1-25-2017

Flexible PFC Control Employing Adaptive Gain, Mode Estimation, & Reactive Power Compensation

Joshua M. Ivaldi
University of Connecticut - Storrs, joshua.ivaldi@uconn.edu

Flexible PFC Control Employing Adaptive Gain, Mode Estimation, & Reactive Power Compensation

Joshua Ivaldi

B.S. University of Connecticut, 2013

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2017

APPROVAL PAGE

Masters of Science Thesis

Flexible PFC Control Employing Adaptive Gain, Mode Estimation, & Reactive Power Compensation

Presented by

Joshua Ivaldi, B.S.

Major Advisor: _____

Dr. Sung-Yeul Park

Associate Advisor: _____

Dr. Ali M. Bazzi

Associate Advisor: _____

Dr. Yang Cao

University of Connecticut

2017

Contents

1 Introduction

The challenge of achieving acceptable power quality has become increasingly difficult; nonlinear loads and loads with poor input power factor operate in an uncertain manner. Insufficient reactive power support and the presence of line current harmonics result in accelerated component aging, increased power losses, decreased line capacity, and degraded system stability [1-4]. Additionally, for electric utility customers, these power quality issues present problems in the form of reduced appliance efficiency and reliability; whereas, industrial consumers are affected by poor power quality penalty costs, electromagnetic compatibility issues, and increased equipment or production downtime [5]. Environmental standards and power quality regulations have become increasingly stringent as a result of these problems and in turn, have accelerated the development of high performance power conversion systems that support efficient and stable power delivery.

Additional financial burdens are imposed on utility companies because flexible AC transmission systems (FACTS) must be installed. FACTS serve as distributed reactive power generators; for example, the Static Synchronous Compensator, Unified Power Quality Conditioner, or Static VAR Compensator are implemented with the goal of supplementing the reactive power consumption of elements present in the local distribution network [6-8]. Active power filters, which attenuate line current harmonics produced by nonlinear loads, are also implemented in support of high power quality systems [9]. Electric utilities face the problem of ensuring the stability of a difficult to predict system with the fundamental expectation to consistently respond to consumer energy demands [10]. Price scheduling, resource planning, and acquisition of distributed generation systems have been suggested as large scale load management options [11].

Given the difficulty of maintaining stability for large scale energy distributors, many studies in the power electronics area have investigated the benefits of localized

reactive power support and harmonic current filtering, often through renewable or reserve energy storage systems [12-18]. Consumer devices supporting localized compensation have been shown to be conducive to maintaining system stability without significantly affecting the performance of power conversion systems [19-21].

AC-DC rectification is a common power transformation in AC power distribution systems, often requiring dedicated circuits with front-end active waveshaping in order to meet harmonics regulations and standards such as IEEE 519, IEC 1000-3-2, or EN 61000 [22-24]. This process is required for DC loads: computer server hardware; home and industrial appliances; PHEV battery storage systems; uninterruptible power supplies, and nonlinear loads (e.g., arc furnaces and variable speed motor drives)[25]. **For passive and active circuits, the nature of this power conversion process presents several key issues in mitigating network pollution while maintaining sufficient performance:**

1. The propensity for fast-scale instability and chaotic behavior [26-28].
2. Inflexible and slow in response to load fluctuations.
3. Degraded input current quality due to circuit nonlinearity.
4. Phase shifting due to the presence of reactive elements.

Extensive research has been applied to active power factor correction (PFC) circuits in order to address and resolve the aforementioned problems. The primary goal is to develop a controller which makes the nonlinear load appear as linear from the perspective of the utility line. However, it is difficult to achieve a solution that addresses all of the intricacies underpinning the nonlinearity of the input current while offering fast and flexible operational capabilities. PFC research typically focuses on the minimization of these shortcomings through innovative techniques: discrete control methods that improve the response of low bandwidth DC regulators to load transients and input current regulation, and the modification or interleaving of existing

topologies for improved performance and efficiency [29-34]. Despite the availability of performance improvement strategies, there is little focus on the enhancement of transmission system efficiency as a whole from the perspective of PFC converters [35]. Although it conflicts with their intended purpose, the input power factor can be intentionally degraded and the reliability and performance of nearby loads and the transmission system can be improved through reactive power support and harmonic current cancelation. This available capacity can be used as a resource which mutually benefits utility companies and consumers.

PFC boost converters are excellent candidates for this strategy, as they are typically employed in consumer and industrial applications, have generally greater peak input current compared to that of the buck and buck-boost topologies, and offer acceptable input current linearity. **This thesis proposes a flexible PFC control algorithm which enables stable operation over a wide range of operating conditions while enhancing network power quality. The contents are presented as follows:**

1. A brief topological comparison of boost PFC circuits in Section 2.

2. Time and frequency domain analysis of the bridgeless boost PFC operating in continuous and discontinuous conduction modes in Section 3.

3. Development of an adaptive PFC control algorithm which adjusts compensator gains with respect to estimated transfer function characteristics and the predicted operating mode in Section 4.

4. Line current harmonic and reactive power compensation is added to the control algorithm and evaluation of power network performance is shown in Section 5.

2 Overview: Boost PFC Circuits

2.1 Boost Topologies

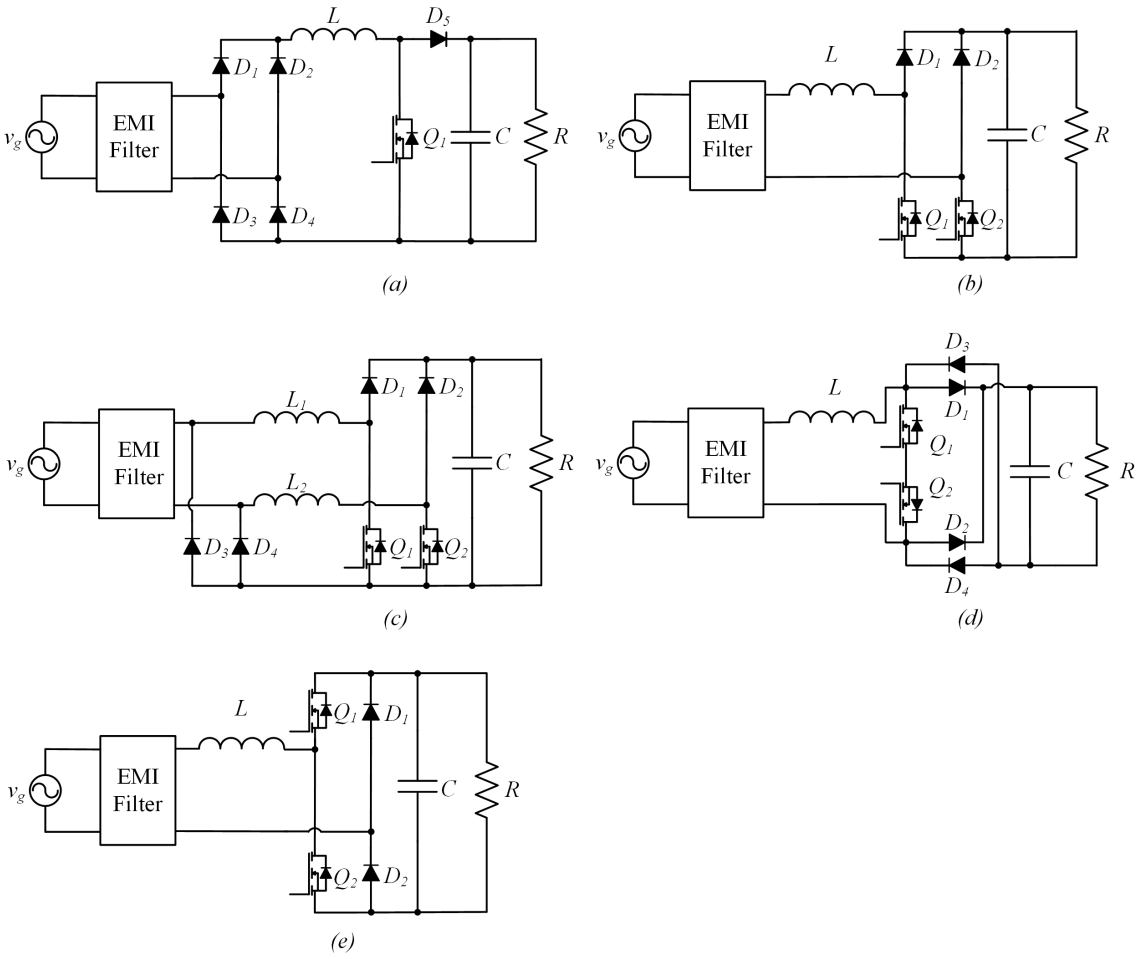


Figure 1. Boost PFC topologies, (a) conventional boost PFC, (b) bridgeless boost PFC, (c) semi-bridgeless boost PFC, (d) bidirectional switch boost PFC, and (e) totem-pole bridgeless boost PFC.

The conventional PFC boost converter, shown in Figure 1a, operating in continuous conduction mode (CCM) is one of the most commonly utilized PFC solutions for applications requiring regulated DC with power consumption greater than 200W. The boost PFC is frequently employed as a voltage pre-regulator circuit in multi-stage converters. The current paths during converter operation are shown below in Figure 2.

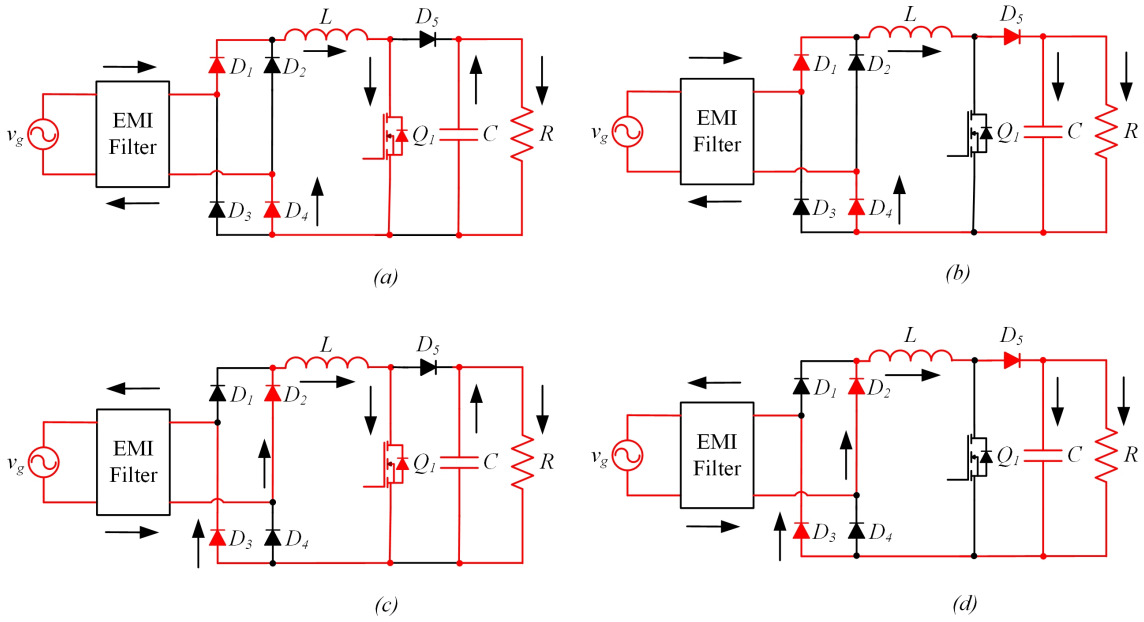


Figure 2. Conventional boost PFC (a) positive half cycle, charging inductor, (b) positive half cycle, discharging inductor, (c) negative half cycle, charging inductor, (d) negative half cycle, discharging inductor.

Due to the connection to the rectifier side ground during all portions of the line and switching cycles, this topology produces low common mode noise. The rectifier bridge is comprised of slow recovery diodes, $D_1 - D_4$ in series with a DC-DC boost circuit. In an effort to reduce the conduction losses, the bridgeless boost power factor correction converter (BBPFCC) has been proposed (pictured in Figure 1b). The BBPFCC and other similar variants were created to increase the efficiency or performance of the rectification process. The motivation behind modifying the conventional boost PFC topology is to reduce the conduction losses by eliminating the fast recovery diode, D_5 in the current path during the switch off state. In this case, D_1 and D_2 are fast recovery diodes due to their connection to the transistors. The operation of the BBPFCC is shown in Figure 3.

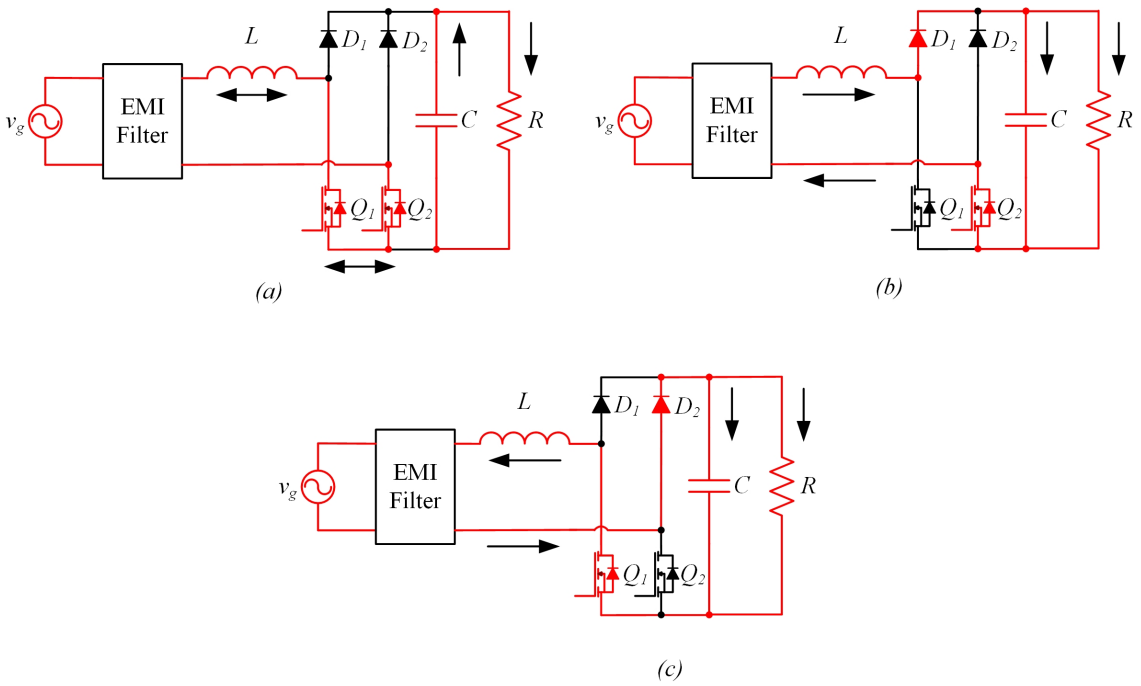


Figure 3. Bridgeless boost PFC (a) charging inductor, (b) positive half cycle, discharging inductor, (c) negative half cycle, discharging inductor.

The BBPFCC transistors can be controlled in two ways: 1) Q_1 and Q_2 share the same switching pattern, which permits a more cost effective gate driving circuit, but due to unnecessary gate driving, increased switching losses are incurred, or 2), in each line half cycle, one transistor will control the current in the inductor while the other transistor remains on, acting as a synchronous rectifier. The second strategy requires an additional gate driving circuit but has the advantage of reduced energy losses from excess gate driving as well as potentially reduced conduction losses [36]. If the source-drain voltage is sufficiently low, the conduction losses can be reduced at light load conditions because the current return path will conduct through the FET channel instead of the antiparallel diode [37].

While the reduced conduction losses are attractive, the elimination of the rectifier bridge diode results in a more demanding EMI filter design. Since the positive line terminal is no longer always directly connected to the DC side ground, increased common mode (CM) noise is generated when compared to the conventional boost

PFC. This is because the parasitic drain to source capacitances appear as pulsating voltage sources relative to the line. In order to reduce the CM noise, the topologies depicted in Figure 1c-e have been proposed.

The semi-bridgeless boost PFC and bidirectional switch boost PFC feature the addition of slow recovery diodes, $D_3 - D_4$, which ameliorate the CM noise problem by providing low frequency current return paths back into the line; however, these topologies suffer from greater implementation costs relative to the BBPFCC. The switching operation of the semi-bridgeless and bidirectional switch circuits are shown in Figures 4 and 5,

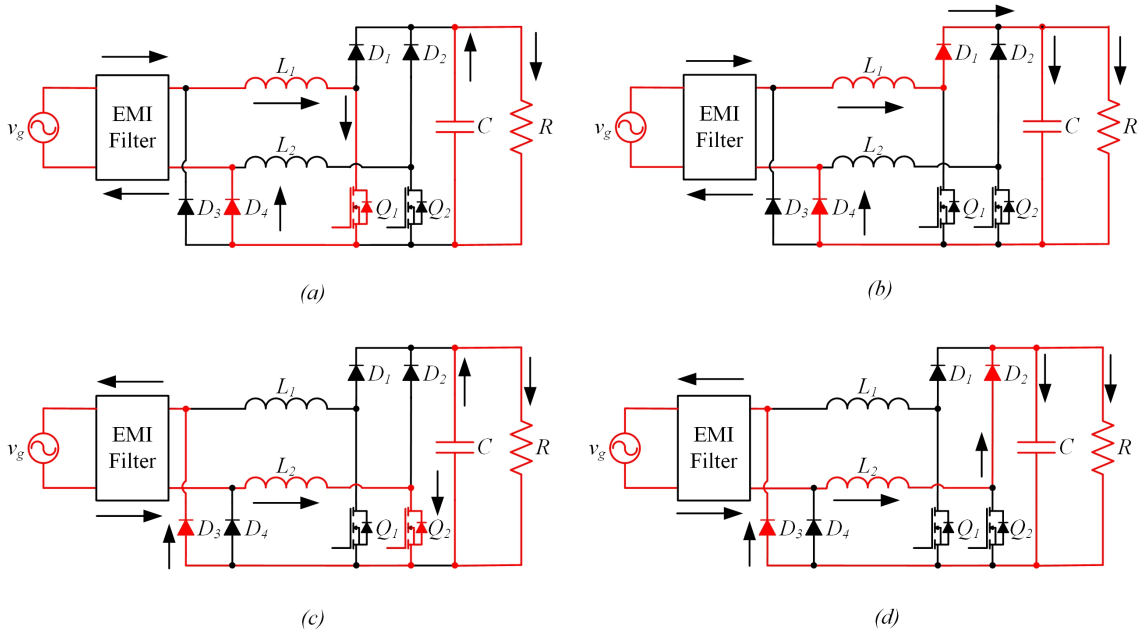


Figure 4. Semi-bridgeless PFC (a) positive half cycle, charging inductor, (b) positive half cycle, discharging inductor, (c) negative half cycle, charging inductor, (d) negative half cycle, discharging inductor.

Designers of semi-bridgeless circuits often couple L_1 and L_2 to serve as a CM choke. Each boost circuit operates during one half of a line cycle, which enhances thermal performance [38].

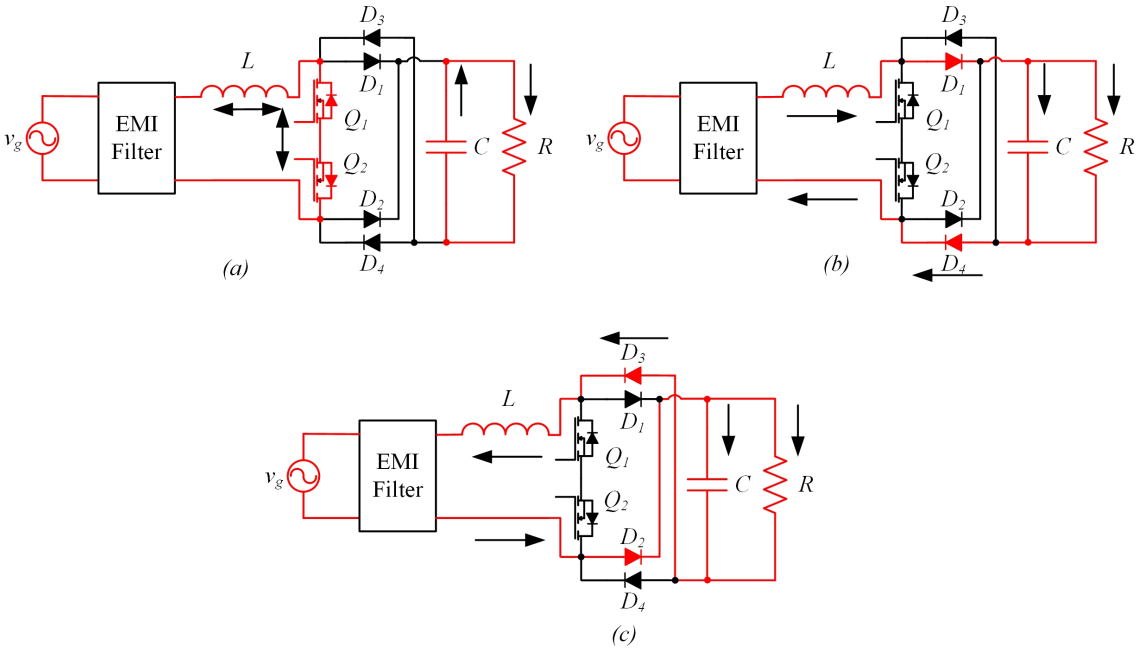


Figure 5. Bidirectional switch PFC (a) charging inductor, (b) positive half cycle, discharging inductor, (c) negative half cycle, discharging inductor.

Similar to the semi-bridgeless circuit, the bidirectional switch boost PFC solves the CM noise problem since AC source is connected to the DC side ground during all portions of the line cycle. The major drawback of this topology is the increased complexity of the gate driving circuit.

Lastly, the totem-pole boost PFC has been presented as a solution to the CM noise problem but is generally suitable for low power, high frequency DCM/CrCM operation due to the reverse recovery characteristics of the semiconductor components and PWM pattern [39]. The operation of the totem pole topology is shown in Figure 6.

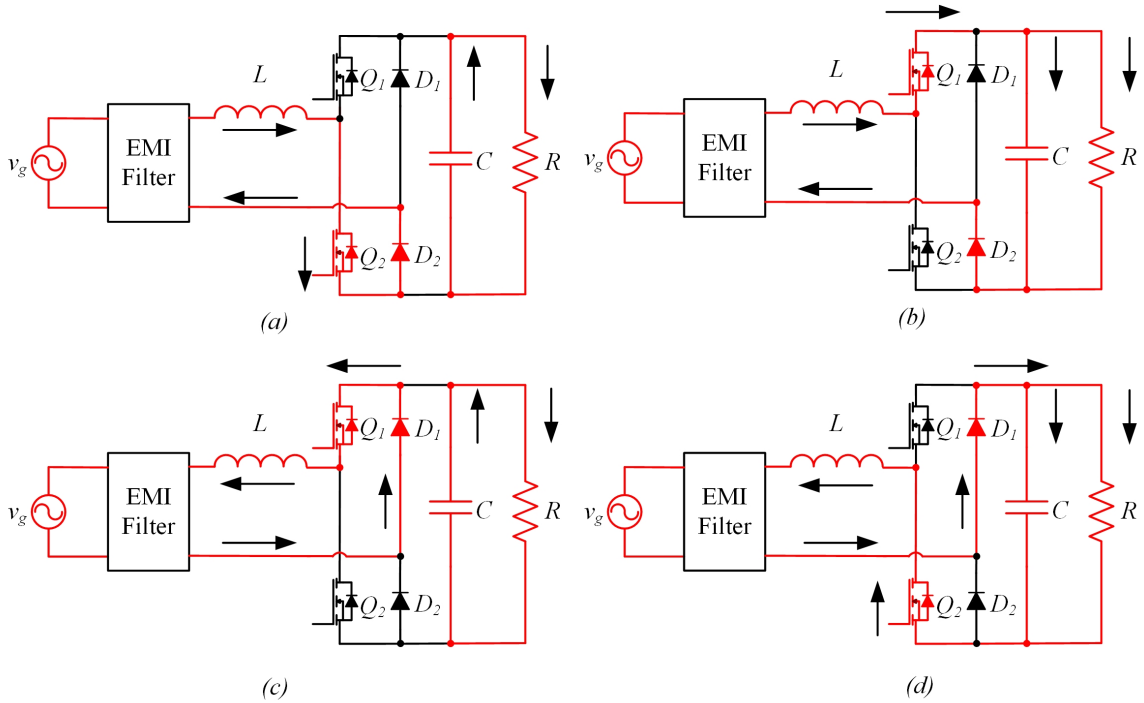


Figure 6. Totem pole PFC (a) positive half cycle, charging inductor, (b) positive half cycle, discharging inductor, (c) negative half cycle, charging inductor, (d) negative half cycle, discharging inductor.

The transistors in the left leg employ complementary switching to charge and discharge the inductor. This topology has found increasing utility with the development of GaN devices which feature extremely low Q_{rr}, and other wide bandgap semiconductor devices.

2.2 PFC Controllers

In this section, the functionalities and merits of controllers for CCM and DCM are discussed. A disadvantage of the rectification process is that it results in a low frequency output voltage ripple occurring at twice the line frequency. As a result, the voltage control loop is designed with a low crossover frequency of 10-15 Hz so that the peak current reference does not become distorted with this harmonic component; this restriction exacerbates the response to transient conditions. The controller design is

additionally complicated by the presence of the right half plane zero (RHPZ) which occurs near the control bandwidth in CCM.

For the aforementioned reasons, the discontinuous conduction mode (DCM) is preferable but only utilized in applications below 200W since the peak input current will not be great enough to damage components. In DCM, the control scheme can be simplified to a single loop control structure based on the output voltage since the input current amplitude follows the input voltage with relatively low nonlinearity [40,41]. This operation mode is preferable because the RHPZ advantageously shifts into the high frequency range and switching losses are reduced due to zero current switching. However, the disadvantage of DCM operation is that the EMI filter design becomes complicated as a result of the high peak input current.

First, the average current control scheme will be discussed. The key waveforms and control structure are shown in Figure 7,

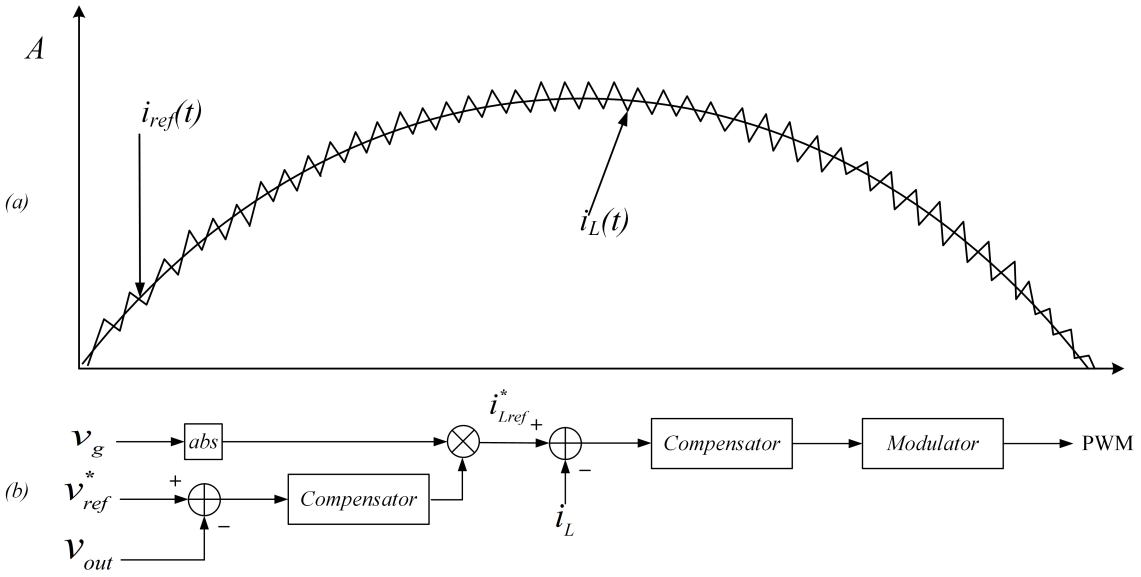


Figure 7. CCM average current control, (a) inductor current and reference waveforms, (b) average current controller

For the sake of example, the input voltage, v_g, output voltage reference, v^*_{ref}, output voltage, v_{out}, and inductor current, i_L, are assumed to be already scaled. Here,

the controller multiplies the input voltage with the peak current reference provided by the voltage control loop. In many cases, this multiplier incorporates the inverse of the squared line voltage so that the performance of the outer loop is independent of the line voltage amplitude. The average current reference signal is compared with the sensed inductor current, and the amplified error signal is used to drive a pulse width modulator. This method is most popular for CCM control due to its excellent input current tracking and low total harmonic distortion; nevertheless, the design procedure is more complicated than other methods due to the dual loop control structure.

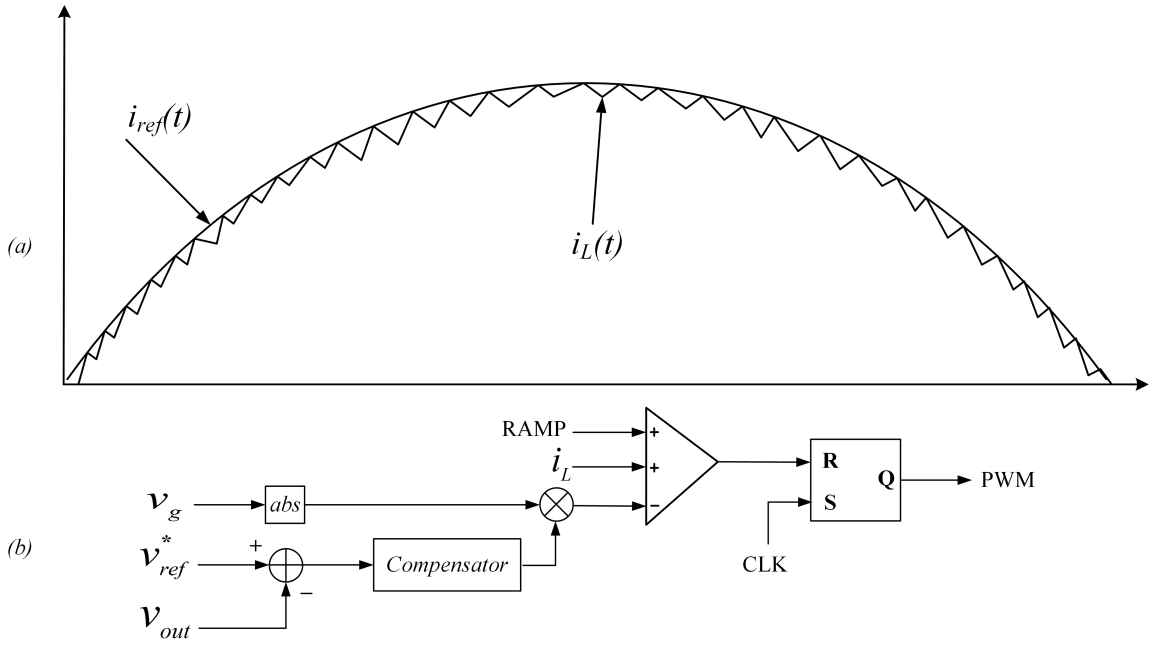


Figure 8. CCM peak current control (a) inductor current and reference waveforms, (b) peak current controller

Figure 8 depicts the peak current controller. Here, the peak current reference is generated by the voltage loop, and when the sensed current exceeds the reference, the PWM output is reset. At the beginning of the next switching cycle, a clock resets the latch output.

Typically, additional ramp signal compensation is required in peak current control in order to avoid stability problems associated with subharmonic oscillations. These

occur when the inductor ripple current does not return to its initial value at the beginning of the next switching cycle and the duty ratio is greater than 50%. The merit of this control scheme is its simplicity; however, it is limited to few applications due to its input current distortion and potential for instability.

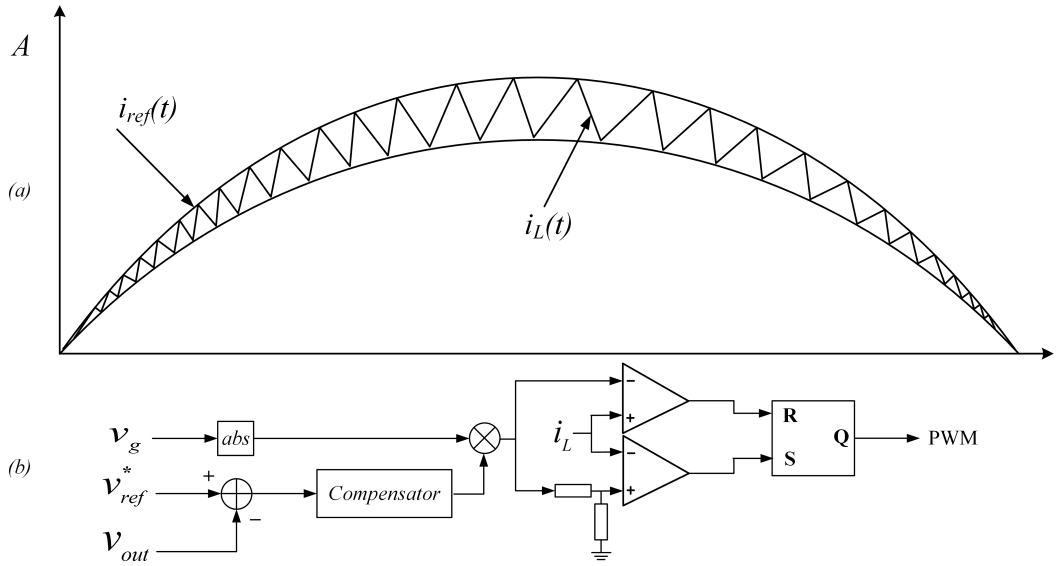


Figure 9. CCM hysteresis current control (a) inductor current and reference waveforms, (b) hysteresis current controller

Figure 9 shows hysteresis control of the inductor current. As usual, the outer loop generates the current reference signal, which is then used as a peak and minimum reference. When the current falls below the minimum reference, the PWM output is set high. Likewise, when the current exceeds the maximum reference, the PWM output is set low. This method utilizes variable frequency control, complicating the filter design procedure.

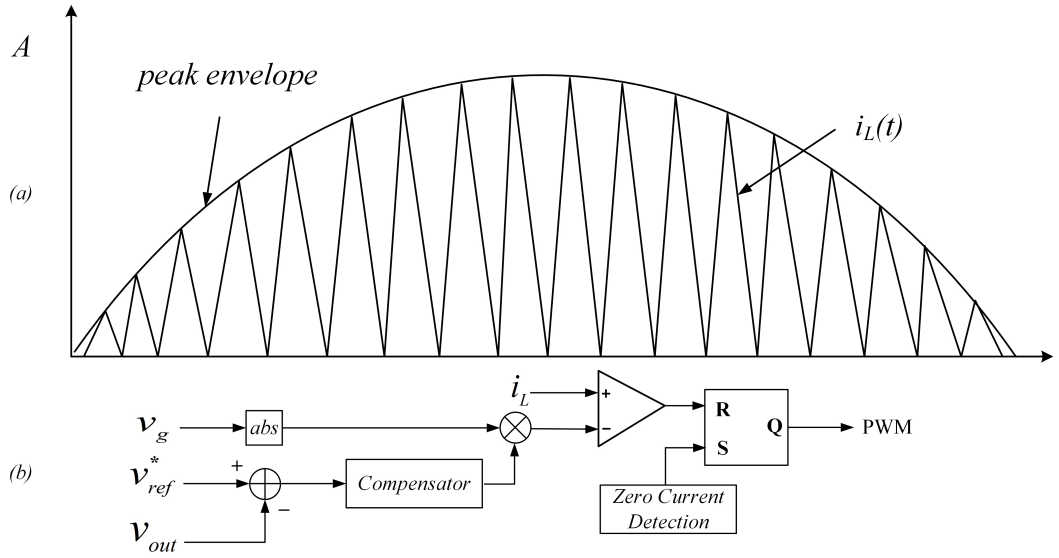


Figure 10. CrCM peak current control (a) inductor current and reference waveforms, (b) peak current controller

Figure 10 depicts the peak current control method used in the critical conduction mode. This method is similar to that of peak current control in continuous conduction mode, but instead, a zero current detector triggers the pulse of every switching cycle. Like the hysteresis controller, this scheme is also variable frequency.

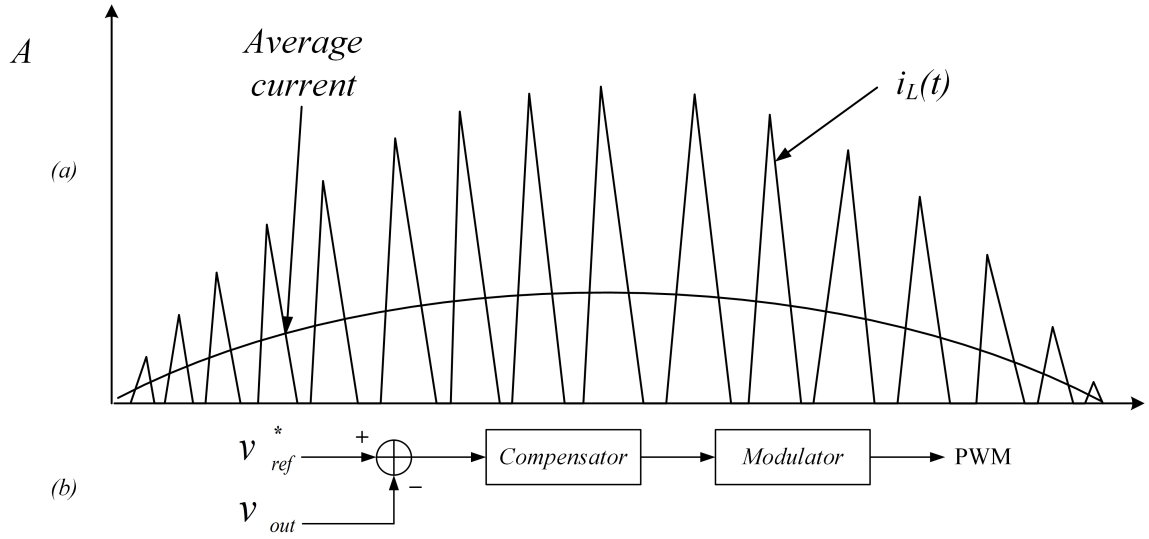


Figure 11. DCM control (a) inductor current and reference waveforms, (b) discontinuous mode controller

Shown in Figure 11 is the most common method for discontinuous conduction mode control. When possible, it is preferred due to its simplicity. Here, the outer voltage loop provides a nearly constant control signal, and the inductor current draw is inherently sinusoidal and in phase with the input voltage waveform.

3 Derivation of Linearized Models

The parameters of the converter designed in this thesis are shown in Table I,

TABLE I

L	C	R_{min}	S	v_g	f_{sw}	v_{out}
$750\mu H$	$450\mu F$	75Ω	$2.13kVA$	$230V_{RMS}$@$50Hz$	$100kHz$	$400V$

L and C are chosen such that the switching ripple characteristics are acceptable and the corner frequency begins well before the switching frequency. This is beneficial because the switching frequency is significantly attenuated in the current loop in CCM. The apparent power rating is arbitrary, but selected to emulate a large home or industrial appliance later in Section 6. The switching frequency is chosen with respect to the typical capabilities of modern digital signal processors. The line voltage amplitude and frequency are chosen since they are common values for single phase AC mains and the input current will be of lower amplitude than its $120V_{rms}$@$60Hz$ counterpart for an equivalent power transfer condition. For the sake of example, a small conversion ratio is chosen so that the system will operate in both CCM and DCM.

The topology chosen for this thesis is shown in Figure 12,

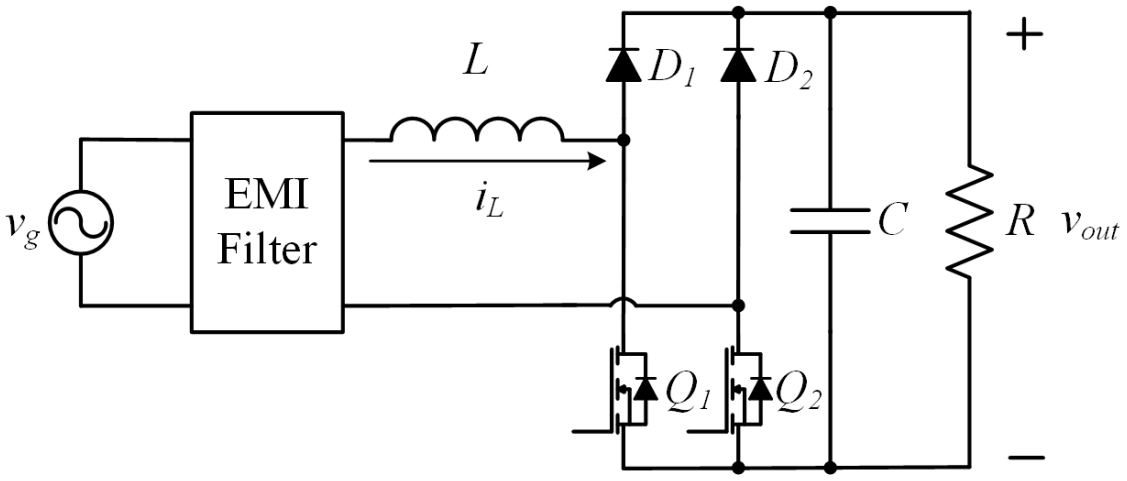


Figure 12. Bridgeless boost power factor correction converter

3.1 The Continuous Conduction Mode

For applications requiring power transfer greater than 200W, CCM type control is generally chosen due to restrictions on the peak current which would occur in DCM. In CCM, the converter must regulate the amplitude and phase of the input current in reference to the output voltage error and line voltage, respectively. Since the transistors drive the states of both the input current and output voltage, the output voltage regulation performance is ultimately a compromise.

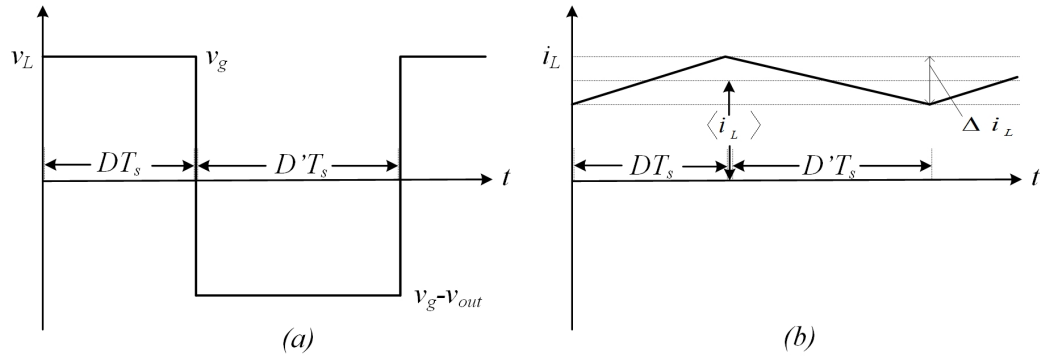


Figure 13. a) Inductor voltage waveform, b) inductor current waveform.

The general operation of the system is as follows, 1) the low bandwidth outer control loop regulates the DC voltage, typically through a linear compensator, 2) the amplified error signal generated by the outer loop creates the peak current amplitude reference, 3) this signal is multiplied by a sinusoidal reference signal based on the line voltage amplitude, and 4) the high bandwidth inner loop drives the input current to follow the waveshape of the input voltage such that the system appears as a resistive load from the perspective of the utility line.

With the convention that the switching period, $T_s = DT_s + D'T_s$, the inductor waveforms are depicted in Figure 13. In this mode, the average current, $\langle i_L \rangle$, is greater than one-half of the ripple current component, Δi_L. Assuming that the input voltage and output voltage are constant over one switching period, integration of the inductor voltage yields,

$$\left\langle v_L \right\rangle = v_g D T_s + (v_g - v_{out}) D' T_s = 0 \tag{1}$$

Where D represents the ratio of switch on time and D' represents the ratio of switch off time during one switching period. When solving for the time-varying duty ratio, $D(t)$, it is evident that the minimum output voltage is equal to the input voltage. As the input voltage is sinusoidal and rectified, the theoretical duty ratio waveform follows,

$$D(t) = 1 - |v_g(t)|/v_{out} \tag{2}$$

Therefore, the PWM control signal is sinewave modulated at the line frequency (shown in Figure 14). In practice, the duty ratio waveform will differ from its depiction in Figure 14, as the current controller must not only drive the inductor current to the desired amplitude for output voltage regulation, but also track the difference in phase between the reference current waveform and the inductor current.

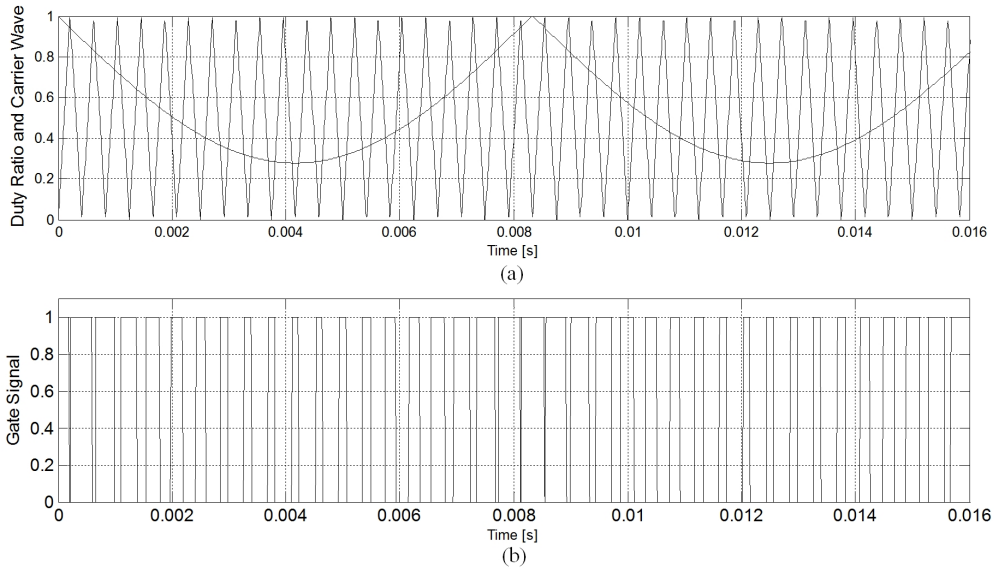


Figure 14. (a) Theoretical duty ratio and carrier wave signals, (b) PWM signal.

Since the input current cannot instantaneously change its amplitude, accurate tracking of the phase difference is critical to the performance of the PFC regulator. For this reason, the converter operating in CCM requires a high bandwidth current controller to follow the current reference signal accurately. In order to further improve the current regulator performance, a common practice is to implement a feedforward compensator that generates the theoretical duty ratio. The merits of this technique are discussed in Chapter 4.

In terms of energy transfer from the input to the output, the switches and fast recovery diodes conduct 180 degrees out of phase, and the inductor discharges during the switch off state. The consequences of energy transfer during this interval are depicted in Figure 15.

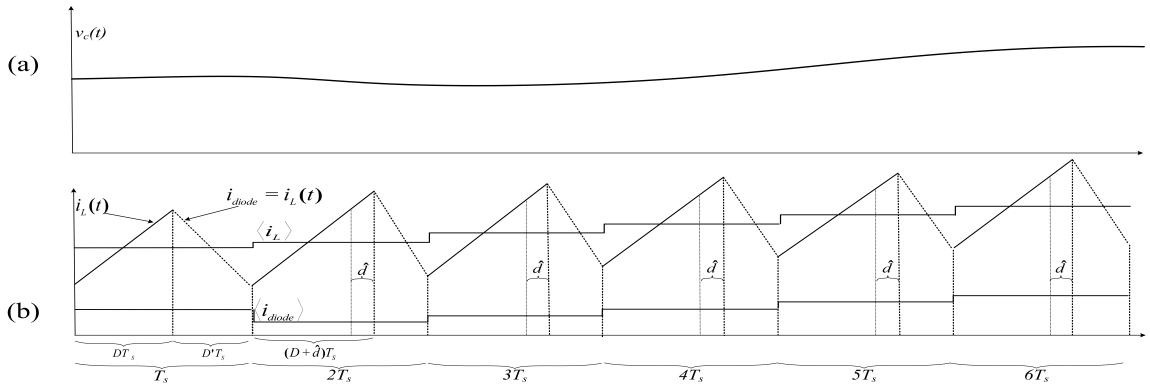


Figure 15. (a) Output voltage response to an increase in D, (b) Inductor and diode current relationship: Instantaneous diode current is depicted as the thick dotted line; instantaneous inductor current is both the solid and thick dotted line; average inductor and diode currents are depicted as thick solid lines.

A disturbance in the system triggers a small change, $\hat{d}$, in the steady state duty ratio D during $2T_s$. The initial effect of this disturbance would result in a greater conduction period in the transistor, initiating a corresponding cumulative increase in the average inductor current in the following switching periods. As a result of an increase in DT_s, the average diode current initially decreases from $2T_s$ to $5T_s$ due to

the reduction in the off time conduction period and only slightly higher peak current value. The temporarily reduced average diode current causes a dip in the output voltage with a delayed recovery due to the response of the slowly increasing average inductor current; This initially appears to the controller as a change in the opposing direction from the reference value. As such, the increase in DT_s indeed results in an increased output voltage due to the greater peak diode current, but the response is delayed in phase by the slow increase in average inductor current. This behavior is the time domain equivalent of a right half plane zero (RHPZ) in the complex frequency domain and ultimately limits the response time of the output voltage to changes in the duty ratio.

In medium to heavy load conditions, the circuit will operate primarily in CCM. Figure 16 depicts the DCM operation region and cusp distortion in this condition. The circuit operates in DCM about the Zero Crossing Point (ZCP) due to the inability to charge the inductor current to a peak value such that it will discharge over the course of $D'T_s$. The cusp distortion following the ZCP is due to the inductor's inability to instantaneously change its current amplitude to follow the reference signal [42].

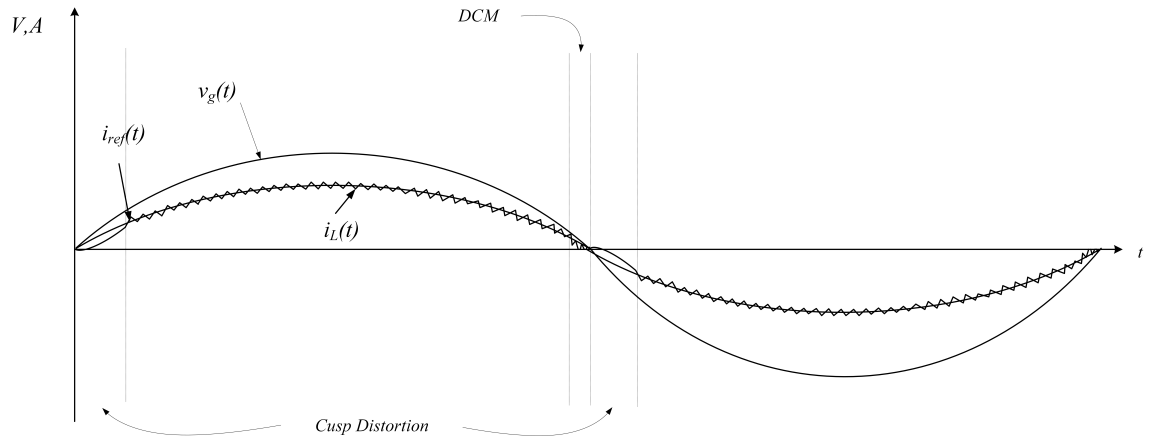


Figure 16. Inductor current waveforms in CCM.

The design process for power converters usually includes the characterization of the nonlinear converter dynamics through small signal linearization. Although these

descriptions are not entirely representative of the system dynamics, they are sufficient for gaining insights regarding selection of compensator gain values. In this case, the simultaneous regulation of the input current and output voltage requires a dual loop control scheme. The system, depicted in block diagram form, is shown in Figure 17.

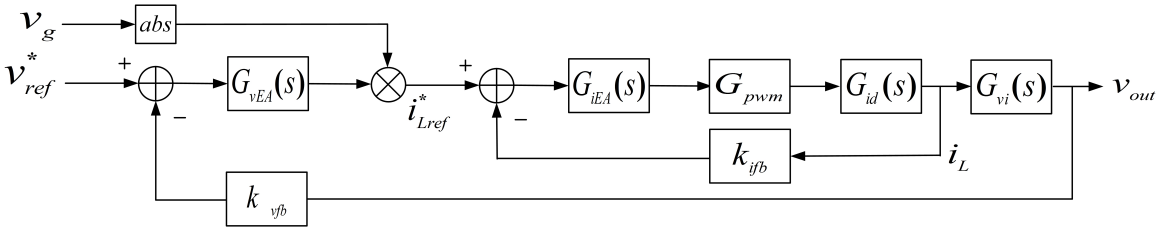


Figure 17. Control block diagram for CCM.

Figure 17 represents the simplest control structure for CCM PFC converters. The outer control loop generates the output voltage error signal. Since there is an output voltage ripple component at twice the line frequency, a common practice is to implement a low pass filter in the feedback loop in series with k_{vfb}, attenuate this frequency with the compensator, or incorporate a predicted ripple voltage value to cancel the effects of the ripple [43-46].

These strategies enhance the performance of the input current regulator, as the ripple component's effects are attenuated when generating the current reference signal amplitude. The voltage error signal determines the peak current amplitude through an error amplifier, $G_{vEA}(s)$, which is multiplied by the amplitude of the input voltage in order to generate a current reference waveform resulting in unity input power factor. The inner control loop regulates the input current against the current reference value through the linear amplifier, $G_{iEA}(s)$, and pulse width modulator, G_{pwm}. The PWM drives the state of the inductor current with respect to the control to inductor current transfer function, $G_{id}(s)$, and in turn, the inductor current drives the state of the capacitor voltage through the inductor current to output voltage transfer function, $G_{vi}(s)$. The two transfer functions cascaded in series results in the control to output voltage transfer function, $G_{vd}(s)$, implying that the output voltage

is controlled indirectly through the duty ratio. In order to design compensators such that stable regulation is achieved, the plant transfer functions must be derived. The following analysis presumes that the AC voltage, v_g, is represented by its RMS value and that the circuit is operating at the rated load condition of 75Ω. To derive the transfer functions, the KVL and KCL equations should be collected for each transistor state during a switching cycle and arranged in the standard state space format, $\dot{x} = Ax + Bu$ and $y = Cx + Eu$. When Q_1 is on,

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_g \tag{3}$$

$$\begin{bmatrix} i_g \\ v_{out} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} v_g \tag{4}$$

Where the return path of the input current is through the antiparallel diode of Q_2. When Q_1 is off,

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_g \tag{5}$$

$$\begin{bmatrix} i_g \\ v_{out} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} v_g \tag{6}$$

Through small signal linearization and separation of terms, the DC steady state values can be found,

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = 0 = \begin{bmatrix} 0 & \frac{-D'}{L} \\ \frac{D'}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_g \tag{7}$$

Which can be rearranged to solve for the RMS inductor current and DC steady state output voltage,

$$\begin{bmatrix} i_L \\ v_C \end{bmatrix} = \begin{bmatrix} \frac{v_g}{D'^2 R} \\ \frac{v_g}{D'} \end{bmatrix} \tag{8}$$

Through the standard process for small signal linearization, the resulting small signal state space equations can be used to determine the system's audiosusceptibility, control to inductor current, and control to output voltage transfer functions. Since the system has two inputs and two outputs, the small signal state space equations can be rewritten as a multiple input multiple output (MIMO) system. Following the subsequent algebraic distribution and cancelation of 2^{nd} order terms, both the input matrix, B, and feedforward matrix, E, can be represented as the column-wise concatenations of the vectors associated with $\hat{v}_g$ and $\hat{d}$, along with a row-wise concatenation of $\hat{v}_g$ and $\hat{d}$ to form the system input vector. The averaged small signal state space equations are determined as

$$\begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-D'}{L} \\ \frac{D'}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & \frac{v_C}{L} \\ 0 & \frac{-i_L}{C} \end{bmatrix} \begin{bmatrix} \hat{v_g} \\ \hat{d} \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \hat{i_g} \\ \hat{v_{out}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} \tag{10}$$

The audiosusceptibility can be determined through the superposition theorem by setting the control input, $\hat{d}$, to zero and extracting only the second output vector row corresponding to $\hat{v}_{out}$.

$$G_{vg}(s) = \frac{1}{D'} \frac{1}{(\frac{\sqrt{LC}}{D'}s)^2 + \frac{L}{D'^2 R}s + 1} \tag{11}$$

The resonant frequency,

$$\omega_0 = \frac{D'}{\sqrt{LC}} \tag{12}$$

The damping factor,

$$Q = D'R\sqrt{C/L} \tag{13}$$

The audiosusceptibility transfer function is depicted in Figure 18. Low passband gain in the audiosusceptibility transfer function directly corresponds to steady output regulation that is unresponsive to fluctuations in the input voltage.

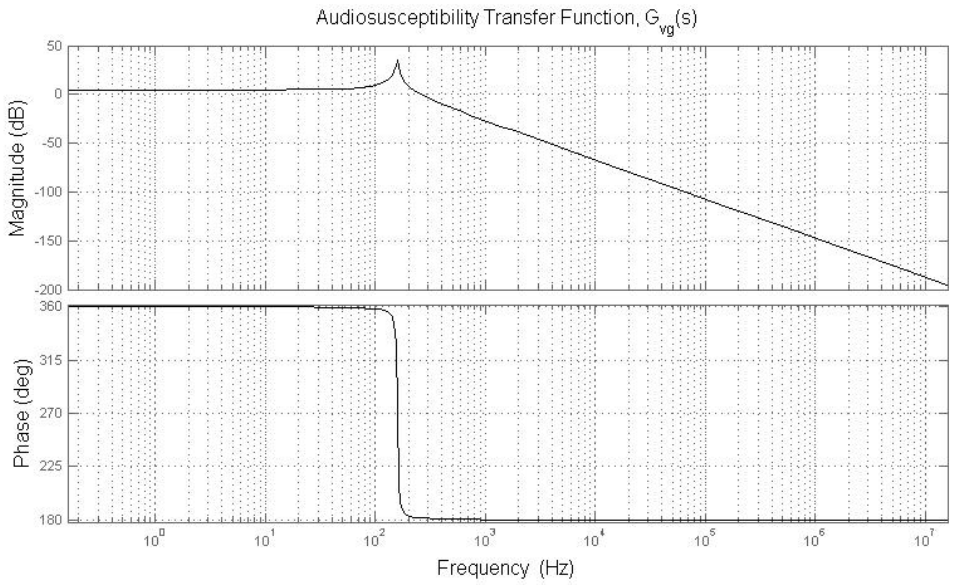


Figure 18. Audiosusceptibility transfer function.

The control to inductor current transfer function, $G_{id}(s) = \frac{\hat{i}_L(s)}{\hat{d}(s)}$, (in Figure 19) can be found by setting the input voltage perturbation, $\hat{v}_g(s)$, to zero and extracting the state matrix elements that correspond to the first output vector row, $\hat{i}_L(s)$.

$$G_{id}(s) = \frac{2v_g}{D'^3R}\frac{1 + \frac{RC}{2}s}{(\frac{\sqrt{LC}}{D'}s)^2 + \frac{L}{D'^2R}s + 1} \tag{14}$$

This is depicted in Figure 19,

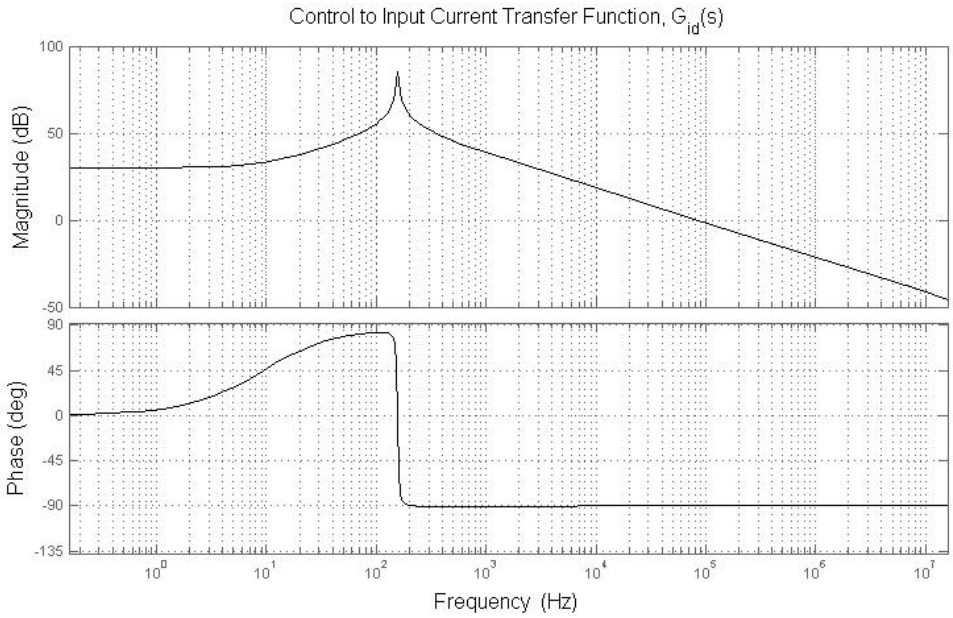


Figure 19. Control to inductor current transfer function.

Following the same process, the control to output voltage transfer function is found,

$$G_{vd}(s) = \frac{v_g}{D'^2} \frac{1 - \frac{L}{D'^2 R}s}{(\frac{\sqrt{LC}}{D'}s)^2 + \frac{L}{D'^2 R}s + 1} \tag{15}$$

The RHPZ frequency is,

$$\omega_{RHPZ} = \frac{D'^2 R}{L} \tag{16}$$

The control to output voltage transfer function is plotted in Figure 20,

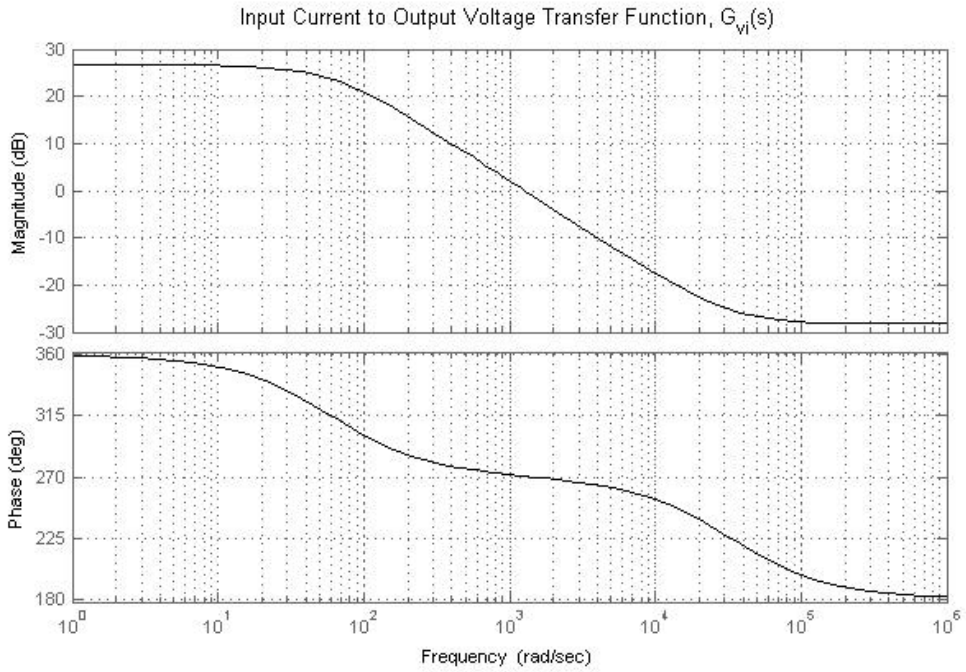


Figure 20. Control to output voltage transfer function.

The RHPZ is a source of system instability and limits the outer loop performance because the crossover frequency must be sufficiently reduced in anticipation of the RHPZ frequency. The RHPZ causes a gain increase of 20dB/dec but introduces a phase lag of 90 degrees. This is a major disadvantage for CCM operation, as the RHPZ effects are difficult to compensate and introduce a sluggish response to changes in the load condition.

Lastly, the outer loop controller design requires the inductor current to output voltage transfer function,

$$G_{vi}(s) = \frac{G_{vd}(s)}{G_{id}(s)} = \frac{D'R}{2}\frac{1 - \frac{L}{D'^2R}s}{1 + \frac{RC}{2}} \tag{17}$$

Shown in Figure 21,

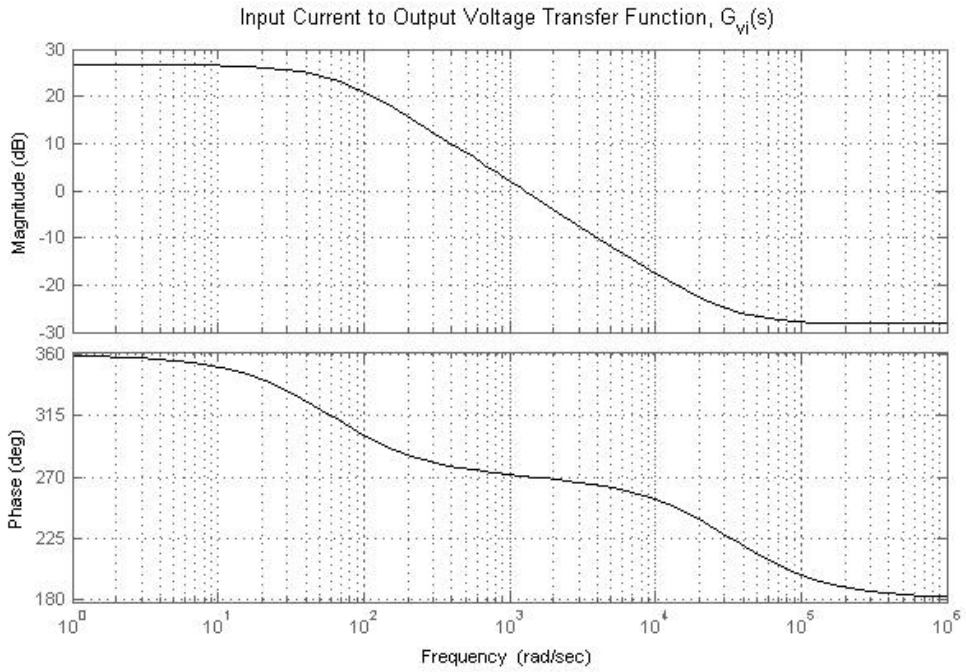


Figure 21. Input current to output voltage transfer function.

3.2 The Discontinuous Conduction Mode

Regardless of the intended operating mode, all boost PFC topologies will operate in DCM to some extent. For power transfer applications below 200W, primary operation in DCM is preferable due to the simplicity of the single loop controller. In applications where DCM is not feasible, DCM will still occur when the inductor does not have enough stored energy to dissipate over the remainder of the switching period following the transistor's turn on interval. This phenomenon occurs near the ZCP of the input voltage or in light load conditions.

The inductor current is categorized into three states in DCM, and the criteria for DCM operation is based on the average inductor and ripple current values,

$$\langle i_L \rangle < \frac{\triangle i_L}{2} \tag{18}$$

The duration of time that the inductor freewheels current, $D_2 T_s$, is dependent on the

peak current amplitude and the output and input voltages. The inductor voltage and current waveforms in DCM are depicted in Figure 22,

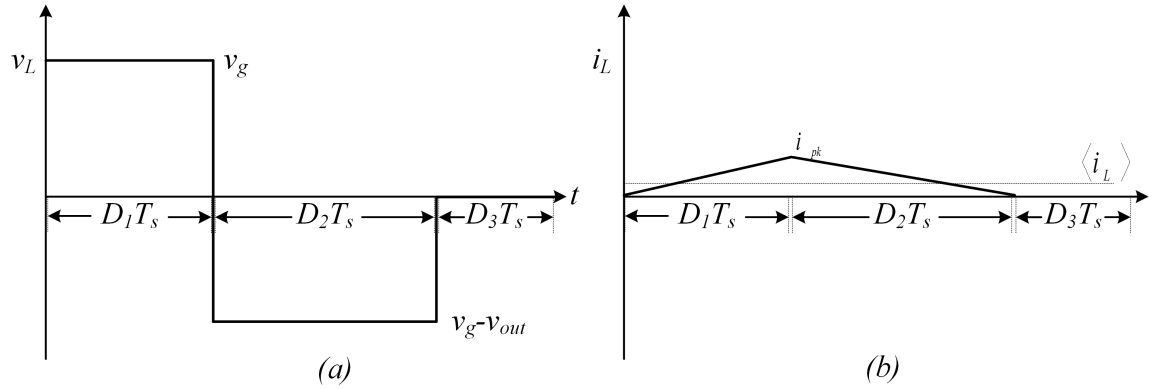


Figure 22. a) Inductor voltage waveform, b) inductor current waveform.

The general assumption is that the inductor's volt-second balance is equal to zero over the course of one switching period. The solution to the integral of the inductor voltage equation can be used to find the voltage conversion ratio,

$$M = \frac{v_{out}}{v_g} = \frac{D_2 + D_1}{D_2} \tag{19}$$

Here, D_2 is an unknown state-dependent quantity. Alternatively, the voltage conversion ratio can be expressed in terms of known parameters; since the capacitor charge balance must equal zero over one switching period, the average diode current is equal to the average load current,

$$\langle i_d \rangle = \frac{v_{out}}{R} = \frac{1}{T_s}\int_0^{T_s} i_d(t)dt = \frac{i_{pk}D_2}{2} \tag{20}$$

where (20) is used to solve for the voltage conversion ratio in terms of known quantities,

$$M = \frac{v_{out}}{v_g} = \frac{1 + \sqrt{1 + \frac{2D_1^2 RT_s}{L}}}{2} \tag{21}$$

In DCM, the degree of increase within the conversion ratio is highly dependent on the load. For example, in very light load conditions, the conversion ratio will rapidly increase with respect to an increase in D. The average inductor current is described as follows,

$$\langle i_L \rangle = \frac{1}{T_s} \int_0^{T_s} i_L(t)dt = \frac{1}{T_s} \left[1/2(D_1 + D_2)T_s i_{pk} \right] = \frac{v_{out} v_g D_1^2 T_s}{2L(v_{out} - v_g)} \tag{22}$$

The average input current draw is a function of the input voltage amplitude and the switch on time, $D_1 T_s$. Figure 23 depicts the average input current, $i_g(t)$; the peak current envelope, $i_{pk}(t)$; and the inductor current with respect to the input voltage amplitude. The average grid current waveform is produced by the differential mode of the EMI filter.

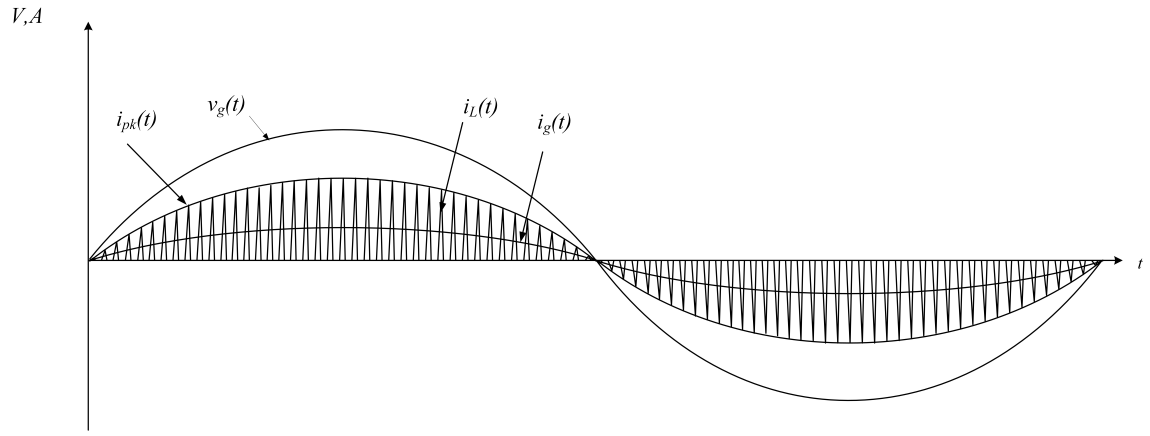


Figure 23. Inductor current waveforms in DCM.

The boost PFC in DCM exhibits self-PFC properties and constant on-time, constant frequency control is a popular control scheme for this reason. Since the input current draw is proportional to the input voltage amplitude, a single loop that only regulates the output voltage is required. The control block diagram for primary DCM operation is depicted in Figure 24.

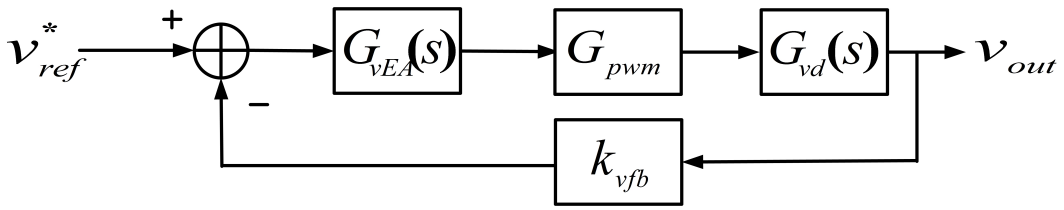


Figure 24. DCM control block diagram.

In order to design a controller for DCM operation, the converter dynamics must be determined. Accurately capturing this behavior is more challenging due to the existence of the state-dependent duty ratio constraint, D_2. The state space matrices in each portion of a switching cycle are as follows,

A	B	C	E
$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix}$	$B_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$	$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$E_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$A_2 = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}$	$B_2 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$	$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$E_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$A_3 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{RC} \end{bmatrix}$	$B_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$E_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

The characterization of converter dynamics in DCM is not as straightforward as the state space average modelling process presented in the CCM section of this chapter. There are several methods for generating DCM models, and these processes yield either reduced or full order models. Reduced order models result from the use of a duty ratio constraint, which implies the inductor dynamics are equal to zero during steady-state operation. As a result, these dynamics are omitted in the transfer functions, and the RHPZ appears to be eliminated as a result. For this reason, reduced order models are regarded as untrustworthy in the high frequency range. Full order models characterize the inductor current as an algebraic expression in terms of the system parameters and provide a higher fidelity characterization of

the high frequency range. If the state space averaging approach in DCM follows the same principle as CCM, then the following must be true,

$$\dot{x} = \big[D_1 A_1 + D_2 A_2 + (1 - D_1 - D_2) A_3\big] x + \big[D_1 B_1 + D_2 B_2 + (1 - D_1 - D_2) B_3\big] v_g \quad (23)$$

The equivalent averaging process applied to the state matrices from (23) describes the average of the matrix parameters, but not necessarily the average of the state variables themselves. For example, the total charge transferred to the capacitor from the inductor in DCM is found through integrating the inductor current waveform during $D_2 T_s$,

$$Q_C = \frac{i_{pk} D_2 T_s}{2} \quad (24)$$

This is the quantity of charge transferred through the rectifier diode. Therefore, the average current charging the capacitor is equal to the average diode current,

$$\frac{Q_C}{T_s} = \frac{i_{pk} D_2}{2} \quad (25)$$

Thus, to preserve the notion of charge balance in the capacitor, the average capacitor current is the sum of its average charging and discharging currents. Rewriting the charging current expression in terms of the average inductor current and substituting into the average capacitor current equation yields,

$$C \frac{dv_c}{dt} = \frac{D_2}{D_1 + D_2} i_L - \frac{1}{R} v_C \quad (26)$$

Whereas, the result from direct state space matrix averaging conflicts with the above description,

$$C\frac{dv_c}{dt} = D_2 i_L - \frac{1}{R}v_C \tag{27}$$

It is evident that the averaging of the physical equations does not yield the same result as averaging the state space matrices. This issue has been resolved: In [47] it was demonstrated that a modification to the state space averaging process can ameliorate the discrepancy in the averaging process. Let,

$$M = diag\left[\frac{1}{D_1 + D_2}, \frac{1}{D_1 + D_2}, ..., 1, 1\right] \tag{28}$$

Where M is an n by n matrix in which n is the number of inductor currents present in the system. Modifying the state space averaging equation with M,

$$\dot{x} = \Big[D_1 A_1 + D_2 A_2 + (1 - D_1 - D_2)A_3\Big] Mx + \Big[D_1 B_1 + D_2 B_2 + (1 - D_1 - D_2)B_3\Big] v_g \tag{29}$$

In the case where D_2 is solved based on the volt-second balance equation, a reduced order model will be produced since the inductor current derivative is zero. This is undesirable for the development of high fidelity circuit models, so the average inductor current will be defined such that the inductor current derivative is nonzero. Using (22) to solve D_2,

$$D_2 = \frac{2Li_L}{D_1 T_s v_g} - D_1 \tag{30}$$

Substitution of (30) into (29) yields,

$$\begin{bmatrix} \dot{i_L} \\ \dot{v_C} \end{bmatrix} = \begin{bmatrix} \frac{2}{D_1 T_s} & (\frac{2i_L}{D_1 T_s v_g} + \frac{D_1}{L}) \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{D_1^2 T_s}{2LC} \end{bmatrix} v_g \tag{31}$$

$$\begin{bmatrix} i_g \\ v_{out} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} v_g \tag{32}$$

Perturbation of the parameters and distribution of terms transforms the state equations into separable DC steady state and AC small signal state space equations. The DC steady state equations are used to find expressions for the average inductor current and capacitor voltage,

$$\begin{bmatrix} i_L \\ v_C \end{bmatrix} = \begin{bmatrix} \frac{D_1^2 T_s v_c}{2L(M-1)} \\ \frac{2LRi_L - D_1^2 T_s v_g}{2L} \end{bmatrix} \tag{33}$$

The AC small signal equations are found as,

$$\begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} = \begin{bmatrix} \frac{2(1-M)}{D_1 T_s} & \frac{-2M}{D_1 T_s R} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix} \begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} + \begin{bmatrix} \frac{DM^2}{L(M-1)} & \frac{2v_C}{L} \\ \frac{-D_1^2 T_s}{2LC} & \frac{-D_1 T_s v_g}{LC} \end{bmatrix} \begin{bmatrix} \hat{v_g} \\ \hat{d} \end{bmatrix} \tag{34}$$

$$\begin{bmatrix} \hat{i_g} \\ \hat{v_{out}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i_L} \\ \hat{v_C} \end{bmatrix} \tag{35}$$

From which the transfer functions can be derived. Using the RMS voltage and a fully DCM ($3k\Omega$) load condition, the audiosusceptibility (Figure 25) is found to be,

$$G_{vg}(s) = \frac{D_1(4M-2)}{2L(M-1)\omega_{p1}\omega_{p2}} \frac{1 - \frac{(M-1)D_1 T_s}{4M-2} s}{(\frac{s}{\omega_{p1}} + 1)(\frac{s}{\omega_{p2}} + 1)} \tag{36}$$

the radian pole frequencies are defined as follows,

$$\omega_p = \frac{\frac{2(1-M)}{D_1 T_s} - \frac{1}{RC} \pm \left[(\frac{2(M-1)}{D_1 T_s} + \frac{1}{RC})^2 - 4(\frac{4M-2}{D_1 T_s RC}) \right]^{1/2}}{2} \tag{37}$$

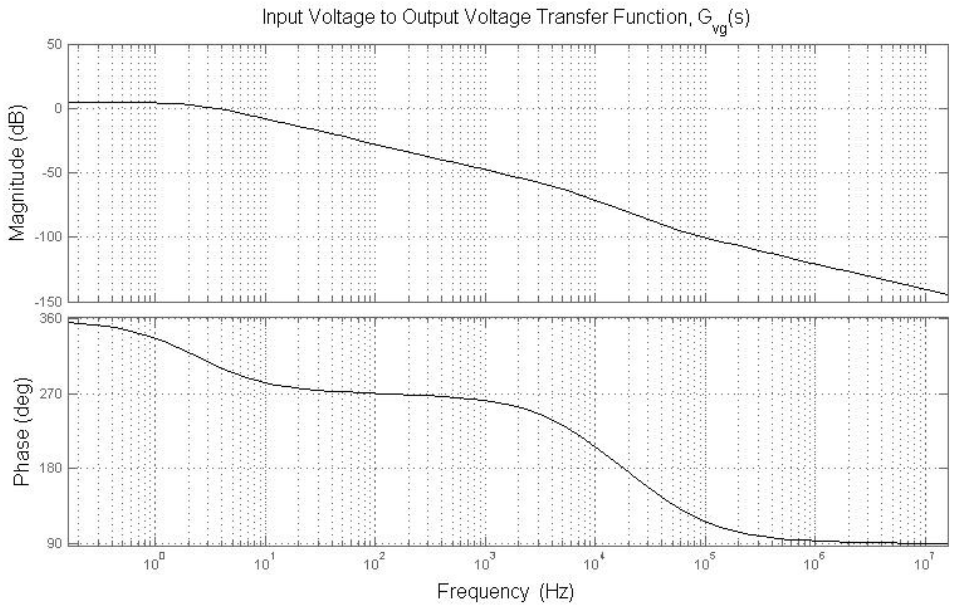


Figure 25. Audiosusceptibility transfer function.

The frequency at which the RHPZ occurs is at least twice the switching frequency, highlighting the distinct advantage of DCM operation. Since the unstable region is in the high frequency range, the design of a low bandwidth control loop is simplified. The control to output voltage transfer function is depicted below (Figure 26),

$$G_{vd}(s) = \frac{2v_g}{LC\omega_{p1}\omega_{p2}} \frac{1 - \frac{D_1 T_s}{2} s}{(\frac{s}{\omega_{p1}} + 1)(\frac{s}{\omega_{p2}} + 1)} \tag{38}$$

Where the pole frequencies are the same as the audiosusceptibility poles.

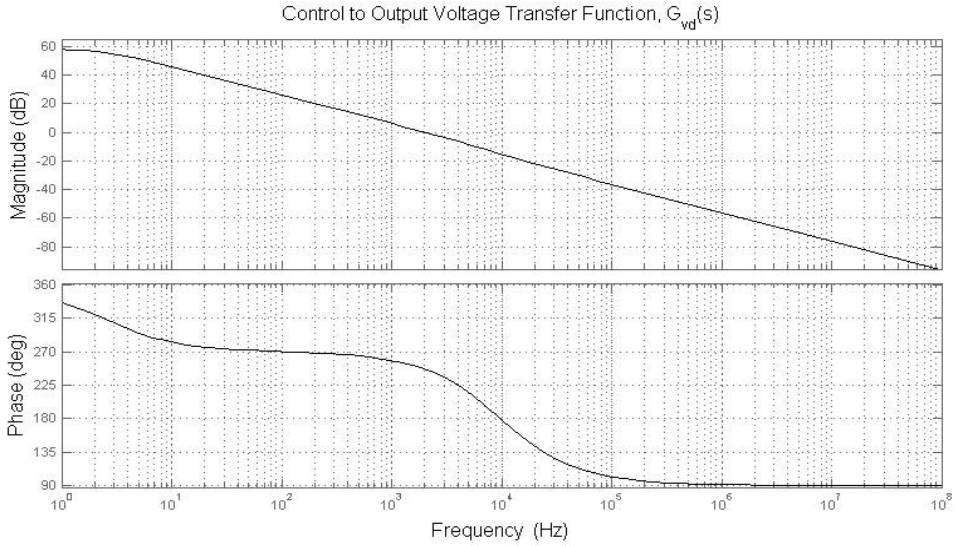


Figure 26. Control to output voltage transfer function.

The control to inductor current transfer function is as follows,

$$G_{id}(s) = \frac{4v_{out}}{LCR\omega_{p1}\omega_{p2}} \frac{1 + \frac{RC}{2}s}{(\frac{s}{\omega_{p1}} + 1)(\frac{s}{\omega_{p2}} + 1)} \tag{39}$$

This transfer function description features a high frequency pole and a low frequency pole-zero pair, depicted in Figure 27,

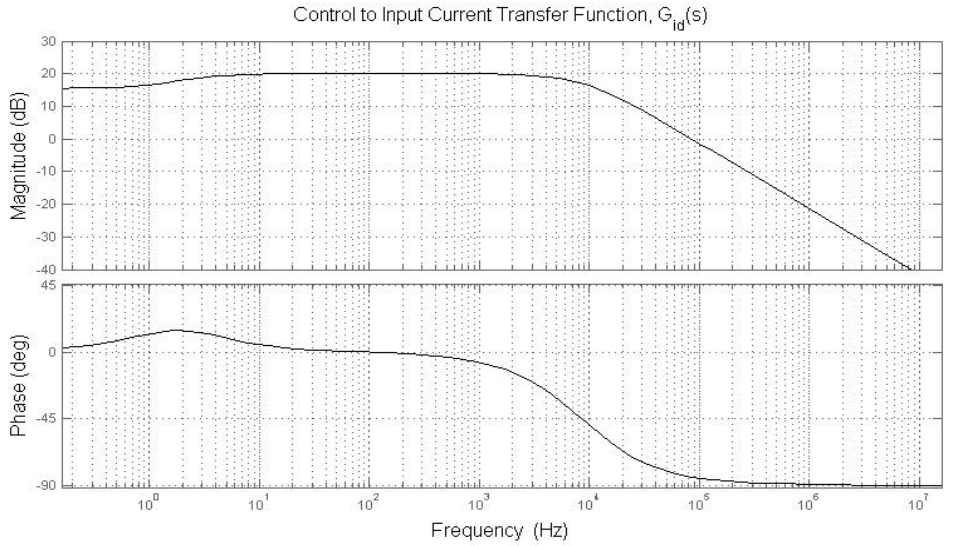


Figure 27. Control to input current transfer function.

The input current to output voltage transfer function is found as,

$$G_{vi}(s) = \frac{G_{vd}(s)}{G_{id}(s)} = \frac{R}{2M} \frac{1 - \frac{D_1 T_s}{2} s}{1 + \frac{RC}{2} s} \tag{40}$$

This transfer function is depicted below,

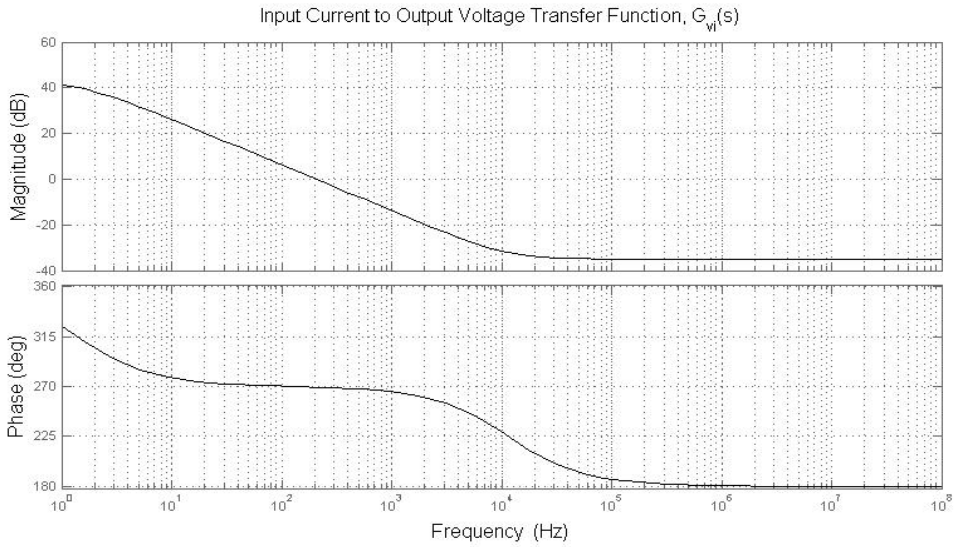


Figure 28. Input current to output voltage transfer function.

In this chapter, the converter has been characterized in both operating modes. Some of the difficulties associated with CCM stability and DCM analysis have been discussed to lay the foundation for the following chapter regarding controller development for a PFC system operating primarily in CCM.

4 Flexible Power Factor Controller

4.1 Stability and Design Challenges in PFC

The simplest compensator design method for any PFC controller is to use RMS values to describe the averaged behavior of the system over the course of a line cycle at the nominal load condition. If the compensator design follows this procedure, the formulated controller will exhibit acceptable windup and steady state performance in the region of the anticipated steady state conditions but will diminish in response to significant load or line fluctuations.

There have been several investigations regarding the undesirable behavior of PFC systems as their parameters transition into states that are not anticipated by the design procedure. Namely, the observation of slow and fast-scale instabilities as a result of increasing load conditions, inappropriately selected compensator gains, small output filter capacitance, and voltage conversion ratio. The effects of improperly selected system parameters (L, R, C, conversion ratio and feedback gains) have been shown [26-28].

In the slow (line and output ripple frequency) time scale, it has been observed that certain configurations of system parameters result in a "Period II" oscillation effect, or even "chaotic" n-periodic oscillations around the desired operating point. Chaos is a loosely defined term that implies n-periodic or entirely unpredictable behavior. Slow scale instability is generally shown in phase-plane trajectories of the capacitor voltage and inductor current. To demonstrate these behaviors, Figure 29 shows the slow-scale instability effects of an improperly designed PFC converter as the load resistance increases. The converter was simulated in PSIM, and the data arrays were used to create phase portraits in MATLAB. Figure 29a shows a stable "Period I" orbit, as the inductor current cycles at the line frequency while the capacitor voltage cycles at twice the line frequency. Figure 29b shows the "Period II" trajectory where

the inductor current peaks are imbalanced between the positive and negative half-cycles and the capacitor voltage cycles at the line frequency. Figure 29c depicts chaotic behavior, where the system operates in the vicinity of the voltage reference, but the inductor current and capacitor voltage orbit is n-periodic. It is clear that the system fails to regulate the input current as the load increases.

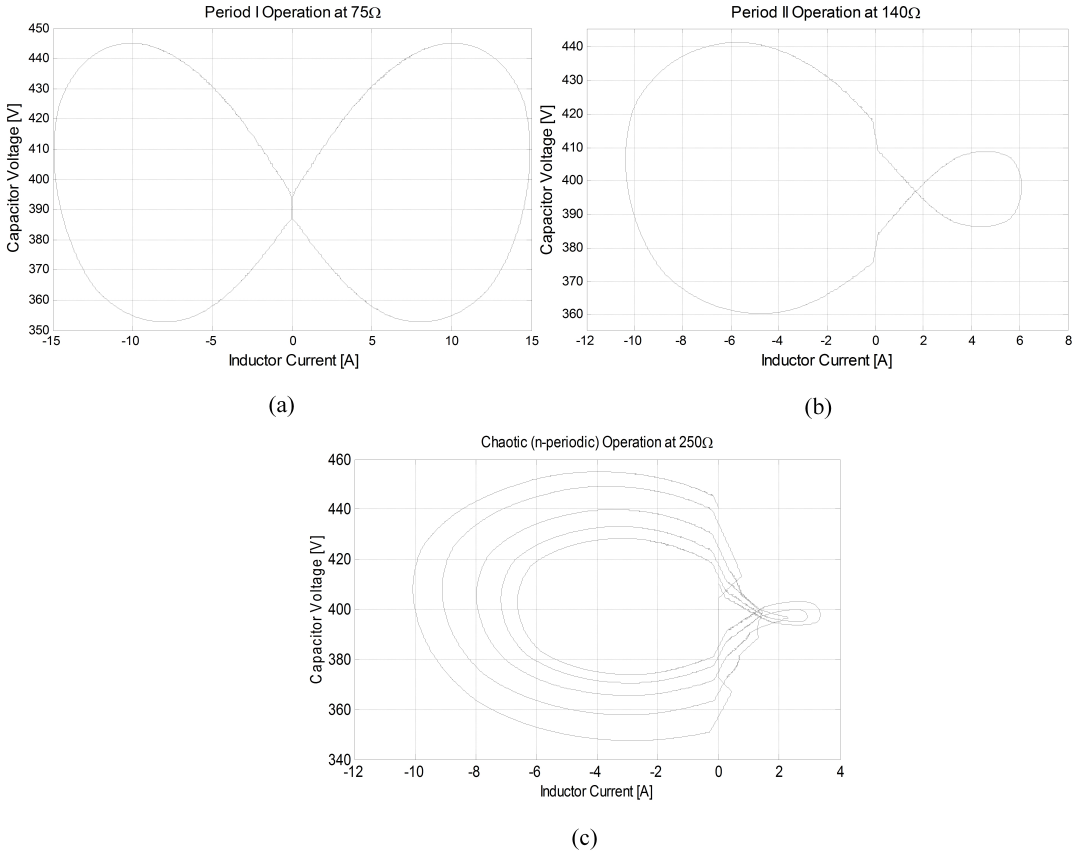


Figure 29. (a) period I orbit, (b) period II orbit, (c) n-periodic (chaotic) orbit.

At the fast (controller bandwidth and pulse width modulator) time scale, certain configurations of system parameters will result in switching period doubling bifurcation effects. In order to induce this condition, the inner current loop gain was increased from the design example of Figure 29 at the rated load condition. Figure 30 shows these effects, which occur about the ZCPs of the inductor current.

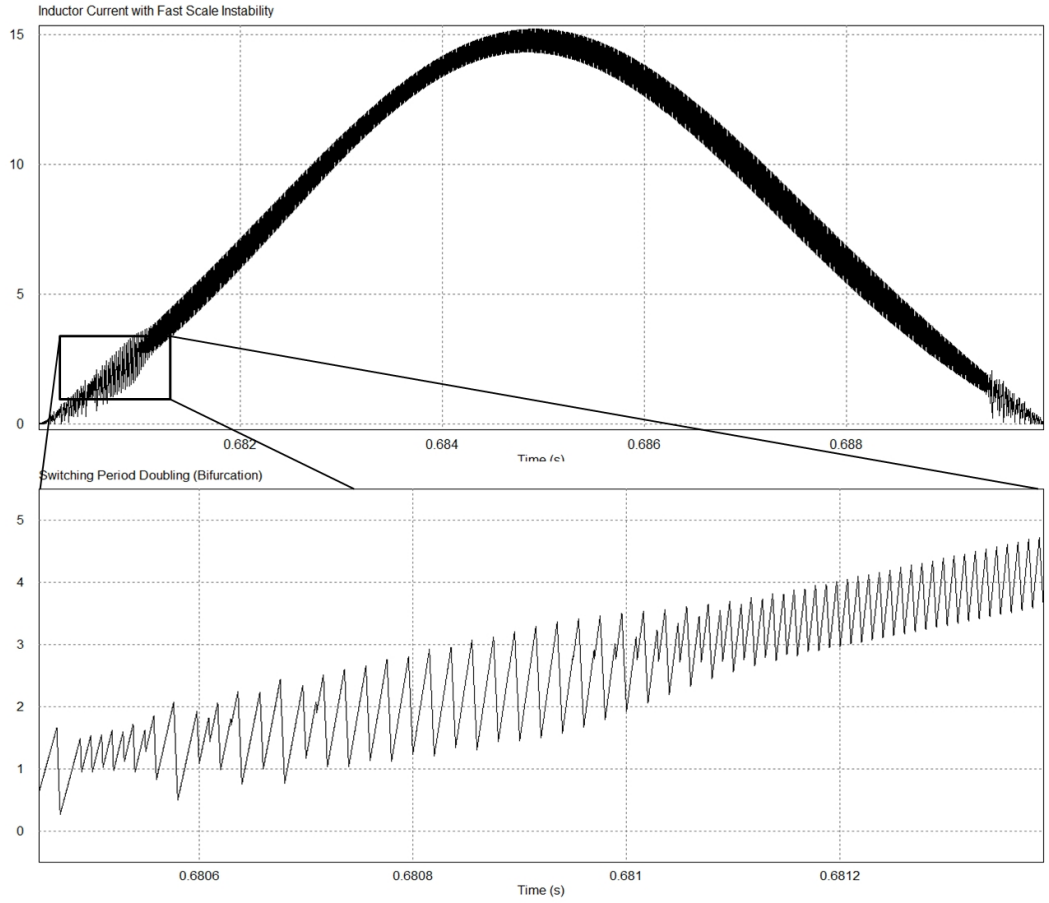


Figure 30. Period doubling bifurcation

This behavior results in increased distortion of the input current, illustrating that the system fails to achieve power factor correction of the nonlinear input current. Although it is not the intent of this thesis to fully analyze the exact sources of instability in the PFC, these behaviors are discussed to show that achieving a fully stable PFC design requires more consideration than only the expected power transfer conditions. Furthermore, it is apparent that the chosen compensator gains or controller structure have a significant impact on the flexibility of the PFC over its operating range.

Given the necessity for stability and satisfactory performance over a wide or unpredictable range of operating conditions, it is desirable to formulate adaptive controllers for PFC regulators. Digital control offers the capability for enhanced performance through source impedance compensation, passivity based control, gain scheduled con-

trol, harmonic injection, and n-degree of freedom control [48-51]. While discrete controllers offer enhanced PFC performance, they do not necessarily account for conduction mode transitions occurring over the course of a line cycle if the load condition increases. Conversely, regarding mixed conduction mode controller design, it is not shown how adaptive gains for each mode can improve the performance of the converter over analog controller counterparts. Furthermore, discrete PFC control algorithms do not improve the power factor at the point of common coupling of a power distribution network. Actively controlling the boost PFC power factor enables further adaptability, as the input power factor may be intentionally degraded for the benefit of line power quality.

In [35], it was demonstrated that input power factor degradation can improve the power quality of the power distribution network through harmonic current cancelation (HCC) and reactive power compensation (RPC). HCC enables the BBPFCC to serve as a unidirectional active power filter which can attenuate the line current harmonics generated by nonlinear loads connected to the power distribution network. RPC can be provided when inductive loads are connected to the power distribution network, although it was shown in [21] that the extent of RPC should be limited due to the generation of harmonics as a result of increased ZCP distortions which occur in capacitive reactive power compensation mode.

The goal of this thesis is to combine the advantages of adaptive gain, conduction mode prediction, and power factor control in order to improve the adaptability and performance of a system generally regarded as slow and inflexible due to the typical structure of the controller. In addition, the benefits of non-unity power factor control employed in BBPFCCs are explored through the extension of the control algorithm to a smart home application.

4.2 Mixed Conduction Mode

Depending on the load condition, the circuit will always operate in DCM or CCM for different portions of the line cycle. In heavy load conditions, the waveform will be predominantly CCM; in medium load conditions, the waveform will consist of some DCM and some CCM, otherwise called Mixed Conduction Mode (MCM); and in light load conditions, the waveform will be predominantly DCM. The dynamics of the circuit change drastically when the conduction mode changes, and stable compensator gain values for one mode may drive the system to instability in the other [52]. If not properly accounted for with a dedicated controller, the MCM condition results in highly distorted current. The inductor current in MCM is depicted in Figure 31,

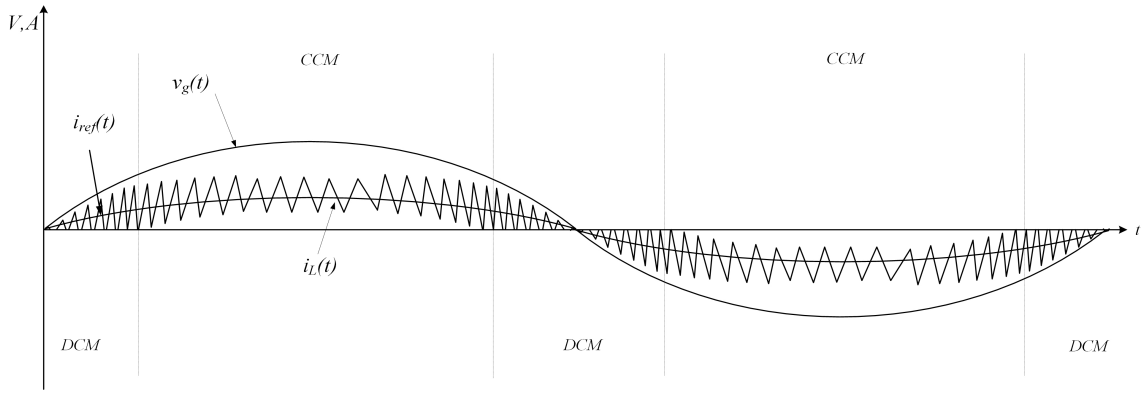


Figure 31. Inductor current waveforms in MCM.

Several research efforts have investigated MCM operation and controller design for boost converters; In [53], it is proposed that the controller operates at a greater switching frequency in CCM and reduced switching frequency in DCM in order to avoid MCM during one line cycle. This method employs 2 parallel controllers which are multiplexed depending on the conduction mode. In [54], DCM sampling correction and CCM/DCM duty-ratio feedforward terms are used in the controller such that unknown output parameters, like the load impedance, are not required for successful MCM operation. In [52], the critical boundary condition and sensed load impedance

are used to determine control gains satisfying the criteria for a stable system. In [55], the proposed controller utilizes a PID gain schedule for various power conditions. In [56], a zero current detection circuit is utilized in order to trigger adaptive switching and a predictive current control law. In [57,58] internal DSP comparators are used to detect DCM and update compensator gains through a multiplexer.

Additionally, the literature presents several methods for determining the operating mode of the circuit when a digital controller is implemented. One utilizes the critical boundary or load condition in order to estimate whether the circuit will operate in DCM or CCM. The load impedance is a required quantity for this determination and must be generated through the output voltage and the RMS input current value or output current. The sampled voltage and current are then used to calculate the impedance, which is passed through a low pass filter such that the low frequency ripple and switching noise are attenuated. Other methods may be used to infer transitions into DCM; such as conduction mode inferencing based on the power transfer condition or simple properties of the current waveform in either conduction mode. In this thesis, the CCM controller employs the commonly used average current control scheme and symmetric PWM and sampling are configured in the DSP firmware. This implies that the sampled current value is approximately equal to the average current value by definition of CCM operation. Using the input current sample, calculated duty ratio, input voltage sample, and output voltage sample, the peak and minimum current values during a switching period, k, can be estimated,

$$i_{pk}[k] = i_L[k] + \frac{d[k]v_g[k]T_s}{2L} \tag{41}$$

$$i_{min}[k] = i_L[k] + \frac{d'[k](v_g[k] - v_{out}[k])T_s}{2L} \tag{42}$$

The validity of (41) and (42) can be tested once the duty ratio for the current switching

period is known. If the circuit is operating in CCM above the critical boundary condition, then the following must be true,

$$sgn(i_{pk}[k]) = sgn(i_{min}[k]) \tag{43}$$

In addition, if the circuit is operating at the critical boundary condition,

$$|sgn(i_{pk}[k])| = (1 - sgn(i_{min}[k])) \tag{44}$$

If either (43) or (44) are true, then the compensator gains should be appropriated to CCM operation. When the signs of the estimated peak and minimum current values are opposite, then it is inferred that the inductor's stored energy at the time of the sample is most likely insufficient for continuous discharging over the remainder of the switching period. These expressions are used to detect the conduction mode of the following switching period and subsequently adjust the compensator gains and sampling scheme to suit the new conduction mode. The duty ratio of the current switching cycle must be known to make an estimation on the conduction mode.

If the circuit is operating in DCM, then symmetric sampling is problematic because the controller may either sample a nonzero current value which is not indicative of the average current or sample a zero current value (if the sample occurs during D_3T_s). This behavior presents a significant problem for average current control because the feedback values do not indicate the state of the system and will not follow the reference. In order to mitigate this problem, a sample correction will be applied in DCM.

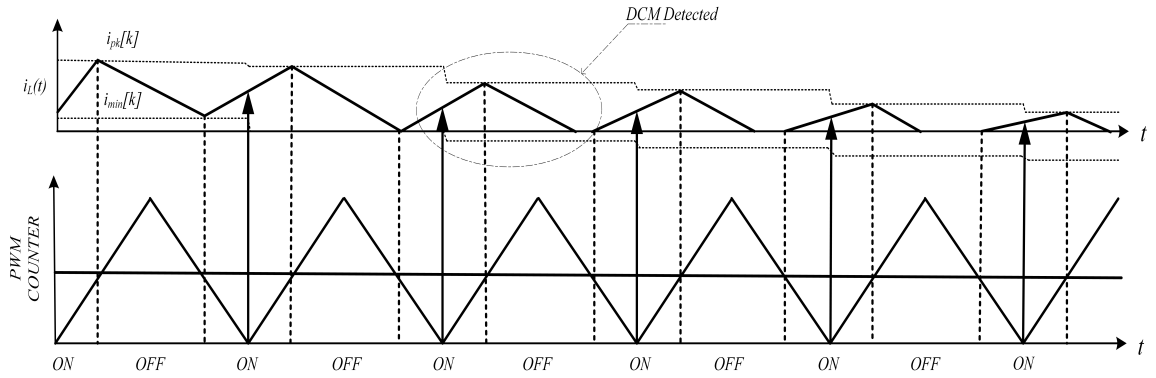


Figure 32. Peak and minimum current estimation is used to predict the conduction mode.

In the first two periods depicted in Figure 32, the sampled values are approximately equal to the average current due to the PWM and sampling pattern. Based on the known duty ratio, the peak and minimum current values are calculated, and their signs are found to be the same. In CCM, at the peak of each PWM count, sampling and consequential updates to the PWM comparator registers are performed. In the third sampling period, the controller detects that a transition into DCM has occurred; the calculated duty ratio under the initial assumption of CCM is kept, however the average current during this switching period must be estimated. If the sample occurs during the middle of the conduction period, then,

$$2i_L[k] \approx i_{pk}[k] \tag{45}$$

Here, the sampled inductor current is one half of the peak current since the duty ratio has been established from the previous sample and subsequent calculation. This is an approximation because the transistor exhibits a finite transition period in which the inductor current will continue to change and A/D resolution will result in quantization of the sampled signal. Since D_2T_s is defined as the time required to fully discharge from the estimated peak current, then from (22) and (45), the average current in a switching cycle can be solved as,

$$\left\langle i_L[k] \right\rangle = d[k] i_L[k] + \frac{2Li_L^2[k]}{T_s(v_{out}[k] - v_g[k])} \tag{46}$$

Using (46) and the 180 degree advance in the DCM sampling point, each inductor current sample can be corrected such that an average current feedback signal is provided for the inner loop controller and the structure does not need to change to that of Figure 14. A similar sampling correction technique was shown in [54].

4.3 Feedforward Compensation

4.3.1 The Continuous Conduction Mode

As suggested in Chapter 3, one method for improving the current regulator performance can be explored by implementing a feedforward controller. Feedforward compensation is open-loop control in that it provides the theoretical control signal for the desired input and output conditions. The strength in utilizing feedforward control can be demonstrated by observing the behavior of the input side of the circuit, shown in Figure 33.

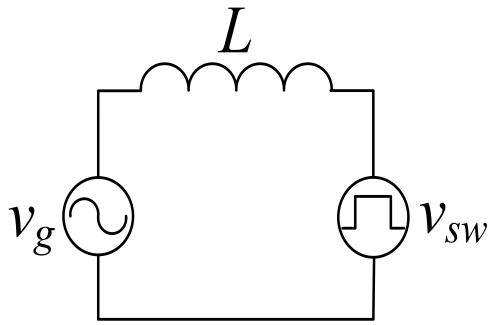


Figure 33. Input side circuit model.

Since the transistors must switch with respect to the line voltage, they can be approximated as a square wave voltage source; as such, the input loop is described by,

$$v_g = L\frac{di_L}{dt} + D'v_{sw} \tag{47}$$

where, if the forward voltage of the rectifier diode is small in comparison to the output voltage, the drain to source voltage during the off state can be substituted as,

$$v_g = L\frac{di_L}{dt} + D'v_{out} \tag{48}$$

Solving for the duty ratio,

$$D = \left(1 - \frac{|v_g(t)|}{v_{out}(t)}\right) + \left(\frac{Ldi_L}{v_{out}(t)dt}\right) = d_{ff}(t) + d_{fb}(t) \tag{49}$$

Where there are now two terms that determine the aggregate duty ratio, $d[k]$. The feedforward duty ratio term, d_{ff}, is equal to the theoretical duty ratio value, and the feedback duty ratio, d_{fb}, directly corresponds to the difference in phase between the reference current waveform and the inductor current. The introduction of a feedforward term into the switching control signal produced by the inner loop compensator alleviates the fundamental displacement factor of the input current and improves distortions about the ZCP [59].

4.3.2 The Discontinuous Conduction Mode

Under the same notions as CCM, the current loop performance in DCM can be improved through feedforward compensation if the controller employs current sample averaging correction. Using (22) to solve for the theoretical duty ratio, given a desired average current value from the outer voltage loop,

$$D = \sqrt{\frac{2Li^*_{ref}(t)(v_{out}(t) - v_g(t))}{v_{out}(t)v_g(t)T_s}} \tag{50}$$

This expression is used to provide the theoretical duty ratio for corrected average

current control in DCM.

4.4 Current Loop Design

4.4.1 The Continuous Conduction Mode

Through inspection of the system transfer functions, it is evident that the frequency response, poles, and zeros shift as the load and line voltage conditions change. This highlights the potential for instability and inflexibility when utilizing static compensator gains. For instance, the CCM transfer function, $G_{id}(s)$; The frequency response characteristics are dependent on the input voltage amplitude, concomitant duty ratio, and load condition. Then the uncompensated inner open loop gain is,

$$|T_{ol}(s)| = 20log_{10}(|G_{id}(s)||G_{pwm}||k_{ifb}|) \tag{51}$$

Figure 34 depicts the effects of the increasing line voltage. The loop crossover frequency remains the same regardless of the line voltage and load condition, and the sensitivity within the control bandwidth is significantly reduced as the input voltage increases and the duty ratio decreases or when the load increases. In the case of increasing peak input voltage, this behavior is a result of the increasing resonant frequency.

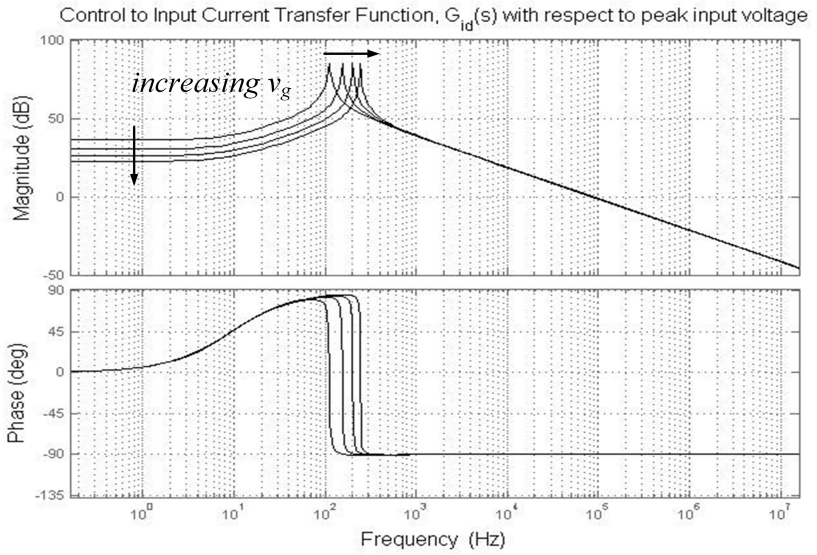


Figure 34. Control to inductor current transfer function with respect to changes in peak v_g

Lag type compensation is desirable because the current loop will operate with low steady state error; however, the addition of a PI compensator introduces an additional 90 degree delay due to its origin pole. Following the general rule that the inner current loop crossover frequency, ω_c, should be approximately one sixth the switching frequency at most, an operational constraint for the PI compensator zero frequency can be defined as:

$$\omega_z = min(10^{\alpha}\omega_0, 10^{\beta}\omega_c) \tag{52}$$

In this design, α is chosen as 1/2 and β is chosen as -1. An offset, controlled with α, from the calculated resonant frequency may be desired, as the system can be tuned to yield higher gain or increased phase near the resonant frequency point. For this reason, one half of one decade above the resonant frequency is chosen, as it is a fair compromise between gain and phase improvement. However, in the scenario where the resonant frequency approaches the desired crossover frequency, the zero frequency should be limited below the desired crossover frequency so that an accept-

able phase margin can be achieved. This upper limitation is selected through β. The PI compensator transfer function is defined as,

$$G_{iEA}(s) = \frac{k_i}{s} + k_p \tag{53}$$

Then the proportional gain is as follows,

$$k_p = \frac{k_i}{\omega_z} \tag{54}$$

The gain magnitude equation for the PI compensator is set equal to the opposite of the uncompensated, estimated loop transfer function gain at the desired crossover frequency. The integral gain can be solved such that the uncompensated loop gain is driven to unity at the desired crossover frequency,

$$k_i = \sqrt{\frac{10^{\frac{-T_{ol}}{10}}}{\frac{1}{\omega_c^2} + \frac{1}{\omega_z^2} + \frac{1}{\omega_c \omega_z}}} \tag{55}$$

Figure 35 shows the mitigation of the resonant frequency effects through adjustments of the PI compensator zero frequency as the peak input voltage increases.

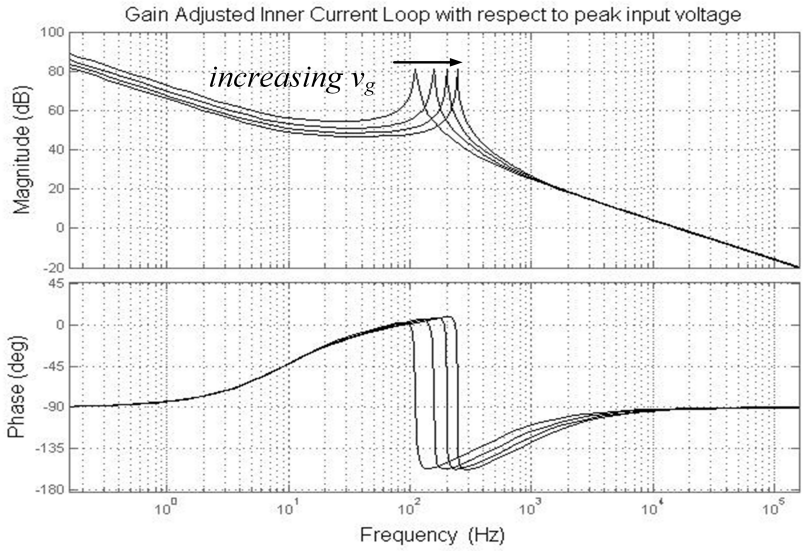


Figure 35. Open loop transfer function with adaptive PI gain.

4.4.2 The Discontinuous Conduction Mode

The same pole and zero shifting phenomenon occurs when the circuit operates in DCM, shown in Figure 36,

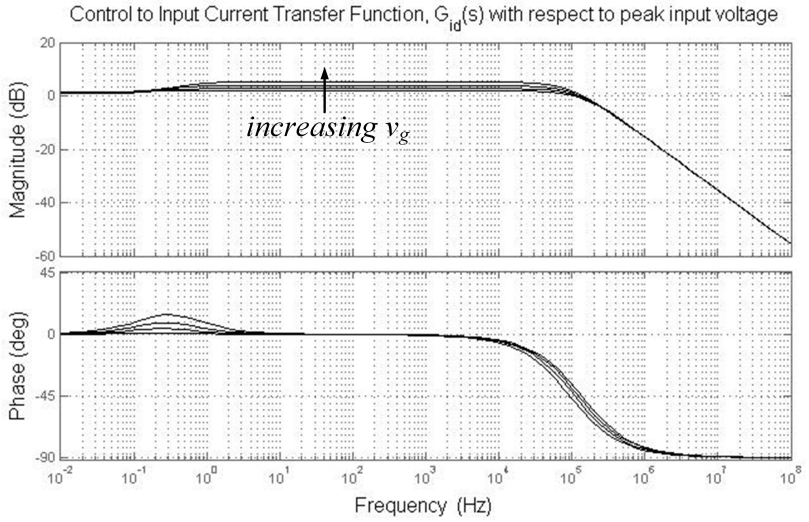


Figure 36. Current loop with respect to changes in peak v_g

As in CCM, the open loop gain at the desired crossover frequency is calculated based on the operating conditions. In this case, there is a low frequency pole which is nearly canceled by the control to inductor current transfer function's low frequency zero during light load conditions. Supposing the "+" result of (11) is ω_{p1}, the high frequency pole can be canceled when it is smaller than the desired crossover frequency. Similarly,

$$\omega_z = min(10^{\alpha}\omega_{p1}, 10^{\beta}\omega_c) \tag{56}$$

Where α and β are the same as the CCM case. Even at the rated load condition chosen in this paper, the high frequency pole is still always greater than the desired cutoff frequency. The characteristic gain of the transfer function is flat in the frequency region between the desired cutoff frequency and the switching frequency. If the compensator sets the crossover frequency to ω_c, then the switching frequency will

not be significantly attenuated. In this case, it is desirable to add an additional gain margin so that the switching frequency is sufficiently attenuated and does not feed back into the control loop. Assuming the circuit is loaded at $3k\Omega$, the open loop transfer function is shown in Figure 37,

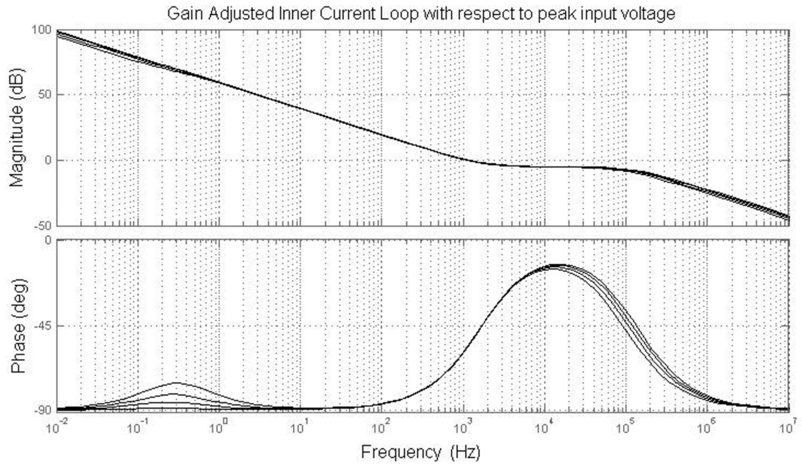


Figure 37. Open loop transfer function with adaptive PI gain.

4.5 Voltage Loop Design

In either conduction mode, the outer voltage loop will be compensated such that it has a very low bandwidth of approximately 10-15Hz. This is required for two reasons, 1) the output voltage ripple effects occurring at twice the line frequency must be sufficiently attenuated so that the current reference signal does not become severely distorted, and, 2) the CCM RHPZ is in the low frequency range and may drive the system to instability. In regards to the outer loop, adaptive gain does not make a significant difference in output voltage regulation performance since the bandwidth is already limited to the low frequency range; the main concern is to ensure stability in the case of both conduction modes.

$$G_{vEA}(s) = \frac{s + \omega_z}{s(s + \omega_p)} \tag{57}$$

Type II compensation is selected in order to improve the rolloff which follows the crossover frequency through the low pass characteristic of the compensator. The pole is placed just below the output voltage frequency so that the ripple and CCM RPHZ are attenuated as much as possible without sacrificing gain in the low frequency region. It is critical that the zero frequency is selected just below the pole frequency and the ratio of the pole and zero frequencies are chosen such that a similar crossover is achieved in both modes. Figure 38 depicts the outer loop gain in CCM, and Figure 39 shows the outer loop gain in DCM,

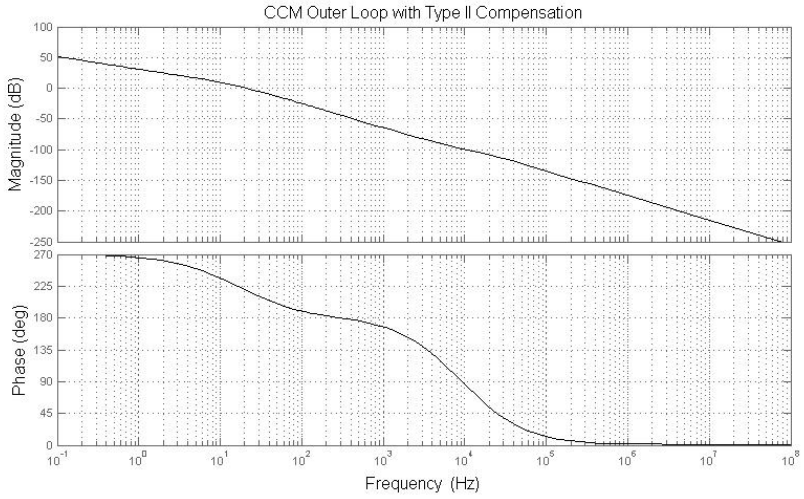


Figure 38. CCM compensated outer loop

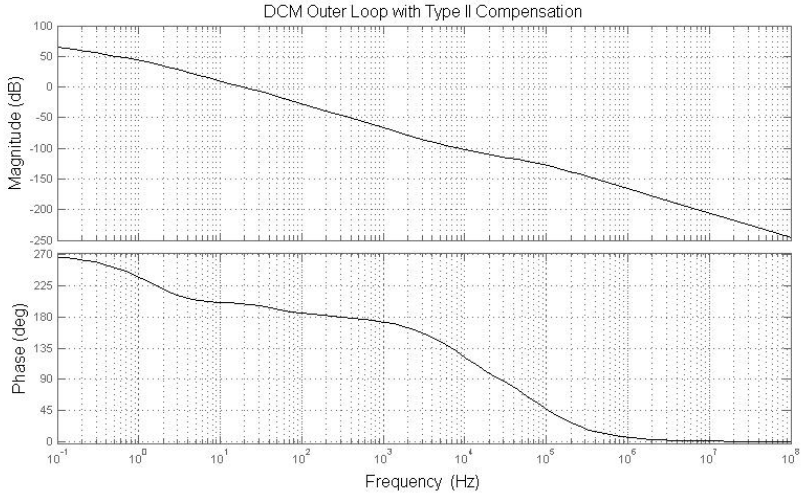


Figure 39. DCM compensated outer loop

5 Simulation and Testing Results

5.1 Simulation Over Various Load Conditions

Using the aforementioned strategies, it is verified that the system is stable over the full range of load conditions. The inductor current, EMI filter current and output voltage at the 100% (75Ω), 50% (150Ω), 25% (300Ω), 10% (750Ω), and 2.5% (3kΩ) load conditions are depicted in Figures 40-44,

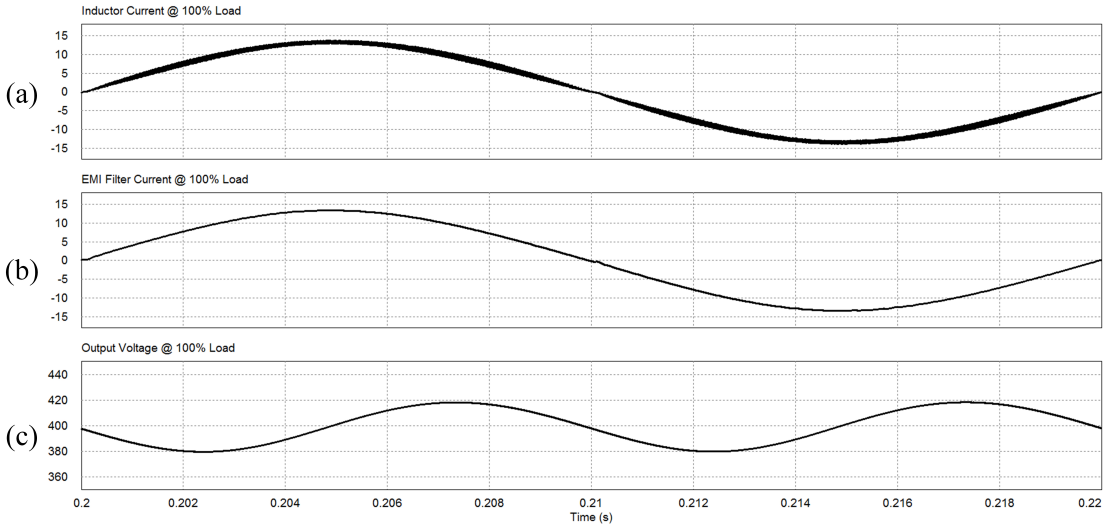


Figure 40. BBPFCC regulation performance at 100% load condition, a) boost inductor current, b) EMI filter input current, c) output voltage.

The power factor of the EMI filter current in Figure 40 is 0.999.

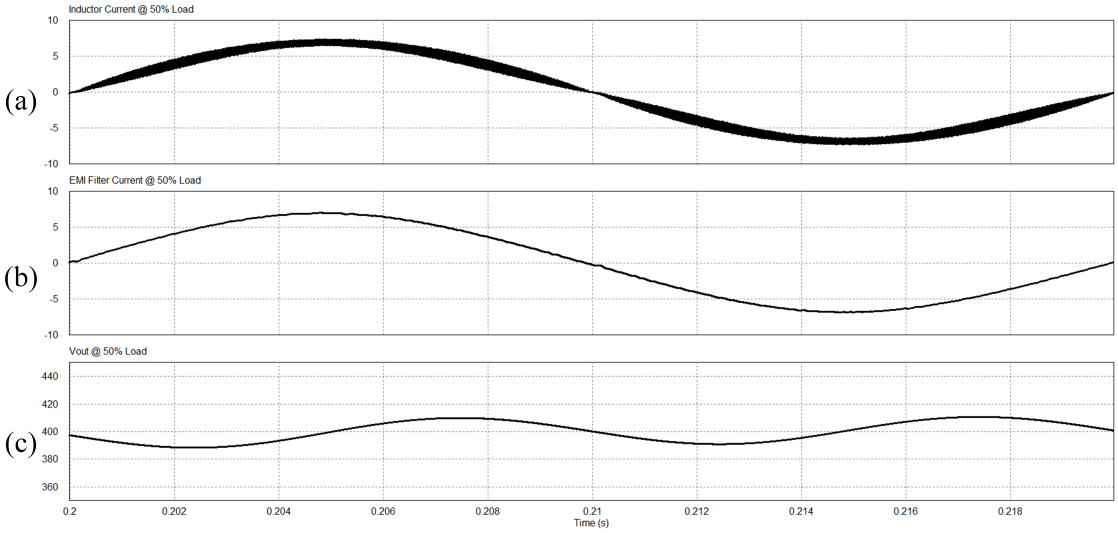


Figure 41. BBPFCC regulation performance at 50% load condition, a) boost inductor current, b) EMI filter input current, c) output voltage.

The power factor of the EMI filter current in Figure 41 is 0.999.

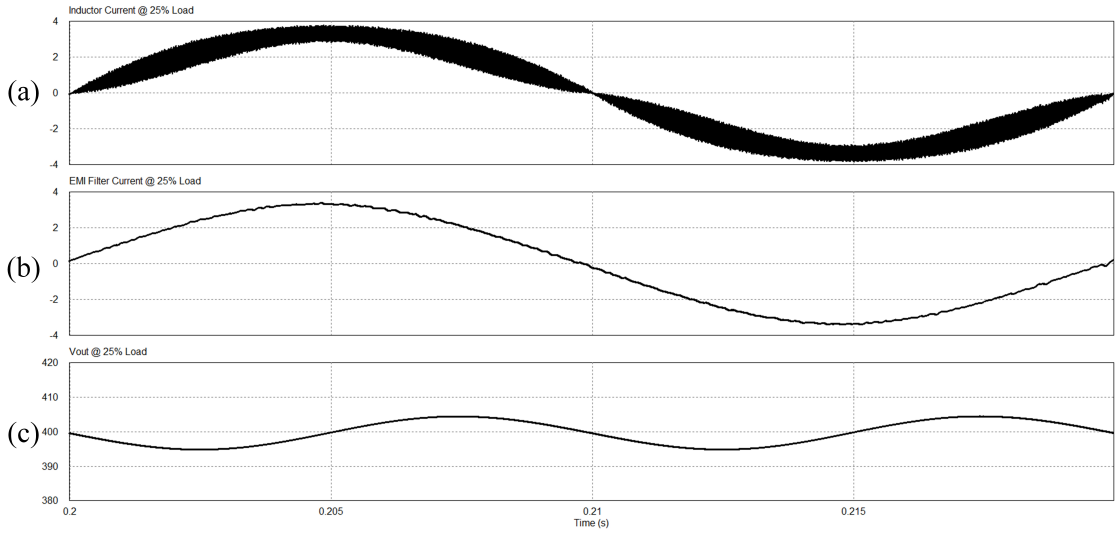


Figure 42. BBPFCC regulation performance at 25% load condition, a) boost inductor current, b) EMI filter input current, c) output voltage.

The power factor of the EMI filter current in Figure 42 is 0.997.

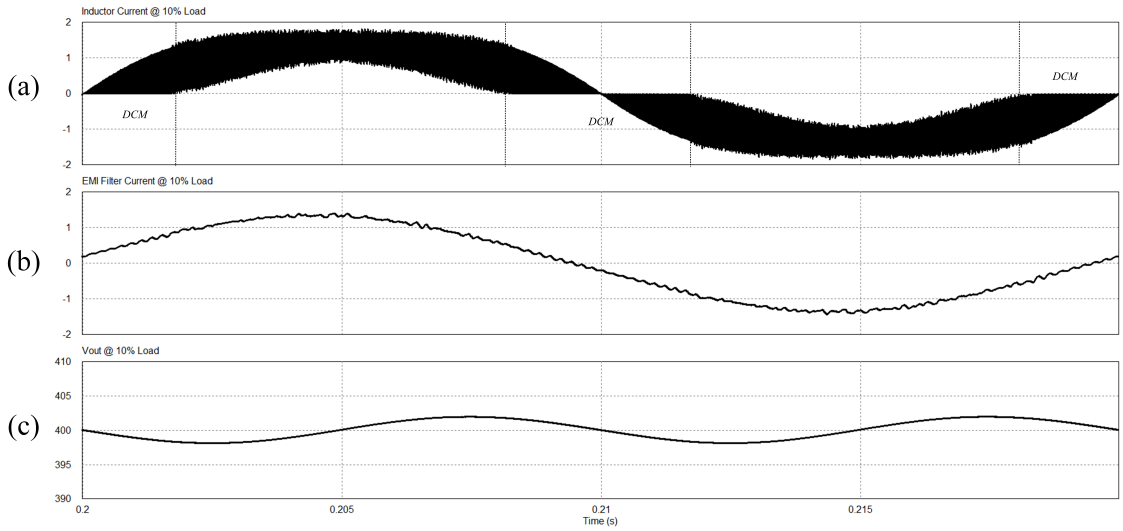


Figure 43. BBPFCC regulation performance at 10% load condition, a) boost inductor current, b) EMI filter input current, c) output voltage.

The power factor of the EMI filter current in Figure 43 is 0.986.

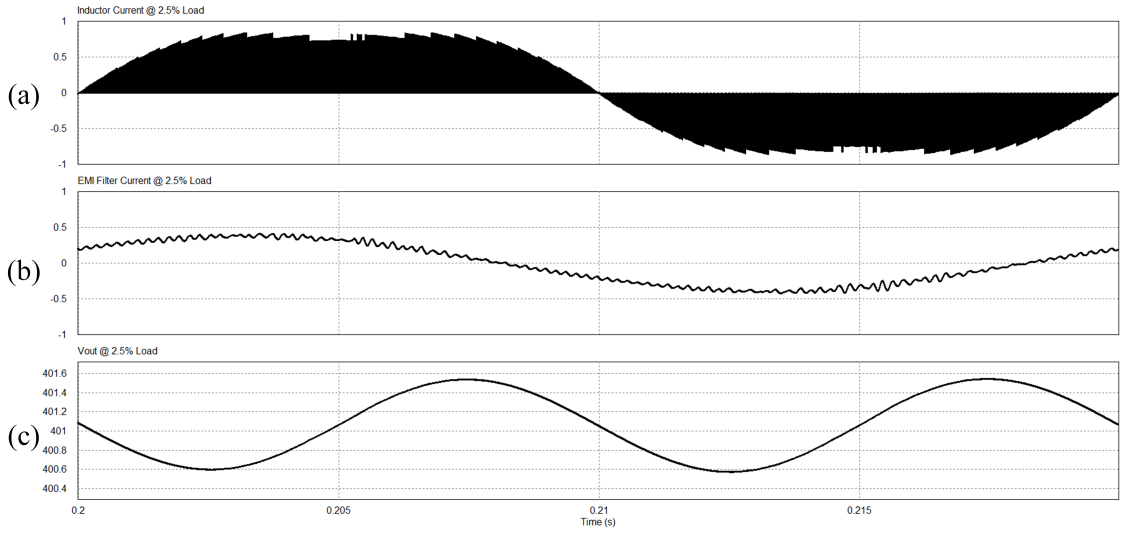


Figure 44. BBPFCC regulation performance at 2.5% load condition, a) boost inductor current, b) EMI filter input current, c) output voltage.

The power factor of the EMI filter current in Figure 44 is 0.844. This is largely due to the phase shifting property of the EMI filter, as the zero crossing of the input current leads the inductor current in phase. Since the crossover frequency of the DCM

loop is lower than that of the CCM loop, the current regulation is worse than that of CCM and the current THD increases slightly.

5.2 Performance Comparison of Adaptive and Static PFC Controllers

5.2.1 Comparison of Input Current Regulation

In this section, the adaptive controller performance is compared with the conventional control scheme, shown in Figure 7. Using both adaptive gain and feedforward compensation, the converters are subjected to the full range of load conditions and several input voltage conditions. The selected input voltage range is $110V_{RMS}$ above the nominal input voltage (with the same conversion ratio as the original design), and the nominal input voltage with a larger conversion ratio. The high input voltage condition is chosen so that the system enters DCM more quickly. The increased conversion ratio condition is chosen to show the effects of the shifted resonant frequency during CCM operation. The result is shown in Figure 45,

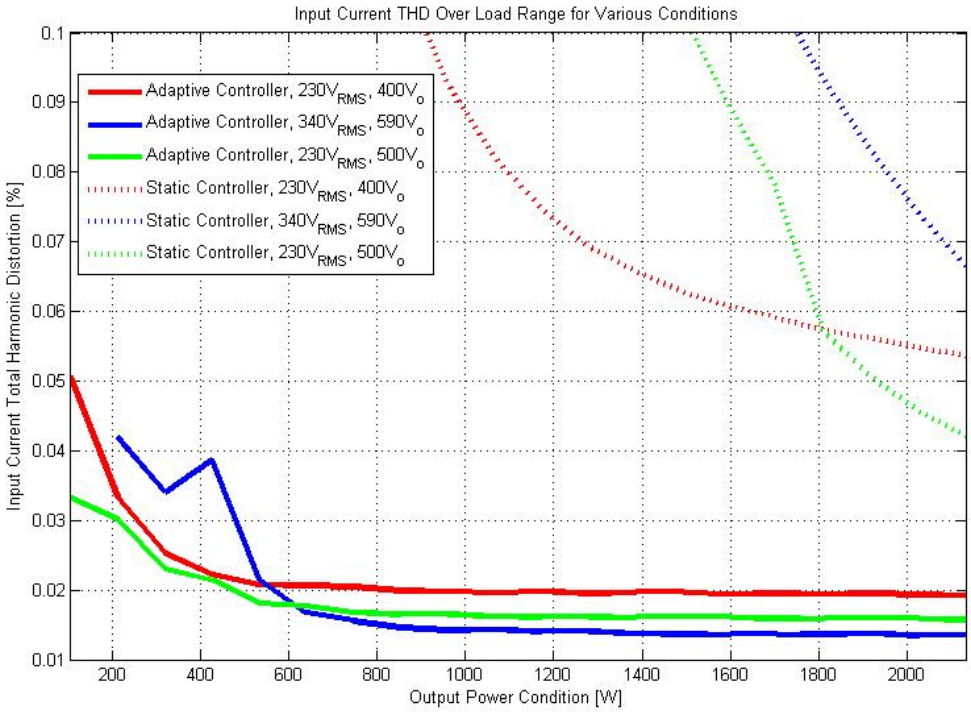


Figure 45. %THD over 5%-100% load at various input voltage and conversion ratio conditions. Curves which end before the load sweep indicate that the system became unstable at lighter load conditions.

The static controller performance deteriorates significantly as the load becomes lighter and the input voltage or conversion ratio increase, indicating that conventional method is insufficient for systems which operate over a wide range of conditions. In the case of the adaptive controller, consistently low input current THD is maintained over the full load range, but deteriorates in the high input voltage condition. Adapting the compensator gains extends the operating range of the converter significantly.

5.2.2 Comparison of Load Step Response

The dynamic response of the adaptive and static controllers is compared with a load step of 100% down to 50%. This is shown in Figure 46,

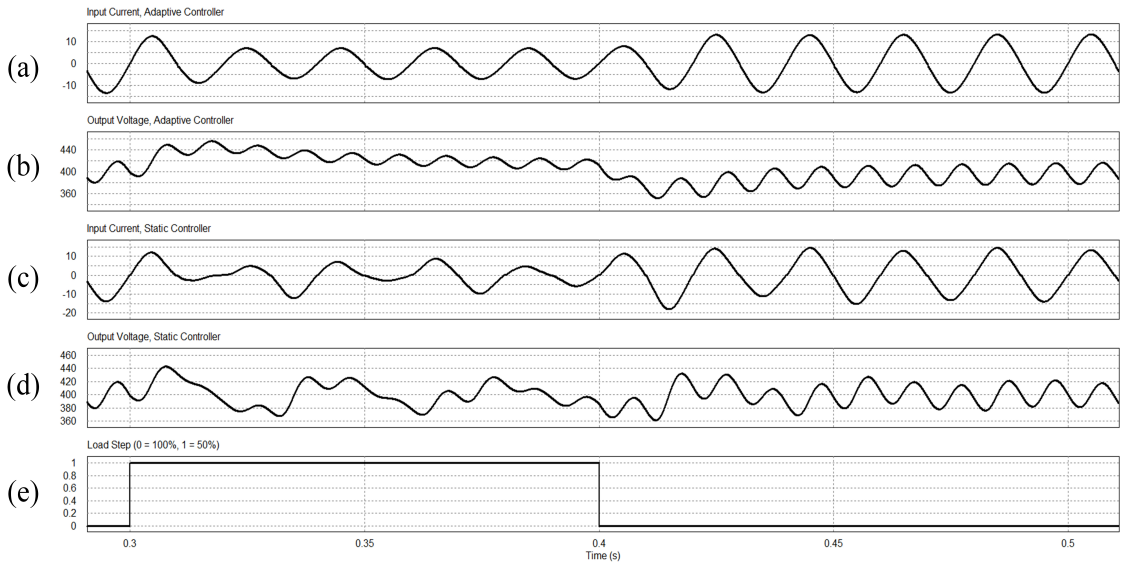


Figure 46. Output voltage of adaptive and static controllers in reponse to 50% load step.

The adaptive controller regulates the input current with significantly less distortion than that of the conventional controller in response to a 50% load step. The adaptive controller favors current performance, as the overshoot is significantly larger and the average output voltage of the static controller is closer to the reference during this interval.

5.3 Validation of Control Algorithm in Hardware in the Loop Simulation

The proposed control algorithm is validated on the Opal RT 4510 system interfaced with a Texas Instruments TMS320F28335 processor. Due to Opal RT's sampling rate, the switching frequency of the converter is reduced (to 10kHz). The test configuration is shown in Figure 47,

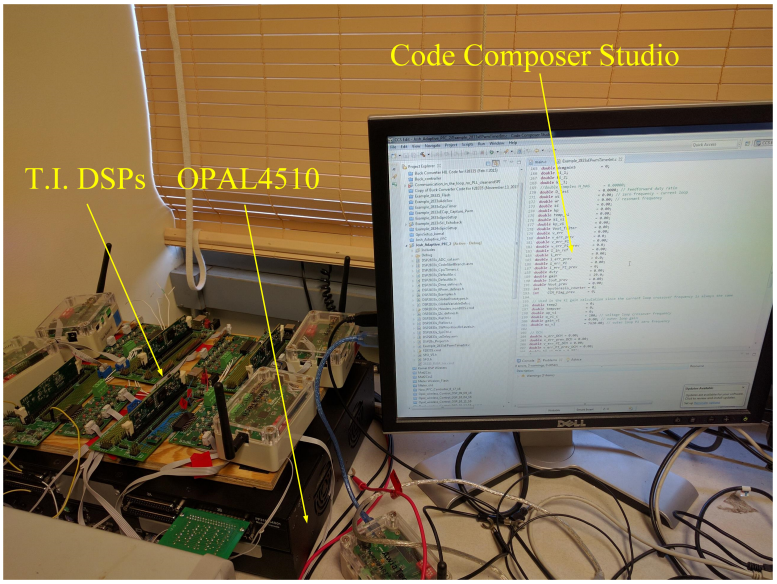


Figure 47. Opal RT and Texas Instruments DSP HIL test bed

Figure 48 depicts the testing results. Figure 48(a)-(c) verify the system operates about the nominal condition. Figure 48(d) depicts operation at a low output voltage and light load condition. Lastly, Figure 48(e)-(f) show operation at different conversion ratio conditions with a smaller peak input voltage. This result demonstrates that the algorithm is feasible for implementation on a modern digital signal processor.

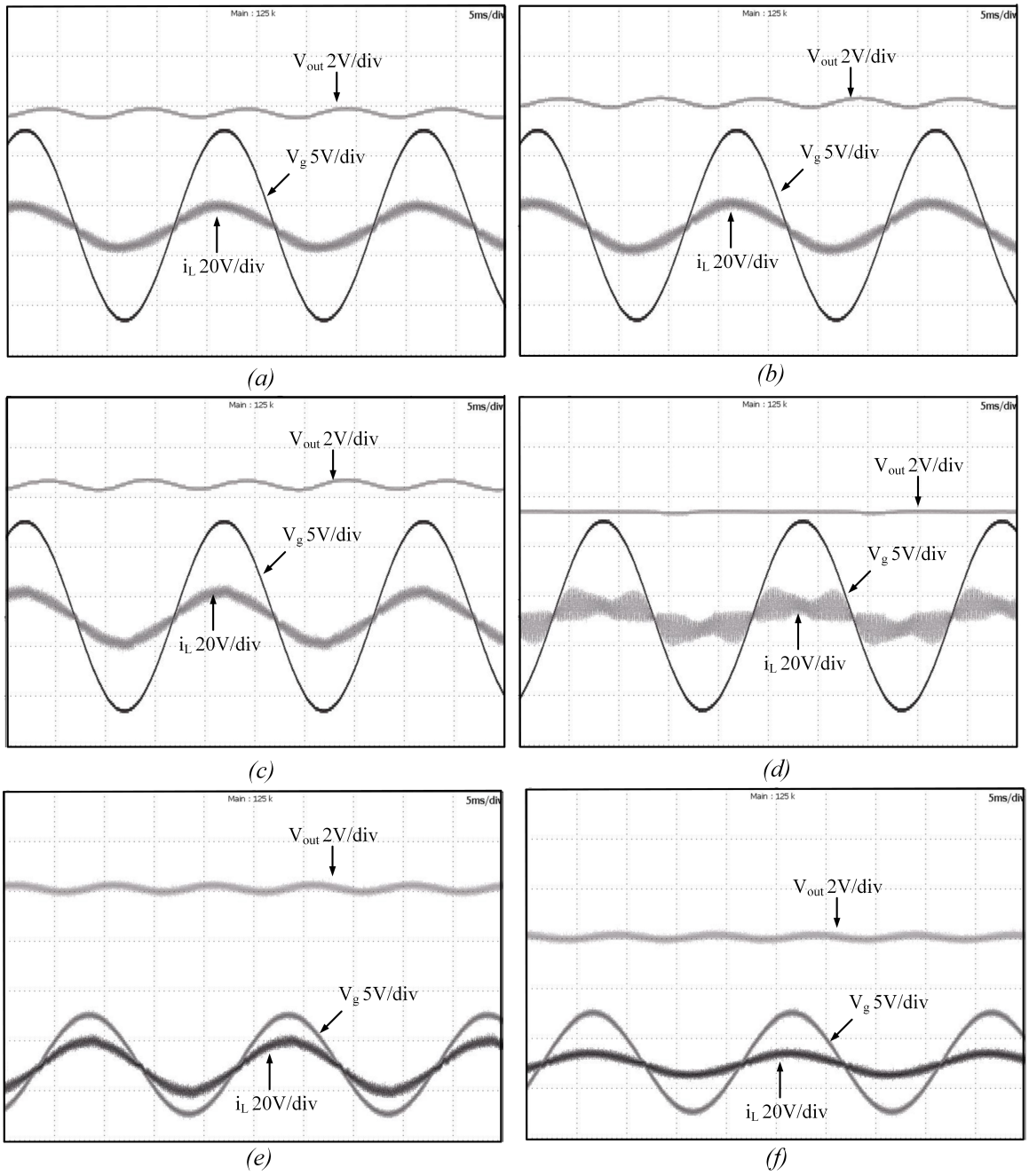


Figure 48. (a) $230V_{RMS}, 75\Omega, 380V_{out}$(b) $230V_{RMS}, 75\Omega, 400V_{out}$ (c) $230V_{RMS}, 75\Omega, 420V_{out}$ (d) $230V_{RMS}, 500\Omega, 380V_{out}$

6 Non-Unity Active Power Factor Control

There is a growing interest in smart home and manufacturing plant technologies, as well as utility-consumer contracts which support efficient and stable power distribution. Through these methods, utilities and industrial consumers of electricity cooperatively benefit from improved system stability and monetary incentives. These benefits are acquired through several avenues, 1) renewable energy production or distributed generation credits, 2) demand response and load shedding through prior notification, and 3) energy efficient appliance and renewable energy subsidies [60-61]. These programs typically require a smart metering interface that records the quantity of energy production by distributed generators or provides a communication interface between the utility and consumer for load shedding notification.

The focus of this thesis is to present solutions to the increasing energy demand and power quality problem through improving PFC performance and stability on the level of individual converters, as well as the performance of the power distribution network through aggregated non-unity active power factor control. Due to their presence in AC power distribution networks, PFC converters operating beneath their rated load conditions are unused resources which can be utilized to compensate line current harmonics and support reactive power. Although full investigation of the benefits is costly and difficult to quantify without a large experimental setup, it is expected that this feature provides improved energy efficiency, reduced degradation of appliances, and the mitigation of poor power factor penalties.

Nonlinear loads, such as rectifiers, cause undesirable harmonics in the line current, which induce deleterious conditions on other devices connected to the point of common coupling (PCC). In order to compensate the harmonics produced by local rectifiers, the BBPFCC will employ harmonic cancellation reference signals provided by a smart metering system.

Furthermore, the available capacity of a BBPFCC can be utilized to compensate

for poor displacement power factor conditions resulting from reactive linear loads. It was shown in [21] that current phase shifting results in increased current distortion due to the unidirectional nature of the topology and input circuit side characteristics. The quantity of reactive power compensation provided by an individual BBPFCC should be limited due to line current THD restrictions. The limitation of reactive power compensation is difficult to characterize because it is dependent on several key parameters, 1) whether the BBPFCC provides inductive or capacitive reactive power since extended cusp distortions in capacitive current mode are worse, 2) the load condition, and, 3) the capability of harmonic current compensation by other devices connected to the PCC. As a result, reactive power compensation as an ancillary service can only be provided in a power distribution network with other systems that support harmonic current compensation functionality.

In order to compensate current harmonics and reactive power, two additional references must be added to the controller detailed in this chapter. In the case of harmonic current compensation, a harmonic reference, $i_{hn}[k]$ is provided to the BBPFCC. This signal is added to the current reference signal generated by the outer voltage loop and reactive power control loop. The reactive power control loop, which employs a low gain PI compensator, similar to the voltage loop, and calculates the error signal and corresponding correction phase angle using the peak voltage, peak current reference produced by the voltage loop (both to produce RMS values), and phase angle difference (tracked through a 1ϕ PLL algorithm). The resultant output of this loop is the current phase reference, $i_{ph}[k]$.

The proposed discrete control algorithm employed in this paper is depicted in Figure 49,

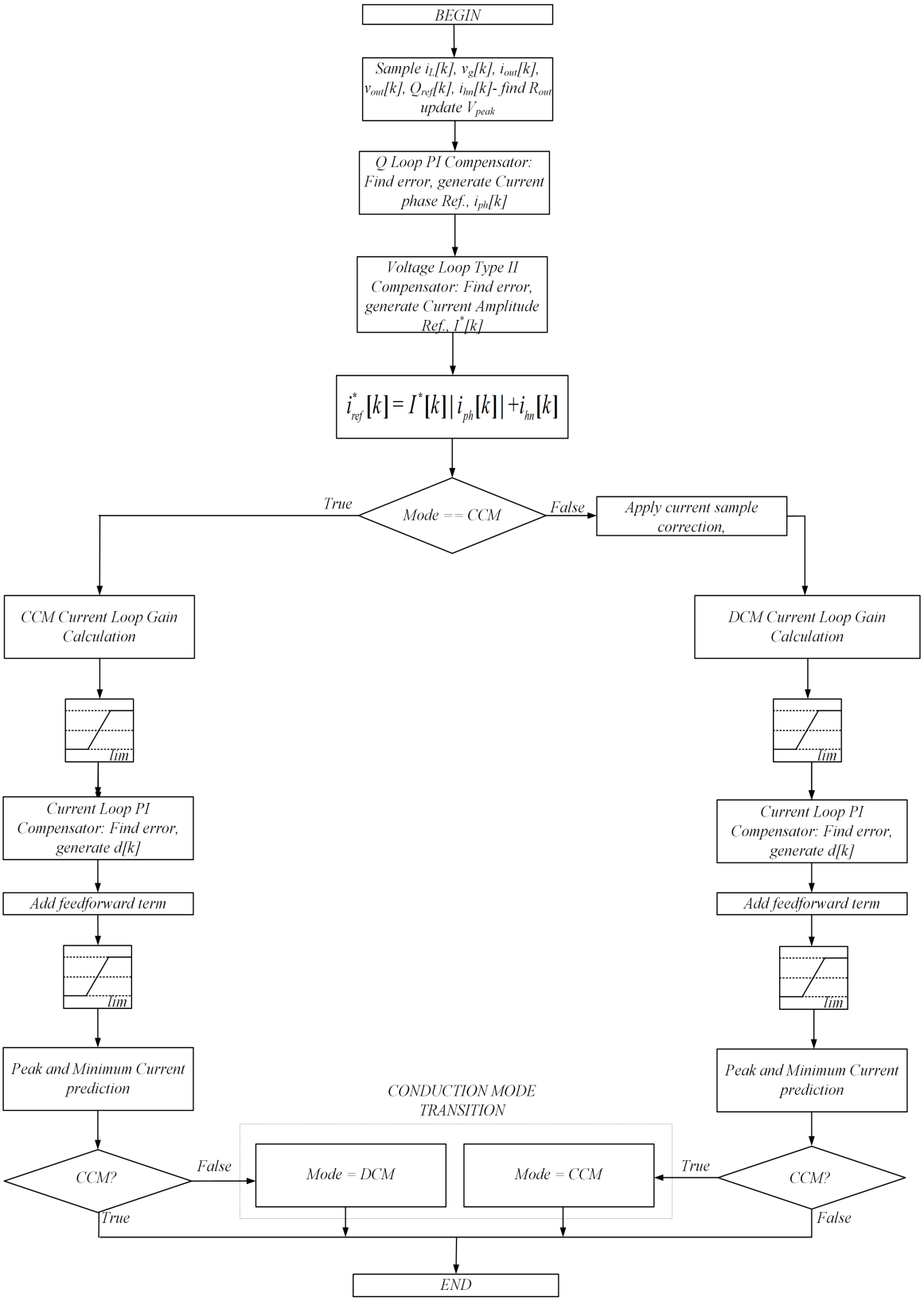


Figure 49. Proposed discrete control algorithm

6.1 Extension of the Proposed Control Algorithm to Smart Home Applications

In this chapter, it is demonstrated that the proposed controller of Figure 39 can be utilized as a stable PFC solution over various operating conditions while improving the stability of the local PCC. This strategy is useful for industrial plants which may suffer from reactive power penalties or equipment downtime due to poor power factor conditions. Although not investigated in this article, it is presumable that similar compensation strategies could be employed in high performance naval and aerospace power distribution systems with AC and DC buses that feed vital equipment. Since the BBPFCC will produce increased line current harmonics, a bidirectional converter, such as a photovoltaic (PV) inverter that compensates line current harmonics is also required.

In the following test scenarios, a supervisory controller embedded in a smart meter feeds harmonic current and reactive power references to the nonlinear subsystems (i.e. unidirectional BBPFCC converters and a bidirectional PV inverter). The nonlinear subsystems will cooperatively support the reactive power demand of a local linear load and compensate the harmonics generated by a local rectifier load. As a result, the peak grid current is reduced. The parameters of each subsystem are shown in Table II,

TABLE II

	$BBPFCC_{1\&2}$	$PVInverter$	$Rectifier$	$LinearInductiveLoad$
$S_{rated}(VA)$	$2.133k$	$1.7k$	220	$1.66k$
$PowerFactor$	0.999	0.999	0.50	0.16
$\%THD$	1.79%	0.79%	167%	0%

The rectifier load causes severe current distortion, and the reactive power consumption of the linear load significantly deteriorates the power factor of the PCC. In

this test, four modes are investigated as follows;

1. 0.15 - 0.25 seconds: Nominal operation
   - Both BBPFCCs operate at 80% load in unity power factor mode.
   - The PV inverter injects 1.6kW into the PCC.
   - The linear load consumes 1.6kVAR and 250W.
   - The rectifier consumes 200W and 25VAR.

2. 0.25 - 0.35 seconds: Reactive power compensation mode
   - Both BBPFCCs operate at 80% load, but each provide capacitive 0.53kVAR.
   - The PV inverter supports 1.5kW and capacitive 0.53kVAR.
   - The linear load consumes 1.6kVAR and 250W.
   - The rectifier consumes 200W and 25VAR.

3. 0.35 - 0.45 seconds: Reactive power & BBPFCC harmonic compensation mode
   - Both BBPFCCs operate at 80% load, but each provide capacitive 0.53kVAR. These converters compensate the rectifier harmonics.
   - The PV inverter supports 1.5kW and capacitive 0.53kVAR. This converter compensates the harmonics generated by the BBPFCCs.
   - The linear load consumes 1.6kVAR and 250W.
   - The rectifier consumes 200W and 25VAR.

4. 0.45 - 0.65 seconds: Load step
   - Both BBPFCCs operate drop to 50% load, but each provide capacitive 0.53kVAR. These converters compensate the rectifier harmonics.
   - The PV inverter supports 1.5kW and capacitive 0.53kVAR. This converter compensates the harmonics generated by the BBPFCCs.
   - The linear load consumes 1.6kVAR and 250W.
   - The rectifier consumes 200W and 25VAR.

The distribution network is simulated in PSIM, and the configuration of all loads are depicted in Figure 50. Note that a short line impedance model is included in the transmission line.

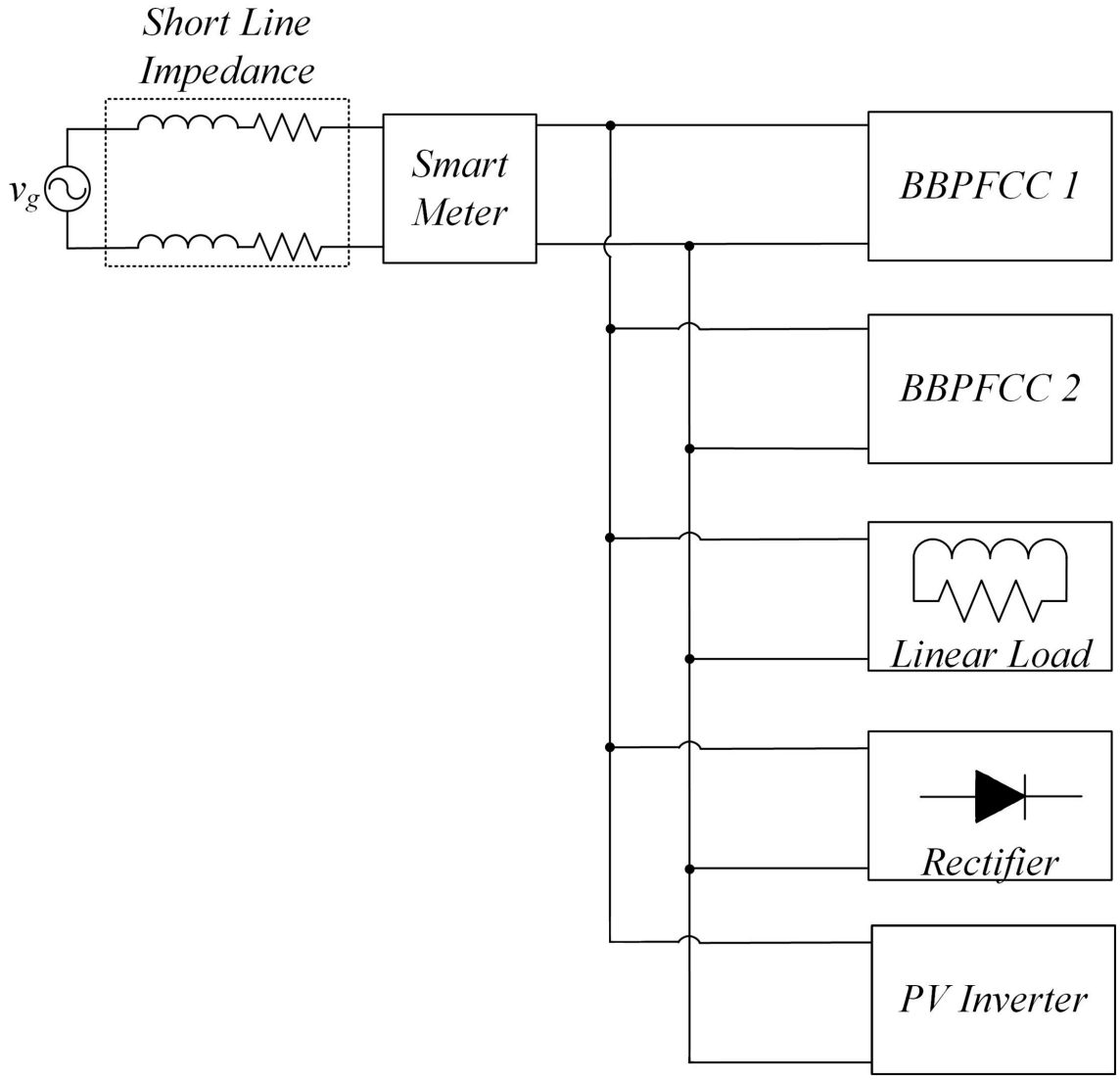


Figure 50. Smart home configuration

The simulation results are shown below in Figures 51-52.

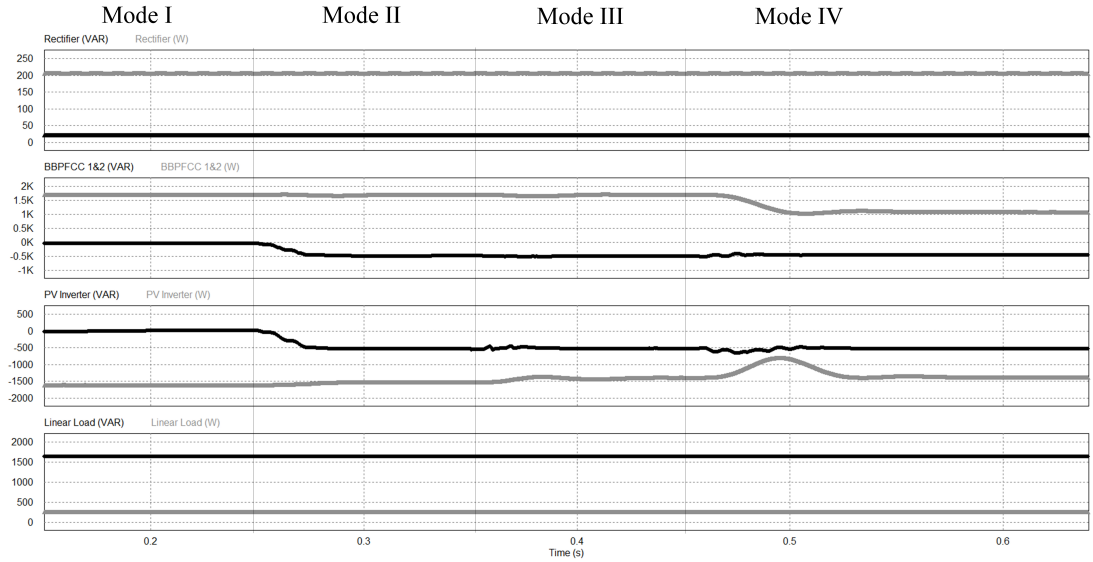


Figure 51. Real and reactive power consumption and compensation

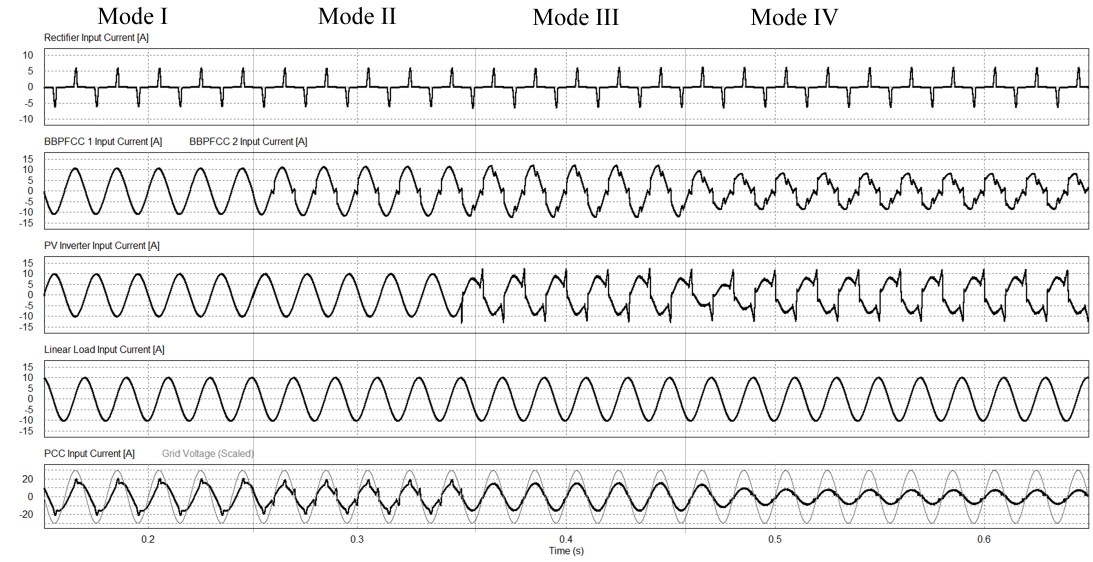


Figure 52. Rectifier, BBPFCC, PV inverter, linear load, and PCC current.

Figure 51 demonstrates that the net reactive power consumption of the smart home can be driven to zero when the BBPFCCs and PV inverter cooperatively compensate for the lagging power factor of the linear load. If the PV inverter was the only compensation system, the reactive power consumption of the linear load would exceed the apparent power capability of the PV system. Consequently, the BBPFCCs enable the smart home to achieve near unity power factor. The input currents of

each device are shown in Figure 52. In Mode II, the BBPFCCs introduce harmonic pollution into the system due to their net compensation of 1.06kVAR; despite the increase in current THD, the peak current draw of the smart home is reduced. In Mode III, the PV inverter compensates for the zero crossing distortions produced by the BBPFCC's. In Mode IV, each BBPFCC experiences a load step from 80% to 50%, in order to demonstrate several key points, 1) although a load reduction results in increased capacity for reactive power compensation, their reference Q should be adjusted relative to the limited compensation capabilities of the PV inverter due to the increased zero crossing distortions, 2) the adaptive algorithm of the BBPFCC has been shown to benefit both the stability of the circuit as well as the stability of the PCC.

The output voltage of the BBPFCCs are overlaid on Figure 53, shown below. This demonstrates that the adaptive algorithm is able to regulate the output voltage successfully while compensating current harmonics and reactive power, and undergoing a significant load step change.

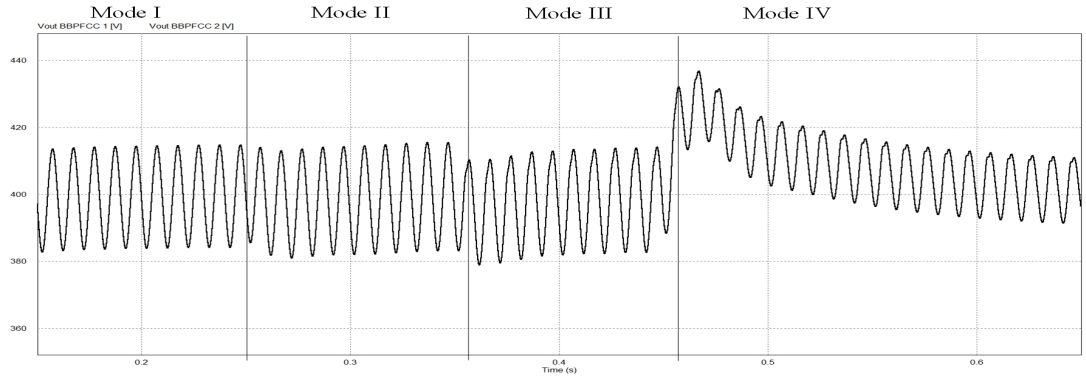


Figure 53. BBPFCC output voltages.

Metrics produced in the simulation are shown in Table III. The peak current draw of the fundamental frequency is reduced during Modes II and III. The average conduction losses occurring as a result of the short transmission line model are computed

Table III: Power Quality Metrics

	BBPFCC 1&2				PV Inverter			
Mode	I	II	III	IV	I	II	III	IV
%THD	1.65	14.5	18.25	30.6	0.48	0.48	36.73	39.97
PF	0.99	0.95	0.94	0.88	0.99	0.94	0.87	0.87
Peak Current (A)	10.7	11.18	12.07	8.17	9.99	10.0	12.1	11.9

	Linear Load				PCC Current			
Mode	I	II	III	IV	I	II	III	IV
%THD	0.03	0.05	0.05	0.07	13.11	26.04	3.18	7.48
PF	0.16	0.16	0.16	0.16	0.82	0.96	0.99	0.98
Peak Current (A)	10.0	10.0	10.0	10.0	20.20	19.39	15.44	7.7

	Rectifier			
Mode	I	II	III	IV
%THD	165.38	165.31	167.11	167.10
PF	0.50	0.50	0.50	0.50
Peak Current (A)	6.06	6.06	6.06	6.06

over 0.05 second intervals, shown below in Figures 54-56. Compared with the average conduction losses computed in Mode I, the reduction in peak current during Mode II results in a 21% increase in power transfer efficiency.

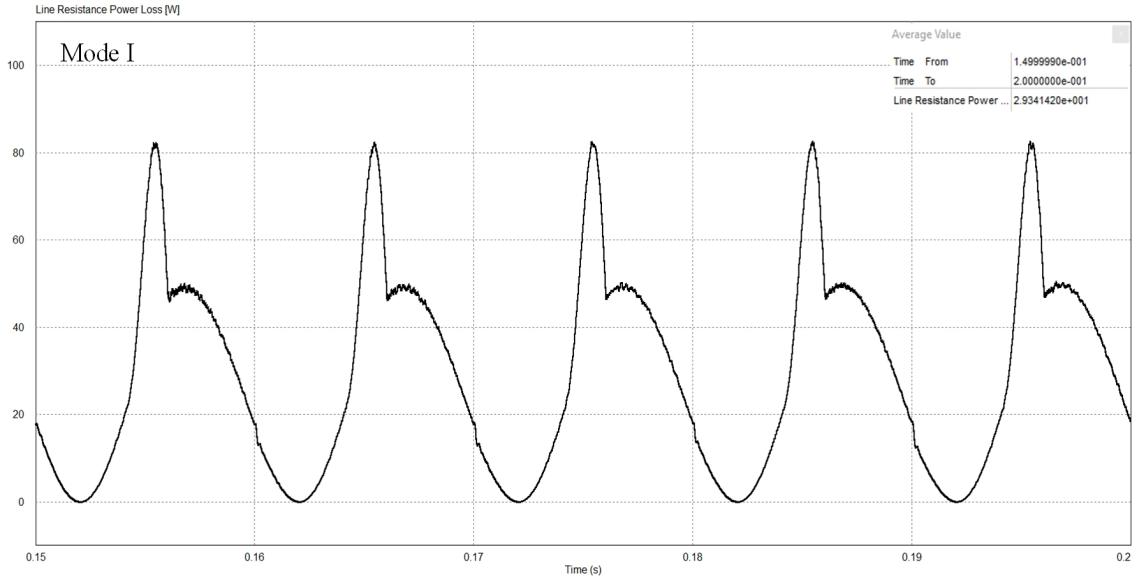


Figure 54. I^2R transmission line losses during Mode I.

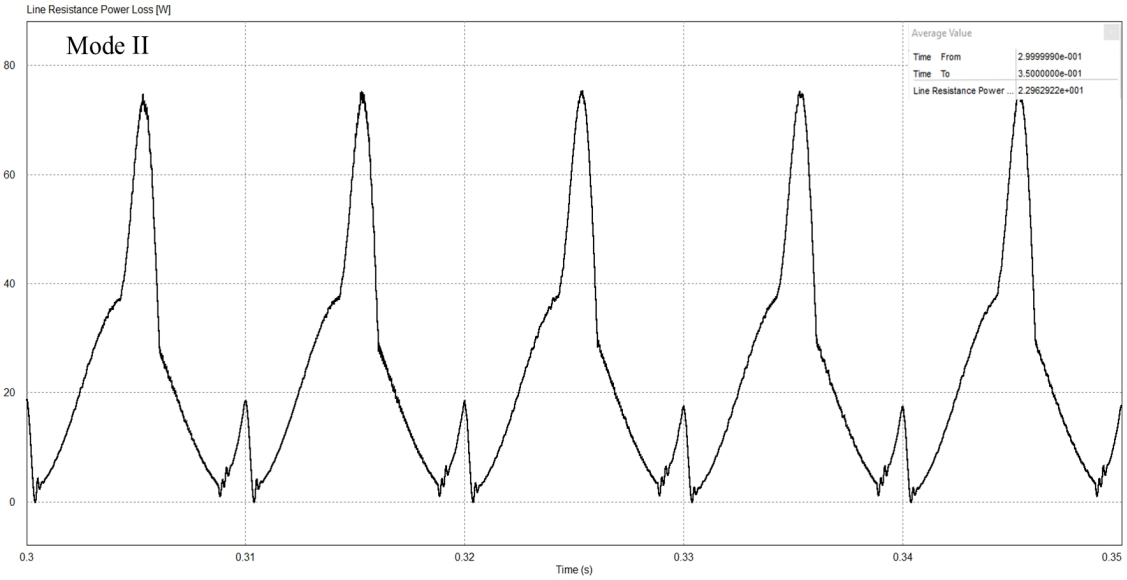


Figure 55. I^2R transmission line losses during Mode II.

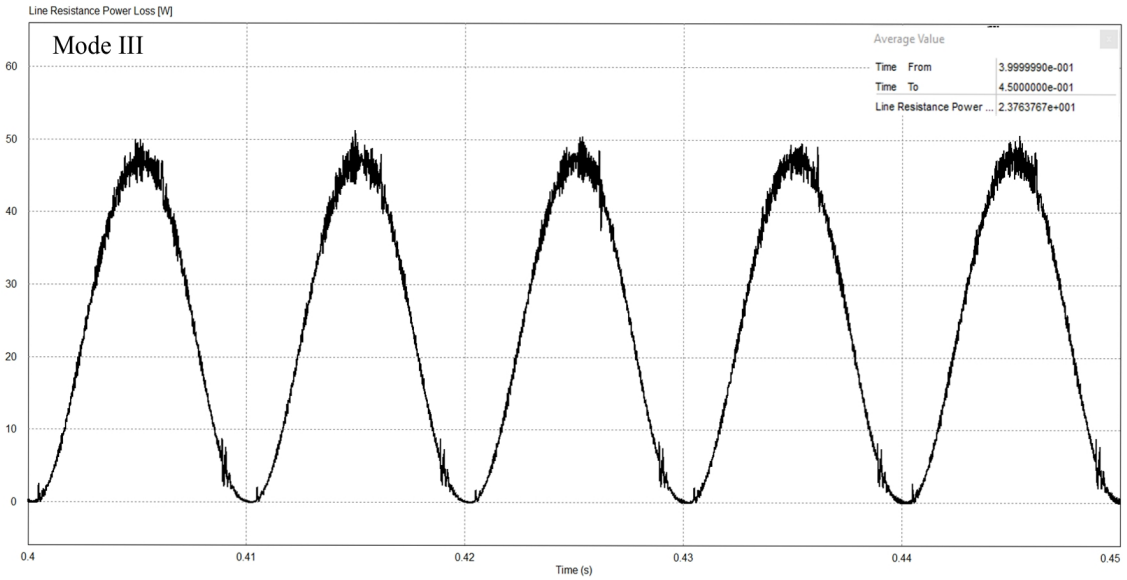


Figure 56. I^2R transmission line losses during Mode III.

7 Conclusion

This thesis investigated the following topics, 1) a brief comparison of boost PFC variants was provided; 2) description of typical controller structures and a subsequent derivation of all CCM and DCM transfer functions through the state space averaging method; 3) slow and fast scale instability issues were described, 4) a method to extend the operating range and flexibility of the PFC was shown. Input current regulation problems were ameliorated with an adaptive gain controller based on the resonant and pole frequencies of the system transfer functions in both CCM and DCM; a simple algorithm was developed to accommodate for conduction mode transitions such that the controller can maintain stability over a wide array of load conditions; harmonic and reactive power compensation references were added to the PFC controller in order to improve the current quality at the point of common coupling.

In a smart home with a metering interface that provides power quality improvement references (harmonic current and reactive power), it was found that the line conduction losses were reduced by 21% if a bidirectional converter is available to compensate for increased zero crossing distortions caused by the bridgeless PFC converters.

8 References

[1] S. Chattopadhyay, M. Mitra, S. Sengupta, *Electric Power Quality* Springer, 2011

[2] M. Bollen, *Understanding Power Quality Problems: Voltage Sags and Interruptions* Wiley, 1999

[3] A. Baggini, *Handbook of Power Quality* Wiley, 2008

[4] Federal Energy Regulatory Commission (2005, Feb.) *Principles for Efficient and Reliable Reactive Power Supply and Consumption* FERC, Available: www.ferc.gov/EventCalendar/Files/20050310144430-02-04-05-reactive-power.pdf

[5] S. Bhattacharyya, S. Cobben. (2011, Apr.) *Consequences of Poor Power Quality - An Overview* InTech, Available: http://www.intechopen.com/books/power-quality/consequences-of-poor-power-quality-an-overview

[6] A. Kumar *Harmonic Compensation of Voltage and Current Using UPQC* GRIN, 2014

[7] N. Hingorani, L. Gyugyi, *Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems* Wiley, 2013

[8] A. Abed (1999, Oct.) *Flexible AC Transmission Systems Benefits Study* California Energy Commission.

[9] S. GJSM S. Rasoolahemmed *Importance of Active Filters for Improvement of Power Quality* IJETT Vol 4. Issue 4, April 2013

[10] U.S. Congress (2005, Jul.) *Energy Policy Act of 2005* Section 1252

[11] U.S. Department of Energy (2006, Feb.) *Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them.*

[12] P. Jahangiri, *Voltage and Reactive Power Regulation by Photovoltaics in Distribution Systems* Iowa State University, 2014, Available lib.dr.iastate.edu/cgi/viewcontent.cgi?article=4726& context=etd

[13] National Renewable Energy Laboratory, (2014, Nov.) *Advanced Inverter Functions to Support High Levels of Distributed Solar* U.S. Department of Energy, 2014, Available http://www.nrel.gov/docs/fy15osti/62612.pdf

[14] A. Ellis, R. Nelson, E. Von Engeln, R. Walling, J. MacDowell, L. Casey, E. Seymour, W. Peter, C. Barker, B. Kirby, J.R. Williams, *Reactive Power Performance Requirements for Wind and Solar Plants* Sandia National Laboratories

[15] M. Mohseni, *Enhanced Reactive Power Support Capability of Fully Rated Converter-Based Wind Generators* IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society, Melbourne, VIC, 2011, pp. 2486-2491.

[16] Bo Sun, T. Dragicevic, M. Savaghebi, J. C. Vasquez and J. M. Guerrero, *Reactive Power Support of Electrical Vehicle Charging Station Upgraded with Flywheel Energy Storage System* PowerTech, 2015 IEEE Eindhoven, Eindhoven, 2015, pp. 1-6.

[17] F. Alsokhiry and K. L. Lo, *Provision of Reactive Power Support Ancillary Services from Distributed Generation Based on Renewable Energy* Renewable Energy Research and Applications (ICRERA), 2013 International Conference on, Madrid, 2013, pp. 1018-1023.

[18] Y. D. Lee and S. Y. Park, *Reactive Power Support Capabilities of Nonsynchronous Interconnection Systems in Microgrid Applications* 2016 IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, 2016, pp. 125-131.

[19] E. Demirok, D.S. Remus Teodorescu *Investigation of Extra Power Loss Sharing Among Photovoltaic Inverters Caused by Reactive Power Management in Distribution Networks* 2016 IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, 2016, pp. 125-131.

[20] M. A. Fasugba and P. T. Krein, *Gaining Vehicle-to-Grid Benefits with Unidirectional Electric and Plug-In Hybrid Vehicle Chargers* 2011 IEEE Vehicle Power and Propulsion Conference, Chicago, IL, 2011, pp. 1-6.

[21] S.M. Park, S.Y. Park *Versatile Unidirectional AC-DC Converter with Harmonic Current and Reactive Power Compensation for Smart Grid Applications* 2016 IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, 2016, pp. 125-131.

[22] N. Mohan, W. Robbins, T. Undeland, *Power Electronics: Converters, Applications, and Design* Wiley, 2006, 487-502

[23] R. Erickson, D. Maksimovic, *Fundamentals of Power Electronics* Kluwer Academic, 2001, 589-691

[24] Siemens (2013, May.) *Harmonics in Power Systems: Causes, Effects and Control* Siemens USA. Available: http://www.industry.usa.siemens.com/drives/us/en/electric-drives/ac-drives/Documents/DRV-WP-drive_harmonics_in_power_systems.pdf

[25] M. Yilmaz and P. T. Krein, *Review of Battery Charger Topologies, Charging Power Levels, and Infrastructure for Plug-In Electric and Hybrid Vehicles* in IEEE Transactions on Power Electronics, vol. 28, no. 5, pp. 2151-2169, May 2013.

[26] C. K. Tse, O. Dranga and H. C. C. Iu, *Bifurcation Analysis of a Power-Factor-Correction Boost Converter: Uncovering Fast-Scale Instability* Circuits and Sys-

tems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on, 2003, pp. III-312-III-315 vol.3.

[27] M. Orabi and T. Ninomiya, *Bifurcation Analysis of Pre-Regulator PFC Boost Converter* Telecommunications Energy Conference, 2003. INTELEC '03. The 25th International, Yokohama, Japan, 2003, pp. 559-564.

[28] A. Gosh, *Nonlinear Dynamics of Power-Factor-Corrected AC-DC Boost Regulator* Lambert Academic Publishing, 2012, pp. 41-56

[29] D. Sun, W. Sun, Q. Wang, X. Shen, S. Lu, *A Novel Digital Controller for Boost PFC Converters with High Power Factor and Fast Dynamic Response*

[30] Y. Fukaishi, K. Higuchi, H. Furuya, Y. Satake, *Design of Robust Digital Controller for Interleaved PFC Boost Converter with DC-DC Converter Load.* 2012 IEEE International Conference on Electron Devices and Solid State Circuit (EDSSC), Bangkok, 2012, pp. 1-2.

[31] Y. Adachi, O. Yoshihiro, K. Tatsuyoshi, H. Kohji, *Design of a Robust Digital Controller for a PFC Boost Converter.* 2012 Proceedings of SICE Annual Conference (SICE), Akita, 2012, pp. 2109-2114.

[32] P. Das, M. Pahlevaninezhad, J. Drobnik, G. Moschopoulos, P. Jain, *A Nonlinear Controller Based on a Discrete Energy Function for an AC/DC Boost PFC Converter* in IEEE Transactions on Power Electronics, vol. 28, no. 12, pp. 5458-5476, Dec. 2013.

[33] Q. Li, O. Thomsen, M. Andersen, *Research on EMI reduction of Multi-Stage Interleaved Bridgeless Power Factor Corrector* Proceedings of The 7th International Power Electronics and Motion Control Conference, Harbin, China, 2012, pp. 1054-1059.

[34] A. Karaarslan, I. Iskender, *Average Sliding Control Method Applied on Power Factor Correction Converter for decreasing Input Current Total Harmonic Distortion Using Digital Signal Processor* in IET Power Electronics, vol. 5, no. 5, pp. 617-626, May 2012.

[35] J. Ivaldi, S.M. Park, S.Y. Park *Integration Strategy for Bidirectional and Unidirectional Converters Aiming for Zero Power Pollution in Residential Applications* 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), Seoul, 2015, pp. 1756-1761.

[36] F. Musavi, W. Eberle and W. G. Dunford, *A Phase-Shifted Gating Technique With Simplified Current Sensing for the Semi-Bridgeless ACDC Converter* in IEEE Transactions on Vehicular Technology, vol. 62, no. 4, pp. 1568-1576, May 2013.

[37] L. Huber, J. Yuntaek, M. Jovanovic, *Performance Evaluation of Bridgeless PFC Boost Rectifiers* in IEEE Transactions on Power Electronics, vol. 23, no. 3, pp. 1381-1390, May 2008.

[38] L. Silva, F. di Seixas, P. Oliveira, *Experimental Evaluation of the Bridgeless Interleaved Boost PFC Converter* Industry Applications (INDUSCON), 2012 10th IEEE/IAS International Conference on, Fortaleza, 2012, pp. 1-7.

[39] L. Zhou, Y. Wu, J. Honea and Z. Wang, *High-efficiency True Bridgeless Totem Pole PFC based on GaN HEMT: Design Challenges and Cost-effective Solution* PCIM Europe 2015; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of, Nuremberg, Germany, 2015, pp. 1-8.

[40] H. Wei, I. Batarseh, *Comparison of Basic Converter Topologies for Power Factor Correction* Southeastcon '98. Proceedings. IEEE, Orlando, FL, 1998, pp. 348-353.

[41] L. Rossetto, G. Spiazzi, P. Tenti, *Control Techniques for Power Factor Correction Converters* Pennsylvania State University, 1994. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.455.3113&rank=2

[42] Jian Sun, *On the zero-crossing distortion in single-phase PFC converters," in IEEE Transactions on Power Electronics,* vol. 19, no. 3, pp. 685-692, May 2004.

[43] A. Pandey, B. Singh and D. P. Kothari, *A simple fast voltage controller for single-phase PFC converters,* IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the], 2002, pp. 1235-1237 vol.2.

[44] T. Takeshita, Y. Toyoda and N. Matsui, *Harmonic suppression and DC voltage control of single-phase PFC converter,* Power Electronics Specialists Conference, 2000. PESC 00. 2000 IEEE 31st Annual, Galway, 2000, pp. 571-576 vol.2.

[45] S. Wall and R. Jackson, *Fast Controller Design for Single-Phase Power-Factor Correction Systems,* in IEEE Transactions on Industrial Electronics, vol. 44, no. 5, pp. 654-660, Oct 1997.

[46] J. Rajagopalan, J. G. Cho, B. H. Cho and F. C. Lee, *High performance control of single-phase power factor correction circuits using a discrete time domain control method,* Applied Power Electronics Conference and Exposition, 1995. APEC '95. Conference Proceedings 1995., Tenth Annual, Dallas, TX, 1995, pp. 647-653 vol.2.

[47] Jian Sun, D. M. Mitchell, M. F. Greuel, P. T. Krein and R. M. Bass, *Averaged Modeling of PWM Converters Operating in Discontinuous Conduction Mode* IEEE Transactions on Power Electronics, vol. 16, no. 4, pp. 482-492, Jul 2001.

[48] P. V. Tran, R. W. Cox and J. M. Anderson, *Using Input Condition Monitoring to Improve the Stability of Digitally Controlled Power Factor Correcting Converters* 2010 IEEE 12th Workshop on Control and Modeling for Power Electronics (COMPEL), Boulder, CO, 2010, pp. 1-6.

[49] S. I. Seleme, A. H. R. Rosa, L. M. F. Morais, P. F. Donoso-Garcia and P. C. Cortizo, *Evaluation of Adaptive Passivity-Based Controller For Power Factor Correction Using a Boost Converter* in IET Control Theory & Applications, vol. 6, no. 14, pp. 2168-2178, September 20 2012.

[50] Y. Ohta, K. Higuchi, T. Kajikawa, T. Matsushima and M. Suzuki, *Robust digital control for PFC boost converter,* SICE Annual Conference (SICE), 2011 Proceedings of, Tokyo, 2011, pp. 1410-1415.

[51] H.S. Athab *Control Strategy for Discontinuous Conduction Mode Boost Rectifier with Low Total Harmonic Distortion and Improved Dynamic Response* American Journal of Engineering and Applied Sciences, 2008. ISSN 1941-7020

[52] T. S. Hwang and S. Y. Park, *Seamless Boost Converter Control Under the Critical Boundary Condition for a Fuel Cell Power Conditioning System* in IEEE Transactions on Power Electronics, vol. 27, no. 8, pp. 3616-3626, Aug. 2012.

[53] R. K. Tripathi, S. P. Das, and G. K. Dubey, *Mixed-mode operation of boost switch-mode rectifier for wide range of load variations,* IEEE Trans. Power Electron., vol. 17, no. 6, pp. 9991009, Nov. 2002.

[54] Koen De Gusseme, D. M. Van de Sype, A. P. M. Van den Bossche and J. A. Melkebeek, *Digitally controlled boost power-factor-correction converters operating in both continuous and discontinuous conduction mode,* in IEEE Transactions on Industrial Electronics, vol. 52, no. 1, pp. 88-97, Feb. 2005.

[55] M. A. Ebrahimzadeh and A. R. Rahmati, *Adaptive and Fast-Response Controller for Boost PFC Converter with Wide Range of Operating Conditions* Power Electronic & Drive Systems & Technologies Conference (PEDSTC), 2010 1st, Tehran, Iran, 2010, pp. 157-162.

[56] F. Z. Chen and D. Maksimovi, *Digital Control for Improved Efficiency and Reduced Harmonic Distortion Over Wide Load Range in Boost PFC Rectifiers,* in IEEE Transactions on Power Electronics, vol. 25, no. 10, pp. 2683-2692, Oct. 2010.

[57] C. W. Clark, F. Musavi and W. Eberle, *Digital DCM Detection and Mixed Conduction Mode Control for Boost PFC Converters,* in IEEE Transactions on Power Electronics, vol. 29, no. 1, pp. 347-355, Jan. 2014.

[58] C. Clark, W. Eberle and F. Musavi, *An adaptive digital controller for the mixed conduction mode boost power factor correction converter,* Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE, Long Beach, CA, USA, 2013, pp. 2712-2719.

[59] D. M. V. de Sype, K. De Gusseme, A. P. Van den Bossche and J. A. Melkebeek, *Duty-ratio Feedforward for Digitally Controlled Boost PFC Converters* Applied Power Electronics Conference and Exposition, 2003. APEC '03. Eighteenth Annual IEEE, Miami Beach, FL, USA, 2003, pp. 396-402 vol.1.

[60] Federal Energy Regulatory Commission (2014, Apr.) *Payment for Reactive Power* FERC, Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjQk72QxojOAhUGLyYKHXXqBSYQFggeMAA&url=http%3A%2F%2Fwww.ferc.gov%2Flegal%2Fstaff-reports%2F2014%2F04-11-14-reactive-power.pdf&usg=AFQjCNEPO4fdqNndamjwCZhstAjhkTIXxw&cad=rja

[61] Office of Energy Efficiency and Renewable Energy (2015, Sep.) *Energy Incentives Program, New York* Available: http://energy.gov/eere/femp/energy-incentive-programs-new-york

9 Appendix A: Publication Record

J. Ivaldi, S.M. Park, S.Y. Park *Integration Strategy for Bidirectional and Unidirectional Converters Aiming for Zero Power Pollution in Residential Applications* 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), Seoul, 2015, pp. 1756-1761.

10 Appendix B: Controller Code & Simulation

10.1 PSIM Code

* NOTE *

Some equations are too long to display in one line on the page with indentation.

Indentation adjustments have been made. Any continuation or conclusion of a calculation which overflowed a line is denoted by "..."

* NOTE *

```
// ************************************************************************** //
//                                  PFC CONTROLLER                            //
// ************************************************************************** //

if( (sample_cnt == 50)||(sample_cnt == 0))
{
if(sample_cnt == 0)
{
Vgrid   =  in[0];   // Sample Vgrid
Vgrid   = abs(Vgrid);
Igrid   =  in[1];   // Sample SW 1 current
Vout    =  in[2];   // Sample Vout
Iout    =  in[3];   // Sample Iout
perturb =  in[4];
step    =  in[5];   // Load Step detection
ihn_ref =  in[6];   // Harmonic current cancelation reference.
ihn_cmp =  in[7];
//iL_pk_prediction = Igrid+duty/50*Ts*Vgrid/(2*L);
//disch_prediction = iL_pk_prediction + (Vgrid-Vout)/L*(1-duty/50)*Ts;
//out[11] = iL_pk_prediction;
//out[12] = disch_prediction;
// Find signedness of peak and min predictions
/*if(iL_pk_prediction >= 0) iL_pk_sign = 1;
else iL_pk_sign = -1;
if(disch_prediction >= 0) disch_sign = 1;
else  disch_sign = -1;
if (( iL_pk_sign == disch_sign)||(abs(iL_pk_sign) == (1-disch_sign)))
{
CCM_Flag = 1;
}
else
{
CCM_Flag = 0;
} */
}
// Must wait until Vpeak detected
if (Vpeak == 0.0)
{
// PEAK DETECTION for startup
Vgrid   =  in[0];   // Sample Vgrid
```

```
Vgrid   =  abs(Vgrid);
if((Vgrid < Vgrid_prev) && (Vgrid_prev > Vgrid_2prev)){ Vpeak = Vgrid_prev;}
Vgrid_2prev = Vgrid_prev;
Vgrid_prev = Vgrid;
//out[1] = Vpeak;
}
// once peak voltage has been found
if (Vpeak > 0.0)
{
// System windup time
SUT = 10000;
// PLL Locking time... 25000x1e-5 second time steps
PLLSUT = 25000;
Q_REF =500.0;
if(StartUpTime<SUT) {CCM_Flag=1; CCM_Flag_prev=1;}// Force CCM on startup

// FIXED SAMPLING, Q and PLL.
// Q Control Feedback Loop
// PLL Loop
if(sample_cnt == 50)
{
PLL_V_ERR =Vgrid/Vpeak*PLL_V_FB;
// Generate the estimated change in radian frequency by passing PLL_V_ERR through a LPF that preserves the DC

...

component of PLL_V_ERROR
PLL_DELTA_OMEGA = LPF_Coefficient_1*(PLL_V_ERR + PLL_V_ERR_PREV)+LPF_Coefficient_2*PLL_DELTA_OMEGA_PREV;
// Generate estimated angular velocity with an offset about the desired angular velocity,
PLL_OMEGA_ESTIMATE = PLL_SAMPLE_PRD*PLL_Kp*PLL_DELTA_OMEGA+PLL_SAMPLE_PRD*OMEGA_GRID;
// Integrate the angular velocity, PLL_OMEGA_ESTIMATE to produce the angular position, PLL_THETA_ESTIMATE
PLL_THETA_ESTIMATE = PLL_THETA_ESTIMATE_PREV + PLL_OMEGA_ESTIMATE;
// Limit PLL_THETA_ESTIMATE about 2*PI
if(PLL_THETA_ESTIMATE >= 2*PI) PLL_THETA_ESTIMATE = PLL_THETA_ESTIMATE - 2*PI;
// Generate the Feedback and Synchronization Signals, PLL_V_FB and PLL_V_SYNC
PLL_V_SYNC = -1.00*cos(PLL_THETA_ESTIMATE+PI/180*Q_ANGLE);
out[0] = PLL_V_SYNC;
PLL_V_FB   =  1.00*sin(PLL_THETA_ESTIMATE);
// Pass values to previous state
PLL_THETA_ESTIMATE_PREV = PLL_THETA_ESTIMATE;
PLL_DELTA_OMEGA_PREV    = PLL_DELTA_OMEGA;
PLL_V_ERR_PREV          = PLL_V_ERR;
// Steady state PLL_DELTA OMEGA ranges from 0.10~0.11
if((PLL_DELTA_OMEGA <= 0.06)&&(PLL_DELTA_OMEGA >= -0.06)) { PHASE_IS_LOCKED = TRUE; }
else {  PHASE_IS_LOCKED = FALSE; }
//out[3] = PHASE_IS_LOCKED;
//out[5] = PLL_V_SYNC;
// When the Phase is Locked & PLL Start Up Time has elapsed, Q control can be enabled.
if(PHASE_IS_LOCKED&&!(StartUpTime<PLLSUT))
{
Q_FB = 0.707*Vpeak*0.707*v_err_TYPEII*sin(PI/180*Q_ANGLE);
out[1] = Q_FB;
Q_ERR = Q_REF - Q_FB;
Q_ERR_PI =  (2*kp_Q+Ts*ki_Q)/2*(Q_ERR) + (Ts*ki_Q-2*kp_Q)/2*(Q_ERR_PREV) + Q_ERR_PI_PREV;
Q_ANGLE = Q_ERR_PI;
Q_ERR_PI_PREV = Q_ERR_PI;
```

```
Q_ERR_PREV = Q_ERR;
PLL_V_SYNC = -1.00*cos(PLL_THETA_ESTIMATE+PI/180*Q_ANGLE);
//       out[0] = PLL_V_SYNC;

}
if((CCM_Flag == 1)||(StartUpTime < SUT))
{
// Set load resistance value.
// determine R[k]
if (StartUpTime > 10000) R = Vout/Iout;
else if (step == 0) R=75;
else R=150;
//out[0]= R;
// Inner Current Loop Gain is fixed at the crossover frequency.
// Use theoretical duty ratio for estimation of system characteristics. Also used for feedforward duty ratio
if (PHASE_IS_LOCKED && !(StartUpTime<PLLSUT)) D_Test = 1-(abs(PLL_V_SYNC)*Vpeak)/400;
else D_Test = 1-(abs(Vgrid))/V_out_ref;
if (D_Test >= 1) D_Test = 0.999;
if (D_Test <= 0.0) D_Test = 0.00001;
if (StartUpTime < 50000 ) { StartUpTime++; }
// CALCULATE OUTER LOOP GAINS
// TYPE II CONTROLLER
wc_vi = 100;
wp_vi = 150;
A0 = 0.1206;
// ************************************************************************** //
//                    Outer voltage loop calculation                          //
// ************************************************************************** //
// Digital PI for outer voltage loop.
v_err = V_out_ref - Vout;
// Digital PI controller discrete time difference equation,
v_err_TYPEII = (A0*Ts*Ts+2*A0*Ts/wz_vi)/(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err+(2*A0*pow(Ts,2)/wz_vi)/(4/(wz_vi*wp_vi)

...

+2*Ts/wz_vi)*v_err_prev+(A0*Ts*Ts-2*A0*Ts/wz_vi)/(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err_2prev+(8/(wz_vi*wp_vi))/

...

(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err_TYPEII_prev+(-4/(wp_vi*wz_vi)+2*Ts/wz_vi)/

...

(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err_TYPEII_2prev;


if(PHASE_IS_LOCKED&&!(StartUpTime<PLLSUT)) I_in_ref = v_err_TYPEII*abs(PLL_V_SYNC);
else I_in_ref = v_err_TYPEII*abs(Vgrid)/Vpeak;
out[1] = I_in_ref;
// Harmonic Current Compensation
if (PHASE_IS_LOCKED&&!(StartUpTime<PLLSUT))
{
if ( ihn_cmp && (ihn_ref < 0.0))   I_in_ref = I_in_ref + ihn_ref;
if ( ihn_cmp && (ihn_ref >= 0.0))  I_in_ref = I_in_ref - ihn_ref;
}
// BEGIN CCM
```

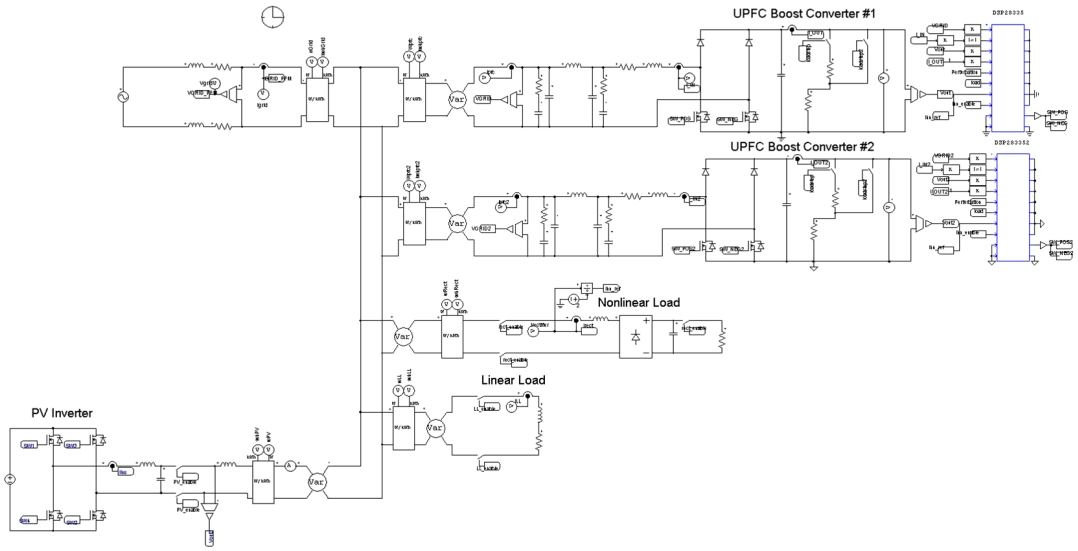
```
// Calculate estimated resonant frequency of the inner current loop
D_RMS = 1-(0.707*Vpeak)/V_out_ref;
wr = (1-D_RMS)/sqrt(L*C);
// Calculate zero frequency of the PI compensator
wz = min(3.16*wr,w/10.00);// Choose either 10^1/2*resonant frequency or w_crossover/10
// Adjust current loop gains based on the sample
// Need to multiply numerator by -1 to extract real result. Same result is imaginary if 1 is chosen
ki = sqrt(tempvar*tempvar/(1/pow(w,2)+1/pow(wz,2)+2/(w*wz)));
kp = ki/(wz);
out[4] = kp;
out[2] = ki;
// ************************************************************************** //
//                      inner current loop calculation                        //
// ************************************************************************** //
i_err = I_in_ref-Igrid;
// Digital PI controller discrete time difference equation,
// y[k] = kp*x[k] + ki*x[k-1] + y[k-1]
i_err_PI = (2*kp+Ts*ki)/2*(i_err) + (Ts*ki-2*kp)/2*(i_err_prev) + i_err_PI_prev;
duty = i_err_PI*50;
duty = duty+D_Test*50;
// Limiter
if(duty > 50)  {duty = 50;}
if(duty < 0   ) {duty = 0 ;}
} else // DCM
{
// determine R[k]
if (StartUpTime > 10000)R = Vout/Iout;
else if (step == 0) R=75;
else R=150;
//out[0]= R;
// TYPE II CONTROLLER
wc_vi = 100;
wp_vi = 150;
A0 = 0.1206;
// ************************************************************************** //
//                     Outer voltage loop calculation                         //
// ************************************************************************** //
// Digital PI for outer voltage loop.
v_err = V_out_ref - Vout;
// Digital PI controller discrete time difference equation,
v_err_TYPEII = (A0*Ts*Ts+2*A0*Ts/wz_vi)/(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err+(2*A0*pow(Ts,2)/wz_vi)/(4/(wz_vi*wp_vi)

...

+2*Ts/wz_vi)*v_err_prev+(A0*Ts*Ts-2*A0*Ts/wz_vi)/(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err_2prev+(8/(wz_vi*wp_vi))/

...

(4/(wz_vi*wp_vi)+2*Ts/wz_vi)*v_err_TYPEII_prev+(-4/(wp_vi*wz_vi)+2*Ts/wz_vi)/(4/(wz_vi*wp_vi)+2*Ts/wz_vi)

...

*v_err_TYPEII_2prev;
if(PHASE_IS_LOCKED&&!(StartUpTime<PLLSUT)) I_in_ref = v_err_TYPEII*abs(PLL_V_SYNC);
else I_in_ref = v_err_TYPEII*abs(Vgrid)/Vpeak;
out[1] = I_in_ref;
```

```
// Harmonic Current Compensation
if (PHASE_IS_LOCKED&&!(StartUpTime<PLLSUT))
{
if ( ihn_cmp && (ihn_ref < 0.0))   I_in_ref = I_in_ref + ihn_ref;
if ( ihn_cmp && (ihn_ref >= 0.0))  I_in_ref = I_in_ref - ihn_ref;
}
if (Vgrid <= 0.00001) Vgrid = 0.00001;
// BEGIN DCM - Inner current loop
// Apply a Correction to the sampled current to find the average current value in DCM...
iL_Average_DCM = duty/50*Igrid+(2*L*Igrid*Igrid)/(Ts*(Vout-Vgrid));
// Expression to include switching transition time -
iL_Average_DCM = 0.5*(duty/50+(L/(Vout-Vgrid))*(2*Igrid+Vgrid*0.0000001/L)/(Ts))*(2*Igrid+Vgrid*0.0000001/L);
D_DCM_FF = sqrt(2*L*I_in_ref*(Vout-Vgrid))/sqrt(Vout*(Vgrid)*Ts);
if (D_DCM_FF>=0.99) D_DCM_FF = 0.99;
if (D_DCM_FF<=0.0) D_DCM_FF = 0.001;
// We need to recalculate the TF when there is
// 1. a different peak input voltage
// 2. a different resistance
// 3. when CCM -> DCM
// Calculate conversion ratio, M
M = Vout/((0.707*Vpeak));
// 0.707*v_err_TYPEII = I in ref RMS
IinRMS = pow(V_out_ref,2.0)/(R*0.707*Vpeak);
D_DCM_RMS = sqrt(2*L*IinRMS*(V_out_ref-0.707*Vpeak))/sqrt(V_out_ref*(0.707*Vpeak)*Ts);
if (D_DCM_RMS>=0.99) D_DCM_RMS = 0.99;
if (D_DCM_RMS<=0.0) D_DCM_RMS = 0.001;
// Calculate the high frequency pole, wp1, in order to cancel it with the compensator's zero
// The low frequency pole, wp2 is canceled by the low frequency zero of the TF
wp1 = (2*R*C*(M-1)-D_DCM_RMS*Ts-sqrt(4*pow(R*C*M,2)-(12*D_DCM_RMS*Ts*R*C+8*pow(R*C,2))*M+pow(D_DCM_RMS*Ts+2*R*C,2)))

...

/(2*R*C*D_DCM_RMS*Ts);
wp2 = (2*R*C*(M-1)-D_DCM_RMS*Ts+sqrt(4*pow(R*C*M,2)-(12*D_DCM_RMS*Ts*R*C+8*pow(R*C,2))*M+pow(D_DCM_RMS*Ts+2*R*C,2)))

...

/(2*R*C*D_DCM_RMS*Ts);
w_dcm = w;
dcmgain1= pow(4.00*V_out_ref,2.00);    //pow(4.00*V_out_ref*D_DCM_RMS*Ts,2.00);
//       out[3] = dcmgain1;
dcmgain2=pow(2.00*R*C*V_out_ref*w_dcm,2.00);//pow(2.00*D_DCM_RMS*Ts*R*C*V_out_ref*w_dcm,2.00);
//       out[5] = dcmgain2;
dcmgain3=pow(L*R*C*w_dcm*w_dcm,2.00);    //pow(L*D_DCM_RMS*Ts*R*C*w_dcm*w_dcm,2.00);
//       out[6] = dcmgain3;
dcmgain4= pow(L*C*R*wp1*wp2,2.00); // pow((L*D_DCM_RMS*Ts+2.00*L*R*C*(M-1.00))*w_dcm,2.00);       //pow((L*D_DCM_RMS*Ts
//       out[7] = dcmgain4;
dcmgain5= pow(L*C*R*(wp1+wp2)*w_dcm,2.0);
//       out[9] = dcmgain5;
gain = 20.00*log10(sqrt((dcmgain1-dcmgain2)/(dcmgain3+dcmgain4-dcmgain5)));
//if (gain > 20.0){ gain = 20.0;}
//if (gain < -20.0) {gain = -20.0;}
//gain = 20;
// DCM crossover frequency adjustment based on R and Vin.
// determined experimentally
if(Vpeak < 400)
```

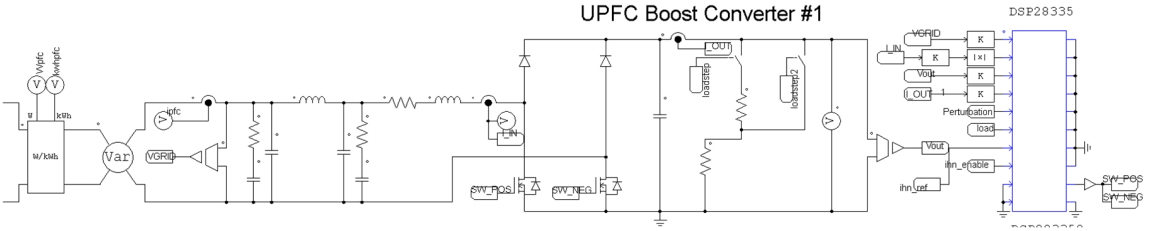
```
{
if((R < 750) ){offset = -3;}
else if ( (R >= 750)&&(R <= 1000) ){ offset = 0; }
else if ( R > 1000 ) {offset = 7; }
}
else
{
if(R<=200){offset = -3;}
else if((R>200)&&(R<=450)) {offset = -7;}
else if(R>450) {offset =-4;}
}
wz = min(wp2*3.16,w_dcm/10.0);
ki_1 = pow(w_dcm,2.00);
ki_2 = pow(10.00,-(gain+offset)/20.00);
ki_3 = pow(wz,2.00);
ki = sqrt(ki_2*ki_2/(1/pow(w,2)+1/pow(wz,2)+2/(w*wz)));
if (ki > 10000.0)  {ki = 10000.0;}
if (ki < -10000.0) {ki = -10000.0;}
kp = ki/(wz);
//out[2] = ki;
if (wz <= 0.0) wz = 0.001;
kp = ki/wz;
//out[4] = kp;
//if(kp >  1) kp =  1;
//if(kp < -1) kp = -1;
mode =0;
// Adjust current sample to DCM average value. We just sampled the 1/2 peak current value.
// by changing the sampling signal 180 out of phase
// ************************************************************************** //
//                    inner current loop calculation                          //
// ************************************************************************** //
i_err = I_in_ref-iL_Average_DCM;
// Digital PI controller discrete time difference equation,
// y[k] = a*x[k] + b*x[k-1] + y[k-1]
i_err_PI = (2*kp+Ts*ki)/2*(i_err) + (Ts*ki-2*kp)/2*(i_err_prev) + i_err_PI_prev;
// IN DCM, we only want to update the duty ratio on a 50 count.
duty = i_err_PI*50;
duty= duty+D_DCM_FF*50;
//      duty = 50-duty;
// Limiter
if(duty > 50){duty = 50;}
if(duty < 0 ){duty = 0 ;}
}


iL_pk_prediction = Igrid+duty_prev/50*Ts*Vgrid/(2*L);
disch_prediction = iL_pk_prediction + (2-duty_prev/50-duty/50)*(Vgrid-Vout)*Ts/(2*L);
//out[11] = iL_pk_prediction;
//out[12] = disch_prediction;

if(iL_pk_prediction >= 0) iL_pk_sign = 1;
else iL_pk_sign = -1;
if(disch_prediction >= 0) disch_sign = 1;
else  disch_sign = -1;

if (( iL_pk_sign == disch_sign)||(abs(iL_pk_sign) == (1-disch_sign)))
```

```
{
CCM_Flag = 1;
}
else
{
CCM_Flag = 0;
}



// Pass to past values...
sample_cnt    = 0;
duty_prev = duty;
i_err_prev    = i_err;
i_err_PI_prev = i_err_PI;
v_err_2prev   = v_err_prev;
v_err_prev    = v_err;
v_err_TYPEII_2prev = v_err_TYPEII_prev;
v_err_TYPEII_prev  = v_err_TYPEII;
Vgrid_2prev = Vgrid_prev;
Vgrid_prev = Vgrid;
Vout_prev = Vout;
Iout_prev = Iout;
prevmode = mode;
ki_prev = ki;
}
}
}
// updown counting PWM to eliminate in samples
if(pwm_count < 50 && upcount) { pwm_count++; }
else if(pwm_count>0 && !upcount) { pwm_count--;  }
else if(!upcount && pwm_count == 0){ upcount = 1; pwm_count++; }
else { upcount = 0; pwm_count--; }
sample_cnt = pwm_count;
if(duty >= pwm_count) { out[8] = 1; }
else out[8] = 0;
out[13] = pwm_count;
out[10] = CCM_Flag;
out[14] = duty;
```

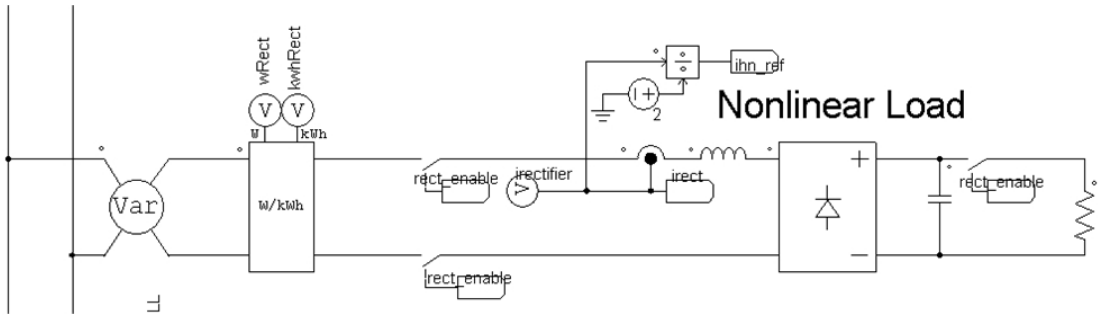
10.2 PSIM Schematics



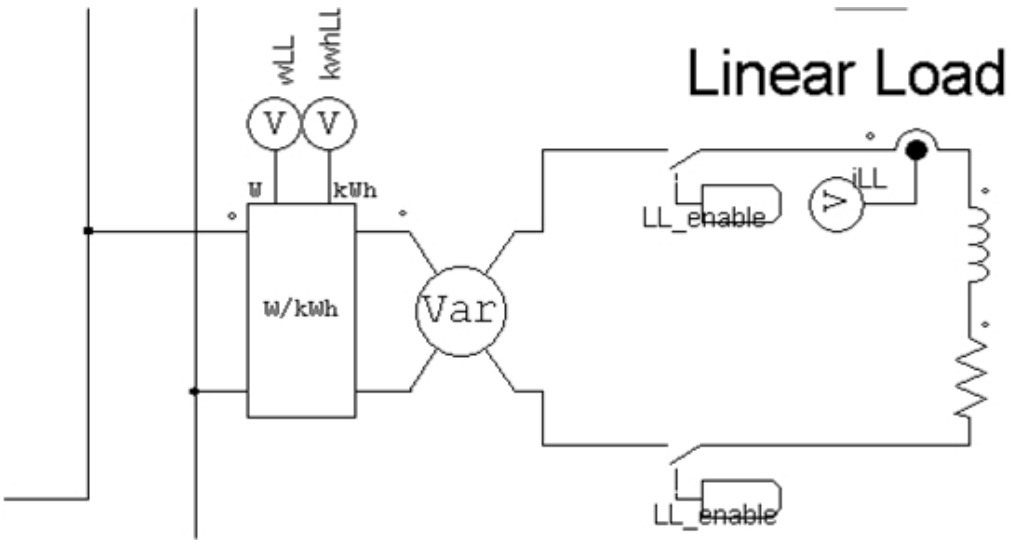
App. B-1 Smart home configuration in PSIM



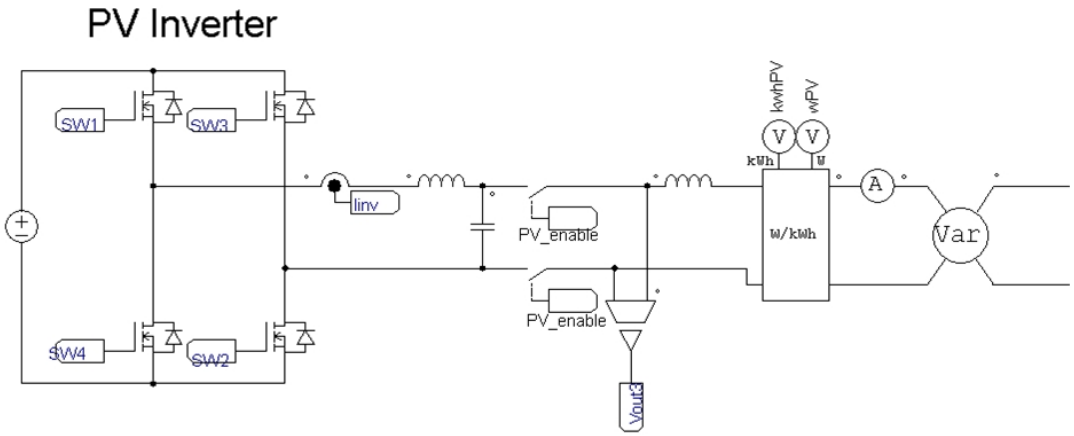
App. B-2 Enlarged shot of PFC boost converter in PSIM



App. B-3 Enlarged shot of rectifier load in PSIM



App. B-4 Enlarged shot of linear load in PSIM



App. B-5 Enlarged shot of PV inverter in PSIM

10.3 DSP Code

```
//#############################################################################
// Description:
//! \addtogroup f2833x_example_list
//! <h1>ePWM Timer Interrupt (epwm_timer_interrupts)</h1>
//!
//! This example configures the ePWM Timers and increments a counter each
//! time an interrupt is taken. \n
//! In this example:
//!  - All ePWM's are initialized.
//!  - All timers have the same period.
//!  - The timers are started sync'ed.
//!  - An interrupt is taken on a zero event for each ePWM timer.
//!  - ePWM1: takes an interrupt every event.
```

```
//!  - ePWM2: takes an interrupt every 2nd event.
//!  - ePWM3: takes an interrupt every 3rd event.
//!  - ePWM4-ePWM6: takes an interrupt every event.
//!
//! Thus the Interrupt count for ePWM1, ePWM4, ePWM5, and ePWM6 should be
//! equal.The interrupt count for ePWM2 should be about half that of ePWM1
//! and the interrupt count for ePWM3 should be about 1/3 that of ePWM1.
//!
//! \b Watch \b Variables \n
//! - EPwm1TimerIntCount
//! - EPwm2TimerIntCount
//! - EPwm3TimerIntCount
//! - EPwm4TimerIntCount
//! - EPwm5TimerIntCount
//! - EPwm6TimerIntCount
//
//
//###########################################################################
// $TI Release: F2833x/F2823x Header Files and Peripheral Examples V140 $
// $Release Date: March  4, 2015 $
// $Copyright: Copyright (C) 2007-2015 Texas Instruments Incorporated -
//             http://www.ti.com/ ALL RIGHTS RESERVED $
//###########################################################################

#include "DSP28x_Project.h"     // Device Headerfile and Examples Include File
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

// These are defined by the linker (see F28335.cmd)
extern Uint16 RamfuncsLoadStart;
extern Uint16 RamfuncsLoadEnd;
extern Uint16 RamfuncsRunStart;
extern Uint16 RamfuncsLoadSize;

// Configure which ePWM timer interrupts are enabled at the PIE level:
// 1 = enabled,  0 = disabled
#define PWM1_INT_ENABLE  1
#define PWM2_INT_ENABLE  1
#define ADC_usDELAY         5000L

// Configure the period for each timer (f_sw=10kHz)
#define PWM1_TIMER_TBPRD   7500
#define PWM2_TIMER_TBPRD   7500
#define HALFPRD                     7500

// Prototype statements for functions found within this file.
__interrupt void epwm3_timer_isr(void);
__interrupt void epwm2_timer_isr(void);
void InitAdc(void);
void ADC_CLK_config(void);
void SPI_CONFIG(void);
void SPI_TRANSMIT(int);
void InitEPwmTimer(void);
void InitEPwm3Gpio(void);
// Global variables used in this example
```

```
Uint32  EPwm1TimerIntCount;
Uint32  EPwm2TimerIntCount;

int V_GRID_DIG = 0;
int I_IN_DIG = 0;
int V_OUT_DIG = 0;
int I_OUT_DIG = 0;
int V_perturb_DIG = 0;
double V_GRID_AN = 0;
double I_IN_AN = 0;
double V_OUT_AN = 0;

#define FALSE               0
#define TRUE                1


double SPItempvar = 0.0;

// SPI
int WriteToSPI = 0;
int SPI_MUX = 3; // write to ki by default


double ipeak = 0;
double Iscale = 10.00;
// I/O
double Vgrid     = 0.00;
double Igrid     = 0.00;
double Vout      = 0.00;
double Iout      = 0.00;

double Vgrid_Abs = 0.00;
// System Parameters
double C = 0.001;
double L = 0.002;
double R = 500; // 75, 150, 300

// CONSTANT DEFINITIONS
const double Ts    = 0.00001; // Sampling period
const double Fsw   = 10000; // Switching period

// Q LOOP CONTROL VARS
double Q_FB           = 0.0;
double Q_ANGLE        = 0.0;
double Q_ERR          = 0.0;
double Q_ERR_PREV     = 0.0;
double Q_ERR_PI       = 0.0;
double Q_ERR_PI_PREV = 0.0;
double kp_Q           = 0.007;
double ki_Q           = 10;
double Q_REF          = 0.0;

// Program Flow Control Flags & Counters
int PHASE_IS_LOCKED = FALSE;
int CCM_Flag         = 1;
int PLLSUT           = 0;
```

```
long long int SUT   = 500000;
long int RMode = 50000;
long int StartUpTime     = 0;
int sample_cnt       = 0;
int pwm_count        = 0;
int upcount          = 1;

// Conduction mode prediction.
int iL_pk_sign = 0;
int disch_sign = 0;
double iL_pk_prediction = 0.00;
double disch_prediction = 0.00;

// Harmonic current compensation variables.
double ihn_ref = 0.00;
int ihn_cmp    = 0;

// Control loop variables...
double w =  0;
double wc_vi               = 100; // 100 rad/s outer loop crossover frequency
double A0                  = 0;   // Gain offset for Type II Controller
double v_err_2prev         = 0;
double v_err_TYPEII_2prev = 0;
double v_err_TYPEII_prev  = 0;
double v_err_TYPEII        = 0;
double iL_Average_DCM      = 0.00;
double DCM_gain            = -19.00;
double D_DCM_FF            = 0.00;
double M                   = 0.00; // Conversion ratio for DCM calculations
double wp1                 = 0.00; // low frequency DCM pole
double wp2                 = 0.00; // high frequency DCM pole
double temp3               = 0;
double Vg_Sign             = 1.0;
double tempvarDCM          = 0;
double w_dcm;
double dcmgain1            = 0;
double dcmgain2            = 0;
double dcmgain3            = 0;
double dcmgain4            = 0;
double dcmgain5            = 0;
double ki_1;
double ki_2;
double ki_3;
//double complex H_MAG      = 0.00000;
double D_Test              = 0.0000; // Feedforward duty ratio
double D_RMS  = 0.00;
double wz                  = 0.00; // zero frequency - current loop
double wr                  = 0.00; // resonant frequency
double ki                  = 0.00;
double kp                  = 0.00;
double temp_vi             = 0.00;
double ki_vi               = 0.00;
double kp_vi               = 0.00;
double Vout_filter         = 0.00;
double v_err               = 0.00;
double v_err_prev          = 0.0;
```

```
double v_err_PI           = 0.00;
double v_err_PI_prev      = 0.0;
double I_in_ref           = 0.00;
double i_err              = 0.00;
double i_err_prev         = 0.0;
double i_err_PI           = 0.00;
double i_err_PI_prev      = 0;
double duty               = 0.00;
double gain               = 29.0;
double Iout_prev                  = 0.00;
double Vout_prev                  = 0.00;
int    hysteresis_counter = 0;
int    CCM_Flag_prev      = 0;

// Used in the ki gain calculation since the current loop crossover frequency is always the same
double temp2              = 0;
double tempvar            = 0.00;
double wp_vi              = 0;
double w_vi_c             = 100; // voltage loop crossover frequency
double gain_vi            = 0.00; // outer loop gain
double wz_vi              = 7620.00; // outer loop PI zero frequency

// DCM
double v_err_DCM = 0.00;
double v_err_prev_DCM = 0.00;
double v_err_PI_DCM = 0.00;
double v_err_PI_prev_DCM = 0.00;
double kp_vi_DCM = 0.001;
double ki_vi_DCM =  0.04;

// Perturbation test
double perturb = 0.00;

// Reference voltage
double Vref = 400.00;
double Vpeak = 325.00;

// Outer loop variables
double COEFF1, COEFF2, COEFF3, COEFF4, COEFF5;


// Load step
int step = 0;


void main(void)
{
// Copy time critical code and Flash setup code to RAM
// This includes the following ISR functions: epwm1_timer_isr(), epwm2_timer_isr()
// epwm3_timer_isr and and InitFlash();
// The  RamfuncsLoadStart, RamfuncsLoadEnd, and RamfuncsRunStart
// symbols are created by the linker. Refer to the F28335.cmd file.

memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (Uint32)&RamfuncsLoadSize);

// Call Flash Initialization to setup flash waitstates
```

```
// This function must reside in RAM
InitFlash();

temp2               = (-1.00*gain/20.00);
tempvar             = pow(10.00,temp2);
i_err_PI_prev       = i_err_PI;
tempvarDCM          = pow(10.00,temp3);
temp3               = (-1.00*DCM_gain/20.00);
w                                   =  Fsw*2*3.14/6;

// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the DSP2833x_SysCtrl.c file.
InitSysCtrl();

// Step 2. Initialize GPIO:
// This example function is found in the DSP2833x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.
// InitGpio();  // Skipped for this example

// Step 3. Clear all interrupts and initialize PIE vector table:
// Disable CPU interrupts
DINT;

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the DSP2833x_PieCtrl.c file.
InitPieCtrl();

// Disable CPU interrupts and clear all CPU interrupt flags:
IER = 0x0000;
IFR = 0x0000;

// Initialize the PIE vector table with pointers to the shell Interrupt
// Service Routines (ISR).
// This will populate the entire table, even if the interrupt
// is not used in this example.  This is useful for debug purposes.
// The shell ISR routines are found in DSP2833x_DefaultIsr.c.
// This function is found in DSP2833x_PieVect.c.
InitPieVectTable();
//InitEPwm1Gpio();
InitEPwm3Gpio();
// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
EALLOW;  // This is needed to write to EALLOW protected registers
PieVectTable.EPWM3_INT = &epwm3_timer_isr;
PieVectTable.EPWM2_INT = &epwm2_timer_isr;
EDIS;    // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize all the Device Peripherals:
// This function is found in DSP2833x_InitPeripherals.c
// InitPeripherals();  // Not required for this example
InitEPwmTimer();    // For this example, only initialize the ePWM Timers
InitAdc();
```

```
// Step 5. User specific code, enable interrupts:

// Initialize counters:
EPwm1TimerIntCount = 0;
EPwm2TimerIntCount = 0;

// Enable CPU INT3 which is connected to EPWM1-6 INT:
IER |= M_INT3;

// Enable EPWM INTn in the PIE: Group 3 interrupt 1-6
//    PieCtrlRegs.PIEIER3.bit.INTx1 = PWM1_INT_ENABLE;
//    PieCtrlRegs.PIEIER3.bit.INTx2 = PWM2_INT_ENABLE;

PieCtrlRegs.PIEIER3.bit.INTx3 = 1;
PieCtrlRegs.PIEIER3.bit.INTx2 = 1;


// Enable global Interrupts and higher priority real-time debug events:
EINT;   // Enable Global interrupt INTM
ERTM;   // Enable Global realtime interrupt DBGM

EALLOW;
GpioCtrlRegs.GPADIR.bit.GPIO0  = 1;     // Set EPWM1A (GPIO0) as output.
GpioCtrlRegs.GPADIR.bit.GPIO30 = 1;             // Min Interrupt Ind
GpioCtrlRegs.GPADIR.bit.GPIO31 = 1;             // Min Interrupt Ind
GpioCtrlRegs.GPBDIR.bit.GPIO32 = 1;             // Min Interrupt Ind
GpioCtrlRegs.GPADIR.bit.GPIO1  = 1;
GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;       // Max Interrupt Ind
GpioCtrlRegs.GPADIR.bit.GPIO2  = 1;
GpioCtrlRegs.GPADIR.bit.GPIO5  = 1;
EDIS;


SPI_CONFIG();

// Initialize to rectifier mode
EPwm3Regs.CMPB = 0;
EPwm2Regs.CMPA.half.CMPA = 0;
//GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;
wz_vi = 2/(R*C)*3.16;
wp_vi = 250;
COEFF1 = (2.0*Ts+wz_vi*pow(Ts,2.0))/(4.0+wp_vi*2.0*Ts);
COEFF2 = (wz_vi*2.0*pow(Ts,2.0))/(4.0+wp_vi*2.0*Ts);
COEFF3 = (wz_vi*pow(Ts,2.0)-2.0*Ts)/(4.0+wp_vi*2.0*Ts);
COEFF4 = (8)/(4+wp_vi*2.0*Ts);
COEFF5 = (wp_vi*2.0*Ts-4.0)/(4.0+wp_vi*2.0*Ts);

// for DAC output
ipeak = 1.414*((Vref*Vref/R)/(0.707*Vpeak));


while(1)
{
//  if(CCM_Flag) GpioDataRegs.GPASET.bit.GPIO30 = 1;
//  else GpioDataRegs.GPACLEAR.bit.GPIO30 = 1;
```

```
if(WriteToSPI)
{
switch(SPI_MUX)
{
case 0 : // Write Recorded Vgrid to SPI
SPItempvar = (int)(4096*((Vgrid/340.00+1.5)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 1 : // Write Recorded Igrid to SPI (Should use Peak current reference to normalize value...)
SPItempvar = (int)(4096*((Igrid/ipeak)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 2 : // Write I reference to SPI (should use peak current reference to normalize value...)
SPItempvar = (int)(4096*((I_in_ref/ipeak)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 3 : // write calculated ki value to SPI (max expected value is ~150)
SPItempvar = (int)(4096*((ki/150.0)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 4 : // Write normalized absolute value of grid voltge to SPI
SPItempvar = (int)(4096*((Vgrid_Abs)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 5 :
SPItempvar = (int)(4096*((iL_pk_prediction/ipeak)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 6 :
SPItempvar = (int)(4096*((disch_prediction/ipeak)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 7 :
SPItempvar = (int)(4096*((duty/HALFPRD)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
case 8 :
SPItempvar = (int)(4096*((D_DCM_FF)/3.0));
SPI_TRANSMIT(SPItempvar);
break;
}
}
}
}

void InitEPwm3Gpio(void)
{
// Basic configurations.
EPwm3Regs.TBPRD                 = PWM1_TIMER_TBPRD;                 // 5kHz. 5uS SYSCLK Prescaled to -> 2.667 * 10^-7
EPwm3Regs.CMPB                  = PWM1_TIMER_TBPRD/2;           // Initialize to 50% Duty. (buffer is inverting)
EPwm3Regs.TBPHS.all                 = 0;                            // Set Phase     register to     zero
EPwm3Regs.TBCTR                 = 0;                            // Clear TB counter.

// Configure Time-Base Control Register.
EPwm3Regs.TBCTL.bit.CTRMODE         = TB_COUNT_UPDOWN;      // Timer Counter Count UP Mode.
```

```
EPwm3Regs.TBCTL.bit.PHSEN       = TB_DISABLE;        // Phase loading disabled.
EPwm3Regs.TBCTL.bit.PRDLD       = TB_SHADOW;         // Period Register (TBPRD) is loded from its shadow register when T
EPwm3Regs.TBCTL.bit.SYNCOSEL    = TB_SYNC_IN;        // Disable EPWMxSYNCO Signal
EPwm3Regs.TBCTL.bit.CLKDIV      = 0;                 // TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV) - CLKDIV
EPwm3Regs.TBCTL.bit.HSPCLKDIV   = 0;                     // TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV) - HSPDIV

// Configure Counter-Compare Control Register.
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;          // Shadow mode. Operates as a double buffer. All writes via the CPU
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;        // Load on CTR = Zero. Time-base counter equal to zero.

// Configure Action Qualifier Output A Control Register.
EPwm3Regs.AQCTLB.bit.CBU        = AQ_CLEAR;
EPwm3Regs.AQCTLB.bit.CBD            = AQ_SET;
//       EPwm3Regs.AQCTLB.bit.ZRO       = AQ_SET;            // Action when counter equals zero - EPWM1A (GPIO0) SET.

EPwm3Regs.ETSEL.bit.INTSEL = 2; // Enable EPWM3 INT on ZERO
EPwm3Regs.ETSEL.bit.INTEN   = PWM1_INT_ENABLE;   // Enable INT
EPwm3Regs.ETPS.bit.INTPRD   = ET_1ST;            // Generate INT on 1st event

//EPwm3Regs.ETSEL.bit.SOCAEN  |= 1; // SOCA triggers ADC Conversions
//EPwm3Regs.ETSEL.bit.SOCASEL |= 1; // trigger occurs when counter = 0
//EPwm3Regs.ETPS.bit.SOCAPRD  |= 1;

EALLOW;
GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 0;                  // Configure GPIO4 as GPIO
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 1;                  // Configure GPIO5 as EPWM3B
EDIS;
}



void InitEPwmTimer()
{
EALLOW;
SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0;       // Stop all the TB clocks
EDIS;

// Setup Sync
EPwm2Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;  // Pass through

// Allow each timer to be sync'ed
EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE;

EPwm2Regs.TBCTL.bit.CLKDIV      = 0x0;//2;                 // TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV) - CLKDIV = 2 is
EPwm2Regs.TBCTL.bit.HSPCLKDIV   = 0x0;//5;                      // TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV) - HSPDIV =
-> 2.6667 * 10^-7 sec/clk tick

EPwm2Regs.TBPRD = PWM2_TIMER_TBPRD;
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;       // Count up
EPwm2Regs.ETSEL.bit.INTSEL = 1;        // Enable INT on PEAK
EPwm2Regs.ETSEL.bit.INTEN = PWM2_INT_ENABLE;    // Enable INT
EPwm2Regs.ETPS.bit.INTPRD = ET_1ST;             // Generate INT on 1st event

// Configure Action Qualifier Output A Control Register.
EPwm2Regs.AQCTLA.bit.CAU        = AQ_CLEAR;
```

```
EPwm2Regs.AQCTLA.bit.CAD            = AQ_SET;
//      EPwm2Regs.AQCTLA.bit.ZRO        = AQ_SET;           // Action when counter equals zero - EPWM1A (GPIO0) SET.



EALLOW;
SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;          // Start all the timers synced
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;                 // Configure GPIO0 as EPWM1A
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;                 // Configure GPIO2 as EPWM2A
EDIS;
}
void InitAdc(void)
{
ADC_CLK_config();

extern void DSP28x_usDelay(Uint32 Count);

// *IMPORTANT*
// The ADC_cal function, which  copies the ADC calibration values from TI reserved
// OTP into the ADCREFSEL and ADCOFFTRIM registers, occurs automatically in the
// Boot ROM. If the boot ROM code is bypassed during the debug process, the
// following function MUST be called for the ADC to function according
// to specification. The clocks to the ADC MUST be enabled before calling this
// function.
// See the device data manual and/or the ADC Reference
// Manual for more information.

EALLOW;
SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 1;
ADC_cal();
EDIS;

// To powerup the ADC the ADCENCLK bit should be set first to enable
// clocks, followed by powering up the bandgap, reference circuitry, and ADC core.
// Before the first conversion is performed a 5ms delay must be observed
// after power up to give all analog circuits time to power up and settle

// Please note that for the delay function below to operate correctly the
// CPU_RATE define statement in the DSP2833x_Examples.h file must
// contain the correct CPU clock period in nanoseconds.

//AdcRegs.ADCTRL3.bit.SMODE_SEL = 0x1;
//AdcRegs.ADCTRL3.all = 0x00E0;           // Power up bandgap/reference/ADC circuits
//DELAY_US(ADC_usDELAY);                  // Delay before converting ADC channels

//EPwm1Regs.ETPS.bit.SOCACNT = 1;
//EPwm1Regs.ETCLR.bit.SOCA = 1;

//AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;          // Reset SEQ1
//AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;
//PieCtrlRegs.PIEACK.all = 0x1;
//      PieCtrlRegs.ETCLR.bit.INT = 1; //need to clear interrupt flag

//AdcRegs.ADCMAXCONV.all = 0x0000;        // Table 2-5, Page 39.  1 Conversion
//AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x3;         //0x4, Setup conv from ADCINA0
//AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1 = 1; // Enable SOCA from ePWM to start SEQ1
```

```
AdcRegs.ADCTRL3.bit.SMODE_SEL = 0x0;
AdcRegs.ADCTRL3.all |= 0x00E0;
AdcRegs.ADCMAXCONV.all = 2;//4;
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0;     //A0 = Vgrid
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 8;     //B0 = Iin
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 1;     //A1 = Vout
// AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 9; //B1 = Iout
//AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 2;   //A2 = Vperturb

DELAY_US(ADC_usDELAY);                   // Delay before converting ADC channels

AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;
AdcRegs.ADCTRL2.bit.RST_SEQ2 = 1;
}
void ADC_CLK_config(void){


EALLOW;
#if (CPU_FRQ_150MHZ)      // Default - 150 MHz SYSCLKOUT
#define ADC_MODCLK 0x3 // HSPCLK = SYSCLKOUT/2*ADC_MODCLK2 = 150/(2*3)   = 25.0 MHz
#endif

SysCtrlRegs.HISPCP.all = ADC_MODCLK;
EDIS;
}
double min(double a,double b){
if(a<b)return a;
else return b;
}
// Interrupt routines uses in this example:
__interrupt void epwm3_timer_isr(void)  // MIN VALUE - Used for DCM Controller
{

EPwm1TimerIntCount++;
// Sample count is 0 and we are in DCM
// Forcing the controller to use CCM on startup...
if (!CCM_Flag&&!(StartUpTime<SUT))
{
WriteToSPI = 1;

int i=0;
AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 0x1;              // start convertsion
for(i=0;i<40;i++){}
AdcRegs.ADCTRL2.bit.SOC_SEQ2 = 0x1;              // start convertsion
V_GRID_DIG = AdcRegs.ADCRESULT0>>4;      // Vgrid
I_IN_DIG   = AdcRegs.ADCRESULT1>>4;   // Iin
V_OUT_DIG  = AdcRegs.ADCRESULT2>>4;   // Vout
// I_OUT_DIG    = AdcRegs.ADCRESULT3>>4;   // Iout
// V_perturb_DIG = AdcRegs.ADCRESULT4>>4;   // Vperturb
AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;          // Reset SEQ1
AdcRegs.ADCTRL2.bit.RST_SEQ2 = 1;          // Reset SEQ1
// Find analog voltage/current values.
V_GRID_AN       = 3.0*(V_GRID_DIG)/4096.00;
I_IN_AN         = (3.00 * I_IN_DIG)/4096.00;
V_OUT_AN        = (3.00 * V_OUT_DIG)/4096.00;
```

```
// Reverse scaling
Vgrid = ((V_GRID_AN-1.5))*20.00/3.00*340.00/10.00;



Igrid = fabs(-1.00*(I_IN_AN-1.5)*20.00/3.00*Iscale/10.00); // 75 ohm = 15A scaling, 150ohm = 15A scaling, 300 ohm = 1
Vout = -1.00*(V_OUT_AN-1.5)*20.00/3.00*500.00/10.00;
Vgrid_Abs   = fabs(Vgrid)/Vpeak;
// ************************************************************************** //
//                    Outer voltage loop calculation                          //
// ************************************************************************** //
// Digital PI for outer voltage loop.
v_err = Vref - Vout;
// Digital PI controller discrete time difference equation,
v_err_TYPEII = COEFF1*v_err+COEFF2*v_err_prev+COEFF3*v_err_2prev+COEFF4*v_err_TYPEII_prev+COEFF5*v_err_TYPEII_2prev;
I_in_ref = v_err_TYPEII*Vgrid_Abs;
// Harmonic Current Compensation
if (!(StartUpTime < SUT))
{               GpioDataRegs.GPBSET.bit.GPIO32 = 1;
if (Vgrid_Abs <= 0.0000) Vgrid_Abs = 0.001;
// BEGIN DCM - Inner current loop
// Apply a Correction to the sampled current to find the average current value in DCM...
iL_Average_DCM = duty/PWM2_TIMER_TBPRD*Igrid+(2*L*Igrid*Igrid)/(Ts*(Vout-Vpeak*Vgrid_Abs));
// Expression to include switching transition time -
//iL_Average_DCM = 0.5*(duty/PWM2_TIMER_TBPRD+(L/(Vout-Vgrid_Abs*325))*(2*Igrid+Vgrid_Abs*325*0.0000001/L)/(Ts))*(2*I
D_DCM_FF = sqrt(2*L*I_in_ref*(Vout-Vgrid_Abs*Vpeak))/sqrt(Vout*(Vgrid_Abs*Vpeak)*Ts);
if (D_DCM_FF>=1.0) D_DCM_FF = 0.99;
if(D_DCM_FF<=0.0)  D_DCM_FF = 0.001;
// Calculate approximate conversion ratio, M
M = Vout/(0.707*Vpeak);
// Calculate the high frequency pole, wp1, in order to cancel it with the compensator's zero
// The low frequency pole, wp2 is canceled by the low frequency zero of the TF
//wp1 = (2*R*C*(M-1)-D_DCM_FF*Ts-sqrt(4*pow(R*C*M,2)-(12*D_DCM_FF*Ts*R*C+8*pow(R*C,2))*M+pow(D_DCM_FF*Ts+2*R*C,2)))/(2
wp2 = (2*R*C*(M-1)-D_DCM_FF*Ts+sqrt(4*pow(R*C*M,2)-(12*D_DCM_FF*Ts*R*C+8*pow(R*C,2))*M+pow(D_DCM_FF*Ts+2*R*C,2)))/(2*

// wp2 = 200000;
w_dcm = w;
dcmgain1=pow(4.00*Vout*D_DCM_FF*Ts,2.00);
dcmgain2=pow(2.00*D_DCM_FF*Ts*R*C*Vout*w_dcm,2.00);
dcmgain3=pow(L*D_DCM_FF*Ts*R*C*w_dcm*w_dcm,2.00);
dcmgain4=pow((L*D_DCM_FF*Ts+2.00*L*R*C*(M-1.00))*w_dcm,2.00);
dcmgain5=pow(4.00*M-2.00,2.00);
gain = 20.00*log10(sqrt((dcmgain1-dcmgain2)/(dcmgain3-dcmgain4+dcmgain5)));
//  if (gain > 25.0){ gain = 25.0;}
//  else if (gain < -25.0) {gain = -25.0;}
//  gain = -15.00;
wz = min(wp2/10.00,w_dcm/10.0);
ki_1 = pow(w_dcm,2.00);
ki_2 = pow(10.00,-(gain+15.00)/20.00); // Want to force fsw to be -15dB
ki_3 = pow(wz,2.00);
ki   = sqrt((-1.00*ki_1*ki_2*ki_2)/(1-ki_1/ki_3));
if (wz <= 0.0) wz = 0.001;
kp = ki/wz;
if(kp > 0.1) kp  =  0.1;
if(kp < -0.1) kp = -0.1;
```

```
//    ki = 100;
//    kp = 0.03;


// Adjust current sample to DCM average value. We just sampled the 1/2 peak current value.
// by changing the sampling signal 180 out of phase
// ************************************************************************** //
//                      inner current loop calculation                        //
// ************************************************************************** //
i_err = I_in_ref-iL_Average_DCM;
// Digital PI controller discrete time difference equation,
// y[k] = kp*x[k] + ki*x[k-1] + y[k-1]
i_err_PI = (2*kp+Ts*ki)/2*(i_err) + (Ts*ki-2*kp)/2*(i_err_prev) + i_err_PI_prev;
// if(i_err_PI < 0.0) i_err_PI = 0.0;
// if(i_err_PI > 1.0) i_err_PI = 1.0;
iL_pk_prediction = 2*Igrid; // in DCM sampling, the sampled current is 1/2 the peak current..
disch_prediction = iL_pk_prediction + (Vgrid_Abs*Vpeak-Vout)/L*(1-duty/PWM2_TIMER_TBPRD)*Ts;
// Find signedness of peak and min predictions
if(iL_pk_prediction >= 0.06) iL_pk_sign = 1; // ADC BIAS
else iL_pk_sign = -1;
if(disch_prediction >= 0.06) disch_sign = 1;
else  disch_sign = -1;

//  if (CCM_Flag != CCM_Flag_prev) { hysteresis_counter++; }

//if (hysteresis_counter<2)
//{
if (( iL_pk_sign == disch_sign)||(abs(iL_pk_sign) == (1-disch_sign))) { CCM_Flag = 1; }
else { CCM_Flag = 0;}

//}
//else {  hysteresis_counter = 0; CCM_Flag_prev = CCM_Flag; }
GpioDataRegs.GPBCLEAR.bit.GPIO32 = 1;
}
}
// Clear INT flag for this timer
EPwm3Regs.ETCLR.bit.INT = 1;
// Acknowledge this interrupt to receive more interrupts from group 3
PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
}
__interrupt void epwm2_timer_isr(void)  //PEAK - Used for CCM Control and Duty Cycle Updates
{

EPwm2TimerIntCount++;
if (StartUpTime < SUT ) { StartUpTime++; CCM_Flag = 1; }

//      CCM_Flag = 1;

if (CCM_Flag == 1 || StartUpTime < SUT )
{
WriteToSPI = 1;
// CALL ADC SAMPLES HERE
int i=0;
AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 0x1;              // start convertsion
for(i=0;i<40;i++){} // Wait for 20 clock cycles...
AdcRegs.ADCTRL2.bit.SOC_SEQ2 = 0x1;              // start convertsion
```

```
V_GRID_DIG = AdcRegs.ADCRESULT0>>4;      // Vgrid
I_IN_DIG   = AdcRegs.ADCRESULT1>>4; // Iin
V_OUT_DIG  = AdcRegs.ADCRESULT2>>4; // Vout
AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;          // Reset SEQ1
AdcRegs.ADCTRL2.bit.RST_SEQ2 = 1;          // Reset SEQ1
// Find analog voltage/current values.
V_GRID_AN      = 3.0*(V_GRID_DIG)/4096.00;
I_IN_AN        = (3.00 * I_IN_DIG)/4096.00;
V_OUT_AN       = (3.00 * V_OUT_DIG)/4096.00;
// Reverse scaling
Vgrid = ((V_GRID_AN-1.5))*20.00/3.00*340.00/10.00;
Igrid = fabs(-1.00*(I_IN_AN-1.5)*20.00/3.00*Iscale/10.00);
Vout = -1.00*(V_OUT_AN-1.5)*20.00/3.00*500.00/10.00;




// Used to force the system to operate in rectifier mode for 2 Seconds
if(StartUpTime >= RMode)
{
GpioDataRegs.GPBSET.bit.GPIO34 = 1;
Vgrid_Abs   = fabs(Vgrid)/Vpeak;
// Inner Current Loop Gain is fixed at the crossover frequency.
D_Test = 1.0-(Vgrid_Abs*Vpeak)/Vref;
if (D_Test >= 1)   D_Test = 0.999;
if (D_Test <= 0.0) D_Test = 0.00001;
// ************************************************************************** //
//                    Outer voltage loop calculation                          //
// ************************************************************************** //
// Digital PI for outer voltage loop.
v_err = Vref-Vout;
v_err_TYPEII = COEFF1*v_err+COEFF2*v_err_prev+COEFF3*v_err_2prev+COEFF4*v_err_TYPEII_prev+COEFF5*v_err_TYPEII_2prev;
I_in_ref = v_err_TYPEII*Vgrid_Abs;
// BEGIN CCM
// Calculate estimated resonant frequency of the inner current loop
D_RMS = (1-0.707*Vpeak/Vref);

wr = (1-D_RMS)/sqrt(L*C);
// Calculate zero frequency of the PI compensator
wz = min(2*wr,w/2.0);//
// Adjust current loop gains based on the sample
// Need to multiply numerator by -1 to extract real result. Same result is imaginary if 1 is chosen
ki = sqrt((-1.00*(w*w)*tempvar*tempvar)/(1.00-w*w/(wz*wz)));
kp = ki/(wz);
// Static testing values... based on RMS for adaptive
//  ki = 0.707*160;
//  kp = 0.039;
// ************************************************************************** //
//                    inner current loop calculation                          //
// ************************************************************************** //
i_err = I_in_ref-Igrid;
// Digital PI controller discrete time difference equation,
i_err_PI = (2*kp+Ts*ki)/2*(i_err) + (Ts*ki-2*kp)/2*(i_err_prev) + i_err_PI_prev;

//  if(i_err_PI < 0.0) i_err_PI = 0.0;
//  if(i_err_PI > 1.0) i_err_PI = 1.0;
```

```
duty = i_err_PI+D_Test;
duty *= PWM2_TIMER_TBPRD;
// Limiter
if(duty > PWM2_TIMER_TBPRD)  {duty = PWM2_TIMER_TBPRD;}
if(duty < 0   ) {duty = 0 ;}
iL_pk_prediction = Igrid+duty/PWM2_TIMER_TBPRD*Ts*Vgrid_Abs*Vpeak/(2*L);
disch_prediction = iL_pk_prediction + (Vgrid_Abs*Vpeak-Vout)/L*(1-duty/PWM2_TIMER_TBPRD)*Ts;
// Find signedness of peak and min predictions
if(iL_pk_prediction >= 0.06) iL_pk_sign = 1; // ADC BIAS
else iL_pk_sign = -1;
if(disch_prediction >= 0.06) disch_sign = 1;
else  disch_sign = -1;
if ( ( iL_pk_sign == disch_sign )||(abs(iL_pk_sign) == (1-disch_sign))) { CCM_Flag =1; }
else { CCM_Flag = 0; }
//if (CCM_Flag != CCM_Flag_prev) { hysteresis_counter++; }
//if (hysteresis_counter<2)
//{
if (( iL_pk_sign == disch_sign)||(abs(iL_pk_sign) == (1-disch_sign))) { CCM_Flag = 1; }
else { CCM_Flag = 0;}
//}
//else {  hysteresis_counter = 0; CCM_Flag_prev = CCM_Flag; } */
// Update Comparator registers
// EPwm3Regs.CMPB = (PWM2_TIMER_TBPRD-duty);
// EPwm2Regs.CMPA.half.CMPA = (PWM2_TIMER_TBPRD-duty);


//EPwm3Regs.CMPB = (0.9*PWM2_TIMER_TBPRD);
//EPwm2Regs.CMPA.half.CMPA = (0.9*PWM2_TIMER_TBPRD);

EPwm3Regs.CMPB = (PWM2_TIMER_TBPRD-duty);
EPwm2Regs.CMPA.half.CMPA = (PWM2_TIMER_TBPRD-duty);


GpioDataRegs.GPBCLEAR.bit.GPIO34 = 1;
}
else // Have not fully run through startuptime
{
EPwm3Regs.CMPB = PWM2_TIMER_TBPRD-0;
EPwm2Regs.CMPA.half.CMPA = PWM2_TIMER_TBPRD-0;
}
}
else
{

GpioDataRegs.GPASET.bit.GPIO30 = 1;


// IN DCM, we only want to update the duty ratio on a 50 count.
duty = i_err_PI+D_DCM_FF;
//duty = D_DCM_FF;
duty *= PWM2_TIMER_TBPRD;
// Limiter
if(duty > PWM2_TIMER_TBPRD){duty = PWM2_TIMER_TBPRD;}
if(duty < 0 ){duty = 0 ;}
// Update Comparator registers
```

```
EPwm3Regs.CMPB = (PWM2_TIMER_TBPRD-duty);
EPwm2Regs.CMPA.half.CMPA = (PWM2_TIMER_TBPRD-duty);


GpioDataRegs.GPACLEAR.bit.GPIO30 = 1;
}
// Pass to past values...
sample_cnt     = 0;
i_err_prev     = i_err;
i_err_PI_prev = i_err_PI;
v_err_2prev    = v_err_prev;
v_err_prev     = v_err;
v_err_TYPEII_2prev = v_err_TYPEII_prev;
v_err_TYPEII_prev  = v_err_TYPEII;
Vout_prev = Vout;
Iout_prev = Iout;
// Clear INT flag for this timer
EPwm2Regs.ETCLR.bit.INT = 1;
// Acknowledge this interrupt to receive more interrupts from group 3
PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
}

void SPI_TRANSMIT(int message)
{
WriteToSPI = 0;
int i = 0;

// Write Command is initiated by driving CS low
GpioDataRegs.GPACLEAR.bit.GPIO12 = 1;  // CS low
GpioDataRegs.GPACLEAR.bit.GPIO15 = 1;  // SCK low
GpioDataRegs.GPASET.bit.GPIO26   = 1;  // LDAC high
GpioDataRegs.GPACLEAR.bit.GPIO24 = 1;  // SI low
// Followed by clocking the four coniguration bits and 12 data bits into the SDI pin on
// rising edges of SCK

GpioDataRegs.GPACLEAR.bit.GPIO24 = 1;  // SI low - DACa
GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPASET.bit.GPIO24 = 1;  // SI hi - buffered
GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPASET.bit.GPIO24 = 1;  // SI hi - 1x gain
GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPASET.bit.GPIO24 = 1;  // SI hi - 1x gain
GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
```

```
for(i = 11; i >= 0; i--)
{

if((message & (0x01<<i))){ GpioDataRegs.GPASET.bit.GPIO24 = 1;  // SI hi

GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);

} else { GpioDataRegs.GPACLEAR.bit.GPIO24 = 1;  // SI lo

GpioDataRegs.GPASET.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO15   = 1;  // SCK clock
DELAY_US(1);
}

}
GpioDataRegs.GPASET.bit.GPIO12 = 1;  // CS high
DELAY_US(1);
GpioDataRegs.GPACLEAR.bit.GPIO26 = 1;  // LDAC low
DELAY_US(1);
GpioDataRegs.GPASET.bit.GPIO26 = 1;  // LDAC low

}
void SPI_CONFIG(void)
{
EALLOW;
GpioCtrlRegs.GPADIR.bit.GPIO12  = 1;            // CS.
GpioCtrlRegs.GPADIR.bit.GPIO15  = 1;            // SCK
GpioCtrlRegs.GPADIR.bit.GPIO24  = 1;            // SI
GpioCtrlRegs.GPADIR.bit.GPIO26  = 1;            // LDAC
EDIS;

GpioDataRegs.GPASET.bit.GPIO12 = 1;  // CS high
GpioDataRegs.GPASET.bit.GPIO26 = 1;  // LDAC hi
}
```