduced by the input source while providing very good load dynamics for the converter in both discontinuous and continuous conduction modes. The adaptive controller is implemented using two voltage sensors that read the input and the output voltages of the converter. The controller's bandwidth is comparable to current mode control, without the need for an expensive current sensor, yet providing the noise immunity seen in voltage mode controllers. The intended input source was a fuel cell but in its absence a DC supply is utilized instead. The system is simulated for both discontinuous and continuous conduction modes and implemented and demonstrated for the continuous conduction mode. The test results are shown to match the simulation results very closely.

Acknowledgements

First and foremost, I want to express my sincere gratitude to my advisor, Dr. Jason Lai, for his guidance, encouragement, and support during my graduate studies. His practical way of teaching, his impressive knowledge, technical skills, and creative thinking have been a source of inspiration throughout the course of my work.

I am grateful to my other committee members, Dr. Krishnan Ramu and Dr. William T. Baumann. Dr. Ramu has shown his cheer, support, and advice to me from the moment I considered starting my graduate studies. The lectures, long conversations with him, and the guidance he has given me are invaluable. Dr. Baumann's teachings of control systems have been some best lectures I have ever received during my entire school studies.

My work would have not been possible without the help of a number of my fellow students. I would especially like to thank Jerry Francis for his support and help with the DSP work, testing, and great friendship. I am extremely grateful to: Chris Smith, Damian Urciuoli, Andy Mclandrich, Amy Johnson, Mike Gilliom, Mike Schenck, and Joel Gouker, all of whom I spent countless late nights and early mornings working on the FEC competition; Xudong Huang, who was my partner in the Delphi project; and other FEEC students including Changrong Liu, Huijie Yu, Junhong Zhang. There are also some past and present CPES students and friends including Lincoln, Troy, Jeremy, Leonard, Carl, Daniel, Doug and Sebastian that are deserving of my appreciation. Additionally, I would like to thank my good friends from the Albanian Club at Virginia Tech, who to some extent have eased the longing I have for my family and friends back home.

Most importantly, I would like to thank my parents and my brother for everything that they have done for me throughout my life and the joy they have brought to me for just having their support. I only hope that what I have accomplished can pay somewhat for the efforts for raising and making me the person that I am today.

Finally, I would like to express a great amount of gratitude to my girlfriend, Carrie, for her companionship and for being the most supportive.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Motivation

The development of new useful energy sources is the key to continued industrial progress, and the continual improvement in the world's standard of living. Discovering new sources of energy, obtaining an essentially inexhaustible energy source, making it available everywhere, and converting it from one form to another without polluting and destroying the environment are some of the great challenges in the world today.

This work was motivated by the work that was done by Virginia Tech team during the Future Energy Challenge (FEC) competition for 2003 [1]. FEC is a competition between universities around the country and abroad primarily supported by the Department of Energy (DOE) and the Department of Defense (DOD). With the energy use rising every year and as the country becomes more dependent on foreign oil, it is in the focus of DOE to develop alternative renewable sources.

One of the competitions in FEC was the Fuel Cell Inverter challenge, with the objective to develop a low-cost power processing system that supports the commercialization of a 5 kW solid-oxide fuel cell (SOFC) power generation system to provide non-utility and ultra clean residential electricity.

One of the objectives of the VT team was to complete a control system that would meet the requirements set by the competition. However, due to time restrictions, a complete control system design was not implemented by the time the competition was held.

The focus of this study was to design a high performance controller for a front-end DC-DC converter to be used in a variable low voltage input power conversion system similar to the FEC requirements. Since at the time of FEC competition and when the work in this thesis was conducted a SOFC was not available, a variable DC supply was utilized instead. The control method presented here is an individual effort to complete the work from the FEC, surpassing the given requirements.

1.2 Objective and Outline

The focus of this thesis is in the conversion of an unregulated low voltage high current input source to a useful source of energy that can be used directly by consumers. As mentioned before, the intended primary energy source from the FEC [1] was a solid oxide fuel cell which in this thesis was replaced by a DC supply that can vary the input voltage in a fashion similar to a fuel cell. In the FEC, the variable DC input voltage was to be converted to two single phase 120 V_{AC} outputs that can be combined to produce a 240V_{AC}, 60 Hz output. Such conversion was done in two stages. First, the nominal voltage of 28V_{DC} input from the DC source is boosted to a 200V_{DC} bus voltage that can be inverted by a single phase inverter to the desired 120/240V_{AC} outputs. The main requirements for the FEC competition are listed in Table 1 [2].

Table 1. Main design requirements for FEC 2003

Design Item	**Target Requirement**
1. Manufacturing Cost	Less then US $40/kW for the 10 kW design in high volume production
2.Output Power Capability-nominal	5 kW continuous
3. Output Power capability - overload	10 kW overload for 1 minute (5kW from fuel cell)
4. Output Voltage	120 V/ 240 V nominal
5. Input Source (SOFC)	22-41 VDC, 29 V nominal
6. Galvanic isolation	Galvanic isolation of the system. The common neutral point of the ac output phases must be earth ground
7. Overall Efficiency	> 90%

To achieve these cost and efficiency targets, the hardware had to be designed from the bottom up, eliminating expensive components with cost-effective solutions. This energy management system has three main parts: the front-end DC-DC converter, the AC inverter and energy storage elements. The system block diagram designed for the FEC 2003 is shown in Figure 1.1. This report focuses on the front end DC-DC which is an important link between the low voltage high current unregulated input voltage source and the 60 Hz inverter.

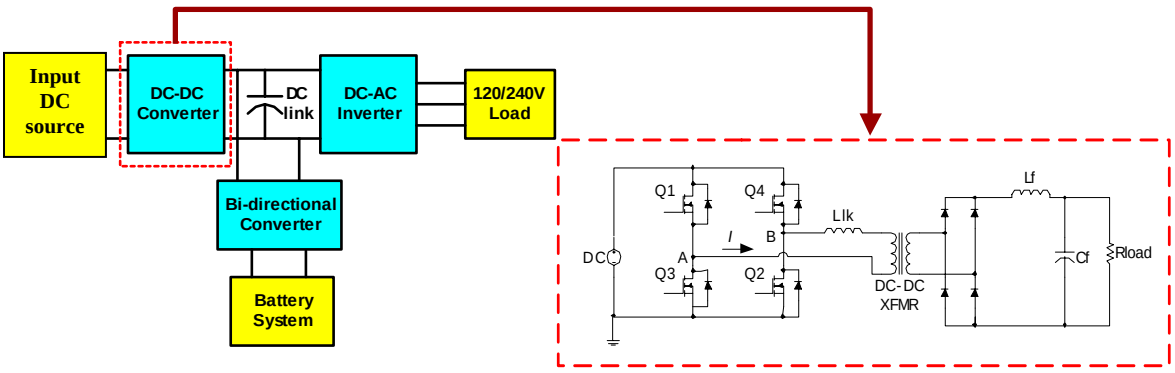

Figure 1.1. Basic block diagram of the power conversion system

The solid-oxide fuel cell that was used as the basis for designing the DC-DC converter, has a voltage-current characteristic, also known as the polarization curve, given in Figure 1.2

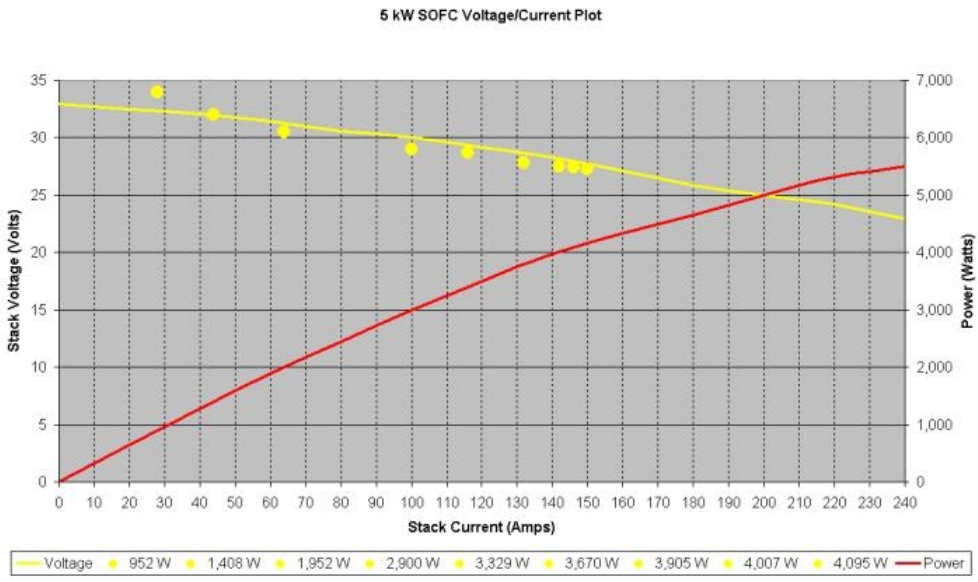

Figure 1.2. Voltage to current plot of a solid oxide fuel cell (SOFC)

According to the power plot Figure 1.2 , the front end DC-DC converter is required to operate at low-voltage high-current at heavy load, and high-voltage-low current at light load conditions. In order to provide a bus voltage large enough for a 120/240V inverter, the input voltage must be boosted significantly. For this application, an isolated topology is favored for converter optimization [3]. Isolated converter topologies provide

advantages in applications requiring large voltage conversion ratios. Transformer isolation can reduce switch and diode device stresses and allows multiple windings or taps to be used to for multiple converter outputs. The full-bridge is a popular design for both buck and boost applications and has become a basis for numerous resonant zero voltage and zero current switching (ZVS, ZCS) schemes. Often in high power applications a phase shift modulation (PSM) switching scheme is used to achieve ZVS and/or ZCS transitions through the interaction of converter parasitic energy storage elements [4]. Another advantage for using the full bridge converter is the fact that when higher power application are requested the full bridge converter can act as a modular block and that it is possible to stack up [5]. For this purpose the chosen topology for the converter to be used in this application is a Full bridge phase shifted PWM converter. The DC-DC converter discussed herein provides a favorable method of boosting the low DC voltage high current input DC for an inverter input. For sufficient inverter input voltage margin, a 200 V converter output was desired for the entire fuel cell input voltage range. To meet these criteria, the converter consists of a switching stage, a transformer stage, and an output rectifier and filter stage

Since the input source is unregulated, the DC-DC converter has to take into account these changes and act appropriately without affecting the output. The output from the DC-DC converter creates the DC link bus voltage. The DC link bus voltage created by the converter needs to be stiff because it is being used by single phase inverter and a bi-directional converter. The converter's output should respond to the load dynamics and still maintain good regulation.

The rest of this chapter gives an overview of the system and different control methods that are used in power converter designs and lays some of the foundations of the adaptive control model that is developed in this thesis. Chapter 2 presents an overview of the design options and discusses in more detail the challenges that arise when a high performance controller is required for the DC-DC converter in the power conversion system application. In chapter 3 the models for the chosen converter topology and the control scheme are developed. This includes the design of a closed loop control for the full bridge converter based on an adaptive model reference scheme that has a very good disturbance rejection and bandwidths comparable to current mode control methods. In

this chapter, the design of the controller is simulated under different load and input voltage conditions. Next, in chapter 4 the design of the hardware and software implementation is discussed. Experimental results are taken and the performance of the converter is compared with the predicted simulation results. Finally, a summary of the design is given, the conclusions are drawn, and future work ideas are presented.

1.3 Background Information

In this section a literature review, on the basic operation of the fuel cells, converters and the control techniques most commonly used is provided. In addition a special section is dedicated to adaptive control from which much of the thesis was based upon.

1.3.1 Fuel Cells

Fuel cells are ideal for distributed power generation applications, environmentally friendly, efficient and desirable for remote locations and developing countries as a mean of providing power at a low cost. Fuel cells are classified by the electrolyte that they employ and the temperature of operation. Solid oxide fuel cells (SOFC) [6] operate at high temperature and the oxygen from air is the oxidant (cathode fuel). SOFCs are solid-state devices operating at temperatures up to 1000°C. They can use a wider choice of fuels and there is no requirement to manage liquid electrolytes. Current is conducted by the movement of oxygen ions through a solid electrolyte. At the cathode, oxygen is reduced to form oxygen ions; at the anode, the transported oxygen ions react with the gaseous fuel to produce water and free electrons for the external circuit. In the case of SOFCs, the temperature has to be high enough to enable oxygen ions to defuse through the electrolyte, which is made possible by the presence of oxygen vacancies in the crystalline structure of the electrolyte. The efficiency of SOFC is in the 50-60%, and although still at a relatively early stage of development, they are regarded as the most promising for generating electricity from hydrocarbon fuels [6].

A fuel cell system consists of the following main stages:

- Fuel Processing
- Water management
- Temperature Control
- Power Conditioning

The power conditioning that is made up of power electronics is the focus in this thesis report. The fuel cell output is in the form of direct current (DC) and the output voltage depends on the stack size. In many applications, alternating current (AC) at higher voltages is required; therefore fuel cell output is transformed from DC to AC by means of power electronics. Different circuit topologies were considered for this application and the design was done in three stages:

1. DC-DC converter
2. DC-AC inverter
3. Bidirectional battery charger

Since the SOFC is used as an input source, the input voltage into the DC-DC converter is 22-31 V_{DC} at a 28 V_{DC} nominal voltage, and a maximum current of 275 A from the fuel cell. The whole system needs to provide galvanic isolation from the input to output for the purpose of safety [2]. Given that the input voltage is low and the required voltage of the whole system is 120V/240 V_{AC} nominal, the main purpose of the DC-DC converter is to provide a sufficient DC link voltage which will be inverted into AC by the inverter.

1.3.2 Control of DC-DC Converters

The structure and complexity of the converter control depends on specific application requirements. A cascaded loop control structure with an inner current loop and a superimposed voltage loop is used as a standard control in DC-DC converters to provide high performance, wide bandwidth output voltage regulation [7].

In all switching converters the output voltage v(t) is a function of input line voltage $v_g(t)$, duty cycle d(t), and the load current $i_{load}(t)$ as well as the converter circuit element values. In DC-DC converter applications it is desired to obtain a constant output voltage v(t) = V in spite of the disturbances in vg(t) and $i_{load}(t)$ (Figure 1.3), and in spite of variations in the converter circuit element values.

The unknown and unmeasurable variations of the process parameters degrade the performance of the control system. Feedback is used in conventional control systems to reject the effect of the disturbances upon the controlled variables and to bring them back to their desired values. To achieve this, first the controlled variables are measured then

compared with the desired values and the difference is fed into controller which will appropriately control these variables to meet the desired specifications.

1.3.3 Voltage Mode Control

A feedback loop can be constructed for regulation of the output voltage. The output voltage v(t) is compared to a reference voltage Vref, to generate an error signal [3, 8]. This error signal is applied to the input of a compensation network, and the output of the compensator drives the control signal d(t) as shown in Figure 1.3. In the case of a full bridge converter, the control signal d(t) is given to a full bridge controller IC which in this case is implemented by the Texas Instruments UCC3895 [9]. This control IC generates the appropriate turn on signals for all four switches in the pattern explained in chapter 2.

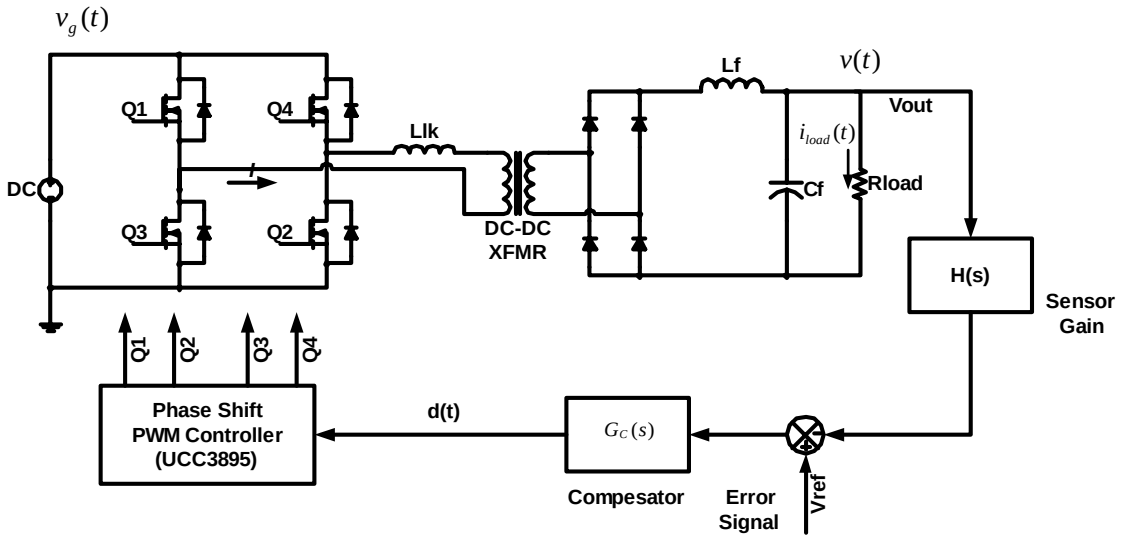

Figure 1.3. Voltage Mode Control

In voltage mode control any change in the source or load is only detected after it has propagated to the output. This slows the control, especially when the source voltage changes as in the case when the input source is a SOFC.

1.3.4 Current Mode Control

Another control scheme that finds a wide application is current mode control. In this type of control the inductor current is used as a feedback state [10]. However the current and voltage reference values are not independent and this specifically requires knowledge of the load [7]. To overcome this obstacle the voltage error signal is used to generate a current error signal as shown in Figure 1.4 [11]. Using the current creates the drawback that the knowledge of how the current affects the voltage is needed. For example, in a resistive system, $v=ir$, and increasing v will increase i. However, in a constant power load, increasing v will decrease i to maintain the relationship $iv=p$, where p is typically constant. This means that the current reference should be smaller for larger voltages. This dependency in undesired, and is a drawback of this type of control. Misrepresenting the load can lead to decreased performance, and possibly instability.

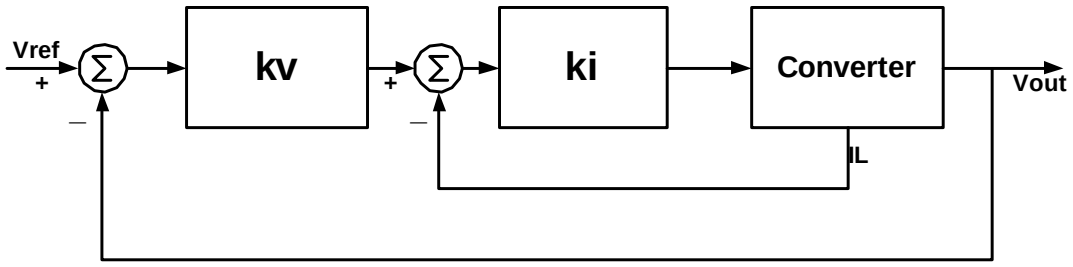

Figure 1.4. Current Mode Control

Current mode control requires knowledge of the inductor current, which is controlled via the inner loop. The outer loop manages the output voltage error by commanding the necessary current. The inner loop makes the converter act as a current source. There are many schemes that deal with current mode control. There are many methods to use to do current mode control, such as peak current mode control [12], average current mode control [13], sensorless current mode control [14]. In this thesis a brief overview of the main schemes is described.

1.3.5 Peak Current Mode Control

Figure 1.5 shows the generic inductor current of a switching converter operating in continuous conduction mode (CCM). The inductor current changes with a slope m_1 during the first subinterval, and a slope $-m_2$ during the second subinterval.

At D > 0.5 there is an inherent instability which is not dependent of the converter topology [3]. The controller can be made stable for all duty cycles by the addition of an artificial ramp with a slope M_a to the sensed current waveform. When $M_a \geq 0.5 \cdot m_2$, then the controller is stable for all duty cycles. The controller now switches the transistor off when this summation crosses the reference value i_{ref}, as shown in Figure 1.5. [12]. The relationship between the ramp, inductor and reference current is given in **(1)**

$$i_a(dT) + i_L(dT) = i_{ref} \tag{1}$$

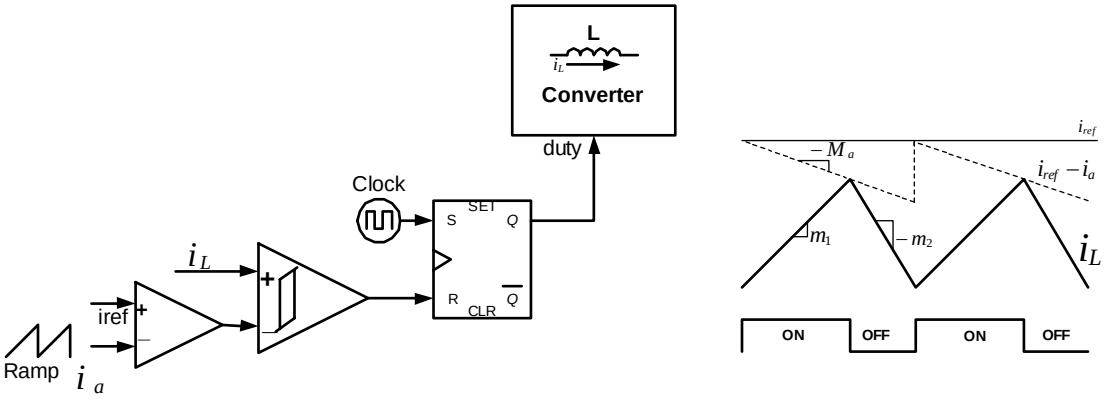

Figure 1.5. Peak current mode control

Some benefits of peak current mode control are:

- Control of the peak inductor current
- Inherent current limiting and sharing
- Good dynamics and performance

Some of the limitations are:

- Limited accuracy in controlling the average inductor current (especially in the DCM case) (The peak and average inductor current are not related)
- Increased sensitivity to line variations
- Increased switching noise problems

1.3.6 Average Current Mode Control

In the average current mode control (ACM) a compensator is added to make the average inductor current track a reference as shown in Figure 1.6 [13] . The triangle carrier waveform in this case remains in place. In this case the variations in the duty cycle are dependent on the value of the averaged current.

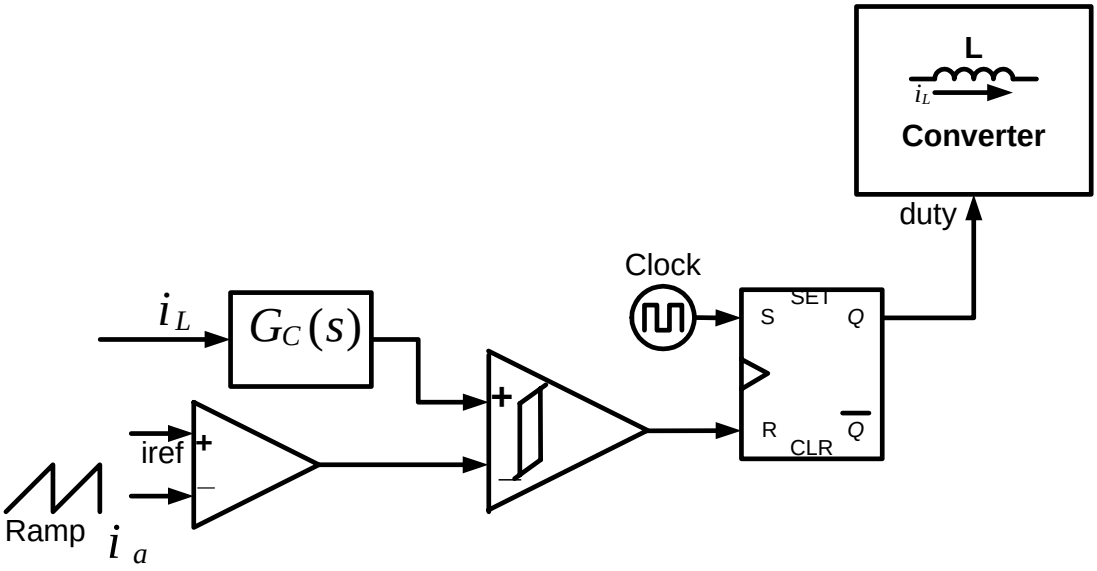

Figure 1.6. Average Current Mode Control (ACM)

1.3.7 Sensorless Current Mode Control

An alternative method for current mode control is to use an observer method. In an observer, a model of the system to be controlled is used in place of the system to provide estimates of the control state values. In the case of sensorless current mode control an observer state is used in the place of inductor current. Since the output should match the reference, the output is a command rather than a dynamic state. So the desired output reference value replaces the output state.

The SCM signal can be used as a direct substitute for peak current mode control. It shares the two key properties of the current mode controls and it provides a direct match to peak current mode control [15]:

- It compensates for changes in the input source
- It requires a stabilizing ramp to reach duty ratios above 50%.

This method, however, has its disadvantages:

- Since an integration step involves an arbitrary constant, the average DC current is not controlled
- For current limiting or current sharing some extra control is needed
- The general version requires an analog representation of the switches or other means to create the observer, as shown in Figure 1.7.

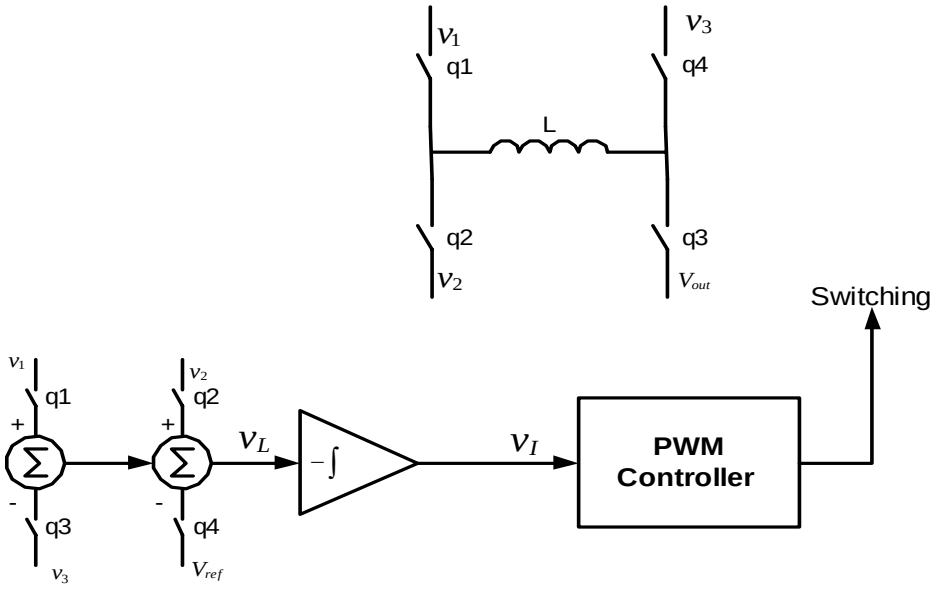

Figure 1.7. General SCM Process

In summary, the sensorless current mode (SCM) control uses the integrated inductor voltage as shown in Figure 1.8 in place of measured inductor current, and substitutes an intended command reference for the states intended to be fixed. The approach is based on an observer, which emulates the converter's operation. To implement the general form, low power switches are added to the PWM IC in order to support the method.

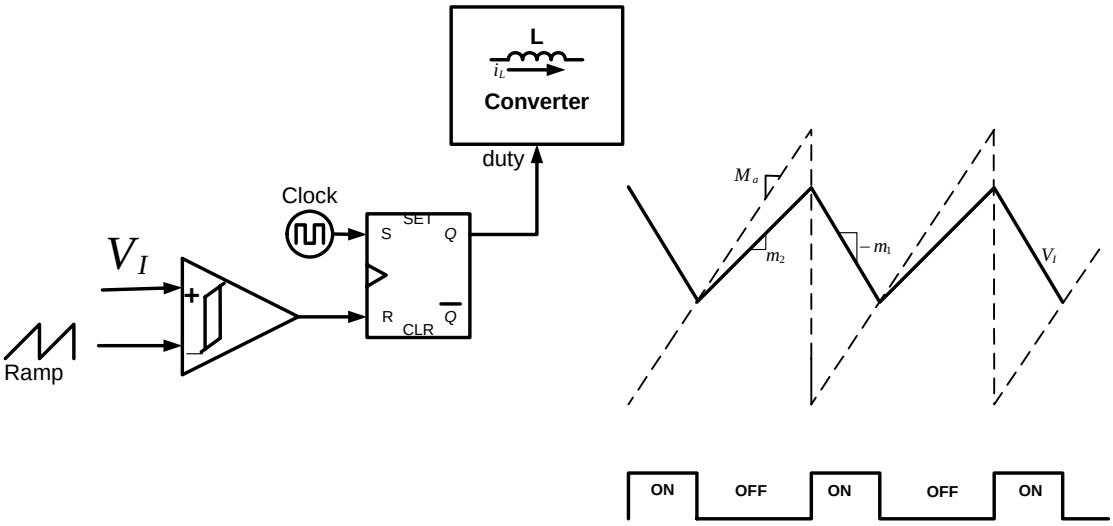

Figure 1.8. Sensorless Current Mode Control

1.3.8 Adaptive control

In the nature of the application, the input source varies with time and its response is highly dependent on external factors. For DC-DC converter to achieve a good output control independent of the line input and load type, other types of control techniques are studied. These methods are highly beneficial in the case when the parameters of the control process are poorly known or vary during normal operation. With PCM, ACM and SCM control techniques the inductor current/voltage needs to be measured and the selection of the ramp control signal needs to be done such that the controller is stable throughout the duty cycle range. In most DC-DC converters, the inductor current is a function of the load current. The converter's operation changes from continuous conduction mode CCM, to discontinuous conduction mode (DCM) when the inductor ripple current is higher than the average value. In DCM, the properties of the converter change radically. The conversion ratio, M, becomes load dependent and the output impedance increases. As a result the control to output transfer function (duty to output voltage) may be unknown when the load is removed.

The control method used in this work is a variation of the adaptive control techniques, which provide a systematic approach for automatic adjustment of the controllers in real time [16]. This method is preferable in order to achieve or maintain a desired level of performance of the control system when the parameters of the plant dynamic model are unknown and/change with time.

The plant (DC-DC converter) dynamic characteristics depend upon the load and the input line. In order to achieve and maintain acceptable level of performance when changes in the model occur, adaptive control has to be considered. In order to design a good controller the following is needed:

- Specification of the desired control loop performances
- Knowledge of the dynamics model of the plant to be controlled

Among various alternative methods, the technique known as adaptive model reference seems to be one of the most feasible approaches for this application [17]. It should be noted that the control system under consideration is an adjustable dynamic system in the sense that its performance can be adjusted by modifying the parameters of the controller or the control signal. The difference between the desired performance and the measured performance (the error) acts through an adaptation law to force the model to match the real system. A general block diagram of the adaptive control is shown in Figure 1.9

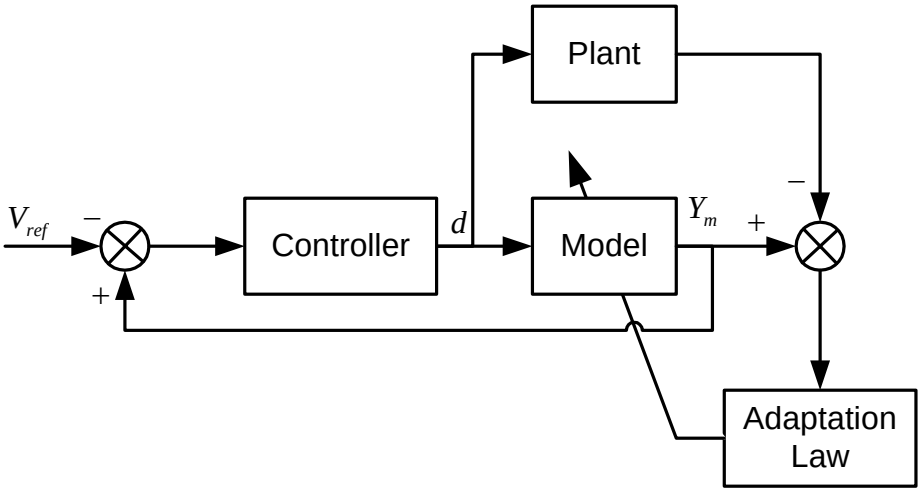

Figure 1.9. Basic configuration of an adaptive model

Chapter 2 - FB-ZVS Converter Model in DCM & CCM

2.1 Principle of operation

In order to reduce the size and the weight of magnetic components it is desirable to increase the switching frequency for DC-DC converters. When conventional PWM converters are operated at high frequencies, the circuit parasitics have negative effects on the converter performance [18]. Switching losses increase in high power applications and snubbers and/or other means of protection are required, which introduce significant losses and lower the efficiency. In the case of the conventional full bridge converter, the diagonally opposite switches (Q1 and Q2, or Q3 and Q4) are turned on and off simultaneously as shown in Figure 2.1.

In the FB-PWM converter, when all four switches are turned off, the load current freewheels through the rectifier diodes [19]. In this case the energy stored in the leakage inductance of the power transformer causes severe ringing with MOSFET junction capacitances. This creates the need for using snubbers that increase the overall losses bringing down the efficiency. If snubbers are not used, the selection of the devices becomes more difficult as the voltage rating for these switches has to be much higher. As the voltage rating goes up, so do the conduction losses and as a result the overall losses increase. At the same time the cost increases as well.

In order to minimize the parasitic ringing, the gate signals of Q2 and Q4 are delayed (phase-shifted) with respect to those of Q1 and Q3 [20], as shown in Figure 2.2, so that the primary of the transformer is either connected to the input voltage or shorted. The leakage inductance current is never interrupted, thus solving the problem of parasitic ringing associated with the conventional full-bridge PWM converter. The energy stored in the leakage inductance can be used to discharge the energy stored in the MOSFET junction capacitances to achieve zero voltage switching (ZVS) conditions for all four switches in the primary side. In this case, the converter requires no additional resonant components.

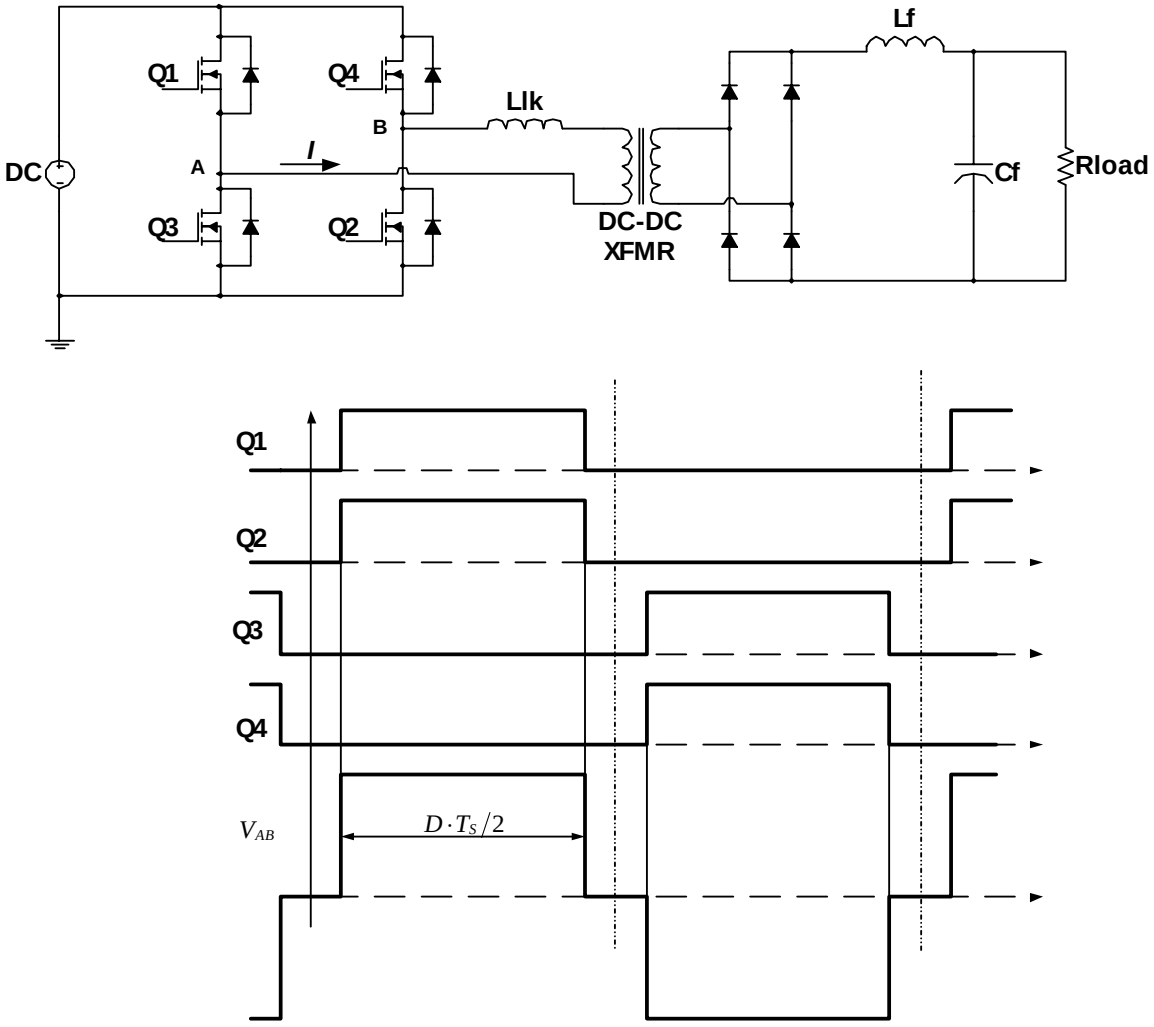

Figure 2.1. Conventional FB-PWM Converter

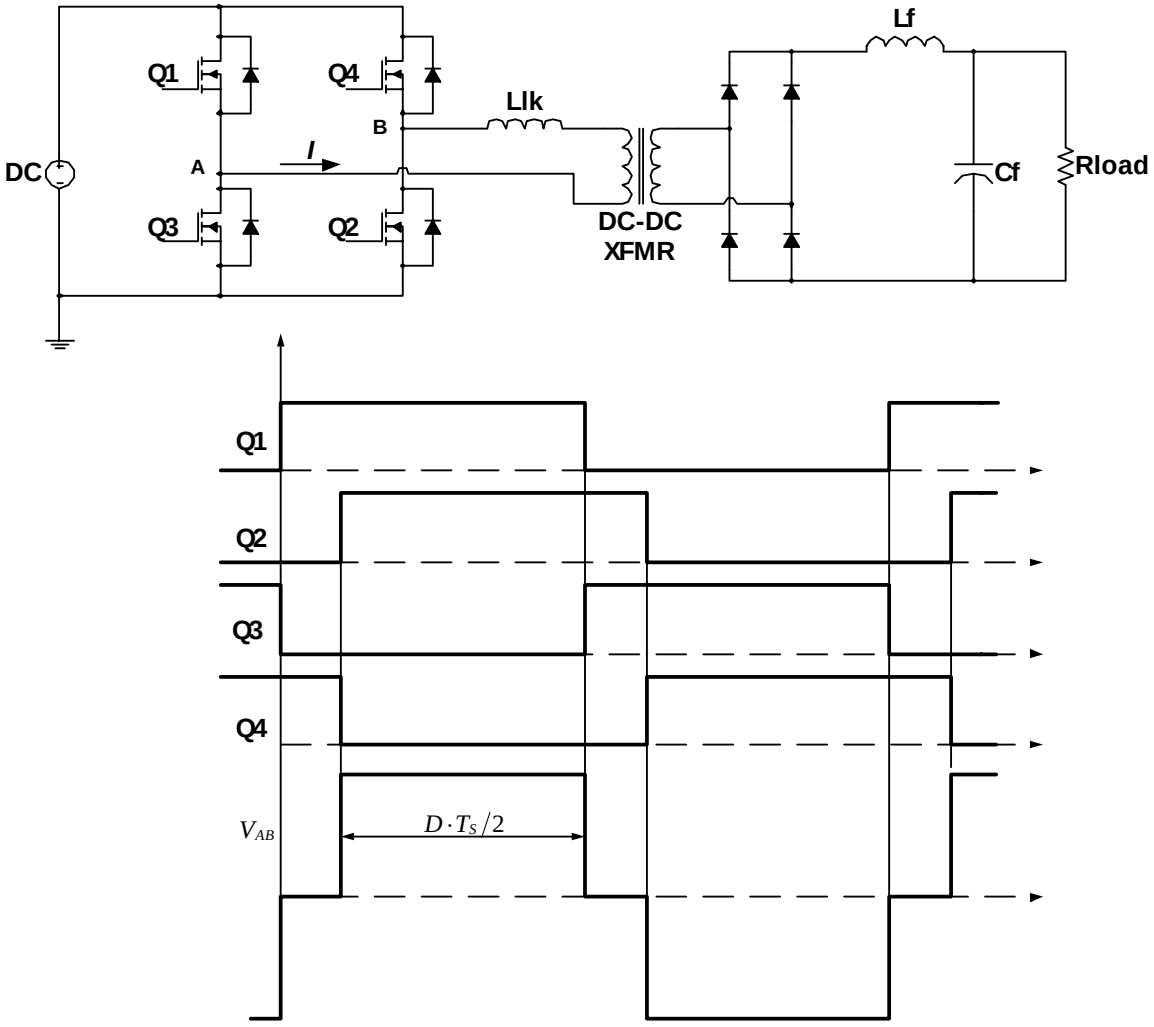

Figure 2.2. Phase Shifted FB-PWM Converter

2.2 Converter Analysis

The FB-ZVS-PWM converter provides ZVS for all four switches in the bridge. However the mechanism by which ZVS is achieved is different for both legs of the bridge Figure 2.3. For transistors Q2 and Q4, the ZVS is provided by the resonance between the leakage inductance, L_{LK} and the output capacitance of the switch.

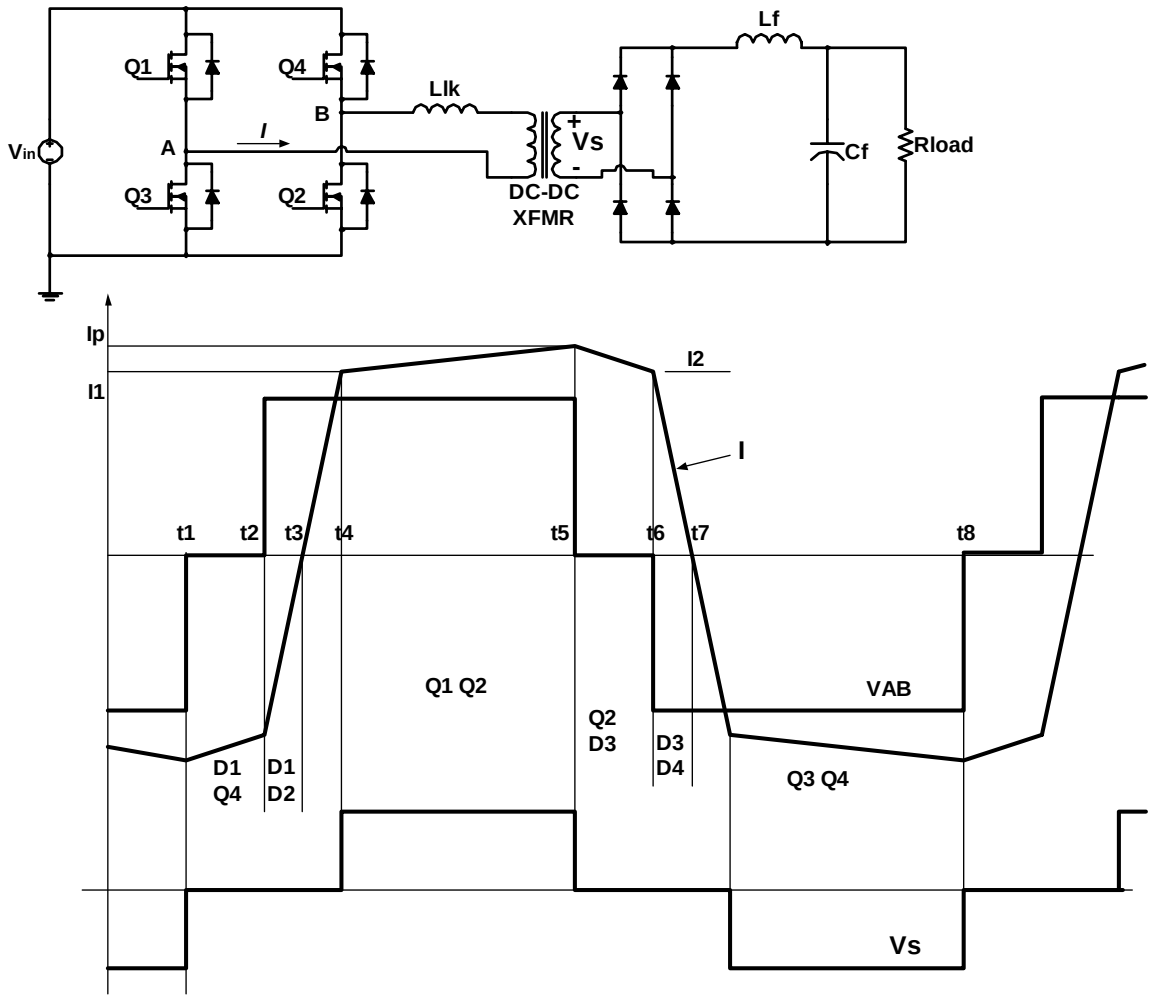

Figure 2.3. Principle of Operation of Phase Shifted FB-PWM Converter

The needed energy for achieving ZVS is given in .

$$E = \frac{1}{2} \cdot L_{LK} \cdot I_2^{\,2} \geq \frac{4}{3} \cdot C_{mos} \cdot V_{in}^{\,2} + \frac{1}{2} \cdot C_{TR} \cdot V_{in}^{2} \qquad (2)$$

In equation22 (2) I_2 is the current through the primary when Q_2 turns off, V_{in} is the input voltage C_{TR} is the transformer winding capacitance. The factor 4/3 is the two times the energy stored in the nonlinear drain to source capacitor, whose capacitance is inversely proportional to the square root of the voltage [20].

The resonance between L_{LK} , C_{mos} and C_{TR} provides a sinusoidal voltage across the capacitances that reaches a maximum at one fourth of the resonant frequency period.

The dead time between Q2 and Q4 has to be set at $\delta t_{\max}$ to ensure that there is sufficient time to charge and discharge the capacitances the dead time required to ensure ZVS with the maximum possible load range can be determined by the following equation (3):

$$\delta t_{\max} = \frac{T}{4} = \frac{\pi}{2}\sqrt{L_{LK} \cdot C} \tag{3}$$

Where $C = C_{mos} + C_{TR}$

Whether ZVS can be achieved for Q2 and Q4 is dependent on the load level of the converter. For light loads, the current though L_{LK} when Q2 and Q4 are turned off may not be enough to turn on the anti-parallel diode.

For switches Q1 and Q3, ZVS is provided by a different mechanism. Before Q1 is turned off the current in the primary is reaching its peak value. The primary current is the filter inductor current reflected to the primary. When Q1 is turned off the energy available to charge the output capacitance of Q1 and discharge the output capacitance of Q3 is the energy stored in L_{LK} and the energy in the output filter inductor. This energy in the output filter inductor is available because the filter inductor current does not freewheel through the rectifier until the voltage across the secondary has fallen to zero.

Since the energy in the filter inductor is large compared to the energy stored in the switch capacitances in the primary, the charging of the switches can be approximated by a linear charging with a constant current. Consequently, the dead time dt_1 required between the turn off of Q1 and turn on of Q3 can be determined from the equation (4):

$$dt_1 \cdot I_p = 4 \cdot C_{mos} \cdot V_{in} \tag{4}$$

Where $4 \cdot C_{mos} \cdot V_{in}$ corresponds to twice the charge stored in the nonlinear output capacitance of the MOSFET and I_p is the peak current in the output filter inductor reflected to the primary. The dead time can be calculated for the minimum I_p chosen to achieve ZVS. If load current is further reduced the ZVS property can not be maintained.

-Critical Current for Zero-Voltage Switching

The ZVS for Q1 and Q3 can be achieved even at light loads because D1 and D3 can always be turned on by the energy stored in the output filter inductance. However, ZVS for Q2 and Q4 can only be achieved for a load current above the critical values it is shown in equation (5):

$$I_{CRIT} = \sqrt{\frac{2}{L_{LK}} \cdot \left(\frac{4}{3} \cdot C_{MOS} \cdot V_{in}^2 + \frac{1}{2} \cdot C_{tr} \cdot V_{in}^2 \right)} \tag{5}$$

The available current through L_{LK} at t_2 can be calculated by:

$$I_2 = \frac{N_S}{N_P} \cdot \left(I_{load} + \frac{\Delta I}{2} - \frac{V_{OUT}}{L_{LK} + L_F} (1 - D) \frac{T}{2} \right) \tag{6}$$

Finally, ZVS is achieved for a load current so that $I_2 > I_{crit}$ which can be expressed as:

$$I_{load} \geq \frac{N_P}{N_S} \cdot I_{CRIT} - \frac{\Delta I}{2} + \frac{V_{OUT}}{L_{Lk} + L_F} \cdot (1 - D) \frac{T}{2} \tag{7}$$

The magnetizing current can only be used to achieve ZVS when the load current reflected to the primary is lower than the magnetizing current (i.e. at light loads) [20]. For such light loads the energy available to charge/discharge the output capacitances of the switches Q2 and Q4 at times t_2 and t_6 respectively (Figure 2.3) is the energy stored in the transformer's leakage inductance plus the energy stored in the leakage inductance of the transformer.

2.2.1 CCM and DCM Operation

DCM occurs with large inductor current ripple in a converter operating at light load and containing current-unidirectional switches. Since it is usually required that converters operate at no load, DCM is frequently encountered. The properties of the converter change radically in the discontinuous conduction mode. The conversion ratio M becomes load dependent and the output impedance is increased. Control of the output may be lost when the load is removed [3]. The average inductor current is related to

output voltage and resistance load by Ohms law in (8) and the inductor current ripple can be determined by equation (9)

$$I = \frac{V}{R} \tag{8}$$

$$\Delta i_L = \frac{(n \cdot V_g - V)}{2L} D \cdot T_S \tag{9}$$

The ripple magnitude depends on the applied voltage $(n \cdot V_g - V)$, the inductance value, and on the transistor conduction time DT_s. However, the ripple does not depend on the load resistance R. The inductor current ripple magnitude varies with the applied voltages rather than the applied currents. If the load resistance is increased so that the DC load current is decreased, the ripple magnitude Δi_L, will remain unchanged. If the load resistance increases there will be a point when $I = \Delta i_L$ is reached. As the load is decreased, the diode current can not be negative therefore the diode must become reverse biased before the end of the switching period. This is what is known as discontinuous conduction mode (DCM). The conditions for operation in the discontinuous conduction modes are given by equations (10) and (11):

$$I > \Delta i_L \quad \text{For CCM} \tag{10}$$

$$I < \Delta i_L \quad \text{For DCM} \tag{11}$$

Where I and Δi_L are found assuming that the converter operates in the continuous conduction mode.

In DCM: $$\frac{D \cdot n \cdot V_g}{R} < \frac{D \cdot D' \cdot T_S \cdot n \cdot V_g}{2 \cdot L} \tag{12}$$

Simplified, this leads to $$\frac{2 \cdot L}{R \cdot T_S} < D' \tag{13}$$

Equation (13) can be expressed as $K < K_{crit}(D)$ where $K = \frac{2 \cdot L}{R \cdot T_S}$ and $K_{crit}(D) = D'$. The converter goes into DCM when $R > R_{CRIT} = \frac{2 \cdot L}{(1-D) \cdot T_S}$ which is

dependent on duty cycle, inductance and switching period. The analysis will continue with the two known quantities based on the inductor volt-second balance and capacitor charge balance in equations (14) and (15) respectively [3].

From inductor volt second balance:

$$< v_L >= \frac{1}{T_S}\int_0^{T_S} v_L(t)\cdot dt = 0 \tag{14}$$

From capacitor charge balance:

$$< i_C >= \frac{1}{T_S}\int_0^{T_S} i_C(t)\cdot dt = 0 \tag{15}$$

Solving for v_L leads to finding the value of the output voltage as shown in (16)

$$V = n\cdot V_g \cdot \frac{D_1}{D_1 + D_2} \tag{16}$$

The inductor current value can be calculated as shown in equation (17) and replacing i_L with (8) the result is shown in (18).

$$i_L = \frac{D_1\cdot T_S}{2\cdot L}(D_1 + D_2)(n\cdot V_g - V) \tag{17}$$

$$\frac{V}{R} = \frac{D_1\cdot T_S}{2\cdot L}(D_1 + D_2)(n\cdot V_g - V) \tag{18}$$

$$\frac{2\cdot L}{R\cdot T_S} = \frac{D_1(D_1 + D_2)(n\cdot V_g - V)}{V} \tag{19}$$

Substituting V from (16) in equation (19) is derived in (20):

$$\frac{2\cdot L}{R\cdot T_S} = \frac{D_1^{\,2}\cdot n^2\cdot V_g^{\,2} - D_1^{\,2}\cdot n\cdot V_g\cdot V}{V^2} \tag{20}$$

Where $K = \frac{2\cdot L}{R\cdot T_S}$

The condition whether the converter is operating in DCM or CCM is dependent only on input voltage, output voltage and the duty cycle. The conversion ratio is derived in (21) and (22) for CCM and DCM operation respectively [3].

$$M = D \quad \text{For CCM} \tag{21}$$

$$M = \frac{2}{1+\sqrt{1+\frac{4\cdot K}{D^2}}} \quad \text{For DCM} \tag{22}$$

The dependency of coefficient K as a function of duty cycle for different input voltage values are shown in Figure 2.4 .

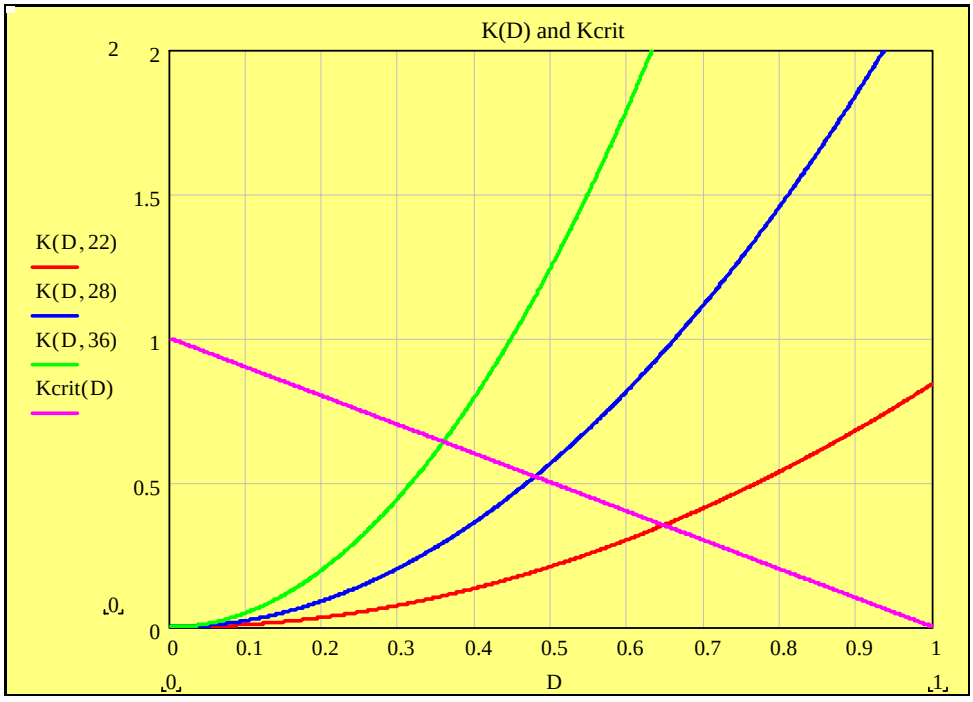

Figure 2.4. Coefficient k as a function of Duty Cycle

The behavior of modulation index as a function of duty cycle for different values of K is shown in Figure 2.5. For DCM case the modulation index is nonlinear (left of the plot) and at CCM it becomes linear.

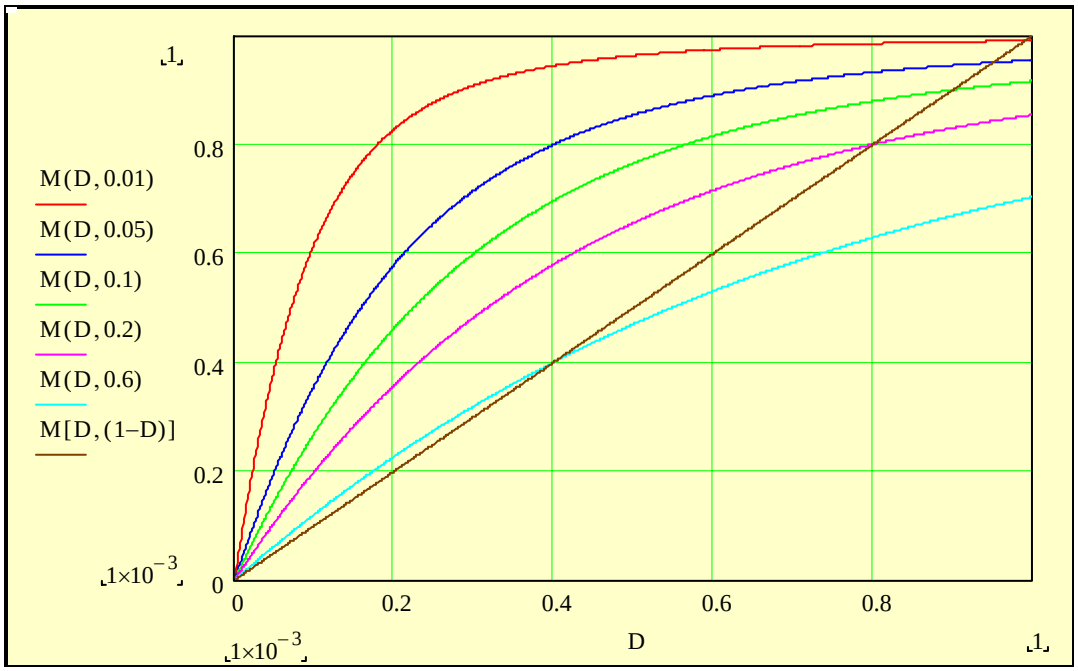

Figure 2.5. Modulation index vs. Duty cycle

As can be seen in Figure 2.4 and Figure 2.5, the modulation index varies nonlinearly when the converter works under DCM conditions. In order to maintain control of the output for the full-bridge converter, the operation point needs to be known. Equation (20) suggests that by measuring only input and output voltage we can predict in what mode the converter is working and make the necessary adjustments to the modulation index. The controller modeling will be based on this fact and is explained in detail in Chapter 3.

2.3 Modeling of the Converter

In order to design a controller to meet the given requirements the small signal model of the converter has to be built. Vlatcovic et.al [21] have developed a small signal model for a phase shift full bridge converter but this model is only valid for the CCM operation. The conventional full-bridge topology is derived from the buck converter, and therefore has a very similar small signal model. The phase-shifted version is also similar to the buck, but slightly more complex. The main difference comes from the effective duty cycle that the phase-shifted converter sees, which is shown in equation (23).

$$D_{eff} = D - \Delta D = D - \left(\frac{\frac{N_s}{N_p}}{\frac{V_{in}}{L_{lk}} \frac{T_s}{2}} (2I_L - \frac{V}{L_f} D' \frac{T_s}{2}) \right) \tag{23}$$

The above relationship suggests that the effective duty cycle is a function of leakage inductance, filter inductance and input and output voltages. Because the full bridge converter behaves just like a regular converter when working in DCM it is possible to design a model that can distinguish between these two modes of operation. To help this case Tsai [22] has developed an average model of the phase shifted PWM converter. This model was developed on the assumptions that all the switches are ideal, inductor ripple current is linear and that the ripple output voltage is zero.

The behavior of the full bridge converter changes radically when the converter's operation changes from CCM to DCM mode. The average switch model from which the model of the converter is derived takes into account this fact of operation point of the converter whether it's working on CCM or DCM mode. The converter's behavior in the discontinuous conduction mode (DCM) is the same as that of a conventional PWM converter since the linear charging/discharging interval of the leakage inductor current no longer exists as illustrated in Figure 2.6.

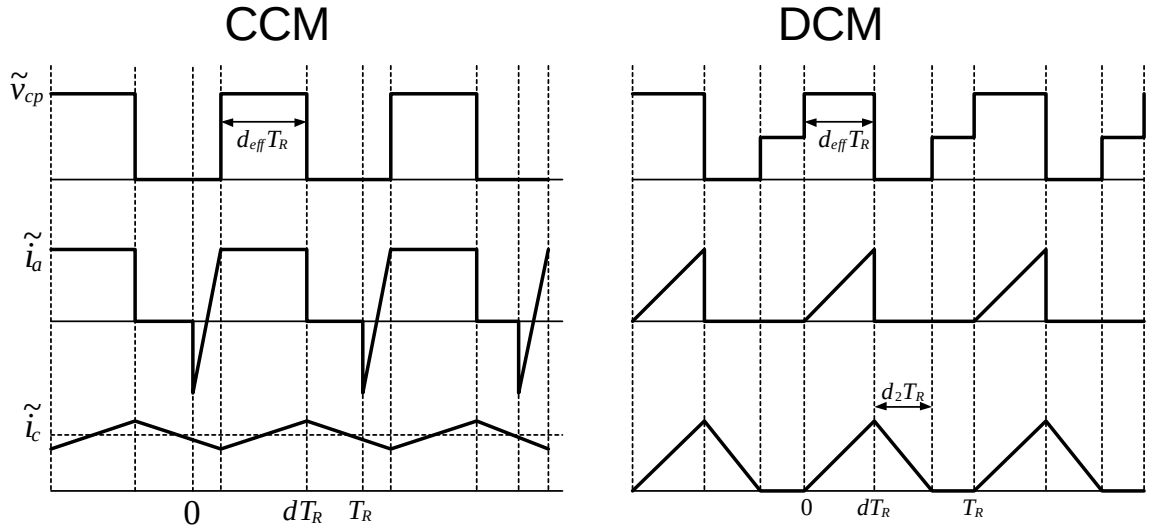

Figure 2.6. Averaged Switch Model terminal quantities for FB-PWM converter

Under the assumptions made above in CCM the relationships are:

$$\begin{cases} \tilde{v}_{cp} = 0, \\ \tilde{i}_a = \dfrac{-i_C}{n} + \dfrac{\tilde{v}_{ap} \cdot t}{L_{LK}} \end{cases} \quad \text{,during } [0,\, d_L T_R] \tag{24}$$

$$\begin{cases} \tilde{v}_{cp} = \dfrac{\tilde{v}_{ap}}{n}, \\ \tilde{i}_a = \dfrac{-i_C}{n} \end{cases} \quad \text{,during } [d_L T_R,\, dT_R] \tag{25}$$

$$\begin{cases} \tilde{v}_{cp} = 0, \\ \tilde{i}_a = 0 \end{cases} \quad \text{,during } [dT_R,\, T_R] \tag{26}$$

Where i_C is the averaged value of $\tilde{i}_c$. Variables without an "$\sim$" indicate the averaged value of the corresponding quantities over a ripple cycle T_R.

The relationship in DCM can be summarized as:

$$\begin{cases} \tilde{v}_{cp} = \dfrac{\tilde{v}_{ap}}{n}, \\ \tilde{i}_a = \dfrac{\tilde{i}_C}{n} \end{cases}, \quad \text{during } [0,\, dT_R] \tag{27}$$

$$\begin{cases} \tilde{v}_{cp} = 0, \\ \tilde{i}_a = 0 \end{cases}, \quad \text{during } [dT_R,\, (d + d_2) \cdot T_R] \tag{28}$$

$$\begin{cases} \tilde{v}_{cp} = v_{cp}, \\ \tilde{i}_a = 0 \end{cases}, \quad \text{during } [(d + d_2) \cdot T_R,\, T_R] \tag{29}$$

From the derivations above the three terminal averaged switch model is derived and shown in Figure 2.7.

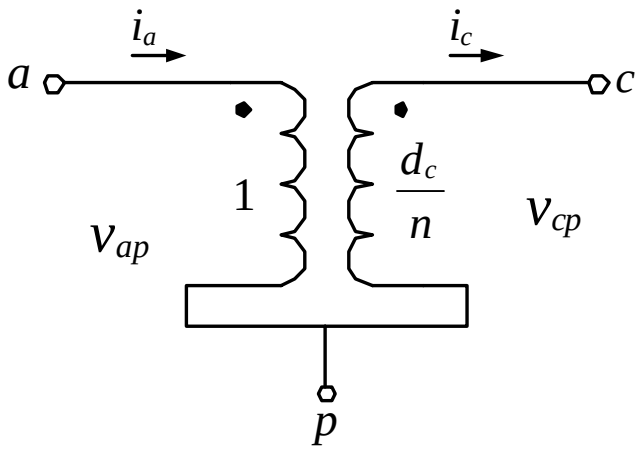

Figure 2.7. Three-terminal averaged ZVS PWM Switch Model

In Figure 2.6, $d_c = \frac{(d - d_L)}{(d + d_2)}$, where $(d + d_2)=1$ for CCM and $d_L = 0$ for DCM case and d_L is the duty cycle loss

Averaging for one ripple cycle:

$$\begin{cases} v_{cp} = \dfrac{v_{ap}(d - d_L)}{n} \\ i_a = \dfrac{i_c(d - d_L)}{n} \end{cases} \text{, for CCM where } d_L = \frac{2 \cdot L_{LK} \cdot i_C \cdot f_r}{n \cdot v_{ap}} \tag{30}$$

$$\begin{cases} v_{cp} = \dfrac{v_{ap} \cdot d}{n \cdot (d + d_2)} \\ i_a = \dfrac{i_c d}{n \cdot (d + d_2)} \end{cases} \text{, for DCM where } d_2 = \frac{2 \cdot L_F \cdot i_C \cdot f_r \cdot m}{d \cdot v_{ap}} \tag{31}$$

Using all the relationships shown above the average model of the full-bridge phase shifted PWM converter was derived and implemented in Saber® and shown in Figure 3.3.

Chapter 3 - Controller Design and Converter Simulation

This chapter lays the foundation of the work in this thesis report. The chapter starts with the average modeling of the dc-dc converter in both discontinuous and continuous conduction modes (DCM and CCM) of operation. In order to accurately design and control the DC-DC converter it must first be modeled and simulated. From the results of these simulations it is possible to predict the performance of the converter and its controller before building the hardware and make sure that all the selected components meet the requirements.

3.1 Controller Design

In a fuel cell system the performance of the DC-DC converter is very dependent on the unknown variations of process parameters. These changes will degrade the performance of the converter and as a result it is desirable that the controller not be affected from these disturbances.

Although the operation of the phase shifted full bridge converter is well known, its behavior is dependent on some factors that are not known well. Such parameters have to do with the nonlinear behavior of the converter itself. The transformers turns ratio, the coupling coefficient, the leakage inductance of the transformer, and the series resistance of the filter inductor are some of the parameters that can not be controlled closely and they vary with the load conditions, input source, and operating frequency.

All these factors could be controlled to a certain degree with relatively easy controllers, but if a tight performance is expected and the output voltage from the converter is required to be stiff, the controller has to be fast and noise immune and should have good disturbance rejection. As discussed in the introduction section, the current control methods are fast but are highly sensitive to noise. On the other hand, a voltage mode controller is noise immune, but usually its performance is slow and it does not correct until the disturbance has already taken effect.

Sensorless current control (SCM) offers these advantages but it is not practical in the case of higher power rating converters and in the case transformers are present. The SCM method has been implemented only for low level power converters where the

sensor isolation is not needed. When a high power inverter is used and the output voltage is higher than that of the DSP voltage levels, the need for isolation is a must.

The SCM method also requires a very fast voltage sensor if the method used is identical to the SCM method used by Krein [14, 15]. This sensor is also needed in order to detect whether the converter is working in DCM or CCM mode, as the integration of the inductor voltage waveform needs to be precise if the SCM method is used.

The control model that is developed in this thesis is a combination of the SCM technique developed by Krein [14, 15] and an adjustable model reference control [16, 17]. An average reference model is first developed and a parameter adaptation law is added into the control loop in order to account for the nonlinearities in the converter. This controller has very good disturbance rejection and still maintains speeds comparable to current mode control methods.

In this case the voltage sensors do not need to measure high frequency switching values. Instead we can measure just slow changing DC values and build our model based on these values. The discussion of operation of the full bridge converter suggests that if we know the value of the input voltage, output voltage of the converter and the duty cycle we can determine if the converter is working in DCM or CCM mode. This relationship between the input output voltage and duty cycle is given in equation (32) below.

$$K = \frac{2 \cdot L}{R \cdot T_S} = \frac{D_1^{\,2} \cdot n^2 \cdot V_g^{\,2} - D_1^{\,2} \cdot n \cdot V_g \cdot V}{V^2} \tag{32}$$

The input and output voltages of the converter can be measured by using relatively simple isolated voltage sensors and the information on the duty cycle already known because this is the signal generated by the DSP. So to develop a good control scheme that meets the requirements it is sufficient to measure just the input and output voltages.

3.1.1 Development of the control scheme

Since the phase shift controller chip TI UCC3895 [9] only uses a reference voltage in order to generate internally the phase shifting function there is no necessity to generate a triangular waveform to be used as in the peak current mode control fashion. Instead an averaged method should compensate for the control of this converter. The building of the control law starts by creating a reference model using the ideas from the

SCM technique. In the full bridge converter the inductor voltage would be the difference between V_d - V_{out}. The average value of V_d is estimated as $V_d = n \cdot V_{in} \cdot M$, where M is the modulation index, n is the number of turns in the transformer and V_{in} is the value of the input voltage. Since the average value of the inductor voltage is zero, this value is equal to the output voltage. This assumes that the inductor is ideal and there is no voltage drop due to the series resistance of the inductor. Since the output voltage has to be equal to a reference voltage the substitution of the output state value by the reference value is made in the same fashion as in SCM. By adding a compensator we achieve an average control scheme that we were looking for. The implementation scheme for this control method is shown in Figure 3.1. In this block diagram the input voltage is read and is multiplied by the number of turns of the full-bridge DC-DC transformer. The information from the input and output voltages are used to determine whether the converter is working in the DCM or CCM mode and the appropriate action is taken. Based on this observation the correct value of the modulation index is delivered which multiplies $n \cdot V_{in}$ value where n is the transformer turns ratio. This is similar to feed-forward control schemes but in this case a plant model is being used.

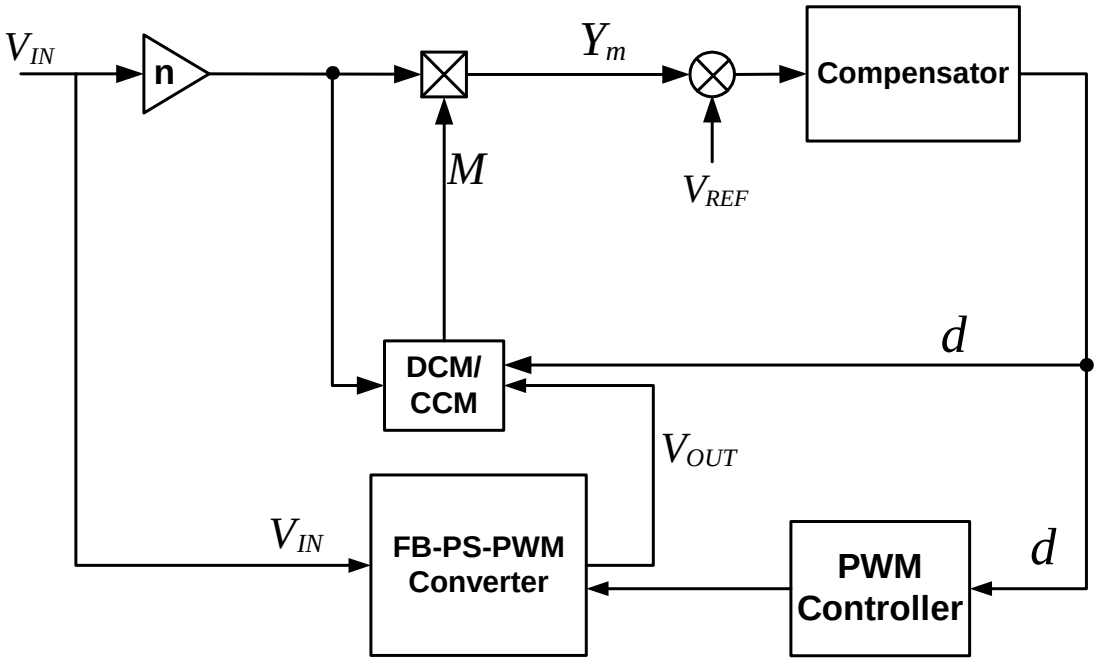

Figure 3.1. Block Diagram Model Reference Controller

If series resistance of the filter inductor is zero, the transformer is ideal and the number of turns is exact, the scheme works well and the output voltage follows the generated Ym model very closely with no steady state error. When the value of the inductor's series resistance is not zero and in most cases not known exactly the voltage drop in the inductor needs to be compensated. At the same time because of the nonlinearity of the transformer for all the load range the scheme has a steady state error that is basically the difference between the plant output and the model Y_m. To compensate for this error an adaptive control mechanism is designed. This mechanism takes into account the difference between the output voltage and the model reference and acts appropriately by changing a coefficient dynamically. The mechanism adjusts the difference between the model and the real output. The speed of this mechanism does not need to be very fast and it is implemented by using a simple integrator with limiting. This mechanism compensates for not only on the inductor voltage drop but also the little difference that might be created by not having the exact number of turns. In this case the transformer number of turns can be adjusted appropriately and in the case when the transformer is not ideal this would be a preferred technique. The implementation of the adjusting mechanism is shown in Figure 3.2

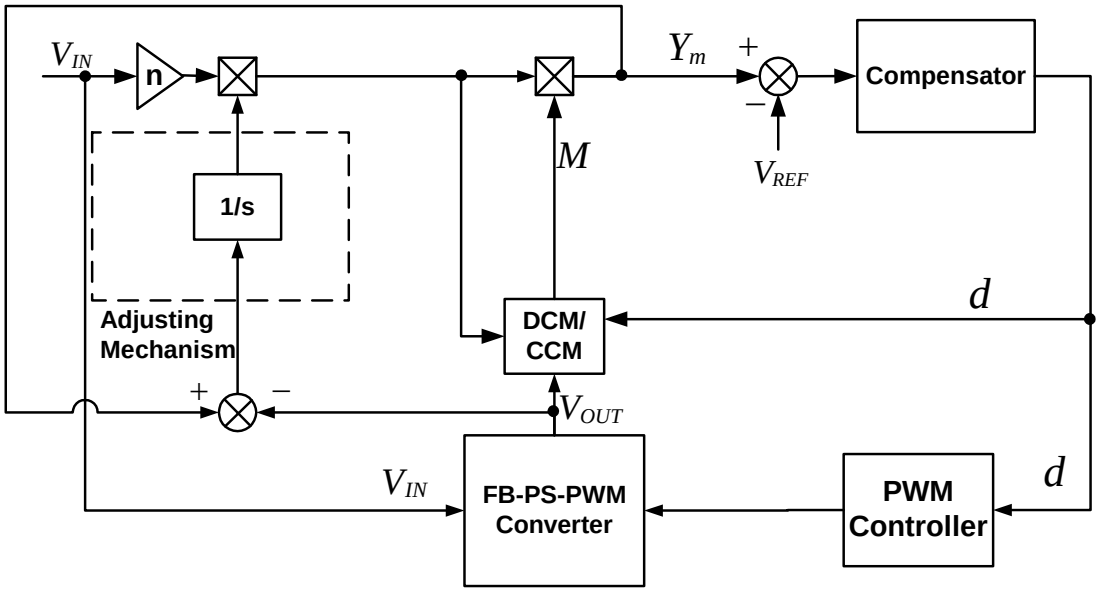

Figure 3.2. Block Diagram of the controller with the adjusting closed loop control in place.

In Figure 3.2 V_{IN} and V_{OUT} are the voltage sensor readings, n is the transformer turns ratio. Duty cycle d in this case is a control value upon which the PWM controller creates the appropriate phase shifting effect. The compensator is a third order compensator and is described in detail in the following section.

3.2 Simulation

Saber® was the chosen tool because of the advantages of having better programming options and better convergence in comparison to Pspice or Simplorer. Using this software tool it is possible to perform transient and frequency analysis in the same circuit.

3.2.1 Average Model Simulations

The average circuit model for the converter is shown in Figure 3.3. The average model of the Full-Bridge Phase shifted PWM converter is done in Saber®. The design work is done starting with the average model because the simulation time is shorter and that with the average model it is possible to simulate for frequency response as well. The average circuit model shown in Figure 3.3 is based on the development if the average switch model and the small signal models [21, 22].

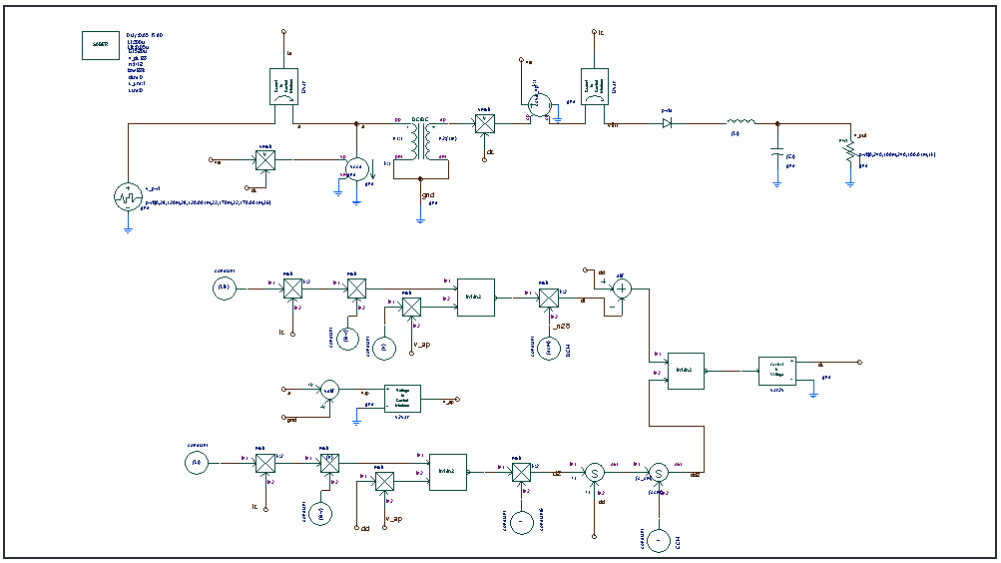

Figure 3.3. Saber® Model for the PS-FB-PWM Converter

First the open loop circuit is simulated in order to verify that the modeling is done correctly for both DCM and CCM operation.

Using the average model circuit shown in Figure 3.4, after determining the steady state operation point the control to output transfer functions for both DCM and CCM operation are simulated and shown in Figure 3.4 and Figure 3.5.

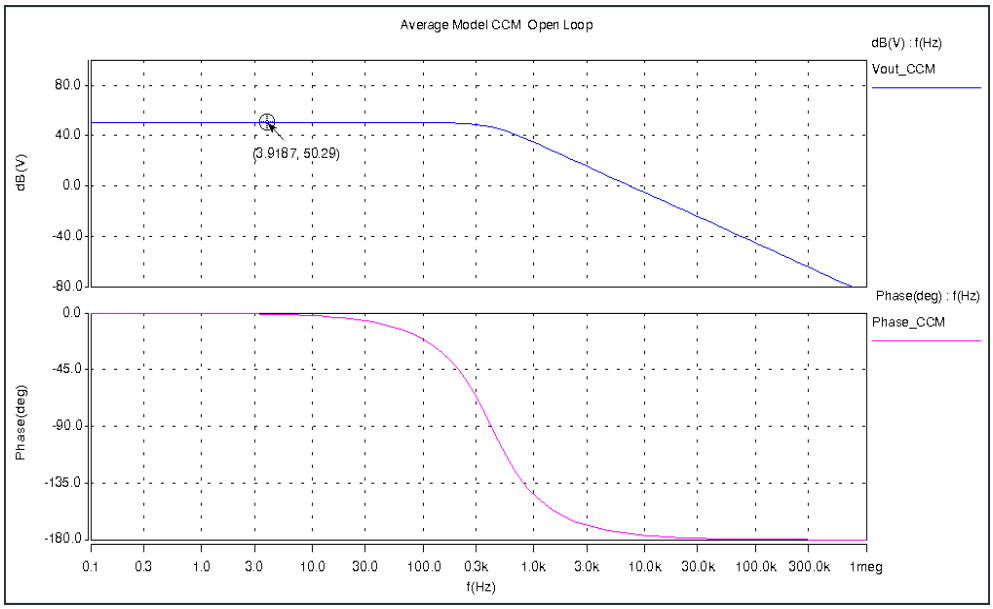

Figure 3.4. Control-to-output transfer function in CCM.

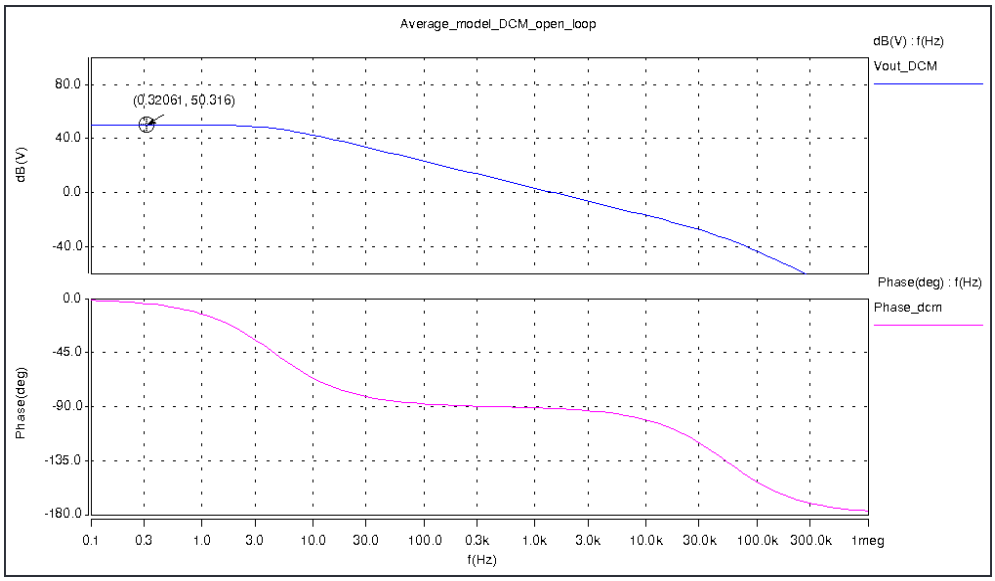

Figure 3.5. Control-to-output transfer function for DCM operation.

These simulations are carried out to verify that the modeling of the converter is done correctly according to Tsai findings [22]. The model is also verified with the

converter parameters as in [22] and the performance is identical. Once the open loop simulation is determined to be correct the controller modeling starts. The controller is modeled based on the block diagram shown in Figure 3.2 and the converter is represented by its average model. Simulations are carried out for both CCM and DCM cases. Since the design in DCM operation is more challenging the primary design is done in this mode of operation and then using the same controller the operation in CCM is verified. This is a common practice in power electronics designs.

The closed loop design procedure started with the design shown in Figure 3.6. Afterwards the adjusting mechanism was implemented and the final simulation circuit of the averaged model is shown in Figure 3.6.

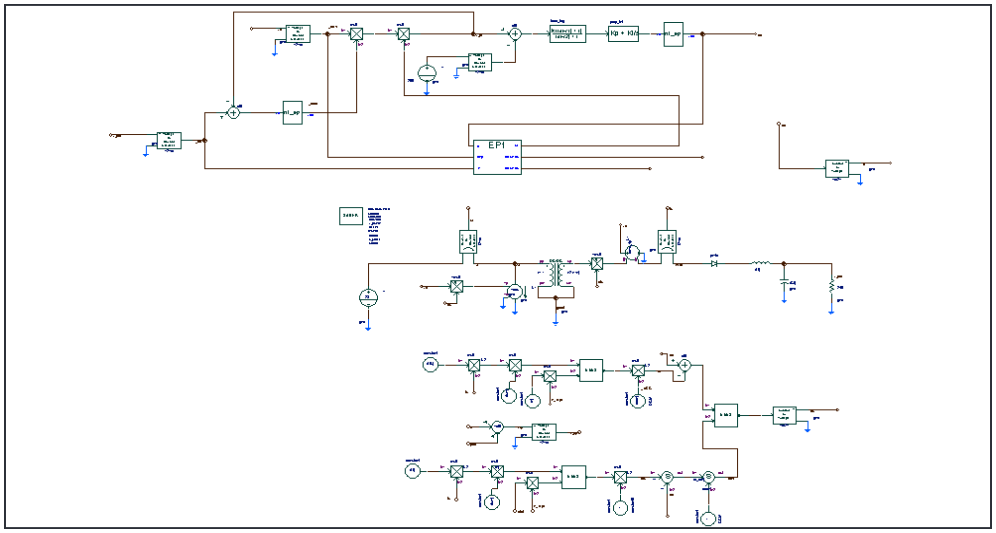

Figure 3.6. Average Model with the Implemented Control Scheme**.**

When designing the compensator before closing the loop a simple PI controller was implemented. The idea was to first find out what the response of the converter would be and take the appropriate actions upon the understanding of its performance. The simulation in DCM mode brought an interesting fact that suggests that the output voltage follows the model very closely although the overall performance is not what it is expected. This performance where the output voltage follows the model is shown Figure. The performance also suggests that the converter is very close to being marginally stable and that the margins are very small.

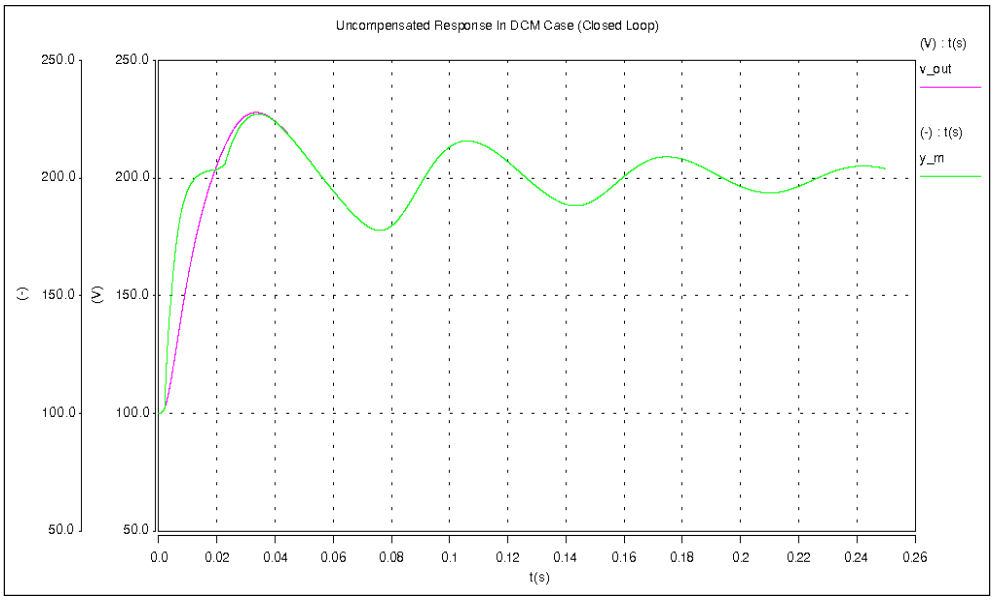

Figure 3.7. Closed loop performance with a PI controller.

In order to verify that this assumption is correct the frequency response of the converter was simulated and shown in Figure 3.8. The phase margin is only 9.2^{o}. This explains why when using a simple PI controller the oscillations are present.

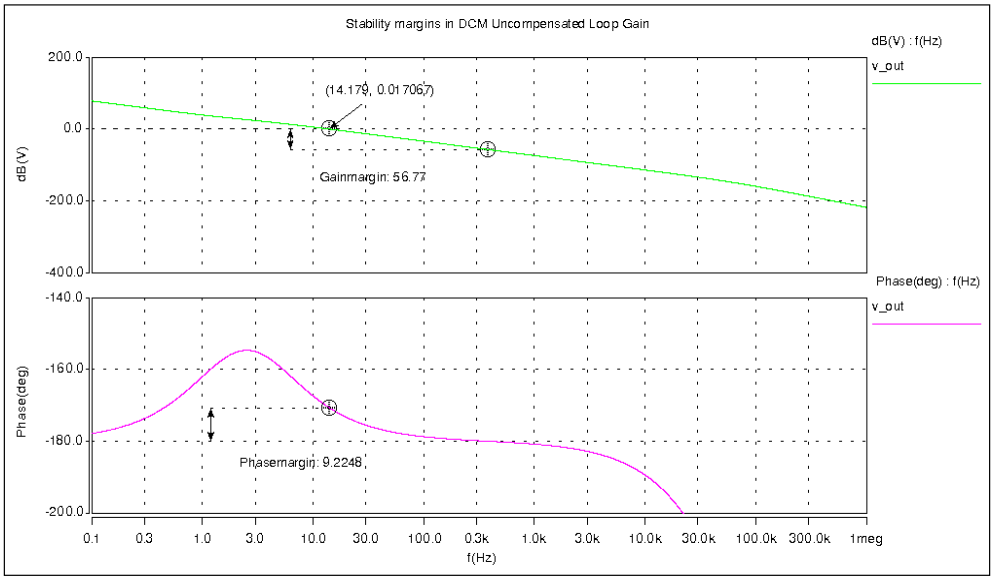

Figure 3.8. Loop Gain Frequency response with simple PI controller.

To improve the performance a lead factor is added into the compensator block [23, 24]. The purpose is to improve the phase margin and increase the crossover

frequency and the bandwidth of the controller. The compensator after adding the lead factor has the following form:

$$C(s) = -\frac{3500(s+1500)(s+15)}{s^2(s+15000)} \tag{33}$$

The performance of the design is greatly improved, the phase margin is at $PM = 66.8^o$, and the gain margin at $GM = 32.6\ dB$ and a crossover frequency $f_c = 1580\ Hz$.

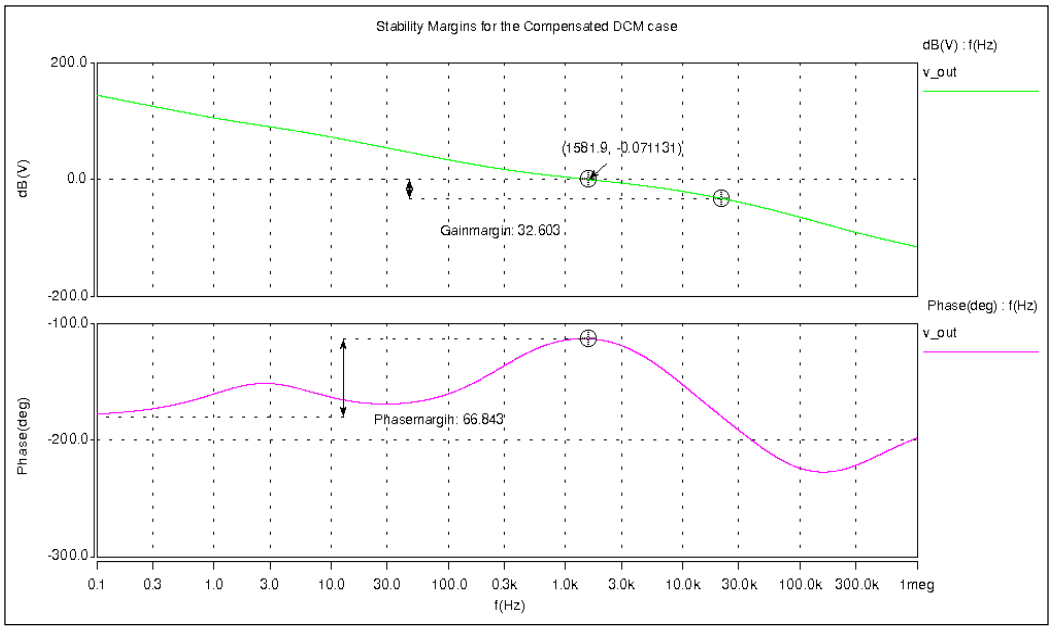

Figure 3.9. Improved performance (lead factor added)

The transient response of the closed loop system is shown in Figure 3.10. In this result an input voltage step and a load step are simulated and as seen by this performance the input voltage step as well as the output load change do not affect the output. However, because the control of this converter is implemented using a DSP there is some delay that is added into the performance.

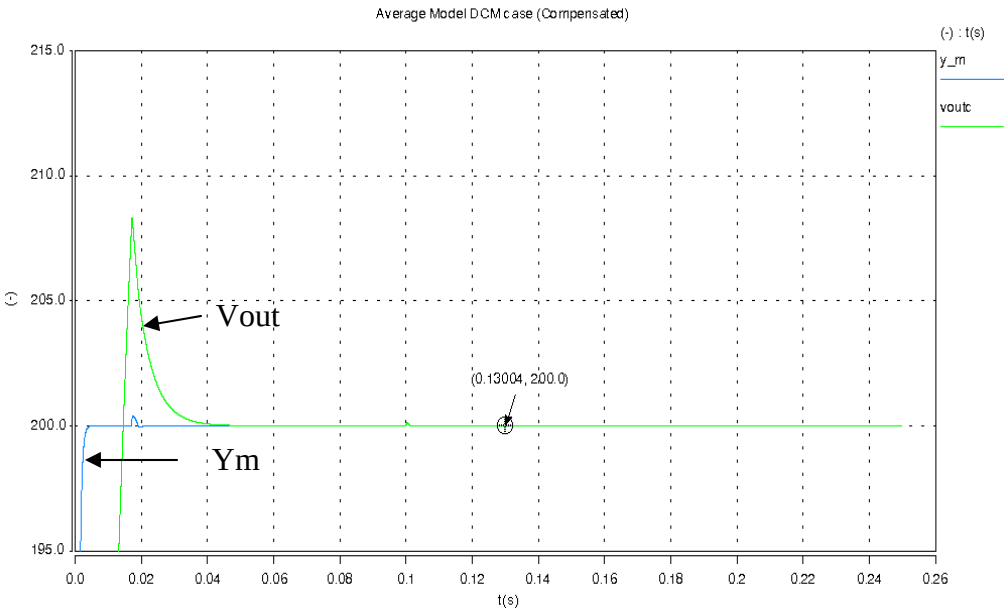

Figure 3.10. Transient performance with the improved controller

This delay first was estimated to be equal to two switching periods. As it turned out this was a good estimate because when the controller was later implemented the overall delay was smaller than what was predicted. By adding a delay of two switching periods the margins change drastically as the delay introduces a drop in the phase. This is shown in the frequency response simulation shown in Figure 3.11 and Figure 3.12. The phase margin is about 27^{o} and to have a good transient response a phase margin of greater than 45 degrees is desired and the gain margin to be greater than 6dB. For this a simple solution is possible by reducing loop gain.

Decreasing the gain will reduce the crossover frequency and in turn the bandwidth but the benefit of higher gain and phase margin offsets such effect. By reducing the gain 2.4 times or 7.6 dB the frequency response is shown in Figure 3.12.

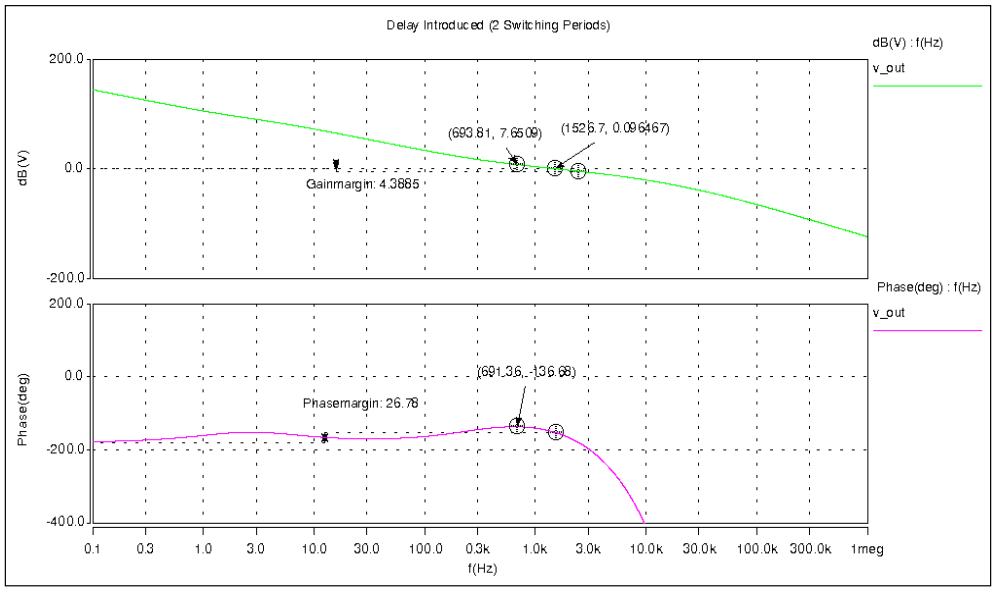

Figure 3.11. Frequency response after delay is added.

The compensator takes the form :

$$C(s) = -\frac{1500(s+1500)(s+15)}{s^2(s+15000)} \tag{34}$$

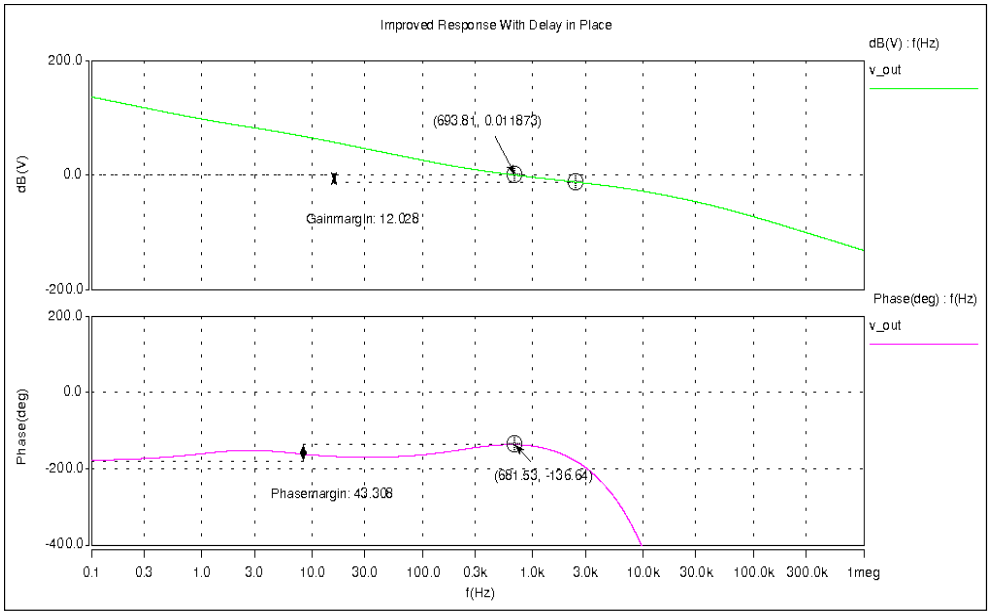

Figure 3.12. Final frequency response

However in the final design implementation the overall compensator the final form was tweaked and it is shown in this is the controller implemented into saber and the DSP:

$$C(s) = -\frac{1400(s+1400)(s+15)}{s^2(s+15000)} \tag{35}$$

3.2.2 Switching Model

The switching model of the converter with the implemented controller is shown in Figure 3.13. The converter model is implemented using ideal switches in order to reduce the simulation time during transient response. The controller model is carried out from the average model. In order to make simulation as close as to real implementation the control block is replaced by a discrete model which is described in more detail at the end of this chapter. The Saber listing of this model is included in the appendix section .

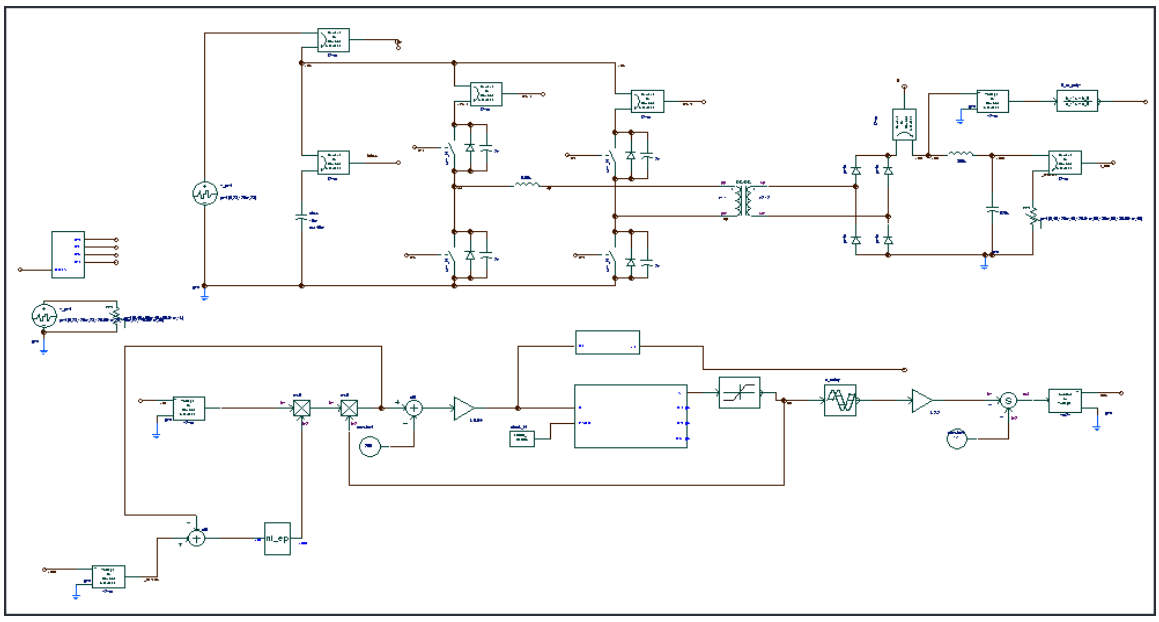

Figure 3.13. Switching model simulation schematic

The model of the UCC3895 phase shifting controller and its corresponding waveforms are shown in Figure 3.14. This model is a very close approximation of the controller IC and all the parameters as voltage levels and limiters are set to the real values

of UCC3895. This is a simple representation but the benefit is that the simulation time is reduced by making this an ideal phase shift controller.

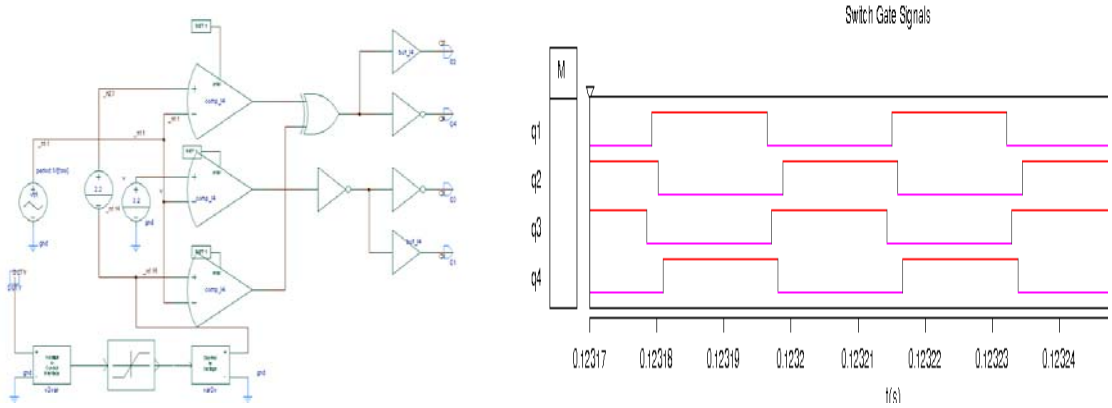

Figure 3.14. Implementation of the UCC3895 and its Corresponding Switch Waveforms**.**

Again a few simulations are carried out in order to verify that the switching model and the average model match with each other. The case when the compensator is just a simple PI+ integrator is simulated and the converter performance is almost identical as in the case of the average model. The output follows closely the reference model as in the case of the average model.

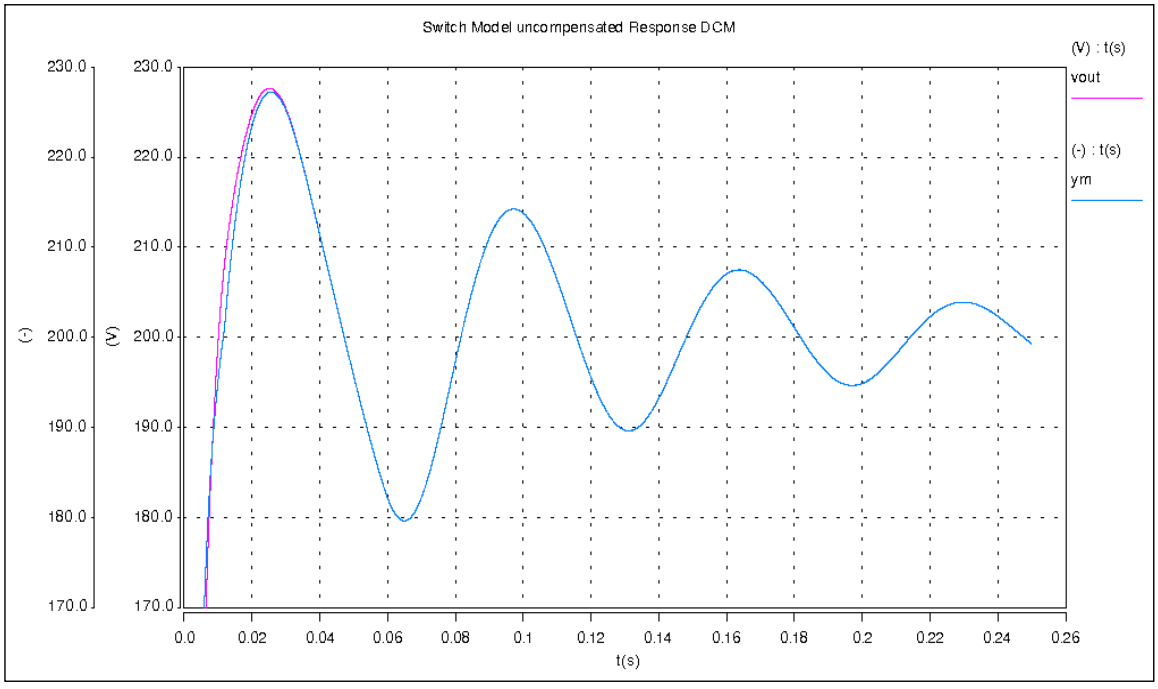

Figure 3.15. Switch model with only PI controller

In Figure 3.15 the converter's performance is tested in the case when the load is changed such that the converter operates from CCM to DCM operation. The load is changed from 1kW to open circuit. Once the output catches up with the model reference they remain identical. The simulation shows the load change from full load to no load operation.

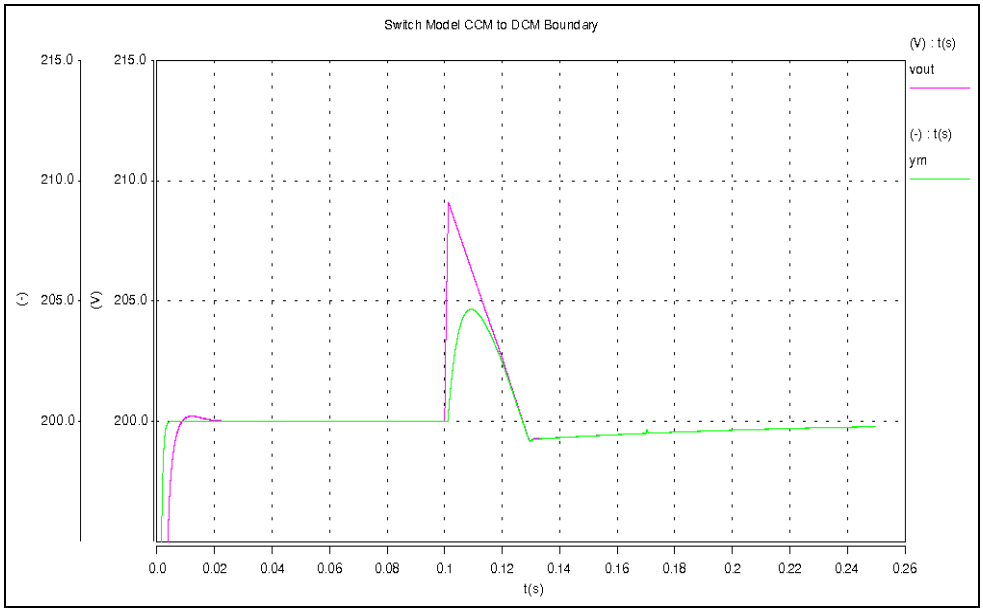

Figure 3.16. Boundary between CCM and DCM operation

The following figure shows the case when the converter is operating in CCM and that we have a 50% step load change. This is shown in order to be compared with the case of the hardware implementation.

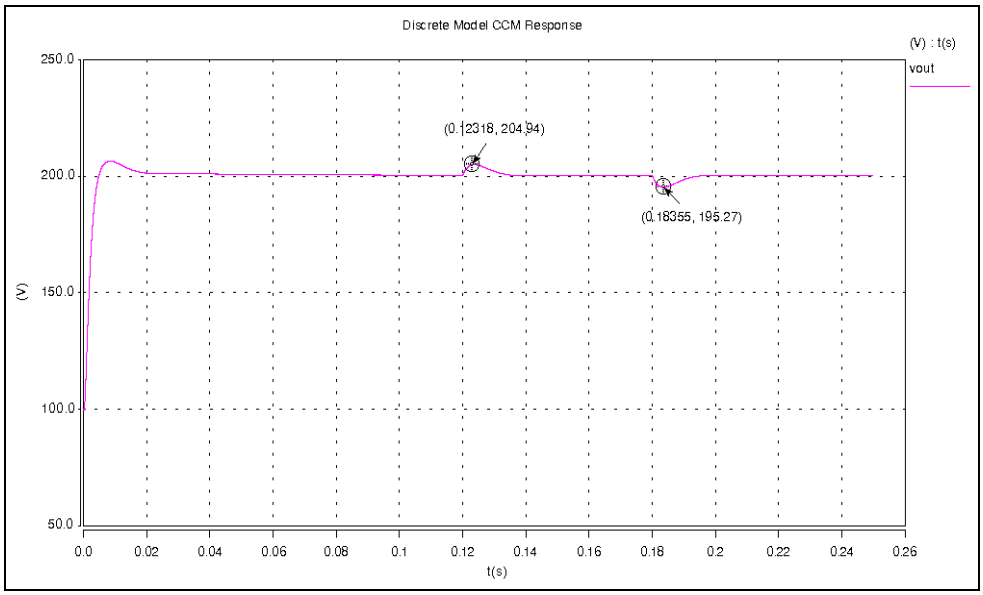

Figure 3.17. 50% load step in CCM model

3.2.3 Representing the Controller as a State Space Model

When implementing the code in the DSP, a state space model was used. One reason for this is that the transfer function representations of the controller may contain numbers that are ill-conditioned. Since the DSP is a fixed point DSP, it is desirable to be able to manipulate the matrices in order to achieve more balanced numbers that are easier to represent.

The general form of the compensator (second order integrator plus lead compensator) is sown in equation (36):

$$u = \frac{k(s+z_1)(s+z_2)}{s^2(s+p_1)}e \tag{36}$$

This system was entered into Matlab, and the *ss* command was used to convert this transfer function to state space. Once in state space form, the system was represented as shown in (37):

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -15000 & 0 & 0 \\ 128 & 0 & 0 \\ 0 & 32 & 0 \end{bmatrix} x + \begin{bmatrix} 128 \\ 0 \\ 0 \end{bmatrix} e \\ y &= \begin{bmatrix} -11.133 & -122.91 & -50.038 \end{bmatrix} x + 0e \end{aligned} \tag{37}$$

In order to convert this to the discrete domain for implementation in a DSP, the discrete version of the controller needs to be determined. The discrete version of the controller can be obtained using Matlab's c2d command. The method used was the zero-order-hold method.

The zero-order-hold method assumes that the input is held constant between samples, which is valid for this type of controller, as the DSP will hold the value of the duty cycle at a constant value between samples.

After converting the controller to the discrete domain, the following state space controller is created:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.687 & 0 & 0 \\ 0.003 & 1 & 0 \\ 0 & 0.0008 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.003 \\ 0 \\ 0 \end{bmatrix} e \\ y &= \begin{bmatrix} -11.133 & -122.91 & -50.12 \end{bmatrix} x + 0e \end{aligned} \tag{38}$$

There are several things that are important in the control design when taking into consideration that the controller will operate in a fixed point DSP. One consideration is the magnitude of the numbers that are used in the controller. In a fixed point DSP, there is a maximum and minimum value that define the range of all allowable numbers in the DSP. It is important that not only the coefficients in the matrices, but also the actual state values are contained within that range. If the values of the state variables are too big, then they cannot be represented or stored in the DSP, leading to problems. The other issue to consider is the resolution. If the state variables are too small in magnitude, then the effects of quantization will dominate the behavior of the controller, which is highly undesirable. In order to "balance" the states, a similarity transformation was used on the states until the simulated results were similar in magnitude for the largest allowable perturbations. When implementing a state space controller, the range of the state variables is important. In state space, a transformation can be made that changes the specific state variables used. This can be shown as:

$$\hat{x} = Tx$$

Then, the controller can be rewritten in terms of the new state variables:

$$x = T^{-1}\hat{x} \Rightarrow \dot{x} = T^{-1}\dot{\hat{x}}$$

$$\begin{matrix} \dot{x} = Ax + Be \\ u = Cx + De \end{matrix} \Rightarrow \begin{matrix} T^{-1}\dot{\hat{x}} = AT^{-1}\hat{x} + Be \\ u = CT^{-1}\hat{x} + De \end{matrix} \Rightarrow \begin{matrix} \dot{\hat{x}} = TAT^{-1}\hat{x} + TBe \\ u = CT^{-1}\hat{x} + De \end{matrix} \tag{39}$$

where $\hat{x}$ represents the new states, and x represents the original states. If the state variables are too large during simulations, then the value can be reduced to a smaller value by scaling T appropriately. During this operation, the input-output characteristics of the system remain the same, and the only difference is the internal dynamics of the controller.

The command ss2ss was used to perform the similarity transform.

The finalized state space controller is given in (39).

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.6875 & 0 & 0 \\ 0.004 & 1 & 0 \\ 0 & 0.004 & 1 \end{bmatrix} x_k + \begin{bmatrix} 2.551 \\ 0.004 \\ 0 \end{bmatrix} e \\ y &= \begin{bmatrix} -0.0117 & -0.0937 & -0.0078 \end{bmatrix} x + 0e \end{aligned} \tag{40}$$

This is the controller that was implemented in the DSP.

Chapter 4 - Hardware Implementation and Testing

4.1 Front End

The front end of the power stage for the full bridge converter was designed by the Virginia Tech team of the 2001 Future Energy Challenge (FEC) [25], and was used here as the focus of this thesis is converter control. This converter was designed for an input voltage of 48 V_{DC} (positron exchange membrane (PEM) fuel cell) at 2 kW. Since it was adopted for a 28V nominal input voltage, the maximum power from the full bridge was kept at 1kW in order to avoid any heating due to larger currents in the primary. The front end utilizes the UCC3895 phase shift controller, and all the gate drivers for the switching devices are incorporated in the design.

4.2 Sensors

Converting DC input from a 22-41 V fuel cell to 200V DC requires sensing circuits for control and protection. Particularly, the DC bus voltage and fuel cell voltage needed to be sensed. The location of the dc sensors is shown in Figure 4.1.

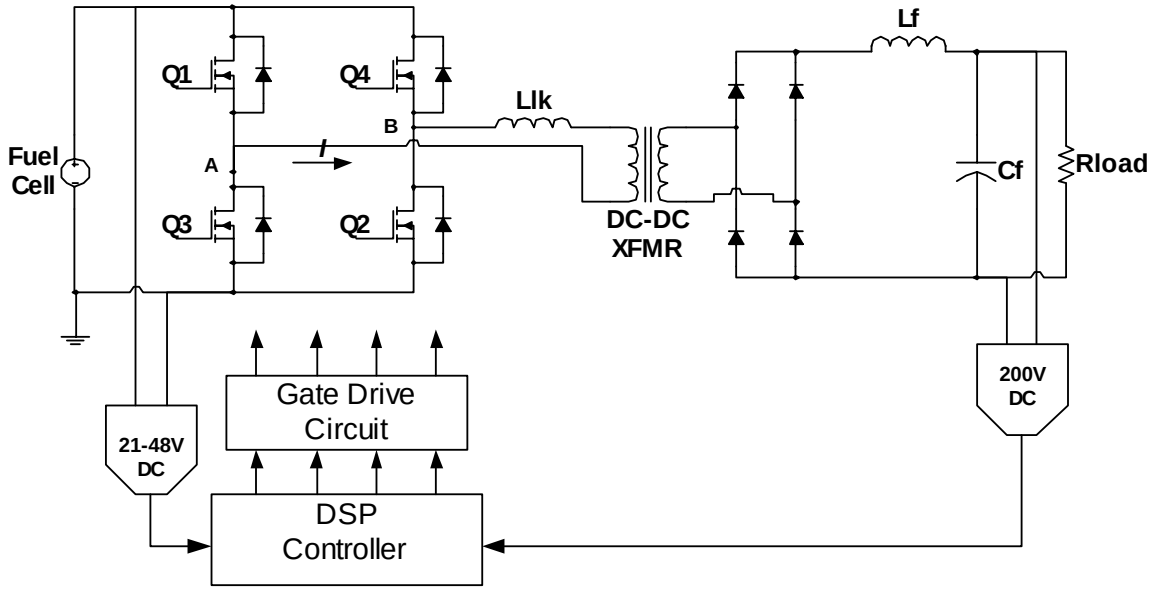

Figure 4.1 Sensor Block Diagram

The fuel cell voltage sensor design follows the same concept as the DC link voltage. The outputs from the DC voltage sensors are read in the DSP controller through

A/D inputs and the digitized signals are used in the control loop by the controller in order to generate the desired PWM signal. The desired PWM signal was then converted into an average voltage level to be taken by the UCC3895 and the switching devices.

The DC voltage sensing circuit consists of two gain stages: an isolation stage and a filtering stage. A block diagram of the voltage sensor is given in Figure 4.2

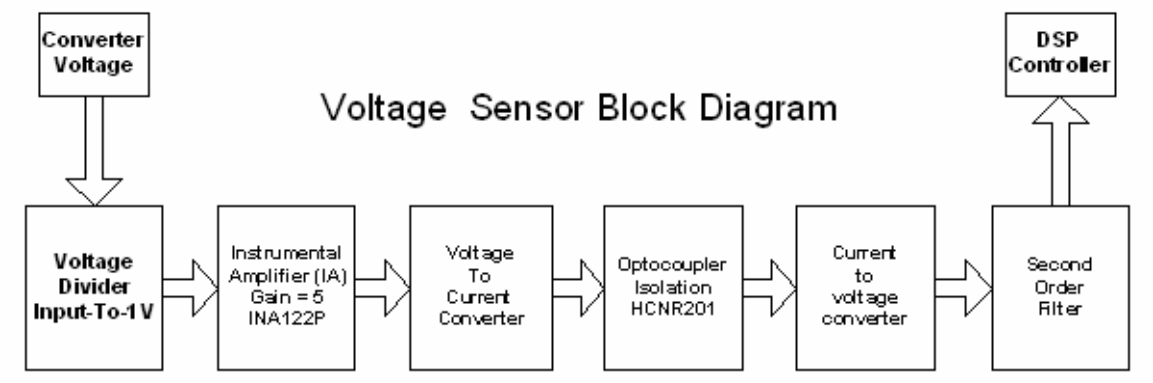

Figure 4.2. DC Sensor Block Diagram

The first gain stage is a voltage divider at each of the terminals of the output (one at the positive terminal and the other at neutral). The second gain stage is an instrumental amplifier (IA), Burr Brown's INA122P [26]. An IA was utilized for its high common mode rejection ratio (CMRR). Without using any external resistors, the gain of the INA122P was 5 V/V which set the overall gain at 0.025 V/V.

Following the IA stage, the isolation stage consists of a voltage to current (V-I) converter which interfaces with the voltage gain and offset stages of the linear opto-coupler HCNR201. A current-to-voltage (I-V) converter is also used to translate the current output of the opto-coupler back into voltage. This yielded an overall gain of 0.0075 V/V. The output of the DC sensor was at 3V, which was below the 3.3V A/D input limit of the DSP.

The last stage of the circuit is a simple KRC filter [27]. This filter reduced the noise injected at the input and any noise pickup by the wires connecting the inverter board and the control board. As mentioned before, this filter should have reduced the noise to levels well below the resolution of the A/D. Thus, there would be greater than 80

dB attenuation at just below the switching frequency of the DC-DC converter. A picture of the actual built sensor board is shown in Figure 4.3.

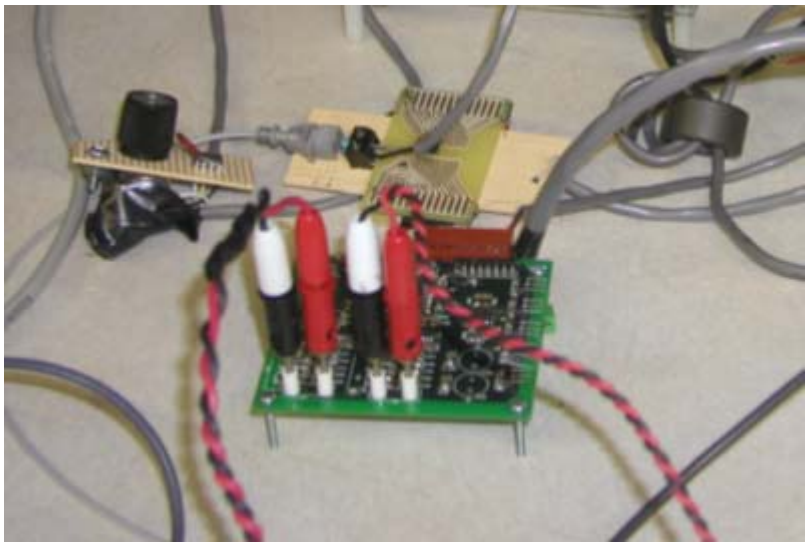

Figure 4.3. Sensor Board

4.3 Transformer

Single phase planar transformers were chosen for the full-bridge transformer configuration. Planar transformers provide unique benefits by differing from traditional wire wound transformers in winding arrangement and core geometry. The winding is made of flat copper layers, where each layer has a certain number of turns wound in a spiral pattern. Insulation between layers is made either of FR4 material or simple insulation paper. The core is usually ferrite and low profile compared to traditional transformers. The winding layers can be connected in series/parallel in order to create the desired number of turns and to obtain an equivalent conductor thickness for the skin depth area. This type of winding geometry allows planar transformers to operate at high switching frequencies, which can result in further core size reduction and reduced winding losses [28]. Stacking the layers simplifies the production of symmetric windings, and keeping greater repeatability in manufacturing. For transformers having center tapped or multiple primary/secondary side windings, the placement of center tap terminals can be made to the same location and the transformer can create perfect

winding balance. Primary and secondary winding layers can also be stacked alternately to produce higher coupling and transformer leakage inductance can be controlled depending on the requirements. A basic schematic of the transformer that was used is shown having one primary winding and two secondaries is shown in Figure 4.4.

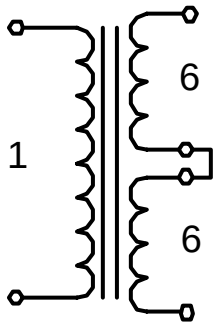

Figure 4.4. Transformer configuration

Planar transformers exhibit high manufacturability leading to more repeatability of transformer parameters. This complements topologies which require multiple closely matched transformers for soft switching and/or delta winding. Finally, planar core geometry is better suited for heat sink mounting which allows higher transformer power densities to be achieved [29, 30].

In this design, a 2kW planar transformer design from Payton Group International was selected. The transformer listed as type T1000AC-1-12; part number 50635 is shown in Figure 4.5. Operating parameters of 50 kHz, 112 A rms maximum primary current, and 1000 V rms minimum dielectric strength were specified. The transformer leakage and magnetizing inductances were measured with an impedance analyzer and the measured values are:

Leakage Inductance: $L_{LK} = 4\ \mu H$

Magnetizing Inductance: $L_M = 1.9\ mH$.

Figure 4.5. 2kW planar transformer

4.4 Output Filter Selection

The output filter inductor and capacitor values were calculated based on maximum ripple current and ripple voltage magnitudes. A peak inductor current ripple no greater than 20% of the average inductor current was desired. These values were calculated ideally using a buck converter output filter design approach [3]. The calculations were done considering the converter is working in CCM.

$$2 \cdot \Delta i_L = \frac{n \cdot V_{in}}{L} \cdot D \cdot T_S \quad \Rightarrow L = \frac{n \cdot V_{in}}{2 \cdot \Delta i_L} \cdot D \cdot T_S \tag{41}$$

Where, N is the transformer turns ratio, D is the duty cycle, V_{in} is the input voltage, Δi_L is the ripple current and T_S is the switching period. Substituting all the values into equation 4.1, the desired inductor value is $L = 300\mu H$. The inductance of a wound core can be calculated from the core geometry by using the following equation [31]:

$$L = \frac{0.4\pi \cdot \mu \cdot N^2 A_e}{l_e \cdot 10^8} \tag{42}$$

L = inductance (Henries), μ = core permeability, N = number of turns, A_e = core cross section, l_e = core magnetic path. For this design *Kool Mμ* powder cores were used. In order to achieve the maximum inductance with the minimum number of turns, 3 cores are stacked on top of each other. The parameters of the inductor design are shown below:

Core part number *77109-A7*, μ = 125, N = 34, l_e = 14.56 cm, A_e = 1.463 cm^2

After the inductor was built, the inductance value was measured in order to verify that the design was correct. The measurement was done with an impedance analyzer and the measurement results are shown in Figure 4.6. The inductance value at 56kHz and no load current is 300uH. At higher current levels, this inductance may be reduced but since this inductor acts as an output filter where the currents are relatively low, the overall inductance should be very close to this measured value.

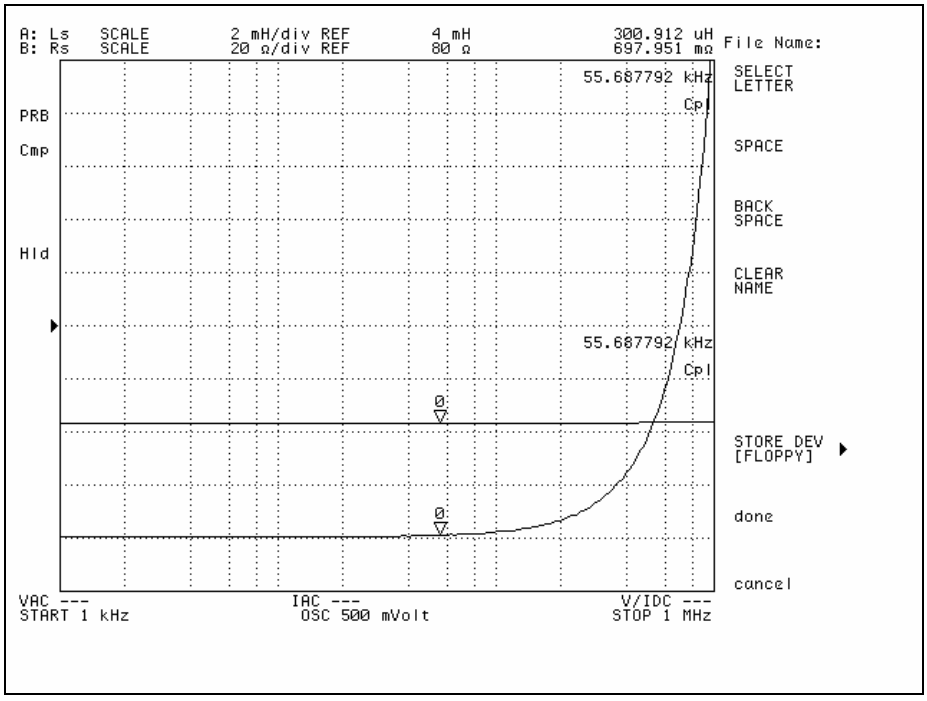

Figure 4.6. Measured Inductor Value

Filter capacitor value is calculated based on the inductor ripple value and the ripple voltage output. This relation ship is shown below:

$$\Delta v = \frac{\Delta i_L \cdot T_S}{8 \cdot C} \quad \Rightarrow C = \frac{\Delta i_L \cdot T_S}{8 \cdot \Delta v} \tag{43}$$

Based on only the above equation the capacitor value was calculated to be $C = 2\mu F$, but a much higher value of 2.2mF was used to provide high side energy storage for voltage regulation during load transients.

4.5 Diode Selection

Based on initial converter simulation results, diode ratings of V_R = 600 V (reverse blocking voltage), and $I_{F(AV)}$ = 50 A (average forward current) were selected. Other specifications such as forward voltage drop, reverse recovery time, peak reverse recovery current, and price were strongly considered during device selection. Three ultra-fast, soft recovery devices offered by different manufacturers were selected for comparison and are presented in Table 2.

Table 2. DC-DC Diode Comparison

Manufacturer	Part Number	V_F	t_{rr}	I	Package
Fairchild	RHRG5060	1.5 V (100 °C)	45ns (max)	50 A	TO-247
International Rectifier	HFA50PA60C	1.9 V (125 °C)	23ns (typ)	50 A	TO-247AC
IXYS	DSEK 60-06A	1.6 V (100 °C)	35ns (typ)	60 A	TO-247AD

A low diode forward voltage drop (V_F) is important to minimize device power consumption. Also, low reverse recovery time and small peak reverse recovery current were important criteria to reduce reverse conduction and provide the most continuous load current.

Overall, the IR and IXYS devices appear to be the best suited for implementation, with the IR device showing slightly more desirable characteristics. The IR devices were chosen for the final implementation.

4.6 DSP and the Controller Implementation

The controller that was chosen for this system is the 40 MHz Texas Instruments TMS320LF2407A DSP [32] (shown in Figure 4.7). This DSP uses fixed-point arithmetic and has 32k of on chip flash EEPROM. This DSP is a good choice because it has 16 onboard PWM units, which are necessary for a multi converter system. The DSP also

has 16 A/D channels, which are important for control of a complex system. In addition, this DSP family includes an integrated development environment that speeds development time and debugging. The DSP code was written in standard ANSI C and assembly.

Figure 4.7. Texas Instruments TMS320LF2407A DSP.

One of the PWM generated signals was used as a duty cycle controller because the phase shift controller takes a voltage level input in order to generate the desired phase shift. The duty cycle output was filtered and the averaged value was sent into the phase shift PWM controller. Due to the relatively large distance from the DSP board to the actual gate drive, circuitry noise problems were encountered when the duty cycle implementation was realized without isolation. In order to provide isolation and to reduce noise pickup, an optical isolation circuit was established. The schematic for the optical isolation circuit is shown in Figure 4.8.

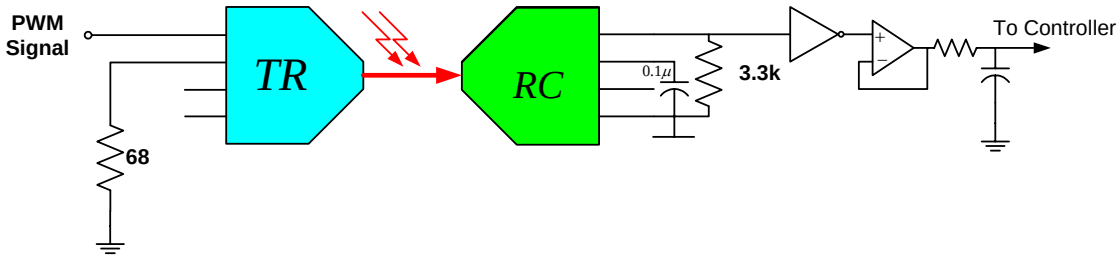

Figure 4.8. Optical isolation circuit for PWM transmission

4.6.1 DSP Control Overview

4.6.1.1 DSP Code Requirements

It is a common misconception that the DSP is just responsible for the control algorithm. There are many aspects of controlling a converter from tracking control to possibly online self-diagnostics that can be implemented in a DSP.

Tracking Control. The DSP should implement a controller that allows the output to track a desired reference.

Protection. The DSP needs to protect the converter from abnormal behavior introduced from either the environment in which the converter exists or abnormal control behavior. These abnormalities may come from fluctuations in the load, such as a load value that is outside the rating, or an input voltage that is too high. The DSP should also protect against failures due to problems with the control algorithm. The duty cycle should be limited to prevent large output voltages during transients.

Startup and Shutdown. During startup and shutdown, the DSP behavior may be different, as it is outside of the desired operating range of the controller. The DSP may need to use different control algorithms to start and stop the DSP.

4.6.1.2 Additional DSP Benefits

Using a DSP has many benefits that make it attractive for use in control systems. Among these are the flexibility of the control design, quick parameter adjustment, and real-time simulation.

Flexibility of Control. When using analog circuits to perform control, the control algorithm is fixed, and is not easily modified. Using a DSP allows the designer to change the control code very quickly. It is often helpful to implement simple, slow control algorithms first to verify the hardware is functioning correctly before moving to a higher performance or complex control algorithm. If hardware were used to do this, this would mean separate hardware designs and implementations for each algorithm. With the use

of software, modifying the control algorithm means changing several lines of code, which will take only several minutes.

Parameter Adjustment. Once the control algorithm is fixed, it is easy to modify the values of references and constants in the control code by directly modifying memory locations. This can be performed while the system is operational, allowing for quick adjustments to be made. If the control algorithm were implemented in analog hardware, this would not be as easy to do.

Backtracking. The use of software allows for easy backtracking in the even that a control algorithm is not working. If the control were implemented in analog circuits, the physical modifications would need to be reversed, which may also introduce additional errors in the process.

4.6.2 Structure of the DSP and Control System Hardware

The DSP communicated with a program residing on a notebook computer called Code Composer Studio [33]. Code composer studio interfaced to a JTAG emulator pod, which was connected to the DSP. The JTAG emulator allowed for incremental stepping of the control code for eased debugging. The JTAG emulator was also the means by which the control code was downloaded to the DSP.

The structure of the control system is shown in Figure 4.9. The control system is isolated from the power stage by means of plastic optical fiber and isolated sensors/ transducers.

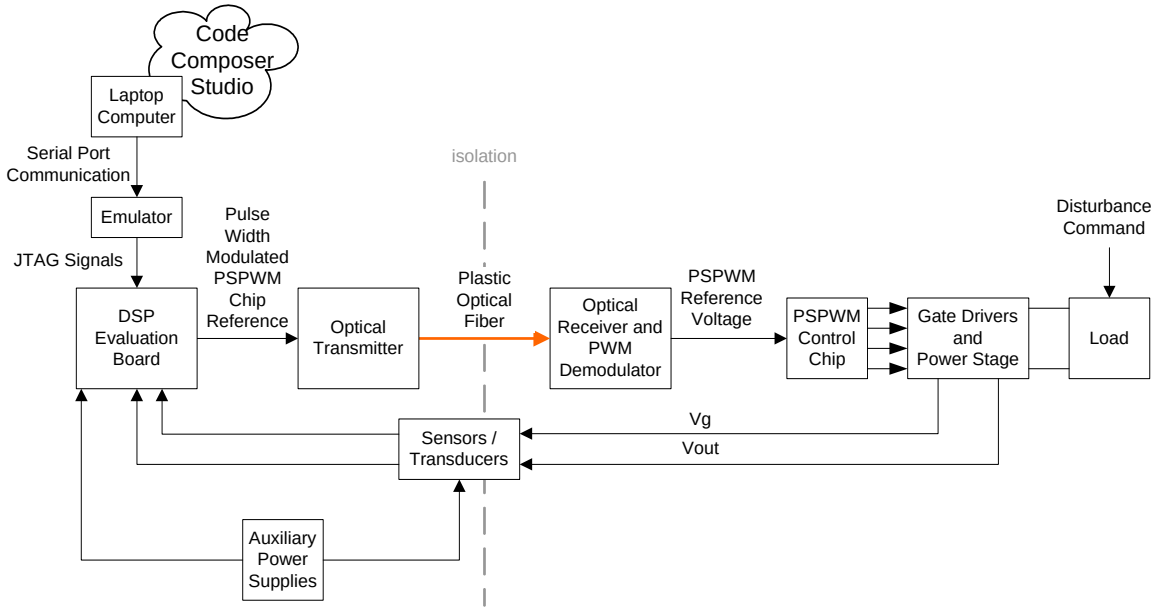

Figure 4.9. Control System Configuration

4.6.3 Control Software Overview

4.6.3.1 Operating Principle

The control of the DSP is composed of an initialization routine followed by periodic interrupts triggered by an internal timer. These interrupts occur once per switching cycle and set the reference for the phase shift modulator. This reference is then sent to the PWM modulator on the DSP so that it can be transmitted over the plastic optical fiber (POF). A flow chart describing the operation of the interrupt is shown in Figure 4.10

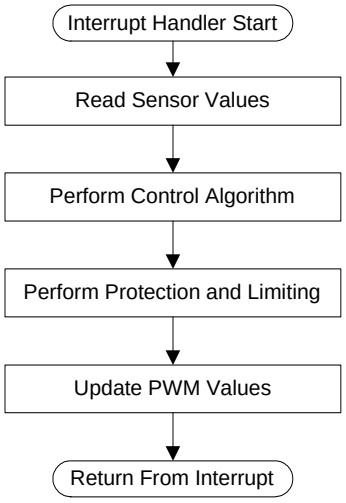

Figure 4.10. Interrupt Handler Flow Chart

The sensor values are read from the analog to digital converter (ADC) onboard the DSP. The output of the ADC is proportional to its input voltage so that an input of 3.3 Volts gives a digital value of 0x7FFF (hexadecimal), and an input of 0.00 Volts gives a digital value of 0x0000. The first thing to at the time of the interrupt is to request the ADC to perform a conversion. This is done by writing to the control register ADCTRL2. After the conversion is finished, indicated by testing ADCTRL2, the results are read into local memory locations, and are shifted to maintain scaling.

All numbers in the DSP use an 8.8 format, where there are seven integer positions, and eight fractional positions. The most significant bit represents the sign, allowing a maximum number of 127 and a minimum number of –128 with a resolution of 1/256 (0.0039). The need for a high resolution partially comes from the use of integrators. If the constant of integration is too small, then the number will not be incremented, as it is smaller than the resolution. This will mean that small errors will not accumulate, and will adversely affect zero steady-state error results. The binary 8.8 format is shown in Figure 4.11.

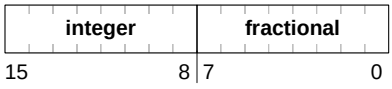

Figure 4.11. 8.8 Digit Format

This format is similar to representing decimal numbers by whole numbers. If a decimal 2.2 format was defined, the number 2.34 would be represented by the whole

number 234. All rules of mathematics still apply. However, when multiplying, the number must be shifted back by 2 to keep the scaling. If 234 was multiplied by 20 (actual number is 0.2), the resulting number would be 4680, which would be shifted by 2 to become 46. This matches the actual computation of 2.34*0.2, which is equal to 0.468. In the binary 8.8 format, numbers are shifted to the right by 8 bits when a multiplication occurs to keep the same scaling.

The next step in the interrupt routine is the control algorithm execution. The control algorithm computes the new value of the duty cycle. A summary of the control algorithm is shown in Figure 4.12. There are three primary components in the control algorithm. There is a linear compensator, which produces the duty cycle. The second component is modifies the output so that it produces the desired voltage at the input to the PSPWM chip in order to generate that duty cycle. The third component is an adaptive model of the power stagewhere N represents the transformer turns ratio. The adaptive parameter is the turns ratio, which increases or decreases at a fixed rate according to the sign of the error between the output voltage reference and the measured output voltage.

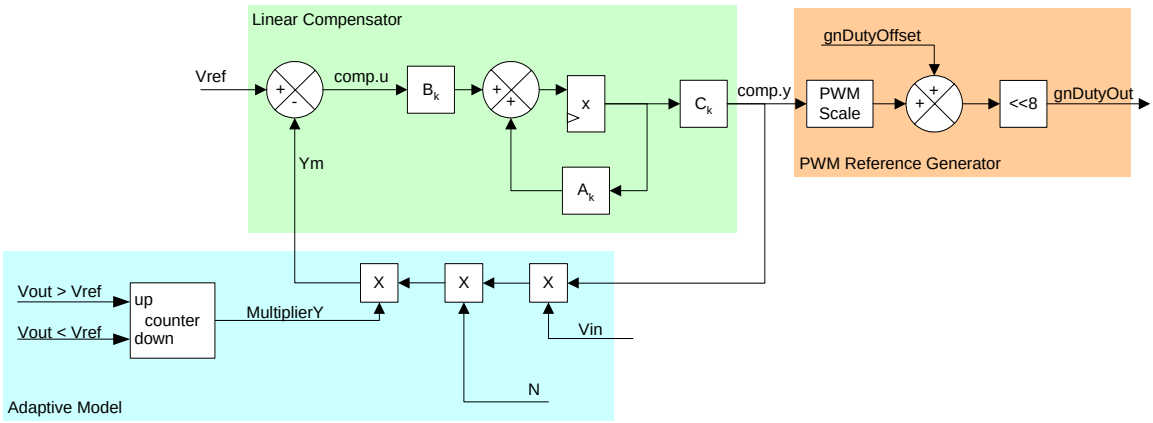

Figure 4.12. Control Algorithm Description

The results obtained in simulation used control algorithms designed in the continuous domain. The control algorithm parameters must be converted to the discrete domain in order to be implemented in a DSP. This was done using the Matlab C2D command. The default conversion method, Zero-Order Hold (ZOH) [34], was used. The result of this command was then converted to 8.8 format, which can be done by multiplying the resulting rational number by 256.

4.6.3.2 DSP Code Files

The files used and their descriptions are shown in Table 3.

Table 3. DSP Code File Description

File Name	Type	Description
adc.c	C Code File	Controls the Analog to Digital Converter
Adc.h	C Header File	
Cvectors.asm	Assembly File	
F2407.h	C Header File	
fir.c	C Code File	Implementation of a Finite Impulse Response Filter
Fir.h	C Header File	
init.c	C Code File	Controls Initialization of the DSP
Init.h	C Header File	
inv_main.c	C Code File	
io.c	C Code File	Control the I/O interface
Io.h	C Header File	
main.c	C Code File	Main initialization and interrupt routine
Matrixmult.asm	Assembly File	
MmfraG.asm	Assembly File	
pwm.c	C Code File	Control the PWM modulator
Pwm.h	C Header File	
qmath.c	C Code File	1.15 Format Signed number management
Qmath.h	C Header File	
sgen.c	C Code File	Generates sinusoidal reference
Sgen.h	C Header File	
spi.c	C Code File	Contains code for using the serial peripheral interface
Spi.h	C Header File	
ss.asm	Assembly File	
ss.c	C Code File	Contains code for state space implementation
ss.h	C Header File	
Ssasm.asm	Assembly File	
std.c	C Code File	
Std.h	C Header File	

4.7 Testing

A fuel cell was not available for testing the converter; and as a result, the experimental data were taken using a power supply as the input source. The system power source was a 65 A, 28 V DC power supply, capable of delivering more than 1kW at its nominal current. In all the tests the load was made of a resistive load bank that could be adjusted by using a contactor. The switching of the load with the DC contactor was seen as the easiest solution, but the mechanical transient of the contacts switching on/off were visible when step load experiments were carried out. The DSP was operated via a serial connection, so changes to the code could be easily made and as the converter was running. A picture of the bench test set-up is shown in Figure 4.13.

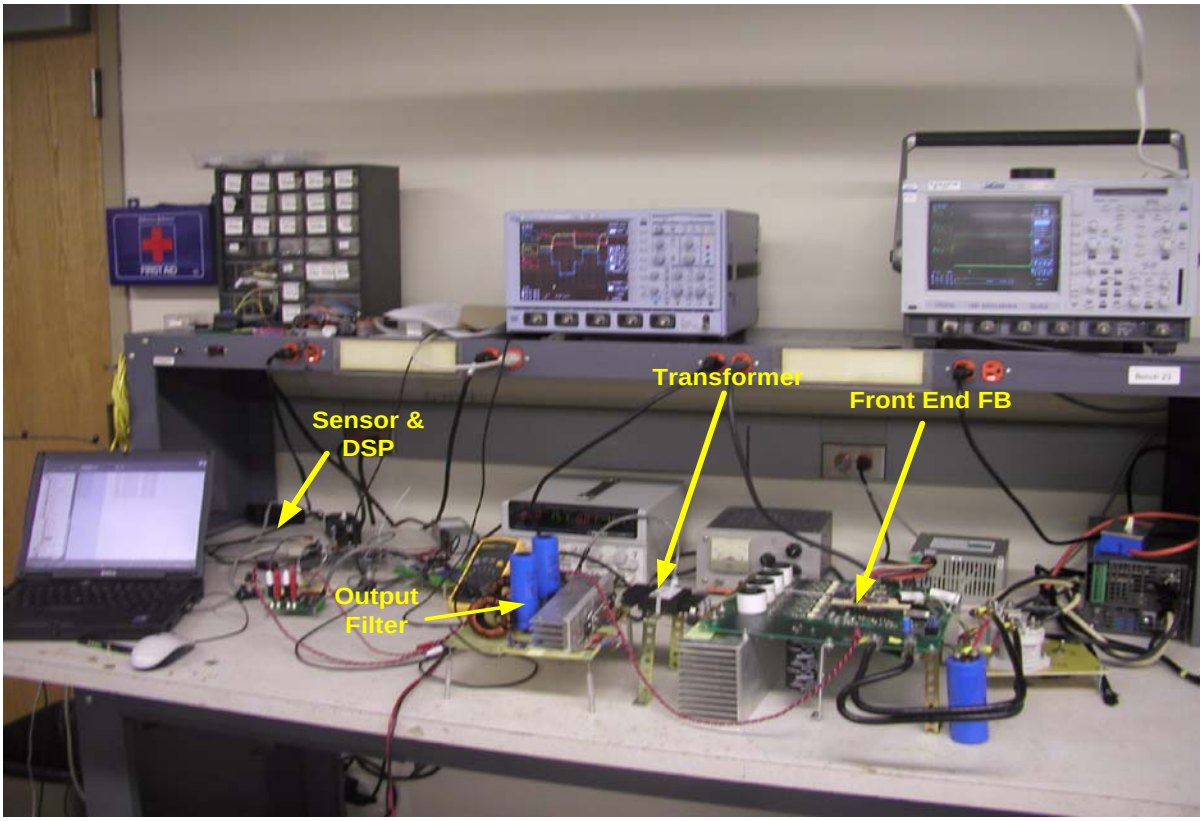

Figure 4.13. Bench Test Set-Up

4.7.1 Device Waveforms

Waveforms of some of the major components were captured to assure correct operation. Figure 4.14 shows the voltage across one of the leading leg Mosfets (channel 1) where it is visible that the device voltage goes to zero before the switch is turned on (channel 3), thus achieving zero voltage switching operation. The waveform in channel 2 represents transformer secondary voltage and channel 4 the input current. The phase shifting operation is shown in Figure 4.15. where the switching voltages are shown channel 1 and 2 and the voltage across the transformer is shownin channel 3. The phase-shift operation is clearly seen because, when the two voltages overlap, the transformer voltage is zero.

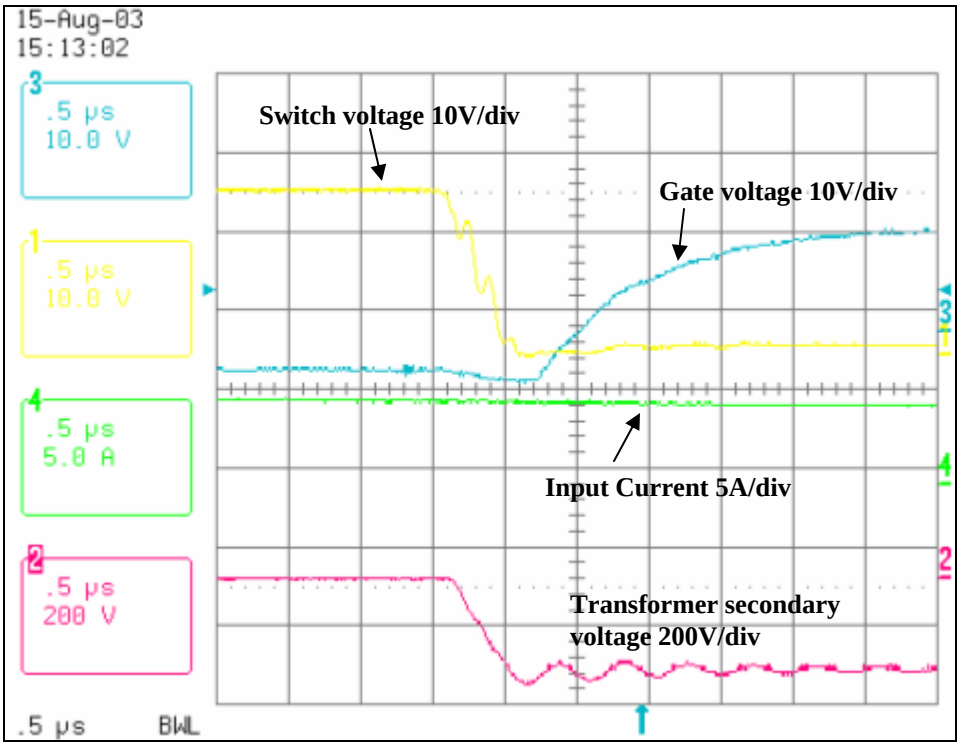

Figure 4.14. Switching Waveforms

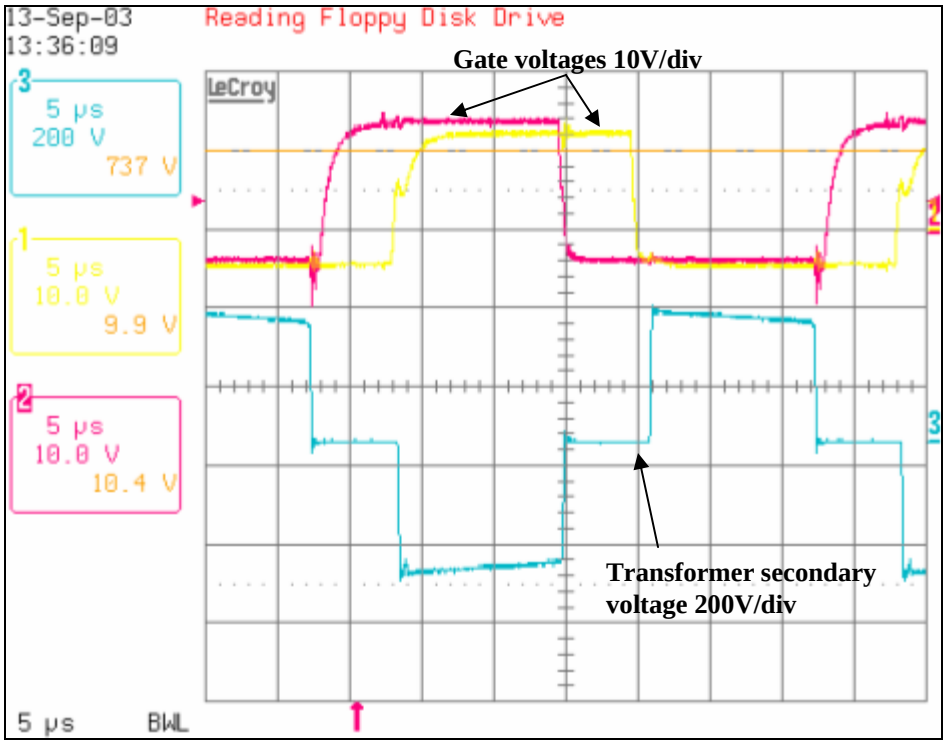

Figure 4.15. Phase Shifting Operation.

4.7.2 Output Waveforms

Some of the major output waveforms are shown Figure 4.16, where channel 1 represents the transformer secondary voltage, channel 2 the diode voltage, channel 3 one of the voltage across the switch and channel 4 the inductor current. The high frequency overshoot ringing on the diode voltage should be noted. This high frequency ringing sometimes can reach peaks large enough to cause failure of the rectifying diodes. The ringing is also transferred to the primary side of the transformer which could bring undesirable noise issues which result in glitches in the operation of the controller and the switching devices. Considering all the issues associated with the high frequency ringing, it is desirable that such issue be avoided.

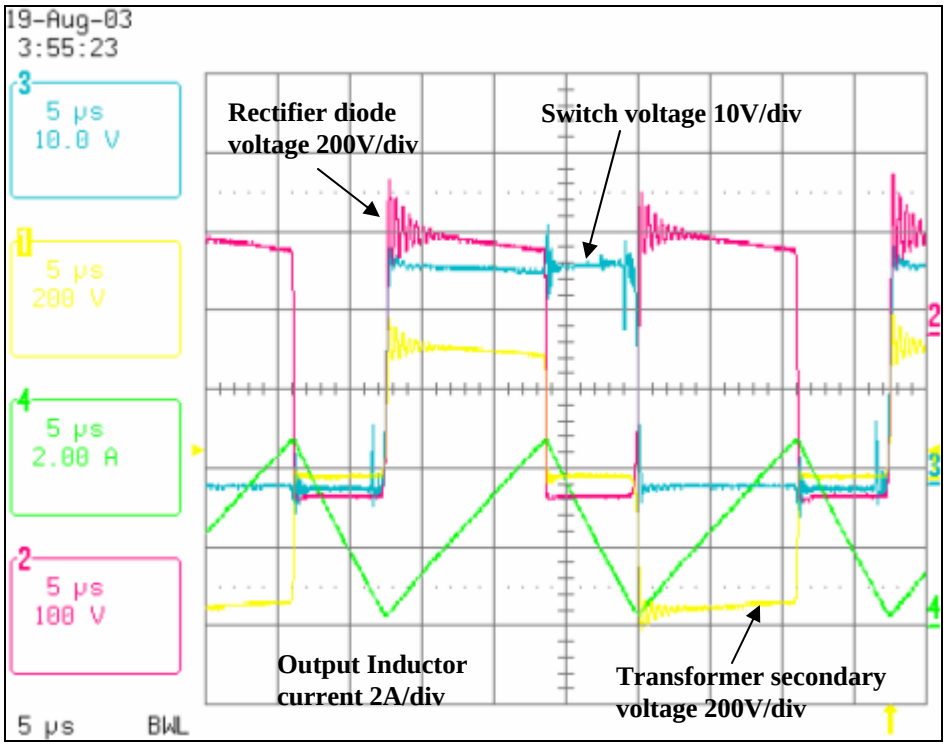

Figure 4.16. Output Waveform and diode ringing

4.7.3 Clamp Design

As it can be seen in Figure 4.16, the diode voltage waveforms contains some high frequency oscillations that it are due to the resonance created by the diode capacitance and the leakage inductance of the transformer. There are different solutions that can be chosen but the most effective and simple solution is the addition of a passive clamp. The passive clamp is made of a diode, resistor, and a high frequency capacitor. Such design is shown in Figure 4.17 where channel 1 represents the transformer secondary voltage, channel 2 the diode voltage, channel 3 one of the voltage across the switch and channel 4 the inductor current .

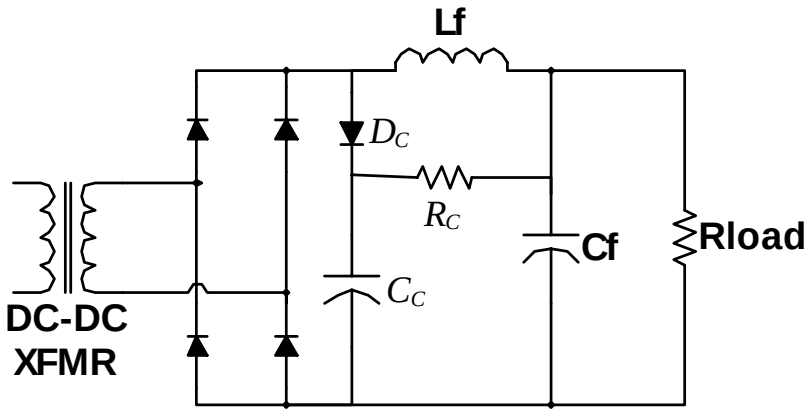

Figure 4.17. Passive Clamp Design

After measuring the ringing frequency being close to f_{ring} *=1 MHz,* the result was evaluated in the simulation and matched. From the simulations the values that were chosen were C_{clamp} = 220nF, R_{Clamp} = 10 kOhm. The results after adding the clamp are shown in Figure 4.18.

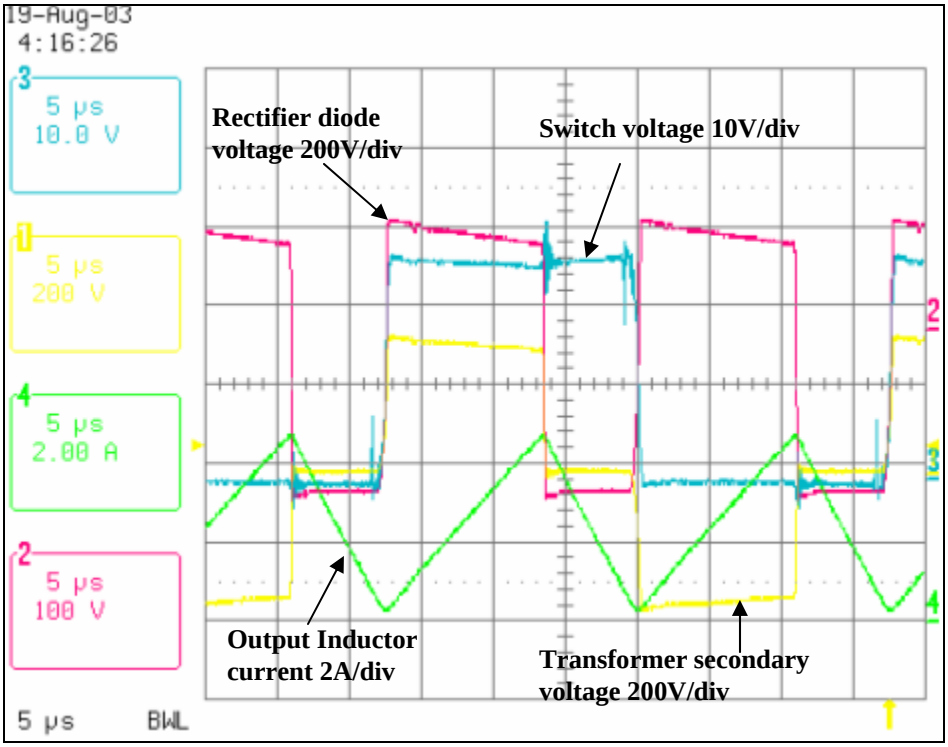

Figure 4.18. Output Waveform After Clamp is Included in the Circuit

4.8 Closed Loop Testing

After the controller is implemented in the DSP, the operation of the closed loop control was tested. There were a few glitches that had to be fixed in the DSP code. These had mostly to do with the fact that the DSP used was a fixed point DSP and some of the parameters in the controller implementation had some overflow issues. The testing was done using a resistive load bank and the load value was changed by using a dc contactor. The mechanical transients of the dc contactor were visible in this case and transmitted in the observed waveforms. The steady state waveforms for input voltages of 28V and 22V are shown in Figure 4.19 and Figure 4.20 respectively. These waveforms were consequently taken as the power supply output voltage was changed slowly.

In the following figures the channel 4 waveform represents the output voltage, channel 2 represents input voltage and channel 3 represents load current.

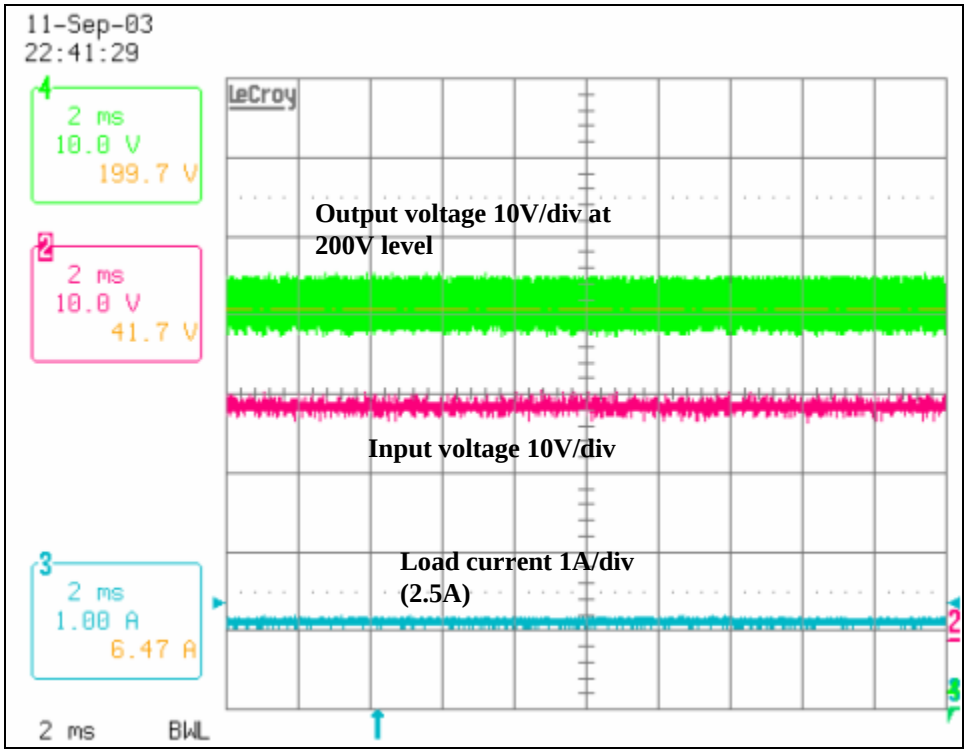

Figure 4.19. Steady State input voltage 28V output voltage 200V

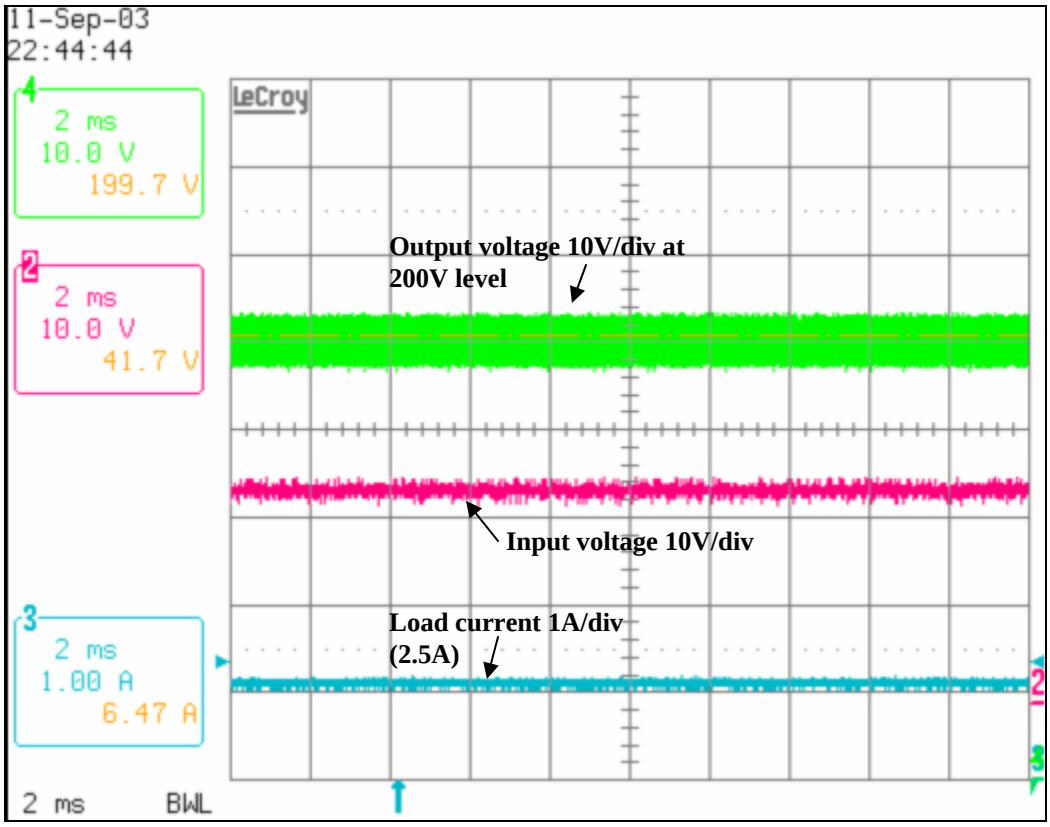

Figure 4.20. Input reduced to 22V and output at 200V

Load step operations were conducted by changing the load in 50% steps from 500W load to 750W loads. These load steps for load increase and decrease are shown in Figure 4.21 and Figure 4.22 respectively. As it can be seen from these results, the change in load does not have any large effect on the output voltage waveform and was very close to the simulation results that are shown in chapter 3.

The same tests were conducted when the input voltage was at 22V. These results are shown in Figure 4.23 and Figure 4.24. The results remain the same as in the 28V case. It should be noted that these tests were conducted in the CCM operation case.

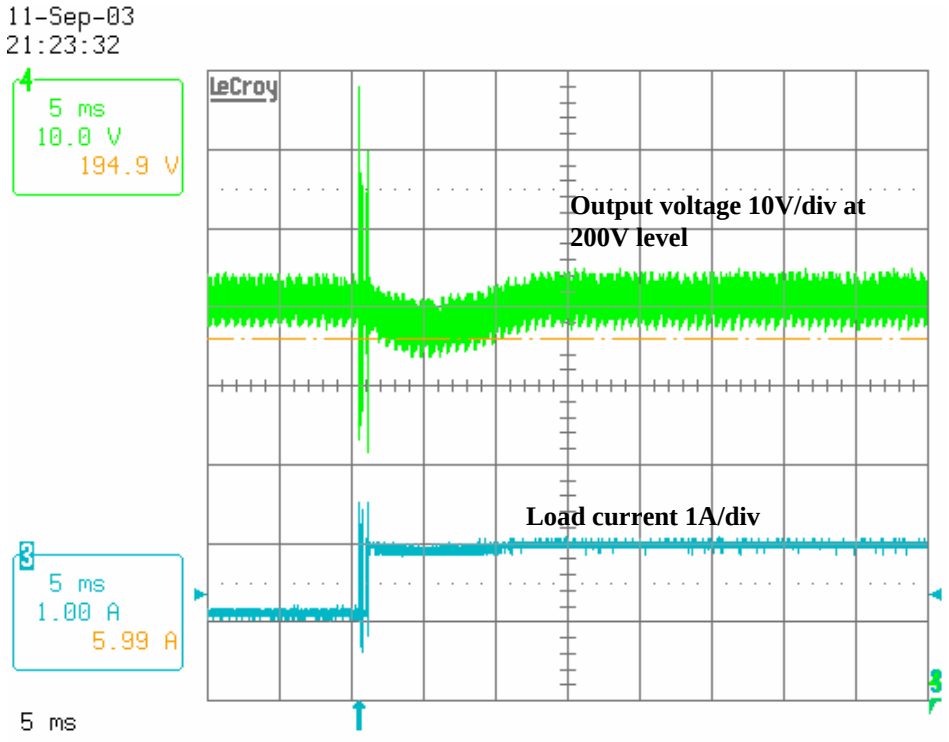

Figure 4.21. Load increase from 500W to 750W

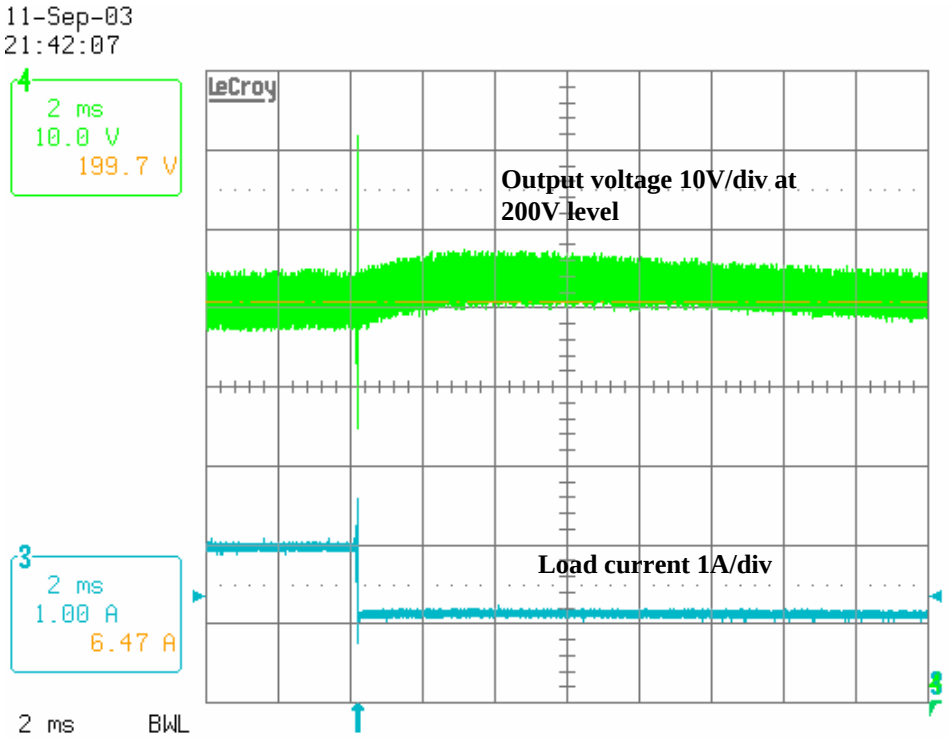

Figure 4.22. 50% load decrease

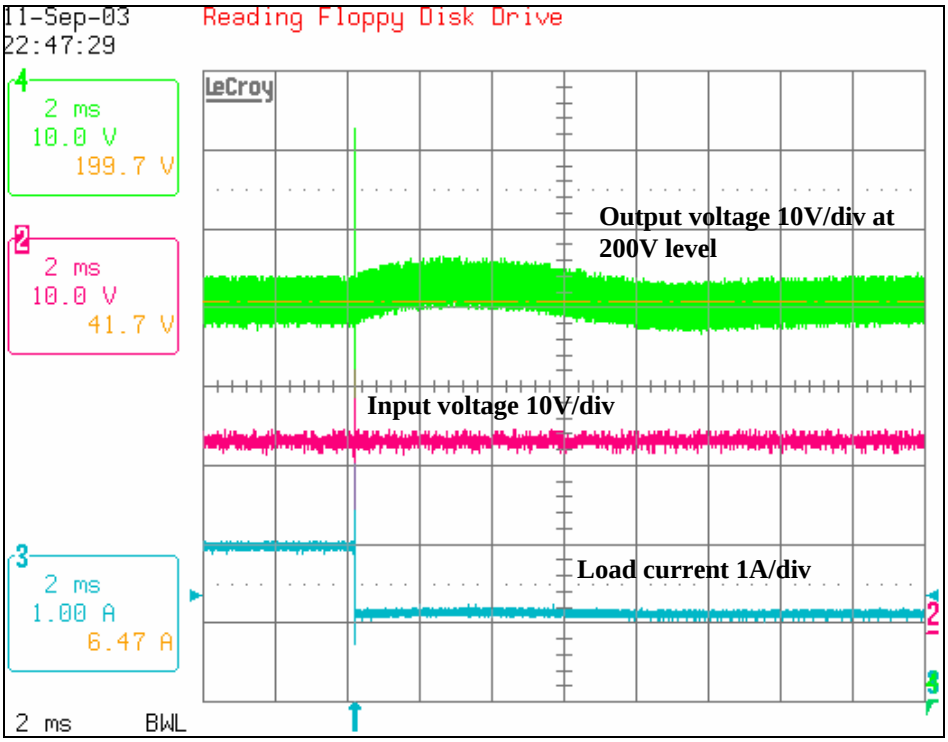

Figure 4.23. 50 % load decrease input voltage at 22V

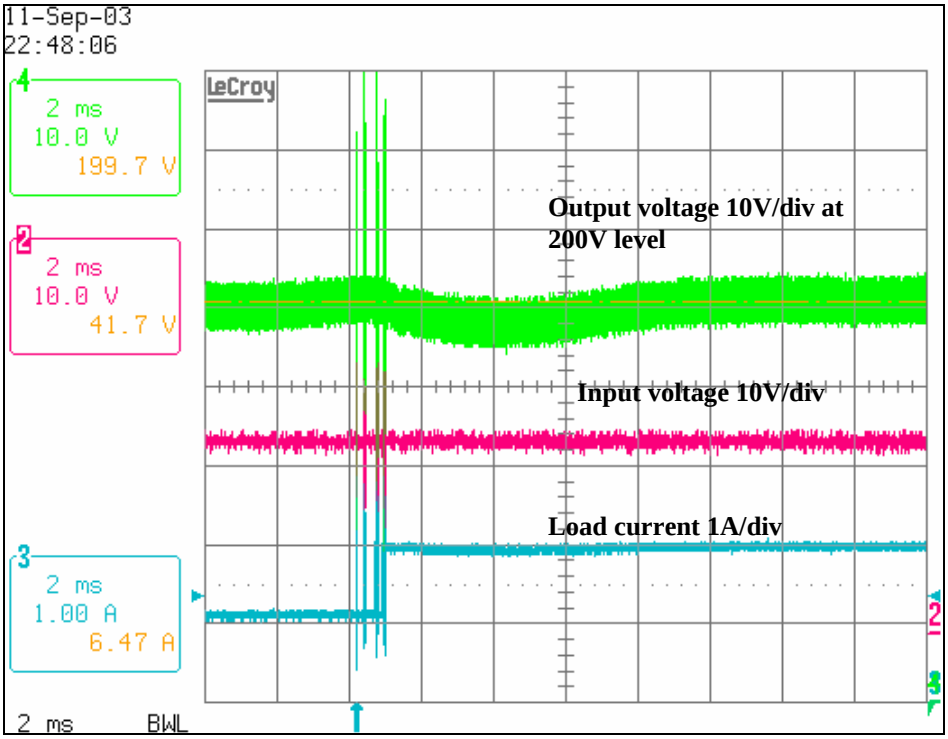

Figure 4.24. 50% load increase at 22V input.

In Figure 4.25 and Figure 4.27 the simulation and test results are shown at approximately the same scale in order to compare them. As it can be see from the simulation and the test the results coincide which makes the point that the modeling of the converter in Saber was done correctly and the experimental results verify this point

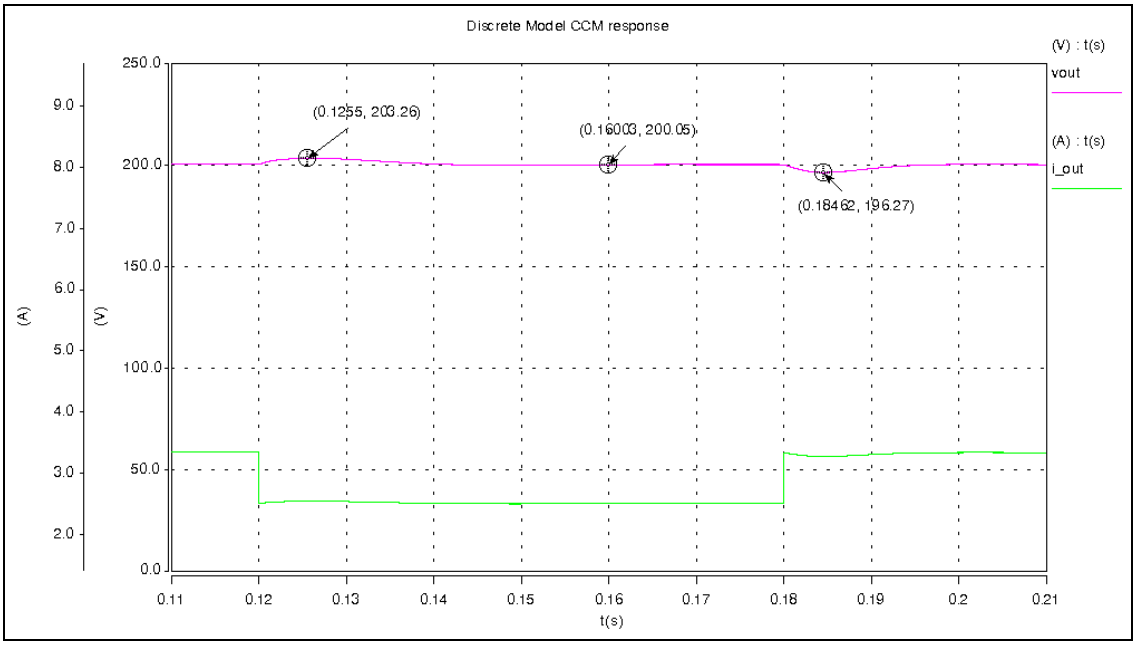

Figure 4.25. Simulation Load steps from 750W to 500W and from 500W to 750W

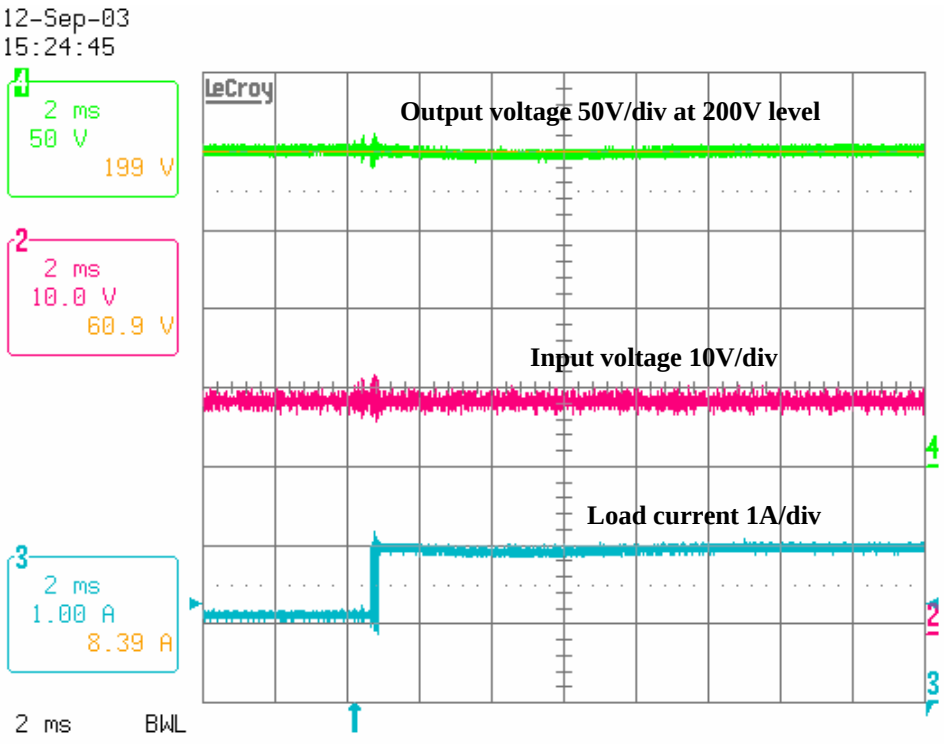

Figure 4.26. Load increase from 500W to 700 W

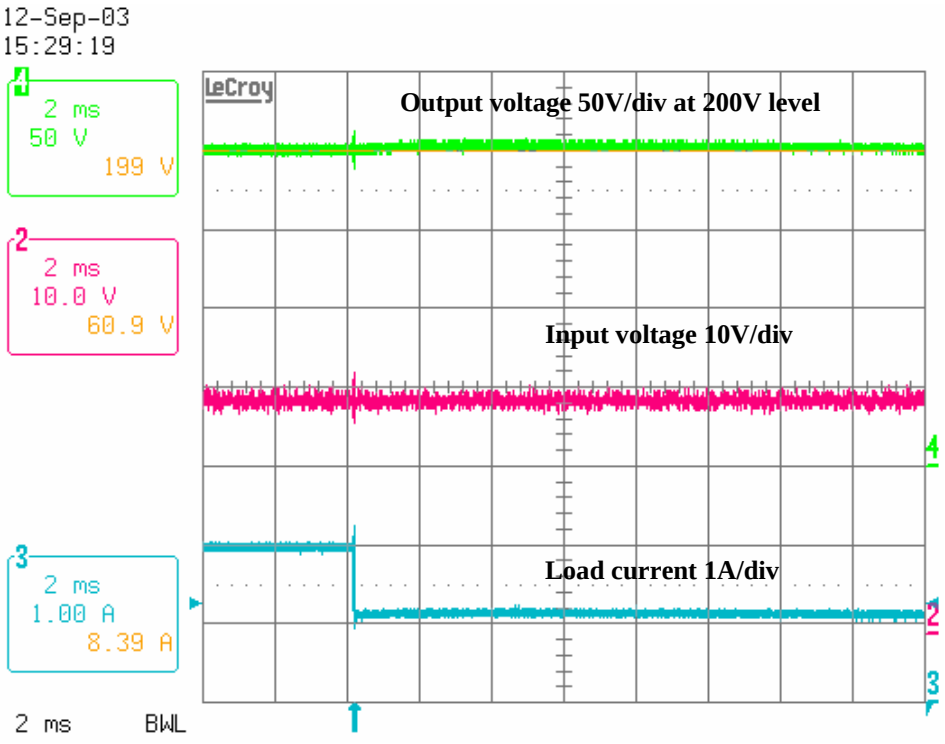

Figure 4.27. Load decrease from 700W to 500W

Chapter 5 - Conclusions

5.1 Summary

The work that was presented in this thesis shows the development of an adaptive control method for a front-end, full-bridge phase shifted PWM (PS-PWM) converter for a solid oxide fuel cell system. In the absence of a fuel cell a DC voltage source is setup in order to vary the voltage input from 22V to 28 V simulating the fuel cell source. The front-end converter supplies a stiff output bus voltage, which can be used by the other stages of stand alone fuel cell system. The control method implemented in this thesis provides not only input disturbance rejections from the input source, but also very good load dynamics. The need for such an advanced controller arises because of the non-linearity that accompanies the full bridge PS-PWM converter.

In the first chapter, different controller strategies were explored, including voltage and current mode based controls. The adaptive controller that was developed here makes use of a model of the system to be controlled, which is used in place of the system itself in order to provide estimates of the control state values.

The second chapter dealt with the issues that evolved when controlling the dc-dc converter and with the creation of models for different operating conditions. A relationship between input and output of the converter in discontinuous conduction mode (DCM) and continuous conduction mode (CCM) were developed. This relationship was then implemented in an adaptive control model, which was able to adjust to the appropriate operating mode based on only the information taken from the input and output voltage of the converter.

The modeling of the converter was founded on the small-signal and average models, providing a convenient and efficient way for studying the performance of the PS PWM converter. Next, in chapter 3, the controller was developed for the converter operating in DCM. The reason for this is that, at this operating mode (DCM), the converter's transient load and line responses suffer the most. After the controller was developed in DCM, the same model was simulated in CCM and the results were satisfactory. Simulations of the controller using the switching model were carried out in both DCM and CCM and it was shown that the results of the simulation closely matched

those of the average model. Using the switching model, some of the circuit parameters were fine-tuned and hardware implementation was carried out.

This control model was relatively simple to implement and required only two simple isolated voltage sensors. The adaptive control model offers high dynamic performance and superior line and load regulations. In chapter 4, the controller was implemented using a DSP for the CCM case only. The reason why the controller was not implemented in DCM was that the Texas Instruments DSP controller used, the LF2407, has a fixed-point processor and the amount of calculations involved in the DCM could not be handled by the DSP. Since the input current in the dc-dc converter was relatively large, the converter's operation in the continuous conduction mode was verified for loads up to 1 kW. The experimental results that were shown in chapter 4 for the transient load steps test was conducted at different input voltage levels and the results matched the simulations.

5.2 Contributions

This thesis has developed a novel approach in controlling a full bridge phase shifted converter. The main contributions are listed below:

- Adaptive controller developed with a high bandwidth
- Excellent load regulation and input line regulation.
- Controller has high bandwitdh comparable to current mode controllers.
- Simple to implement with only two inexpensive isolated voltage sensors.
- High noise immunity comparable to volteage mode controllers.
- It can be implemented for other types of convertes such as boost and buck

As shown by this report, the performance of the adaptive controller has a bandwidth comparable to current mode controllers, yet also has the noise immunity advantages of a voltage mode type of controller.

The performance is superior to a voltage mode controller and it uses the same controller for both discontinuous and continuous conduction modes. This adaptive method allows the same controller to be used for different operating conditions (DCM and CCM) and its promising results suggest that it can be applied to other types of

converters where circuit parameters, due to non-linearities, are not well defined. Overall, the results are satisfactory and matched the prediction from modeling and simulations.

5.3 Future work

As mentioned in the summary, this control method was developed for a front-end converter to be used in a variable low input voltage applications. Since a fuel cell source has these characteristics it would be beneficiary if this converter can be tested using a SOFC and evaluate its Since all of the control parameters were developed based on simulation results, it would be beneficial if some of the control transfer functions could be measured directly. Also, the operation in DCM needs to be verified and such controller needs to be implemented using a floating point DSP in order to account for the amount of calculations needed to implement the controller in such operating conditions.

The adaptive controller was implemented only for one single front end dc-dc converter, and for usage in higher power levels, as in the case of a multiphase dc-dc converter, the control method needs to be analyzed further. In order to complete the study, a variety of other controls methods need to be implemented and a comparison needs to be made in order to study the feasibility of the proposed controller.

This control method in principle opens the possibility to be implemented in other systems and it could provide a way for controlling other converter topologies.

Appendix

Saber Mast Listings

Digital Controller Listing

element template DIGITAL_CONTROLLER SYNC U X1 X2 X3 Y

```
input nu U
output nu Y
state nu X1, X2, X3
state logic_4 SYNC


{
        state nu XX1, XX2, XX3
        state nu XX1p, XX2p, XX3p, YYp

        when (dc_init ) {
                schedule_event( time, XX1, 0 );
                schedule_event( time, XX2, 0 );
                schedule_event( time, XX3, 0 );
        }

        when( event_on(SYNC)) {

                if( SYNC == l4_1 ){
                        XX1p = 176/256*XX1 +0*XX2 + 0*XX3 +653/256*U
                        XX2p = 1/256*XX1 + 1*XX2 + 0*XX3 +1/256*U
                        XX3p =0*XX1 + 1/256*XX2 + 1*XX3 +0*U
                        schedule_event( time, XX1, XX1p )
                        schedule_event( time, XX2, XX2p )
                        schedule_event( time, XX3, XX3p )
                        schedule_event( time, X1, XX1 )
                        schedule_event( time, X2, XX2 )
                        schedule_event( time, X3, XX3 )

                        #schedule_event( time, Y, YYp )
                }
        }

        equations{
                Y = -3/256*XX1 -24/256*XX2  -2/256*XX3
        }
```

}

DCM and CCM algorithm

element template ep_ctl1 nVg D V M DCMSCL CCMSCL

#...declaration of connections:

input nu nVg   #product of n and Vg as estimated by adaptive controller
input nu D   #current duty cycle
input nu V   #courrent output voltage
output nu M   #calculated M (gain from Vg to V)
output nu DCMSCL
output nu CCMSCL

export val nu kcrit
export val nu Mccm
export val nu Mdcm
export val nu k
export val nu v_clamp
export val nu k_clamp

#++
+++++++++

Start the definition

{
val nu Msel
val nu Hdcm
val nu Hccm
#++
+++++++++

values {
	if(V <= 1p){
		v_clamp = 1p
	}
	else if(V>=nVg){
		v_clamp = nVg
	}
	else {
		v_clamp = V
	}

```
    k = (D*D*nVg*nVg - D*D*nVg*v_clamp)/(v_clamp*v_clamp)

    if(k<=1f){
            k_clamp = 1f
    }
    else if (k>=100) {
            k_clamp = 100
    }
    else {
            k_clamp = k
    }



    Mdcm = (2*D)/(D+sqrt((D*D)+4*k_clamp))
    Mccm = D

    kcrit = 1-D

    if( k_clamp <= kcrit ){
    Msel = Mdcm
    }
    else{
            Msel = Mccm
}

    if (Mccm < 1m){

    Hdcm = Msel/1m

    }

    else {

    Hdcm = Msel/Mccm

    }

    ### inverse of Hdcm
    if (Msel < 1m){

    Hccm = Mccm/1m

    }
```

```
        else {

        Hccm = Mccm/Msel

        }

        }

equations {
        M = Msel
        DCMSCL = Hdcm
        CCMSCL = Hccm
        }

}
```

Integrator with limiting

```
element template int_ep  vin vout =  gain, low_lim, up_lim, initial

#...declaration of connections:

input nu vin
output nu vout
number gain
number low_lim
number up_lim
number initial

# Start the definition

{

val nu eff_inp
var nu ival

values {

        if (vout > up_lim & gain*vin >0) {
                eff_inp = 0
                }
        else if (vout<low_lim & gain*vin<0){
                eff_inp = 0
                }
        else {
        eff_inp = gain*vin
        }

        }

control_section {

        # Initial condition for v
        initial_condition(ival,initial)

           }

equations {
        ival: d_by_dt(ival)= eff_inp
        vout = ival
        }
}
```

Matlab Code

```
%
s = tf('s');
Ts = 1/40000;

%Gneeded= Gleadlag*GIntegrator*Gpi;
Gneeded= -1400*(s+1400)*(s+13)/(s+15000)/s/s
zpk(Gneeded)
%G = ss(Gpi*GIntegrator*Gleadlag);
%Gss = ss(G);
G= ss(Gneeded)
Gd = c2d(G, Ts);
Gdss = ss(Gd);


%define transformation matrix to balance matrices
T = 1000*diag([0.95,1.32,6.5]);
Gdss2 = ss2ss(Gdss, T)

%examine closed loop response
M = feedback( G*12*28,1, +1 );
step(-200*M);


%Discretize for implementation in a DSP using 9.7 signed format
Gdssc = Gdss2;
GSCL = 2^8;


Gdssc.a = round(Gdssc.a*GSCL);
Gdssc.b = round(Gdssc.b*GSCL);
Gdssc.c = round(Gdssc.c*GSCL);
Gdssc.d = round(Gdssc.d*GSCL);

Gdsstrunc = Gdssc;
Gdsstrunc.a = Gdssc.a/GSCL;
Gdsstrunc.b = Gdssc.b/GSCL;
Gdsstrunc.c = Gdssc.c/GSCL;
Gdsstrunc.d = Gdssc.d/GSCL;

bode(Gdsstrunc, Gdss2);
legend('Gdsstrunc','Gdss2');

%%%%%%%%%%%%%

intMult=1/s;
```

```
Multdig=ss(c2d(intMult,Ts));
Multdig2 = ss2ss(Multdig, 1/10);
```

DSP listings

std.c code

```
#include "std.h"

static inline void set(volatile unsigned int* address, char mask)
/* mask: 1 in bits you want to set */
{
        *address |= mask;
}

static inline void clear(volatile unsigned int* address, char mask)
/* mask: 1 in bits you want to clear */
{
        *address &= ~mask;
}

static inline char test(volatile unsigned int* address, char mask)
{
        return !((*address & mask)==0);
}
```

adc.c

```
#include "adc.h"
#include "pwm.h"
#include "std.h"
#include "io.h"

extern isr_20khz;

int initADC(void)
{
        *ADCTRL1  = 0x4000; /* reset ADC module */

        *ADCTRL1 = ADCTRL1_INIT_STATE;
        *ADCTRL2 =  ADCTRL2_INIT_STATE;
        *MAX_CONV = MAX_CONV_INIT_STATE;


        *CHSELSEQ1 = 0x0543; /* just 5, 4 and 3 */
        *CHSELSEQ2 = 0;
        *CHSELSEQ3 = 0;
        *CHSELSEQ4 = 0;
```

```
        /* set interrupt for 50kHz sampling */
        #define T1UFINT 0x200
        set(EVAIMRA,T1UFINT);     /* Enable Timer1(EVA) Underflow interrpt */
        set(IMR, BIT1);                 /* enable interrupts for timer1 */
        *T1CON = (FREE_RUN_FLAG + \
                                TIMER_CONT_UPDN + \
                                TIMER_CLK_PRESCALE_X_1 + \
                                TIMER_ENABLE_BY_OWN + \
                                TIMER_ENABLE + \
                                TIMER_CLOCK_SRC_INTERNAL + \
                                TIMER_COMPARE_LD_ON_ZERO );
//      *T1PR = 1000;
        *T1PR = 500;
        return 1;
}

int updateADC(void)
/* returns 1 if the conversion was ready, 0 if not */
{
        if((*ADCTRL2 & ADC_SOC_SEQ1) == 0) /* conversion done */
                *ADCTRL2 = ADC_SOC_SEQ1; /* start a new conversion */
        else return 0; /* not ready yet */

        return 1;
}
```

fir.c

```
#include "FIR.h"

void initFIR(FIRFILT_ORD10 *fir)
{
        fir->dbuffer_ptr=fir->dbuffer;
        fir->coeff_ptr=fir->coeff;
        fir->init(fir);
}
```

init.c

```
#include "init.h"
#include "pwm.h"
#include "adc.h"
#include "sgen.h"

extern ADC_SAMPLER ADC;
```

```
extern SGENT_1 sgen;

void init()
{

        asm (" setc INTM");/*Disable all interrupts                                    */
        asm (" clrc SXM");    /*Clear Sign Extension Mode bit
        */
        asm (" clrc OVM");    /*Reset Overflow Mode bit*/
        asm (" clrc CNF");    /*Configure block B0 to data mem.                        */

        WSGR=((BVIS<<9)+(ISWS<<6)+(DSWS<<3)+PSWS);
                                    /* set the external waitstates    WSGR            */


        *WDCR=((WDDIS<<6)+(WDCHK2<<5)+(WDCHK1<<4)+(WDCHK0<<3)+W
DSP);
                                    /* Initialize Watchdog-timer
        */

        *SCSR1= ((CLKSRC<<14)+(LPM<<12)+(CLK_PS<<9)+(ADC_CLKEN<<7)+
                  (SCI_CLKEN<<6)+(SPI_CLKEN<<5)+(CAN_CLKEN<<4)+
                  (EVB_CLKEN<<3)+(EVA_CLKEN<<2)+ILLADR);
                                    /* Initialize SCSR1      */

   *SCSR2 = (*SCSR2 | 0x004B) & 0x000F;
                  /*
                  bit 15-7     0's:   reserved
                  bit 6              1:       input qualifier
                  bit 5        0:     do NOT clear the WD OVERRIDE bit
                  bit 4        0:     XMIF_HI-Z, 0=normal mode, 1=Hi-Z'd
                  bit 3        1:     disable the boot ROM, enable the FLASH
                  bit 2     no change   MP/MC* bit reflects state of MP/MC* pin
                  bit 1-0      11:     11 = SARAM mapped to prog and data
                  */

        *MCRC =
((MCRC13<<13)+(MCRC12<<12)+(MCRC11<<11)+(MCRC10<<10)
                  +(MCRC9<<9)+(MCRC8<<8)+(MCRC7<<7)+(MCRC6<<6)
                  +(MCRC5<<5)+(MCRC4<<4)+(MCRC3<<3)+(MCRC2<<2)
                  +(MCRC1<<1)+MCRC0);
                        /* Initialize master control register C          */

        *MCRB = ((MCRB9<<9)+(MCRB8<<8)+
                  (MCRB7<<7)+(MCRB6<<6)+(MCRB5<<5)+(MCRB4<<4)+
                  (MCRB3<<3)+(MCRB2<<2)+(MCRB1<<1)+MCRB0);
```

```
                                    /* Initialize master control register B          */

          *MCRA =
((MCRA15<<15)+(MCRA14<<14)+(MCRA13<<13)+(MCRA12<<12)+
        (MCRA11<<11)+(MCRA10<<10)+(MCRA9<<9)+(MCRA8<<8)+
        (MCRA7<<7)+(MCRA6<<6)+(MCRA5<<5)+(MCRA4<<4)+
        (MCRA3<<3)+(MCRA2<<2)+(MCRA1<<1)+MCRA0);
                          /* Initialize master control register A          */

   /*** Setup the core interrupts ***/
   *IMR = 0x0000;                  /* clear the IMR register */
   *IFR = 0x003F;                 /* clear any pending core interrupts */
   *IMR = 0x0000;                  /* enable desired core interrupts */

   /*** Setup the event manager interrupts ***/
   *EVAIFRA = 0xFFFF;              /* clear all EVA group A interrupts */
   *EVAIFRB = 0xFFFF;              /* clear all EVA group B interrupts */
   *EVAIFRC = 0xFFFF;              /* clear all EVA group C interrupts */
   *EVAIMRA = 0;               /* disable all EVA interrupts */
   *EVAIMRB = 0;
   *EVAIMRC = 0;

   *EVBIFRA = 0xFFFF;              /* clear all EVB group A interrupts */
   *EVBIFRB = 0xFFFF;              /* clear all EVB group B interrupts */
   *EVBIFRC = 0xFFFF;              /* clear all EVB group C interrupts */
   *EVBIMRA = 0;               /* disable all EVA interrupts */
   *EVBIMRB = 0;
   *EVBIMRC = 0;

   initPWM(); /* setup PWM stuff go to pwm.c*/
   initADC(); /* setup analog to digital stuff. go to adc.c */
         initSGEN(&sgen);
         initPorts();  /* initialization of ports starts here go to io.c*/

   return;

}
```

inv_main.c

```
#include "f2407_c.h"
#include "sgen.h"
#include "pwm.h"
#include "adc.h"
#include "io.h"
#include "std.h"
```

```
#include "Qmath.h"

/* EVERYTHING DECLARED HERE, BEFORE MAIN, IS GLOBAL */

/* declare the parts of DSP used */
PWMGEN PWM = PWM_DEFAULTS;
ADC_SAMPLER ADC = ADC_DEFAULTS;
SGENT_1 sgen=SGENT_1_DEFAULTS;

/* bit deffinitions */
bit_io LED;
bit_io BB_sel;
bit_io GDreset;
bit_io DSP2_Fault;

/* global variables */
signed int sinRef;

void main()
{

        int i = 100;

        init();

   bit_off(LED); /* turn on the "no fault" led */

   bit_off(GDreset); /* reset gate drives */
   while(i--);
   bit_on(GDreset);
   bit_on(DSP2_Fault);

   PWM.mfunc_c1 = 0x0000;
   bit_off(BB_sel); /* boost */

   while(1)
        {
        /*
   /*   bit_off(BB_sel); /* boost */

   /*   PWM.mfunc_c2 = 0xFFFF - PWM.mfunc_c1;   */
   PWM.mfunc_c1 = 0;
   PWM.mfunc_c2 = 0xFFFF;

   }
```

```
}

void interrupt isr_20khz()
{
   /* sin wave refernce */
   static unsigned int H_Scale = 0x6E00; /*5500*/

         sgen.calc(&sgen);
         sinRef=sgen.out;

         /* do gain multiply, sinRef = 1.15, H_Scale = 1.15, result = unsigned, 0-1 */
   PWM.mfunc_c4 = norm((((signed long)(sinRef))*((signed long)(H_Scale))) >> 15);
         /* use oppisate sign signref signal */
   PWM.mfunc_c6 = ~PWM.mfunc_c4 + 1;

   updatePWM(&PWM);
set(EVAIFRB, BIT2); /* clear GPT2 interrupt */
}



void interrupt fault()
{
  bit_on(LED);  /* turn off the led */
  bit_off(DSP2_Fault);

  while(1); /* wait indefinitely */
}
```

io.c

```
#include "io.h"
#include "f2407_c.h"

char* var2hex(unsigned int var, char* buffer)
{
         char table[16] = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
         buffer[3] = table[(var & 0xF)];
         (buffer[2]) = table[(var & 0xF0)>>4];
         (buffer[1]) = table[(var & 0xF00)>>8];
         (buffer[0]) = table[(var & 0xF000)>>12];
         return buffer;
}

char* var2percent(int var, char* buffer)
{
```

```
   return buffer;
}

void initPorts()
{
        /* set port directions */
        *PADATDIR = *PADATDIR | 0xFF00; /* set to all outputs. Port A */
        *PCDATDIR = *PCDATDIR | 0xFF00; /* set to all outputs. Port C */

        //EP
        *PBDATDIR = *PBDATDIR | 0xFF00; /* set to all outputs. Port B */
}


#include "f2407_c.h"
#include "sgen.h"
#include "pwm.h"
#include "adc.h"
#include "io.h"
#include "std.h"
#include "Qmath.h"
#include "SS.h"

/* EVERYTHING DECLARED HERE, BEFORE MAIN, IS GLOBAL */

/* declare the parts of DSP used */
ADC_SAMPLER ADC = ADC_DEFAULTS;
SGENT_1 sgen=SGENT_1_DEFAULTS;
extern SS comp;
extern signed long gnDutyOut;
extern signed long gnVout;

/* global variables */
int sinRef;
```

main.c

```
int H_Scale = 0x1000;
int scale = 0;//0x7F00;
unsigned int Vdc = 60*50;
signed int in[500];
signed int out[500];

unsigned int PWM1_Duty;
unsigned int PWM2_Duty;
```

```
unsigned int PWM3_Duty;
unsigned int PWM4_Duty;
unsigned int PWM5_Duty;
unsigned int PWM6_Duty;
unsigned int PWMA_Period;
unsigned int PWMB_Period;

signed int gnOffset = 0x00000000;     /*offset between input and output*/

signed int gnTrigger;
signed int postctr;

//inline void UpdateController( )
void UpdateController( )
{
//          //24192
//          if( gnVoltageReference > (unsigned int)(comp.u[1])){
//                    if( gnX < gnHighLimitDuty ) {
//                              gnX++;
//                    } else{
//                              gnX = gnHighLimitDuty;
//                    };
//          }
//          else{
//                    if( gnX > gnLowLimitDuty) {
//                              gnX--;
//                    } else{
//                              gnX = gnLowLimitDuty;
//                    };
//          }
//
//          comp.y[0] = (unsigned int)(0xFFFF) - (unsigned int)(gnX);
}


//global constant integer Increment for triangle waveform
const unsigned int gcnIncrement = 30;
unsigned int gnIncreasing;
void main()
{
          unsigned int i = 100;
          gnOffset = 0;
   gnTrigger = 0;
   postctr = 128;
          initSS(&comp);
          init();       /* init routine called by main to initialize dsp. continue to init.c */
```

```
//          gnX = 1900;
```

ss.c

```
#include "ss.h"
SS comp;

Tstate gnYm;  // Model Plant output voltage
Tstate gnVin;  // Input voltage
Tstate gnVout;// Output voltage
Tstate gnN;     // Number of turns
volatile Tstate gnDutyOut;  // Output Duty

Tstate gnLowLimitDuty =  27;   //Limit of low duty cycle
Tstate gnHighLimitDuty = 200; //High limit for duty cycle

static const Tstate gnMultiplierMax = 4800;
static const Tstate gnMultiplierMin = 1600;

Tstate gnMultiplier;
Tstate gnMultiplierY;

//Tstate gnVref = 51200;
Tstate gnVref = 47000;
//for testing
int gnTesting;
Tstate gnVoutFake;


// gbEnableAdaptive = 0 when voltage feedback is smaller
// than the gnAdaptiveReference variable (converter is starting or stopping)
// gbEnableAdaptive = 1 otherwise
const Tstate gnEnableAdaptive = 40000;


void initSS()
{
   gnYm = 0;
   gnVin = 28*256;
   gnVout = 0;
   gnN = 12*256;
   gnMultiplier = 256*16;

          comp.u[0] = 0;

          comp.x[0] = 201;
```

```
        comp.x[1] = 1;
        comp.x[2] = -18576;

        comp.y[0] = 0;

        gnTesting = 0;
 }

void calcN()
{

//      gnMultiplier = gnMultiplier + ((gnVout-gnYm)>>12);

   if( gnVout >= gnEnableAdaptive ) {
      gnVin = comp.u[2]/28;
            gnVin = gnVin*10;
      //gnVin = 7182;


         if( gnVout == gnYm ) {
               //do nothing
         }else if( gnVout > gnYm ) {
               gnMultiplier++;
               gnMultiplier++;
         }else
         {
               gnMultiplier--;
               gnMultiplier--;
         }
   }

        if( gnMultiplier > gnMultiplierMax ){
               gnMultiplier = gnMultiplierMax;
        }

        if( gnMultiplier < gnMultiplierMin){
               gnMultiplier = gnMultiplierMin;
        }
//       gnMultiplierY = gnMultiplier;
        gnMultiplierY = ((gnMultiplier * 16)>>8);
}

Tstate gnDutyOffset = 55;
void calcSS()
{
        Tstate xx0, xx1, xx2;
```

```
  if( gnTesting > 0 ){
        gnVout = gnVoutFake;
  }
        calcN(); //calculating MRAC parameter

        //comp.u[0] = 24100;
        //comp.u[0] =  comp.u[0] - (gnYm>>1);

        comp.u[0] = gnYm-gnVref;

//      comp.u[0] = ((comp.u[0]>>8)*51);
        comp.u[0] = ((comp.u[0]>>8)*25);

        xx0 = ((176*comp.x[0])>>8);
        xx0 = xx0 + ((653*comp.u[0])>>8);

        xx1 = ((comp.x[0])>>8);
        xx1 = xx1 + (comp.x[1]);
        xx1 = xx1 + ((comp.u[0])>>8);
        xx2 = ((comp.x[1])>>8);
        xx2 = xx2 + (comp.x[2]);

        comp.y[0] = ((-3*xx0)>>8);
        comp.y[0] = comp.y[0] + ((-24*xx1)>>8);
        comp.y[0] = comp.y[0] + ((-2*xx2)>>8);

        comp.x[0] = xx0;
        comp.x[1] = xx1;
        comp.x[2] = xx2;



//      comp.y[0] = ( comp.y[0] > gnHighLimitDuty ? gnHighLimitDuty : comp.y[0] );
//      comp.y[0] = ( comp.y[0] < gnLowLimitDuty ? gnLowLimitDuty : comp.y[0] );

        gnYm = (comp.y[0]*gnVin)>>8;
        gnYm = (gnYm*gnMultiplierY)>>8;
                gnYm = (gnYm*gnN)>>8;

        //gnDutyOffset = gnVref*64/47000;
        //This part is fixed to: vo = 0.8d+0.08
        gnDutyOut = comp.y[0]*334>>9; //322  366
        gnDutyOut = gnDutyOut + gnDutyOffset; //27
        gnDutyOut = gnDutyOut << 8;
        gnDutyOut = (unsigned int)(0xFFFF) - (unsigned int)(gnDutyOut);
```

References

[1] *Future Energy Challenge* website http://www.energychallenge.org/

[2] *Future Energy Challenge 2003* Competition Specifications, March 3, 2003.

[3] Erickson, R. W., *Fundamentals of Power Electronics*, (Chapman & Hall, 1997).

[4] L. H. Mweene, C. A. Wright, and M. F. Schlecht, " A 1kW 500 kHz Front-End Converter for a Distributed Power Supply System", *IEEE Transactions on Power Electronics,* Vol. 6, No. 3, July 1991.

[5] S.Y.R. Hui, Y.K.E. Ho and H. Chung, "Modular single-stage, three-phase full-bridge converter with inherent power factor correction and isolated output", *IEE Proc. Electr. Power Appl.*, Vol. 145, No. 4, July 1999, pp. 407-414

[6] M.A. Laughton, "Fuel Cells", *Power Engineering Journal*, February 2002, pp.37-47.

[7] Dr. D. Boroyevich, *"Modeling and Control of Three-Phase PWM Converters"*, ECE 5274 Class notes, Spring 2002.

[8] Dr. F. C. Lee, *"Power Converter Modeling and Control*", ECE 5254 Class Notes, Fall 2002.

[9] UCC3895 data sheet, *BiCMOS Advanced Phase Shift PWM Controller*, Texas Instruments Jan. 2001.

[10] Middlebrook, R.D., "Modeling current-programmed buck and boost regulators", *IEEE Transactions on Power Electronics* ,Vol. 4 , Iss. 1 , Jan. 1989, pp. 36 – 52.

[11] Dr. J. S. Lai, *"Electric machines and Transients"*, ECE5334 Class Notes Fall 2001, Spring 2002

[12] *Ridley, R.B.;* "A new, continuous-time model for current-mode control", IEEE Transactions on Power Electronics ,Volume: 6 , Issue: 2 , April 1991 Pages:271 – 280

[13] Tang, W.; Lee, F.C.; Ridley, R.B., **"** Small-signal modeling of average current-mode control*"* *IEEE Transactions on Power Electronics* ,Vol. 8, Issue: 2 , April 1993, pp.112 – 119

[14] P. Midya, P. T. Krein, M.F. Greuel, "Sensorless Current Mode Control – An Observer – Based Technique for DC-DC Converters", *IEEE Transactions on Power Electronics,* Vol. 16, No. 4, pp. 522 – 526, July 2001.

[15] P. Midya, P.T.Krein, "Sensorless Current Mode Control: A Line Disturbance Immune Controller for DC to DC converters", *Professional Education Seminars, APEC 2003*, Vol. 2, February 2003.

[16] I.D. Landau, R. Lozano, and M. M'Saad, *Adaptive Control*, (Springer-Verlag London Limited, 1998).

[17] Landau, Yoan D., *Adaptive Control: The Model Reference Approach*, (Marcel Dekker, Inc.,1979)

[18] Pavier, M.; Sawle, A.; Woodworth, A.; Monteiro, R.; Chiu, J.; Blake, C "High frequency DC:DC power conversion: The influence of package parasitics," Applied Power Electronics Conference and Exposition, Vol 2, February 2003, pg. 699-704.

[19] Mohan, N., Undeland, T.M., Robbins, W.P., *Power Electronics: Converters, Applications, and Design, 2nd ed*. (John Wiley & Sons, Inc. 1995).

[20] J. A. Sabate, V. Vlatkovic, R. B. Ridley, F. C. Lee, and B. H. Cho, "Design considerations for high-power full-bridge zero-voltage-switched PWM converter," in *Proc. IEEE Applied Power Electronics Conference*, 1990, pp. 275-284.

[21] V. Vlatkovic, J. A. Sabate, R.B. Ridley, F. C. Lee, and B. H. Cho, "Small-signal analysis of the phase-shifted PWM converter," *IEEE Transactions on Power Electronics*, vol. 7, no. 1, pp. 128-135, Jan. 1992.

[22] F. S. Tsai, "Small-signal and transient analysis of a zero-voltage-switched, phase-controlled PWM converter using averaged switch model," *IEEE Transactions on Industry Applications*, Vol. 29, No. 3, pp. 493-499, May/June 1993.

[23] Norman S. Nise, *Control Systems Engineering*, 3rd Edition, John Wiley & Sons, Inc, 2000.

[24] Dr. W.T. Baumann, "Control Systems I", ECE 4405 Class notes Fall 2003

[25] T. A. Nergaard, J. F. Ferrell, L. G. Leslie, and J. S. Lai, "Design considerations for a 48 V fuel cell to split single phase inverter system with ultracapacitor energy storage," *in Proc. IEEE Power Electronics Specialist Conference*, 2002.

[26] INA122P datasheet, Texas Instruments, www.ti.com

[27] Franco, Sergio, Design with Operational Amplifiers and Analog Integrated Circuits, 3rd ed. (McGraw Hill, 2002).

[28] Ramakrishnan, S.; Steigerwald, R.L.; Mallick, J.A , "A comparison study of low-profile power magnetics for high-frequency, high-density switching converters"

Applied Power Electronics Conference and Exposition, 1997. APEC '97 Conference Proceedings 1997., Twelfth Annual ,Volume: 1 , 23-27 Feb. 1997 Pages:388 - 394 vol.1

[29] Payton Group International., "What is Planar Technology", *Expert Technical Report*, http://www.paytongroup.com/whatis.htm

[30] Quinn, Conner, et al., "A Review of Planar Magnetic Techniques and Technologies," *Applied Power Electronics Conference and Exposition*, Vol 2, March 2001, pg. 1175- 1183.

[31] Magnetics Inc, *Kool Mu Handbook*, www.mag-inc.com

[32] Texas Instruments Literature SPRU357B, "TMS320LF/LC240xA DSP Controllers Reference Guide - System and Peripherals," December 2001.

[33] Code Composer Studio, Version 4.12, Texas Instruments

[34] Gene F. Franklin, J. David Powell, Michael L. Workman, Digital Control of Dynamic Systems, 3ed ed Addison Wesley Longman, Inc. 1998

Vita

Elton Pepa was born in Tirana, Albania in April 1974. He finished high school in his hometown and started his pursuit of a degree in electrical engineering at Tirana Polytechnic University in Tirana, Albania. In fall 1997, he transferred to Virginia Polytechnic Institute and State University, where he received a Bachelor of Science degree in Electrical Engineering. Upon his graduation in May 1999, he joined Black and Decker Corp. in Towson, Maryland as a design electrical engineer where he worked in electrical machine design. In fall 2001, he decided to go back to Virginia Tech to pursue his master's degree in Electrical Engineering with a concentration in power electronics. As a graduate student, Elton worked as graduate research assistant under the supervision of Dr. Jason Lai in both CPES and FEEC labs. After graduation, he joined Aker Wade Power Technologies in Charlottesville, Virginia where he currently works as a Power Electronics Engineer.