Spring 2013

Quantization noise analysis of a closed-loop PWM controller that includes Σ-Δ modulation

Anupama Balakrishnan

Department:

QUANTIZATION NOISE ANALYSIS OF A CLOSED-LOOP PWM CONTROLLER THAT INCLUDES Σ-Δ MODULATION

by

ANUPAMA BALAKRISHNAN

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2013

Approved by

Jonathan W. Kimball, Advisor
Mehdi Ferdowsi
Keith Corzine

ABSTRACT

Σ-Δ modulation is a popular noise shaping technique which is used to move the quantization noise out of the frequency band of interest. Recently, a number of authors have applied this technique to a pulse width modulation (PWM) controller for switching power converters. However, previous analysis has not incorporated the effects of analog-to-digital converter (ADC) resolution or feedback control on the Σ-Δ modulator. In this work, quantization due to ADC resolution and PWM resolution are analyzed, considering the effects of noise-shaping and feedback. A number of simulations have been performed to explore the impact of various design choices on output noise. The study variables included the order of the Σ-Δ modulator, resolution of ADC, resolution of DPWM, the plant and the compensator. The theoretical model developed is used to generate the expected system Power Spectral Density (PSD) curves for each design choice and simulations techniques are used to validate the analysis. Experimental analysis has been performed on a digital voltage-mode control (VMC) synchronous buck converter and the output voltage PSD curves are generated using the welch method and compared with the theoretical and the simulation results. The experimental PSD curves for the 1^{st}- order modulator match the simulation and theoretical PSD curves. This suggests that the theoretical model is a useful approximation and similar methods can be used to analyze the contribution of the quantizers to the output noise of a closed-loop controller system.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Jonathan Kimball for his efforts in conducting the preliminary studies and developing the very high precision simulation models that have been used in this work. I would like to acknowledge the fact that without his efforts this work would not have been possible. I owe my sincere gratitude to Dr. Kimball for his continued support throughout the course of my graduate education. His dynamic personality, thorough knowledge, down-to-earth approach, and humble nature will continue to amaze and inspire me.

I would like to thank my committee members, Dr. Mehdi Ferdowsi for his excellent instruction in Advanced Power Electronics and Dr. Keith Corzine for his interest in my research topic.

I would especially like to thank my parents, Balakrishnan Ranganathan and Lalitha Balakrishnan for their love and support. I am grateful for the many sacrifices they have made for me. I would also like to thank my brother, Dr. Anand Balakrishnan, for his constant guidance, support and encouragement.

Lastly, I would like to acknowledge and appreciate the faculty, staff and friends of Missouri S&T for making this journey a comfortable and a memorable one.

TABLE OF CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

1. INTRODUCTION

1.1. INTRODUCTION TO DIGITAL CONTROL SYSTEMS

The simple, prototype friendly and low-cost analog circuits are soon giving way to their digital counterparts owing to the stringent demands of the modern day integrated circuit (IC) driven industries. The low-cost, low-weight, and low-power demands of the modern day electronic devices require systems which use fewer components but are able to implement advanced control or power management techniques. The ubiquitous nature of control systems implies that the above restrictions are applicable to control systems as well and has thus paved the way for the need of digital control systems.

Digital control systems use digital Integrated Circuits (ICs) such as Microcontrollers, DSPs (Digital Signal Processor), FPGAs (Field Programmable Gate Array) or other ASICs (Application Specific IC) which are used as to implement control algorithms to control a process or a plant. These ICs often include other peripheral units like the Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs) and digital Input-Output (I/O) pins in a single chip. The ADC receives the analog reference and input variables, quantizes the variables and sends them to the controller (IC) for further processing. The DAC converts the digital control signal to an analog signal and applies it to the analog plant. With advancement in manufacturing processes, these ICs can be mass-produced, which decreases their cost significantly. This ability to use low-cost ICs as controllers enables the digital controllers to enjoy a few of many advantages over their analog counterparts.

1.2. ADVANTAGES OF DIGITAL SYSTEMS

Digital systems address many issues faced by the analog systems. Analog systems are susceptible to environmental factors like time and temperature. Their performance degrades with age and thermal influence. Analog systems have low reliability due to high part count. Sometimes, the high part count may be due to redundancy in the system. They also suffer from low flexibility, loading issues and the analog controllers, in particular, are difficult to tune. On the other hand, digital systems are more reliable, use fewer components, and provide design flexibility.

Some of the other advantages of digital control systems are:

- Design Portability – The control algorithms are realized using software. This allows for ease of exporting and implementing the control algorithms on other systems.
- Programming Flexibility – Software realization also allows for ease of modifications in software with change in system specifications. This also means that different algorithms can be implemented in the same controller for different system specifications.
- Expandability – Safety, communication and monitoring circuits can be added on as per system requirements. The control algorithm can be expanded to include the function of these add-ons. Also, multiple control systems can be integrated together to suit specific design needs.
- Inherent Noise Rejection - Digital control involves using digitally-encoded information which is represented by a range of values. For example, a digital signal can be represented such that it can take any value

between 0 and 3.3V. Depending on the technology used (for example, TTL or CMOS) the noise margins for the logic '0' and logic '1' are established. This allows for noise rejection in digital systems.

- Accuracy – Error checking and correction can be implemented in digital systems to ensure accuracy. Also, resolution can be improved by increasing the number of bits.

The above advantages make digital systems popular in spite of a few limitations like complex and tedious system design, software errors, and information loss during reconstruction, delays introduced in the system and effect on system stability margin [1]. Recent advancements and continued research efforts have helped minimize these limitations and have made digital systems an attractive alternative to analog control systems. Some unique advantages of digital control technique/discrete time domain, in particular, have been listed in the following section.

1.3. IMPORTANCE OF DIGITAL CONTROL

Owing to its discrete time nature, sometimes, a digital control system is the only/easy solution to certain control problems. A few of such situations are listed below [2]:

- For naturally occurring/inherently discrete time plants (like those involving computing systems), continuous time modeling may not be possible. Two such naturally occurring plants, IBM Lotus Domino Server and Supply Chain Control, are discussed in [3]. Similar models in banking and the criminal justice system are discussed in [4]. For such plants, differential equation models describing the plant do not exist. One

possible way for modeling the system is by a process called identification which involves experimentation, data collection and curve fitting [3]. Let us assume that this process results in a plant model where the relationship between the states is similar to

$$x(k+1) = ax(k) + bu(k) \quad (1)$$

If the system identification process leads to $a < 0$ then for such plants, there is no suitable continuous time design.

- A Linear Time-invariant (LTI) system can be represented in the z-domain using pole-zero plot [5]. By using z-transformation, the difference equation of a LTI system can be represented in the form of system poles and system zeros. This representation is very useful in preliminary filter design. A pole can be used to amplify the signal and a zero can be used to attenuate the signal. This simple filter design technique is not possible in continuous time domain.
- Certain useful controller and control designs are possible only in the discrete domain. For example, the dead-beat controller which is designed to achieve zero-error at the sample points in a finite number of sampling periods and the control design with negative PID tuning parameters can be implemented only in discrete domain [2].
- Digital controllers can be easily realized in z-domain using recursive techniques by representing them as negative powers of z (z^{-1}).

System identification procedures which build mathematical models of dynamical systems from measured data determine discrete time models for almost all systems. Thus it is easier to understand most control strategies in discrete domain.

1.4. INTRODUCTION TO DIGITAL CONTROL OF POWER CONVERTERS

From the above discussions, it is clear that digital control systems offer numerous advantages over their analog counterparts. Continued research and development efforts in the field of design of digital controllers have made digital controllers viable in almost all applications. This transition has been acknowledged by the power supply industry, lately, which has led to extensive investigations and significant progress in the area of digital control of power converters.

Though analog controllers are more prevalent in the DC-DC switching power supply industry due to their simplicity, the recent trend of using low voltage (3.3 V – 1V) supplies for the ICs to improve the speed and performance [6], has increased the need for controllers which can implement complex algorithms to regulate output voltage while maintaining the speed and dynamic response under fast load changes.

Digital control systems offer design portability, programming flexibility, and provisions to implement sophisticated control algorithms without compromising on reliability and safety of the system. This enables efficient output voltage regulation and improves the dynamic performance of the power converters [6]. The ability to implement the efficient Pulse Width Modulation (PWM) technique [7], used in analog switching power supplies for regulating voltage, using Digital Pulse Width Modulator (DPWM) in the digital control system is an added advantage.

Even though digital controllers suffer from some drawbacks like limit cycle oscillations due to DPWM, complex digital design techniques, and high cost and power consumption at high frequencies, digital controllers for power converters have some unique advantages like efficiency optimization, auto-tuning and non-linear control [8], which cannot be obtained using analog controllers. The advantages derived from these unique capabilities have led to extensive investigations and solutions have been found for almost all the drawbacks listed above.

1.5. REVIEW OF RESEARCH EFFORTS

In late 1990's, analog technique for controller design was popular due to its simplicity. But design techniques proposed in [9], [10], and [11] incorporate the analog techniques in digital design, making digital design readily understood. In [9], the author compares the output of five digital control schemes with the output of an equivalent analog controller and concludes that bilinear transformation yields a similar output voltage to that of analog controller. Similar studies presented in [10] and performance comparison is made in terms of bandwidth and phase margin of control loop. Analog-to-digital redesign technique, an approach used in both [9] and [10] requires minimum design in z-domain but suffers from discretization effects. The direct digital design approach provides superior performance, as demonstrated in [10]. In [11] a direct digital design method is proposed which is similar to traditional analog compensation design techniques. These preliminary efforts have made digital controller design techniques for power converters more intuitive.

The research work in the field of digital control of power converters is mainly focused on two areas. One is improving the stability and output voltage of a converter by

eliminating limit cycles, and improving the effective resolution of the DPWM by using add-on modules so as to achieve high resolution of output voltage with reasonable clock frequency. Second is identifying and taking advantage of the unique capabilities of the digital controller to improve performance by developing new control strategies. The second part is out of the scope of this research work and thus literature review pertaining to this part has been omitted for present discussions.

Limit-cycle oscillations have been a subject of research from as early as 1961 when C. B. Bettin studied the "Effect of quantization on the stability of feedback control systems [12]." Later, the effect of finite resolution of the computation data word on the stability of second-order digital filters was identified in [13]. In [13] the author concludes that limit-cycles oscillations may occur due to this limitation and also calculates the frequency and amplitude of these oscillations. The presence of limit cycles in digitally controlled Pulse Width Modulated converters is mentioned in [14]. In this work, the author introduces the idea of the necessity of appropriate resolution selection of ADC and DPWM to reduce the possibility of multiple switching period limit cycles and non-periodic steady state behavior. This idea has been expanded in [15], in which the author derives three specific conditions to be satisfied in order to eliminate limit cycles in DPWM power converters. Elimination of limit cycles is essential because they cause output voltage oscillations whose frequency is lower than the switching frequency. Also, due to their unpredictable nature, the amplitude and frequency of such oscillations cannot be determined making it difficult to estimate the output noise and electro-magnetic interference (EMI) produced by the converter. To eliminate limit cycles it should be ensured that the resolution of DPWM is always greater than the resolution of ADC, the

integral term should be present in the control law and the loop gain which includes the effective gain of the ADC should not be zero. The dither technique used to increase the effective resolution of the DPWM is discussed. In [16], a graphical method is used to study the existence of a dc-solution to eliminate limit cycles. Also, a dynamic system model is derived using the generalized describing function method and the amplitude and offset-dependent gain model. The conclusion is that resolution of the two quantizers, the ADC and DPWM, are important to ensure no limit cycles.

Due to the discrete nature of a digital control signal, the accuracy of the output depends much upon the resolution (i.e. number of bits) of the DPWM and ADC modules. The relationship between the required resolutions of the output voltage, ΔV_{out}, and ADC resolution is given in [6] as

$$N_{ADC} = log_2\left(\frac{V_{in}}{\Delta Vout}\right) \tag{2}$$

Also, as per the previous discussion on limit cycles, the necessary condition to eliminate limit-cycle oscillations given in [15] is

$$N_{DPWM} > N_{ADC} \tag{3}$$

where, N_{DPWM} and N_{ADC} are the number of bits of the DPWM and ADC modules respectively. This implies that the resolution of DPWM should be at least 1 bit greater than the resolution of ADC. According to the author in [15], this 1 bit difference, which provides two levels of DPWM for one level of ADC, is satisfactory for most applications.

Therefore, the resolution of DPWM depends on the accuracy of the output desired and the resolution of ADC. This relationship is important because it determines the system clock frequency. The clock frequency is related to the resolution of DPWM as

$$f_{clk} = f_{sw} * 2^{N_{DPWM}} \quad (4)$$

where, f_{clk} is the system clock frequency and f_{sw} is the converter switching frequency. So, for a converter with 11 bit ADC, N_{DPWM} is 12 bits and the system clock frequency is 4096 MHz or 4 GHz. This clock frequency is unreasonable and cannot be implemented efficiently. Because of this limitation on practically achievable clock frequency, it is desired that high resolution of DPWM be achieved with reasonable clock frequency.

A number of authors have proposed many different methods to increase the effective resolution of the DPWM which allows high resolution operation of DPWM at reasonable clock frequency. A simple method is proposed in [17], using counter-comparator scheme for a synchronous Buck converter. The high DPWM resolution is achieved by fast-clocked counter comparator which uses a clock frequency of $2^N * f_{sw}$ where N is the number of bits in the command word. Though this method is simple and can be easily implemented, it requires very high clocks for high values of N. This method has increased power consumption and die area at high switching frequencies and multi-phase applications due to the need of independent counter circuits and other fast logic circuits [18]. Another method called the tapped delay line technique is proposed in [19]. This method uses a clock which operates at switching frequency, a delay line and a multiplexer. The control pulse, set by the clock, sets the PWM high and propagates down

the delay line where it is used to set the PWM output low when the pulse reaches the output selected by the multiplexer. This scheme has a significant improvement in terms of power consumption over the previous fast-clocked counter comparator method but still suffers from unsuitability for multi-phase applications. A ring-oscillator-MUX implementation of the DPWM module is proposed in [18]. A MUX, which is controlled by the commanded duty cycle, is used to set the taps which control the rising and falling of a square wave propagating through the 128 stage differential ring oscillator. The rising edge of the square wave sets the PWM high and when the rising edge reaches a specified tap value the PWM is set low. The method has low power consumption, less area and has a symmetric structure which enables multi-phase application.

The methods discussed in [17]-[19] for improving the effective resolution of DPWM results in high power dissipation or large area [15]. The digital dither method introduced in [15] involves the addition of high frequency periodic or random signals to a quantized signal which is filtered to produce averaged DC levels with increased resolution [15]. The duty cycle is varied by an LSB over a few switching cycles and when the averaging action is implemented on the duty by the converter's LC filter, the average duty cycle has a value between two adjacent quantized duty cycle levels [15]. The effective resolution can be increased by using longer dither patterns. Though this technique uses limited die area to successfully eliminate limit cycle oscillations by increasing the effective resolution of DPWM, it suffers from dither ripple [15]. Dither ripple is the AC ripple at the output of LC filter which is caused due to dithering of duty cycle and is superimposed on the ripple from converter switching action [15]. With the use of longer dither patterns, the AC ripple at the output increases it contains low

frequency components which cannot be filtered by the low pass LC filter [15]. This limits the number of dither bits that can be used and consequently the resolution of DPWM.

The underlying issue with the deviation of the DPWM module has been identified and corrected by proposing a novel control algorithm by the author in [20]. The author suggests that output error in digitally controlled systems arises due to the propagation of the DPWM deviation into the main circuit and proposes the concept of correcting the quantization error within the DPWM module. For this, a proposed add-on module is inserted before the PWM module in the system which acts as an accumulator for the quantization error. The quantization error is deposited and accumulated and it is compensated in the next cycle according to the accumulated error. This technique is similar to the averaging by dithering technique used in [15] the only difference being the use of look up table with pre-programmed patterns in [15] and the use of first order ∑-Δ generator in [20] as identified in [8].

The operation of the Σ-Δ modulator is based on the noise-shaping concept discussed in [21] and [22]. The modulator is designed with a noise shaping transfer function (NTF) of the form

$$NTF = (1 - z^{-1})^n \tag{5}$$

where *n* is the order of the modulator and is equal to 1 for first order modulator, 2 for second order modulator and so on. The NTF defines the output spectral characteristics of the noise [21].

The single-bit first-order Σ-Δ modulator suffers from slow convergence and a low-frequency periodic behavior of the DPWM output which introduces low-frequency noise [25]. The application of a multi-bit first order Σ-Δ modulator in high-frequency low-power dc-dc conversion is discussed in [23]. The multi-bit structure provides fast convergence towards the high-resolution value thus reducing the averaging period. It also forces the steady state error (which is essentially the difference between high resolution reference value from ADC and low-resolution command value of DPWM) to be zero (due to the presence of pole at origin) thus effectively increasing the internal resolution of the DPWM. The sigma delta pre-processing approach [24] is also discussed in [24], wherein a comparison is made between a quantizer and a sigma-delta modulator of same resolution. The Power Spectral Density (PSD) obtained, clearly differentiates the noise-shaping characteristic of the sigma-delta modulator from the white noise characteristic of the quantizer. Also, the effective resolution or Equivalent Number of Bits (ENOB) is calculated by using the Signal-to-Noise Ratio (SNR) approach.

The single-bit first-order Σ-Δ modulator suffers from slow convergence and a low-frequency periodic behavior of the DPWM output which introduces low-frequency noise [25]. This behavior, known as tones, is demonstrated in [25]. Though multi-bit structures address the convergence issues, the low-frequency tones still exist. In [26], the advantages of a one bit second order Σ-Δ modulator over the first order modulator, namely the suppression of low frequency tones and faster convergence, have been recognized and a single-bit analog second-order Σ-Δ modulator has been implemented effectively to improve output regulation. The performance of first order and second order

Σ-Δ modulators have also been compared in [25] and experimental results which validate the observations are provided.

The recognition of the advantages of the Σ-Δ modulators and their simple structural implementation has led to their wide spread application in the field of data conversion at relatively low frequency ranges [27]. Another application of delta modulation techniques for resonant link inverters has been studied in [28]. It has been shown that Σ-Δ modulated inverters switch faster and have better spectral response than the conventional PWM inverters. Also, to reduce acoustic noise in motor drives, Σ-Δ modulator based current regulated delta modulation technique is used [27]. Σ-Δ modulators are also used in multilevel inverters [29]. Σ-Δ modulators can be used in the design of switching power supplies [30]. The noise shaping characteristic of the modulator is exploited to provide a better noise performance controller. Σ-Δ modulators can also be used to control the EMI from switch-mode power supplies as demonstrated in [27].

This review has led to a greater understanding of the working of digital control architecture and the important challenges faced due to the discrete time nature and consequently the quantization effects present in digital systems. It has also redirected the course of this research towards further work in the area of Σ-Δ modulators by stressing on the simplicity and ease of implementation of Σ-Δ modulator structures and their applications in numerous fields. Numerous other applications may exist which have not been covered in this review.

1.6. MOTIVATION FOR CURRENT WORK

Current studies have majorly focused on the open-loop spectral characteristics of the Σ-Δ modulator and the performance of a closed-loop system in which the modulator is being used. But the spectral characteristics of the output of a closed-loop digitally controlled converter have not been analyzed. This analysis is important because it will help us answer a few questions related to the effect of feedback on modulation, the necessity of Σ-Δ modulators in closed-loop systems, output quality improvement in a closed-loop system due to the addition of Σ-Δ modulators and the performance comparisons of the first-order and second-order Σ-Δ modulators in a closed-loop system.

It is well known that Σ-Δ modulators improve open-loop performance but the presence of the compensator in a closed-loop system leads to the necessity of further analysis. The question is, is Σ-Δ modulator required to improve the output quality in a closed-loop system or is the compensator sufficient? Also, the effect of band width of the system on the performance of the Σ-Δ modulator and consequently the output noise should be analyzed and the contribution of individual quantizers to output noise PSD should be determined.

Preliminary simulation studies have been performed to explore the impact of various design choices on output noise. Some of the variables in the study were the resolutions of ADC and DPWM, the plant, the compensator, and the order of Σ-Δ modulator (N). These studies have led to a number of conclusions which have been discussed in Section 4. The goal of this thesis is to validate the analysis and simulation studies with experimental results.

1.7. THESIS ORGANIZATION

Section 1 introduced the concept of digital control and discussed the advantages and importance of digital control. The application of the concept of digital control to power converters was introduced. The research trends involved were discussed which directed the course of this work towards implementation of Σ-Δ modulator structures.

In Section 2 the application of digital control to power converters is discussed. The plant is modeled and discretized. Using the transfer function model of the plant and direct control design technique the digital compensator is designed. The implementation of analog/digital interface systems is discussed. The resolutions of these interface systems and their effect on output voltage accuracy and converter stability is discussed. Finally, the implementation of Σ-Δ modulator structures in power converter applications and its advantages are discussed.

In Section 3 the quantization noise introduced by the ADC and DPWM blocks are analyzed and the PSD curve of the output voltage of the converter is generated by taking noise shaping characteristic of the Σ-Δ modulator structures and the feedback control into account.

In Section 4 simulation analyses are conducted to validate the theoretical analysis and in Section 5 experimental analyses are conducted to validate the simulation and analysis.

2. DIGITAL CONTROL OF POWER CONVERTERS

2.1. INTRODUCTION

In this section, the theoretical concepts involved in the design of digital control systems for power converters are reviewed. The target specifications are used to design the plant and a behavioral study of the plant is conducted. A mathematical model of the plant is developed. The state-space averaging technique is used to model its steady-state behavior and the small-signal model is developed to model its behavior for perturbed input and control signals. From the small-signal model a transfer function is determined to be used as the model of the plant for further analysis in the present work. This plant model is discretized to enable direct control design. The digital interface systems of a digital control system, ADC and DPWM, are discussed in detail. The use of Σ-Δ modulators as a pre-processor block and its advantages are discussed.

2.2. BEHAVIORAL STUDY OF THE PLANT

The plant, a synchronous buck converter, is shown in Figure 2.1. R_{in} and R_{out} represent the ESR of the input and output capacitors, C_{in} and C_{out}, respectively; R_L is the DCR of the inductor L; Q1 and Q2 are two N-Channel power MOSFETs with their $RDS_{(on)}$ specified as R_{on}; R_{load} is the load resistance. V_{in} is the input voltage of the converter. The converter is designed to step-down a 5 V input voltage to 1 V with 20 mV voltage ripple and 20% current ripple. The switching frequency is chosen to be 500 kHz. The specifications and component values along with the internal parasitic resistances of each component of the converter are summarized in Table 2.1.

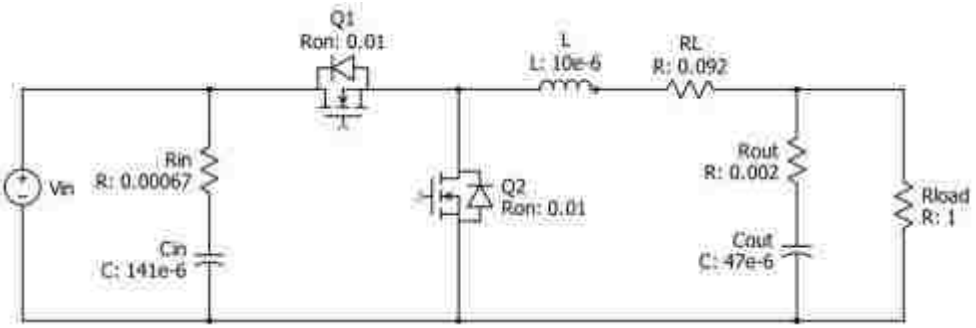


Figure 2.1 Synchronous Buck Converter

The ratio of output voltage to input voltage gives the duty ratio of a buck converter. For the specified converter, the duty ratio is 20% (D = 0.2). As the converter spends 80% (D' = 0.8) of its time in the OFF-state, synchronous rectification is chosen instead of the conventional diode to reduce losses. For example, for a load current of 1A the conduction loss in a conventional diode with a forward voltage drop of about 0.4 V at peak current of 1 A is about 0.32 W whereas for a power MOSFET with $RDS_{(on)}$ of 0.01 Ω, the conduction loss is about 0.008 W . Though synchronous rectification is a trade-off between efficiency and cost, the usage of only two switches with low $RDS_{(on)}$ prioritizes efficiency over cost in this particular case.

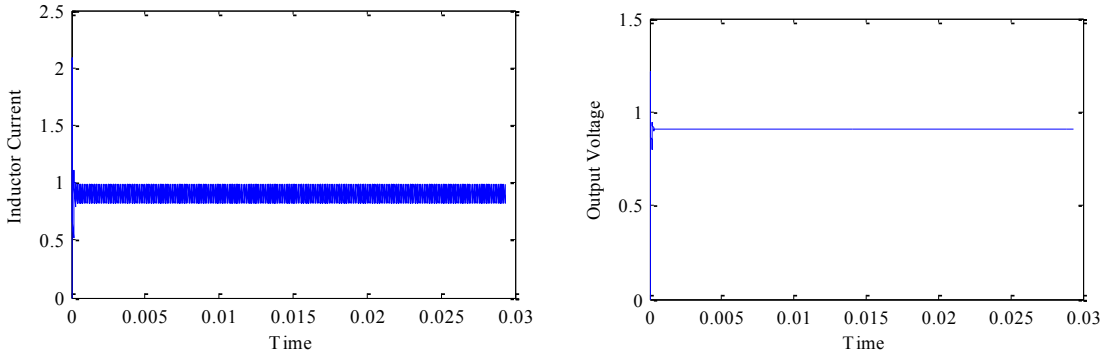
A preliminary simulation study was conducted using PLECS[1]. PLECS facilitates circuit level simulation. The advantage is that the parasitic resistances of the components can be included in the model to observe their effect. The PLECS circuit is same as shown in Figure 2.1 but with a pulse generator added to emulate the PWM signals for the two switches. The first switch, Q1, receives a pulse signal with a frequency of 500 kHz, amplitude of 1 and a duty of 0.2.

[1] PLECS is a registered trademark of Plexim Gmbh.

Table 2.1 Table Summarizing the Specifications and Component Values

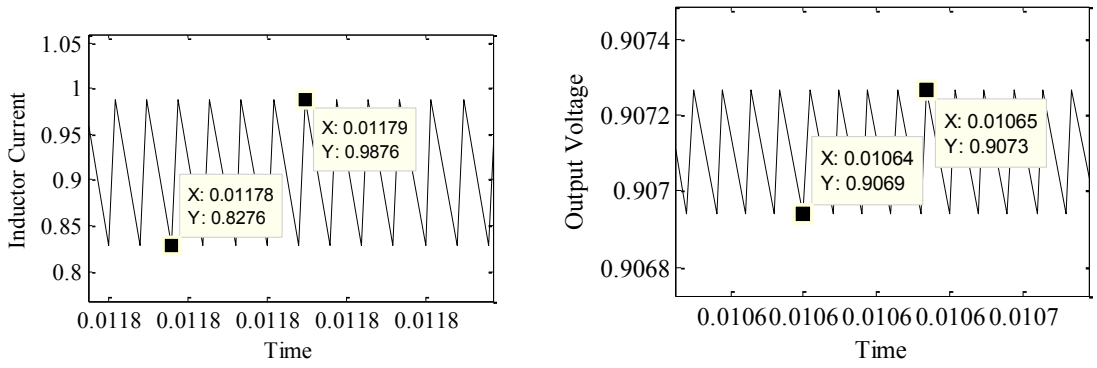
Specifications	
Input Voltage	5 V
Output Voltage Reference	1 V
Switching Frequency	500 kHz
Target Voltage Ripple	20 mV
Target Current Ripple	2 A (20%)
Component Selection	
Component	**Value**
Input Capacitor (Cin)	141 µF
Input Capacitor ESR (Rin)	0.00067 Ω
Switches	SI7160 DP, N CH MOSFET
RDSon	0.01 Ω
Inductor	10 µH
Inductor DCR	0.092 Ω
Output Capacitor (Cout)	47 µF
Output Capacitor ESR (Rout)	0.002 Ω
Load Resistance	1 Ω

The second switch, Q2, receives a signal with the same frequency and amplitude as Q1 but with a duty of 0.8. The inductor current (I_L) and output voltage (V_{out}) are observed. The I_L and V_{out} are plotted in Figures 2.2 (a) and (b). Figure 2.3 (a) and (b) show the I_L and V_{out} ripple along with their maximum and minimum magnitudes as pointed by the data tip. For a load resistance of 1 Ω and output voltage of 1 V, the load current, which is same as the inductor current, should be 1 A. The inductor current observed, depicted in Figure 2.3 (a), is about 0.9 A with ripple maximum of 0.9876 A and ripple minimum of 0.8276 A. The peak-to-peak value of ripple is 0.16 A. For the observed mean value of 0.907 A, the percentage of current ripple is 17.6 %. Similarly, the mean value of output voltage observed, depicted in Figure 2.3 (b), is 0.9 V with ripple maximum of 0.9073 V and ripple minimum of 0.9069 V. The peak-to-peak value of the ripple is 0.0004 V.



(a) (b)

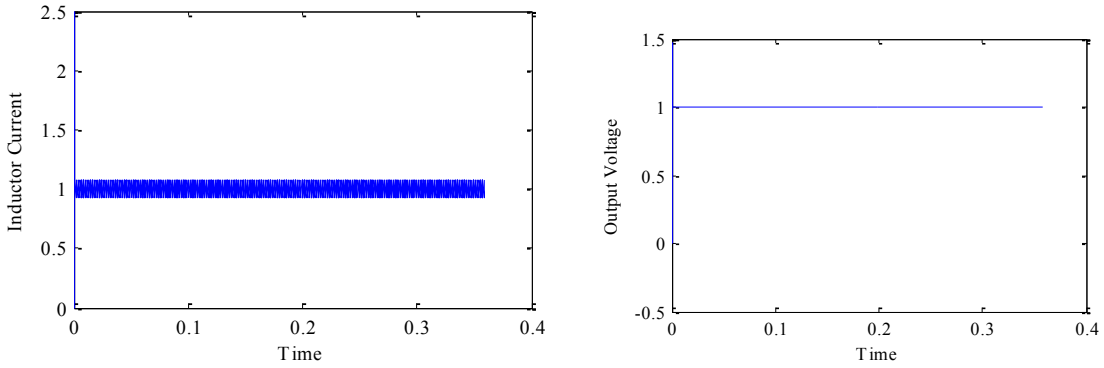
Figure 2.2 Inductor Current and Output Voltage Waveforms for the Specified Synchronous Buck Converter; (a) Inductor current; (b) Output Voltage



(a) (b)

Figure 2.3 Inductor Current and Output Voltage Ripple for (a) Inductor current ripple; (b) Output Voltage ripple

These observations indicate that the ripple values are within the specified limits and thus validate the inductor and output capacitor selection. However, due to voltage drop across the parasitic resistances a 10 % reduction in the desired output voltage can be observed. The results with the parasitic resistances set to zero, shown in Figure 2.4, validate this claim. The observed mean value of inductor current is 1A, and mean value of output voltage is 0.999 V.

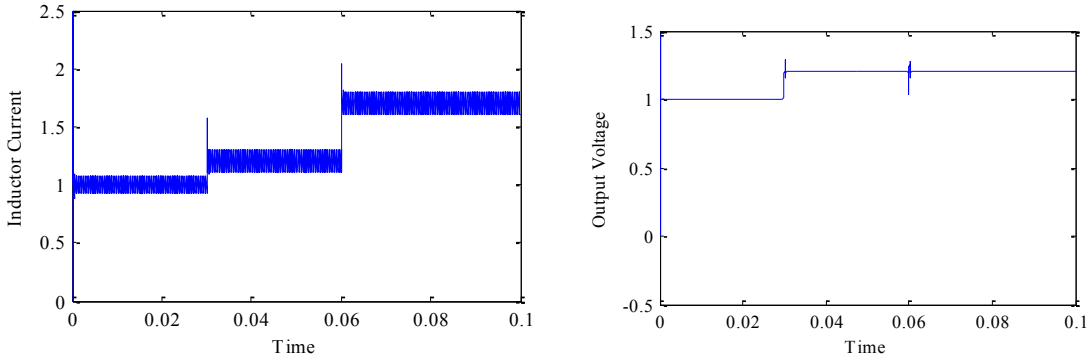


(a) (b)

Figure 2.4 Inductor Current and Output Voltage Waveforms with Ideal Components; (a) Inductor Current; (b) Output Voltage

Another study was conducted to observe the effect of variations in input voltage and load on the output of the synchronous buck converter. To emulate the variation in input voltage, a step change in input voltage from 5 V to 6 V was introduced using a step source at 30 ms. To emulate the load variations, the load was first modeled using a Norton resistance of 1 Ω (equal to Vout divided by Imin) and a Norton current source of 0.5 A (equal to Imax minus Imin, where Imax = 1.5 A) and then the step change was applied to the current source at 60 ms. The results are shown in Figure 2.5.

Few conclusions can be drawn from the preliminary simulation studies of the plant. First, the components chosen are appropriate for the desired specifications. Second, due to the inherent presence of parasitic resistance in all the components, feedback type control is required to regulate the output voltage so that it equals the desired or reference voltage. Third, regulation is necessary during load or input voltage variations so as to minimize the transients across the load connected to the converter.



(a) (b)

Figure 2.5 Inductor Current and Output Voltage Waveforms for Step Variations in Input Voltage and Load; (a) Inductor Current; (b) Output Voltage

2.3. MODELING AND DISCRETIZATION OF CONTINUOUS TIME PLANT

From the preceding discussions it is clear that power converters require regulation/feedback to maintain a constant output voltage regardless of variations in the load or input voltage. This regulation is accomplished by employing a closed-loop system with a controller- consisting of a sensor to sense the output voltage, a compensator to compensate for the changes and an appropriate device which can generate the control signal in accordance with the compensator. In order to design a control loop which is stable and will help the converter meet the specifications, it is necessary to understand the behavior of the converter and develop a mathematical model which can generalize its behavior.

An example of modeling is discussed in this section. The plant, synchronous buck converter, is modeled in this section to facilitate the design of the controller in the next section. In the first step, some known approximations are used to develop a DC or average-value model of the converter to understand its dominant or steady-state behavior.

Later, an AC small signal model is developed to understand the converter's behavior during perturbations.

2.3.1. Average-Value Model of Synchronous Buck Converter. The determination of long-term or steady-state behavior of a power converter is essential because it defines the state of equilibrium or stability of the converter. The steady-state, which is the dc component of currents and voltages in the converter circuit, is determined by averaging the waveforms. In order to simplify the steady-state analysis, linear-ripple approximation, the principle of inductor-volt balance and the principle of capacitor charge balance are used [34]. Also, the loads associated with the converter are designed based on its steady-state value and any unprecedented change in the steady-state value may damage the load. Thus, the knowledge of steady-state behavior is essential for appropriate filter and control circuit design.

The steady-state behavior of a power converter can be studied by deriving the converter's average-value model. One such average-value model formulation method is the State Space Averaging (SSA). The SSA method helps simplify the analysis by considering the average characteristics of the converter circuit instead of the cycle-by-cycle switching. The analysis is limited to within the Nyquist frequency (which is one half of switching frequency) thereby eliminating the switching process from the analysis. This further simplifies the steady-state analysis and helps to quickly estimate the DC operating value (steady-state) and the stability (Loop gain) of the converter.

For a linear time invariant system, the SSA uses the state space equations in the matrix form, to model power converters.

The general form of the state space equations are:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6)$$

$$y(t) = Cx(t) + Du(t) \quad (7)$$

where,$x(t)$ is called the state of the system and is defined as the set of variables required to completely understand the system. These internal sate variables are the smallest possible subset of system variables (generally equal to the order of the system and specifically, though not always, equal to the number of energy storage elements in a power converter) which can represent the entire state of the system at any given time. These state variables should be linearly independent so as to be solvable. $u(t)$ is the input vector matrix which contains the inputs of the converter and $y(t)$ is the output vector matrix. A, B, C and D are the constant/coefficient matrices where A is called the state matrix, B is called the input matrix, C is called the output matrix and D is the called the feed-forward matrix.

The determination of the state space equations for a power converter is a four step process. First, the switching states in the converter are identified and the converter is represented in those states by replacing the switches with their On-State models in the circuit diagrams. Second, for each state, the circuits are analyzed by using appropriate circuit analysis techniques like the mesh analysis and nodal analysis and the element equations are formulated. Third, the input vector, state vector and output vectors are identified and the equations are represented in the state space matrix form. Finally, the duty ratio of the converter is determined and the weighted average of the coefficient

matrices, for the time spent in each state is determined. The matrices can then be solved to obtain the steady state solution using MATLAB[2].

The derivation of SSA for the synchronous Buck converter is done in the following steps:

- Step 1: Switching Mode Representation:

The synchronous Buck converter has two modes of operation - Mode 1, when Q1 is ON and Q2 is OFF, Mode 2 when Q1 is OFF and Q2 in ON. The converter is in Mode 1 for 20% of the switching time and in Mode 2 for 80% of the switching time. The circuit models for Mode 1 and Mode 2 are shown in Figure 2.6. In its ON state, a MOSFET behaves like a resistor between the drain and the source terminals represented as $RDS_{(on)}$ in the data sheet. During Mode 1, switch Q1 is ON and is represented as Ron which is equal to the $RDS_{(on)}$ of the MOSFET and in Mode 2, switch Q2 is ON and is represented as Ron. The currents flowing in the circuit for each mode are as shown. Also, the output voltage ($V_{out}(t)$) and the output capacitor voltage ($v_{Cout}(t)$) are shown.

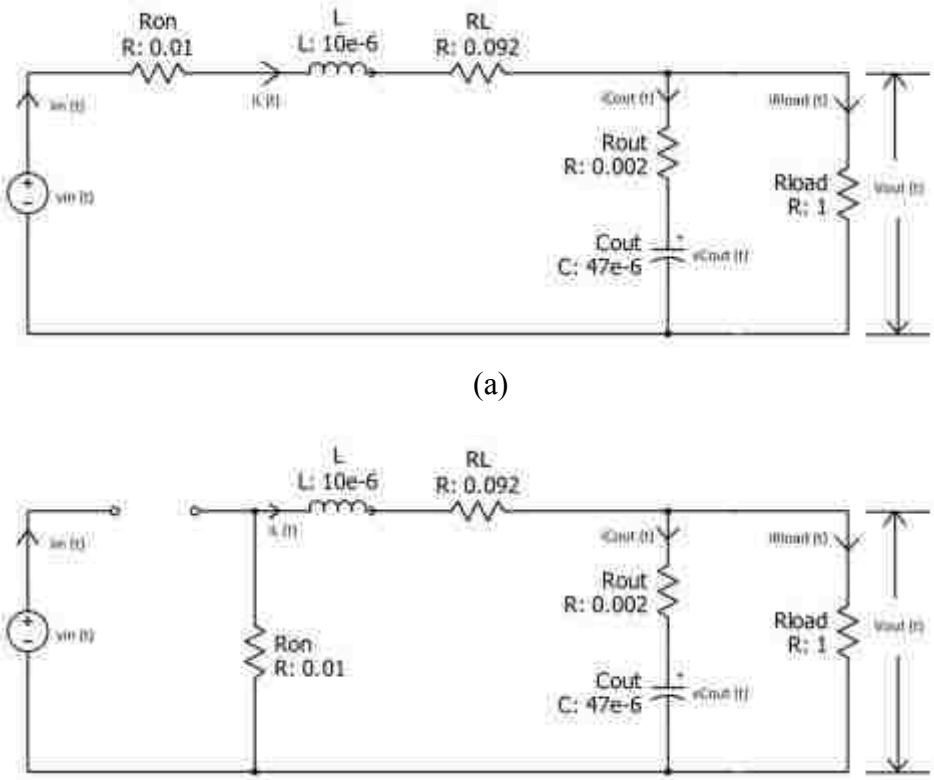
- Step 2: Formulation of circuit equations:

Using the linear-ripple approximation and the basic circuit analysis techniques the element equations can be formulated for each mode of operation. The output voltage waveform depicted in Figure 2.4 (b) consists of a DC component (V_{out}) plus a small AC ripple component ($v_{ripple}(t)$).

The output voltage $V_{out}(t)$ can thus be expressed as

$$V_{out}(t) = V_{out} + v_{ripple}(t) \tag{8}$$

[2] MATLAB is the trademark of MathWorks Inc.



(a)

(b)

Figure 2.6 Circuit Representation of the Two Operating Modes; (a) Mode 1 with switch Q1 ON; (b) Mode 2 with switch Q2 ON

In a well-designed converter, the output voltage ripple is very small (0.4 mV here when compared to 1V output) since the objective is to produce a DC output voltage. Hence, the small ripple component can be safely neglected and small-ripple approximation can be applied to the output voltage to obtain the output voltage equation as:

$$V_{out}(t) = V_{out} \tag{9}$$

Similar approximation can be applied to the inductor current $i_L(t)$. For cost-efficient converter design, the peak current flowing through the components in the circuit

should be minimized. Higher peak currents require devices with higher peak current rating. This increases the cost and size of the devices, especially the semiconductor switches and the inductor. As the peak current is the sum of the average current (DC component) and one half of peak-to-peak ripple, the ripple component should be minimized to limit the peak current. Typically, in a well-designed converter, the inductor ripple is limited to 10% to 30% of full-load average value and thus the linear-ripple approximation can be applied to the inductor current in this converter. The voltage and current linear-ripple approximations greatly simplify the analysis of the converter waveforms for determining the steady state behavior.

During Mode 1:

Using linear-ripple approximation and Kirchhoff's Voltage Law (KVL), the voltage equation around the loop, for Figure 2.6(a), can be written as:

$$-V_{in} + (i_l(t) * R_{on}) + \left(L * \frac{di_L(t)}{dt}\right) + (i_L * R_L) + V_{out} = 0 \quad (10)$$

$$\left(L * \frac{di_L(t)}{dt}\right) = V_{in} - (i_L(t) * R_{on}) - (i_L(t) * R_L) - V_{out} \quad (11)$$

Using Kirchhoff's Current Law (KCL), the current equations at the nodes for Figure 2.6 (a) can be written as:

$$i_L(t) = i_{Cout}(t) + i_{Rload}(t) \quad (12)$$

Applying current linear-ripple approximation to the inductor current, we obtain the DC currents in the converter as:

$$I_L = i_{Cout}(t) + I_{Rload} \tag{13}$$

The DC component of load current is equal to:

$$I_{Rload} = \frac{V_{out}}{Rload} \tag{14}$$

Also, the output voltage is the sum of voltage across the capacitor C_{out} and the resistor R_{out}. By applying voltage and current division rules, the output voltage can be expressed as:

$$V_{out} = \left(V_{Cout} * \frac{Rload}{Rload+Rout}\right) + \left(I_L * \frac{Rout*Rload}{Rout+Rload}\right) \tag{15}$$

Substituting (14) and (15) in 13 and solving for $i_{Cout}(t)$, the element equation can be obtained as:

$$C_{out} * \frac{dvCout\,(t)}{dt} = I_L * \frac{Rload}{Rload+Rout} - \frac{VCout}{Rload+Rout} \tag{16}$$

Similarly, applying the inductor ripple approximation to (11) and substituting the value of V_{out}, the element equation for inductor current can be obtained as:

$$L * \frac{diL(t)}{dt} = V_{in} - \left(\left(R_{on} + R_L + \frac{Rout*Rload}{Rout+Rload}\right) * I_L\right) - \left(\frac{Rload}{Rload+Rout} * V_{Cout}\right) \tag{17}$$

During Mode 2:

Applying the above analysis for the circuit in Figure 2.6 (b), the element equations during mode 2 of operation can be written as:

$$L * \frac{diL(t)}{dt} = -\left(\left(R_{on} + R_L + \frac{Rout*Rload}{Rout+Rload}\right) * I_L\right) - \left(\frac{Rload}{Rload+Rout} * V_{Cout}\right) \quad (18)$$

$$C_{out} * \frac{dvCout\,(t)}{dt} = I_L * \frac{Rload}{Rload+Rout} - \frac{VCout}{Rload+Rout} \quad (19)$$

- Step 3: Representing in state space matrix form:

For the synchronous buck converter, the state matrix consists of the states of the two energy storage elements inductor and output capacitor as the inductor current and capacitor voltage represent the whole state of the converter. Thus, the state matrix is equal to

$$x(t) = \begin{bmatrix} i_L \\ v_{Cout} \end{bmatrix} \quad (20)$$

The input to the system is the input voltage Vin and the output observed is the output voltage Vout. Thus the input matrix is:

$$u(t) = [Vin\,] \quad (21)$$

and the output matrix is:

$$y(t) = [Vout\,] \quad (22)$$

Using the above matrices and the element equations, the state space equations can be formulated as:

For Mode 1:

$$\begin{bmatrix} L & 0 \\ 0 & Cout \end{bmatrix} \begin{bmatrix} \frac{diL(t)}{dt} \\ \frac{dvCout(t)}{dt} \end{bmatrix} = \begin{bmatrix} -(Ron + RL + \frac{Rload*Rout}{Rload+Rout} & \frac{-Rload}{Rload+Rout} \\ \frac{Rload}{Rload+Rout} & \frac{-1}{Rload+Rout} \end{bmatrix} \begin{bmatrix} IL \\ Vc \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [Vin] \tag{23}$$

$$Vout = [\frac{Rload*Rout}{Rload+Rout} \quad \frac{Rload}{Rload+Rout}] \begin{pmatrix} IL \\ VCout \end{pmatrix} + [0]\,[Vin] \tag{24}$$

For Mode 2:

$$\begin{bmatrix} L & 0 \\ 0 & Cout \end{bmatrix} \begin{bmatrix} \frac{diL(t)}{dt} \\ \frac{dvCout(t)}{dt} \end{bmatrix} = \begin{bmatrix} -(Ron + RL + \frac{Rload*Rout}{Rload+Rout} & \frac{-Rload}{Rload+Rout} \\ \frac{Rload}{Rload+Rout} & \frac{-1}{Rload+Rout} \end{bmatrix} \begin{bmatrix} IL \\ Vc \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [Vin] \tag{25}$$

$$Vout = [\frac{Rload*Rout}{Rload+Rout} \quad \frac{Rload}{Rload+Rout}] \begin{pmatrix} IL \\ VCout \end{pmatrix} + [0]\,[Vin] \tag{26}$$

- Step 4: Determining weighted average of coefficient matrices:

From the above state space equations, the coefficient matrices for each mode of operation can be identified as:

For Mode 1:

$$A1 = \begin{bmatrix} -(Ron + RL + \frac{Rload * Rout}{Rload + Rout} & \frac{-Rload}{Rload + Rout} \\ \frac{Rload}{Rload + Rout} & \frac{-1}{Rload + Rout} \end{bmatrix}; \; B1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix};$$

$$C1 = \begin{bmatrix} \frac{Rload*Rout}{Rload+Rout} & \frac{Rload}{Rload+Rout} \end{bmatrix} \; and \; D1 = [0]$$

For Mode 2:

$$A2 = \begin{bmatrix} -(Ron + RL + \frac{Rload * Rout}{Rload + Rout} & \frac{-Rload}{Rload + Rout} \\ \frac{Rload}{Rload + Rout} & \frac{-1}{Rload + Rout} \end{bmatrix}; \quad B2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

$$C2 = \begin{bmatrix} \frac{Rload * Rout}{Rload + Rout} & \frac{Rload}{Rload + Rout} \end{bmatrix} \; and \; D2 = [0]$$

Let D1 be the duty ratio of the converter during Mode 1. In Continuous Conduction Mode (CCM), the duty ratio during Mode 2, D2, is D2 = 1- D1. Using D1 and D2, the weighted average coefficient matrices can be found. The average coefficient matrices for CCM are:

$$\text{A} = \begin{bmatrix} -(Ron + RL + \frac{Rload*Rout}{Rload+Rout} & \frac{-Rload}{Rload+Rout} \\ \frac{Rload}{Rload+Rout} & \frac{-1}{Rload+Rout} \end{bmatrix}$$

$$B = \begin{bmatrix} D1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{Rload*Rout}{Rload+Rout} & \frac{Rload}{Rload+Rout} \end{bmatrix}$$

$$D = [0]$$

Substituting the above matrices in equations (6, 7) we can obtain the average state equation and output equation for the converter operating in CCM. The equations are:

$$\begin{bmatrix} L & 0 \\ 0 & Cout \end{bmatrix} \begin{bmatrix} \frac{diL}{dt} \\ \frac{dVCout}{dt} \end{bmatrix} = \begin{bmatrix} -(Ron + RL + \frac{Rload*Rout}{Rload+Rout}) & \frac{-Rload}{Rload+Rout} \\ \frac{Rload}{Rload+Rout} & \frac{-1}{Rload+Rout} \end{bmatrix} \begin{bmatrix} IL \\ VCout \end{bmatrix} + \begin{bmatrix} D1 \\ 0 \end{bmatrix} [Vin] \tag{27}$$

$$Vout = [\frac{Rload*Rout}{Rload+Rout} \quad \frac{Rload}{Rload+Rout}] \begin{pmatrix} IL \\ VCout \end{pmatrix} + [0]\,[Vin] \tag{28}$$

As stated earlier, the principle of inductor volt-second balance and principle of capacitor charge balance [35] can be applied to the above equations to simplify the steady state analysis. From the principles, the following relations can be defined for steady-state operation:

$$L\frac{diL}{dt} = 0\,, \quad Cout\frac{dVCout}{dt} = 0 \tag{29}$$

Therefore, equation (28) can be reduced to

$$0 = \begin{bmatrix} -(Ron + RL + \frac{Rload*Rout}{Rload+Rout}) & \frac{-Rload}{Rload+Rout} \\ \frac{Rload}{Rload+Rout} & \frac{-1}{Rload+Rout} \end{bmatrix} \begin{bmatrix} IL \\ Vc \end{bmatrix} + \begin{bmatrix} D1 \\ 0 \end{bmatrix} [Vin] \quad (30)$$

Using equations (29) and (30), the relationship between the input voltage and output voltage can be found as:

$$\frac{Vout}{Vin} = D1 * \frac{Rload}{Rload+RL+Ron} \quad (31)$$

The voltage relationship is multiplied by a correction factor due to the inclusion of parasitic resistances of the inductor and the MOSFET.

A better way to represent the average value model is by generating averaged circuits that correspond to the model equations [35]. These circuit models can then be simulated using circuit simulators to study the behavior of the system. Average circuit model makes the simulation faster and a time behavior similar to the actual circuit model can be obtained with less complexity owing to the absence of switching ripples. The average circuit model for the synchronous buck converter can be derived using the element equations (16) and (17). A detailed description of the procedure is given in [34]. The final circuit is shown in Figure 2.7.

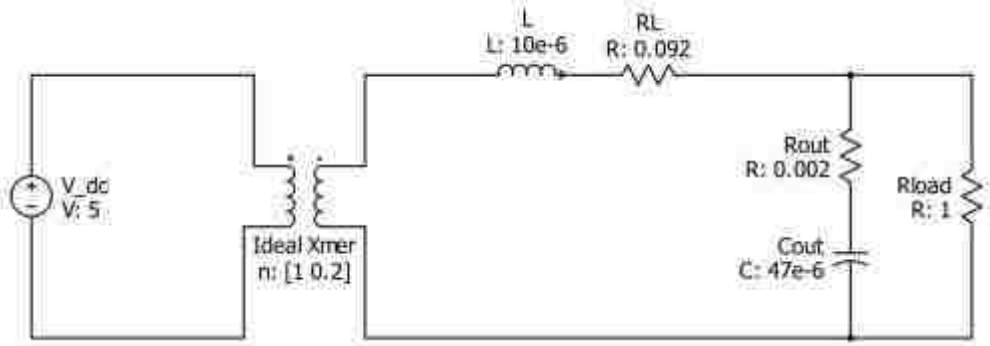
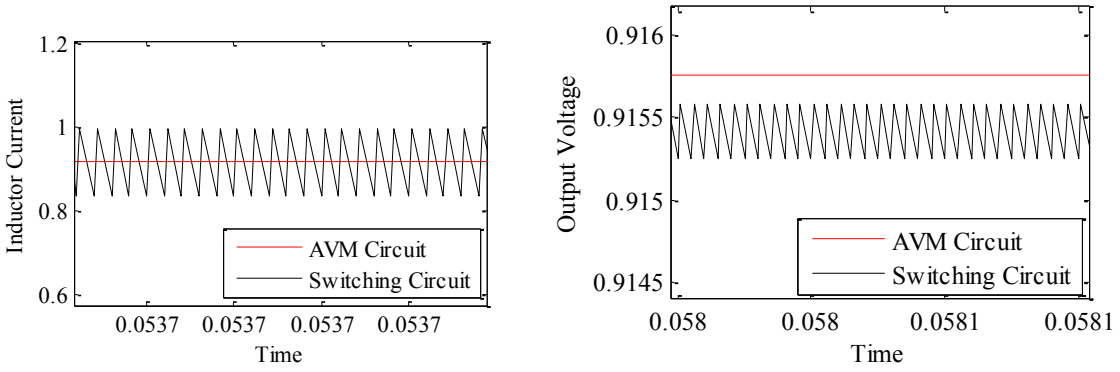


Figure 2.7 Average Value Model of Buck Converter

In the average model, the switches are replaced with an ideal transformer whose turns ratio is equal to the 1/D1. The results are depicted in Figure 2.8. In the figure, a comparison is made between the average model (AVM) circuit and switching circuit. It can be observed that both the circuits exhibit same time behavior. The results also validate the average model developed.



(a) (b)

Figure 2.8 A comparison of Average Value Model (AVM) Circuit to Switching Circuit; (a) For Inductor Current; (b) For Output Voltage

2.3.2. Small-Signal Model of Synchronous Buck Converter. The small-signal model of the power converter is usually derived from the average model, to simplify the process, rather than the converter itself because small deviations are not possible in the

switching function which is the only difference between the converter and its average model. The process involves replacing the state variables and the control parameters in the average equations with their linearized quantities (sum of the DC term and time varying term), neglecting the product of the perturbations and collecting and equating the like terms on both sides to find the small signal element equations.

For the synchronous buck converter, using the detailed derivation in [34], the small-signal model can be derived. The element equations obtained from the average state for the synchronous buck converter given by equation (28) are:

$$L\,\frac{dI_L}{dt} = -\left(Ron + RL + \frac{Rload*Rout}{Rload+Rout}\right) * I_L + \frac{-Rload}{Rload+Rout} * V_{Cout} + D1 * V_{in} \quad (32)$$

$$Cout\,\frac{dV_{Cout}}{dt} = \left(\frac{Rload}{Rload+Rout}\right) * I_L + \frac{-1}{Rload+Rout} * V_{Cout} \quad (33)$$

The above non-linear differential equations can be linearized by adding perturbation terms to the state variables and the duty ratio. When perturbation is added, the resulting equations can then be split into a linear average periodic steady-state solution and a perturbed expression that represents small signal dynamics. This can be achieved by neglecting any perturbation products and collecting the remaining terms on both sides and equating the like terms. The average state variables and the duty are replaced by the following expressions:

$$I_L = I_L + \widetilde{i_L} \quad (34)$$

where, I_L is the DC component and $\tilde{\imath}_L$ is the perturbation in I_L.

Similarly,

$$V_{Cout} = V_{Cout} + \widetilde{v_{Cout}} \tag{35}$$

$$D1 = D1 + \widetilde{d1} \tag{36}$$

Substituting equations (35-37) in equations (34, 35) and collecting the like terms we get:

$$L\,\frac{d\tilde{\imath}_L}{dt} = \left[-\left(Ron + RL + \frac{Rload * Rout}{Rload + Rout}\right) * I_L + \frac{-Rload}{Rload + Rout} * V_{Cout} + D1 * V_{in}\right] + \left[-\left(Ron + RL + \frac{Rload*Rout}{Rload+Rout}\right) * \tilde{\imath}_L + \frac{-Rload}{Rload+Rout} * \widetilde{v_{Cout}} + D1 * \widetilde{v_{in}} + \widetilde{d_1} * V_{in}\right] \tag{37}$$

$$Cout\,\frac{d\widetilde{v_{Cout}}}{dt} = \left[\left(\frac{Rload}{Rload+Rout}\right) * I_L + \frac{-1}{Rload+Rout} * V_{Cout}\right] + \left[\left(\frac{Rload}{Rload+Rout}\right) * \tilde{\imath}_L + \frac{-1}{Rload+Rout} * \widetilde{v_{Cout}}\right] \tag{38}$$

Equating the like terms on both sides, the steady-state or equilibrium solution can be found as:

$$0 = \left[-\left(Ron + RL + \frac{Rload*Rout}{Rload+Rout}\right) * I_L + \frac{-Rload}{Rload+Rout} * V_{Cout} + D1 * V_{in}\right] \tag{39}$$

$$0 = \left[\left(\frac{Rload}{Rload+Rout}\right) * I_L + \frac{-1}{Rload+Rout} * V_{Cout}\right] \tag{40}$$

and the small-signal dynamics can be represented by the perturbation equations as:

$$L\,\frac{d\widetilde{i_L}}{dt} = \left[-\left(Ron + RL + \frac{Rload*Rout}{Rload+Rout}\right) * \widetilde{i_L} + \frac{-Rload}{Rload+Rout} * \widetilde{v_{Cout}} + D1 * \widetilde{v_{in}} + \widetilde{d_1} * V_{in}\right] \tag{41}$$

$$Cout\,\frac{d\widetilde{v_{Cout}}}{dt} = \left[\left(\frac{Rload}{Rload+Rout}\right) * \widetilde{i_L} + \frac{-1}{Rload+Rout} * \widetilde{v_{Cout}}\right] \tag{42}$$

$$\widetilde{vout} = \left(\widetilde{v_{Cout}} * \frac{Rload}{Rload+Rout}\right) + \left(\widetilde{i_L} * \frac{Rout*Rload}{Rout+Rload}\right) \tag{43}$$

Using the above equations the small-signal model for a synchronous buck converter can be derived as shown in Figure 2.9. The general procedure described in detail in [34] has been followed to arrive at the model.

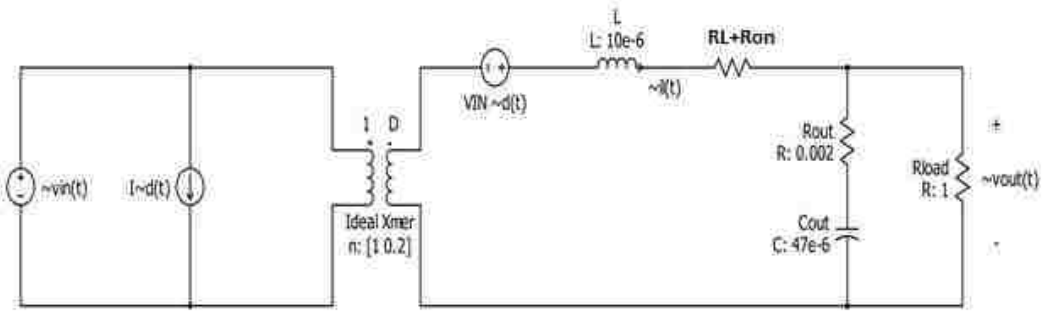


Figure 2.9 Small-Signal Model of Synchronous Buck Converter

The small-signal model can be used to find the transfer function models of the converter for line regulation, load regulation and duty ratio control.

2.3.3. Control-to-Output Transfer Function. The control-to-output transfer function describes the effect of the variations in the control input on the output. In a power converter with an output voltage regulator system, the control input is the duty ratio and the output is the output voltage. Thus the control-to-output transfer function is

represented as $G_{vd}(s)$. To find the effect of duty ratio perturbations on the output voltage, all other perturbations in the system model are set to zero and the small-signal model is solved for $\widetilde{v_{out}(s)}$ as a function of $\widetilde{d(s)}$.

$$G_{vd}(s) = \frac{\widetilde{V_{out}(s)}}{\widetilde{d(s)}} \; for \;\; \widetilde{v_{in}(s)} = 0 \tag{44}$$

Using equations (41-43), the control-to-output transfer function for the synchronous buck converter can be derived as:

$$\frac{\widetilde{V_{out}(s)}}{\widetilde{d(s)}} = \frac{V_{out}}{D}\left[\frac{1+(s*Rout*Cout)}{1+s*\left(\frac{L}{Rload+Ron+RL}+\frac{Cout*Rlod*(Ron+RL)}{Rload+Ron+RL}+Cout*Rout\right)+s^2\left(L*Cout*\frac{Rload+Rout}{Rload+Ron+RL}\right)}\right] \tag{45}$$

To validate the equation, a simulation model, as shown in Figure 2.10, is used to simulate the perturbation of the PWM control signal of the converter and the Bode of $G_{vd}(s)$ is compared with the simulation result as shown in Figure 2.11. The amplitude of the perturbation signal is set at 10% of its DC value (0.2) and data points are collected for a number of different frequencies. The data for which the simulation Gvd in Figure 2.11 is plotted is given in Table 2.2.

Table 2.2 Data to Plot $G_{vd}(s)$

Hz	20log10	Hz	20log10
100	13.23321783	10000	12.49546092
1000	13.35168032	100000	-30.82569092
2000	13.59447639	200000	-40.62035169
4000	15.02260954	400000	-32.97911611
6000	16.91215522	600000	-33.25441869
8000	16.3300756	800000	-37.06656153
		1000000	-52.31991539

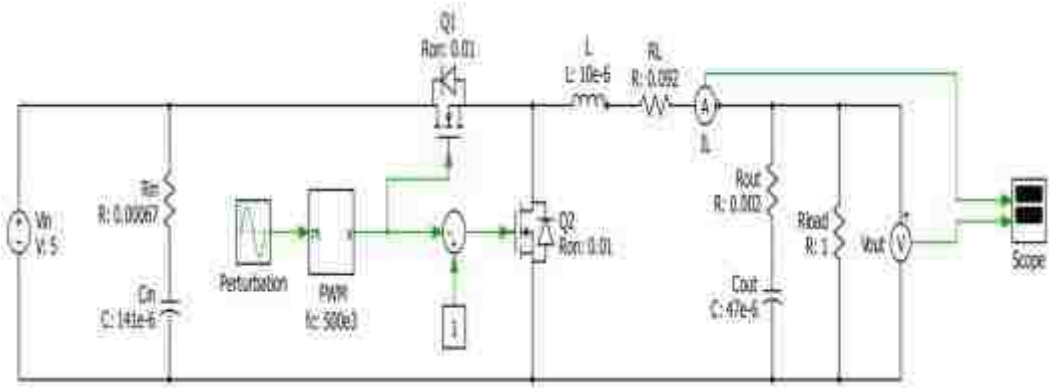


Figure 2.10 Perturbing the PWM Control Signal of the Converter

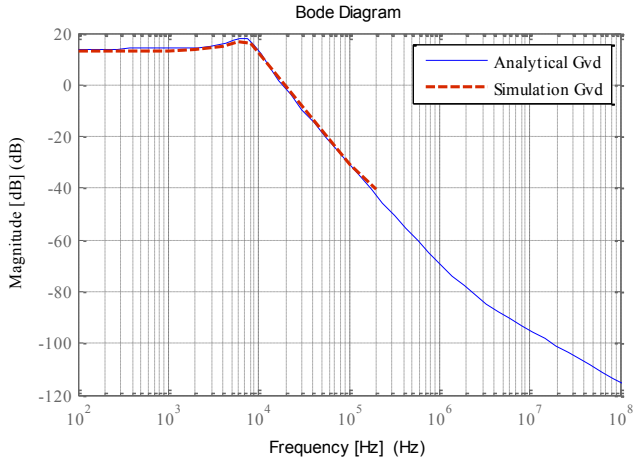


Figure 2.11 Comparison of $G_{vd}(s)$ with the Simulation Result

This exercise proves that the transfer function $G_{vd}(s)$ derived from the small-signal model of the plant models the effect of duty signal variations on the output of the plant accurately. Thus, this transfer function can be used as the model for the converter to design an appropriate compensator for closed-loop voltage regulation.

2.3.4. Discretization of Continuous Time Plant. In order to build an effective controller for a plant, it is essential to analyze the plant and the controller in the same domain. Usually, in a digital control system, the plant is modeled using differential

equations with continuous time variables whereas the controller is developed in a discrete time domain and understands only the transition between the sampling intervals [3]. To achieve superior performance of the controller, direct design technique is used [10]. In this technique, the continuous-time plant is first discretized and then the discrete-time controller is designed based on the discrete-time plant transfer function and other specifications. This method brings the plant and the controller on the same basis and helps the controller to take better control decisions.

Bilinear transformation, or Tustin's method, is a first-order approximation of the natural logarithmic function which is used to transform continuous time system to discrete-time and vice versa. When the substitution $z = e^{st}$ is used in the Laplace equation, the Z transform can be obtained. Thus it is an exact mapping from the s-plane to the z-plane. The above substitution can also be represented as:

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{46}$$

This transformation is utilized by a function in MATLAB called "c2d" which can be used to obtain the discrete-time model for the continuous-time plant model. The arguments required are the plant model, sampling time and the transformation method. The arguments used for the plant in this project are the control-to-output small-signal model $G_{vd}(s)$, sampling time of 2e-6 and Tustin's method. The result obtained is given in the MATLAB command window snippet below:

```
Gvd_Z =

  0.01238 z^2 + 0.02263 z + 0.01025
  ---------------------------------
        z^2 - 1.93 z + 0.9388

Sample time: 2e-06 seconds
Discrete-time transfer function.
```

Figure 2.12 Discrete-Time Transfer Function of $G_{vd}(z)$

2.4. DISCRETE CONTROL DESIGN

The discrete control-to-output transfer function $G_{vd}(z)$ can be used to study the stability property of the plant. For LTI discrete-time dynamic systems, two methods can be used to define the stability property. One, the response of the system to an impulse signal $\delta(0)$ applied at the input can be studied. Or for the transfer function models, the placement of poles on the z-plane can be used. These two methods are equivalent. The conditions for stability for these two methods are summarized in Table 2.3.

Table 2.3 Conditions to Define Stability Property of a Discrete System.

System Stability	Impulse Response Method	Pole Placement Method
Asymptotically Stable	Steady-State Response is Zero	All poles lie inside the unit circle
Marginally Stable	Steady-State Response is not equal to zero but is bounded	One or more poles on the unit circle.
Unstable	Steady-State Response is not bounded	At least one pole is outside the unit circle.

SISO tool in MATLAB is used to obtain the root-locus, bode and the analysis plots for a given system. The transfer function model of the plant is exported to the tool and the pole and zero placement technique is used to design the digital controller to obtain the desired response characteristics of the system. The Bode plot is used to determine the phase margin and gain margin of the system. The open-loop Bode and the root-locus of the plant are shown in Figure 2.13. The impulse response of the open-loop plant is shown in Figure 2.14. The plant has a zero-pole-gain (zpk) value given by equation (47) with a pair of complex-conjugate poles at 0.9650 + 0.0870i and 0.9650 - 0.0870i. From the impulse response and the pole placement it can be determined that the plant is asymptotically stable.

$$zpk(Gvd\_Z) = \frac{0.12379\,(z+1)(z+0.8282)}{(z^2-1.93z+0.9388)} \tag{47}$$

From the behavioral study of the plant and the plant Bode analysis the objectives of controller design can be identified as zero steady-state error, output regulation, and preserving the asymptotic stability of the plant while increasing the Phase margin (PM) and bandwidth. In discrete domain, the SISO tool uses the relation given by equation (48), where T_s is the sample time, to design the discrete compensator.

$$w = \frac{z-1}{T_s} \tag{48}$$

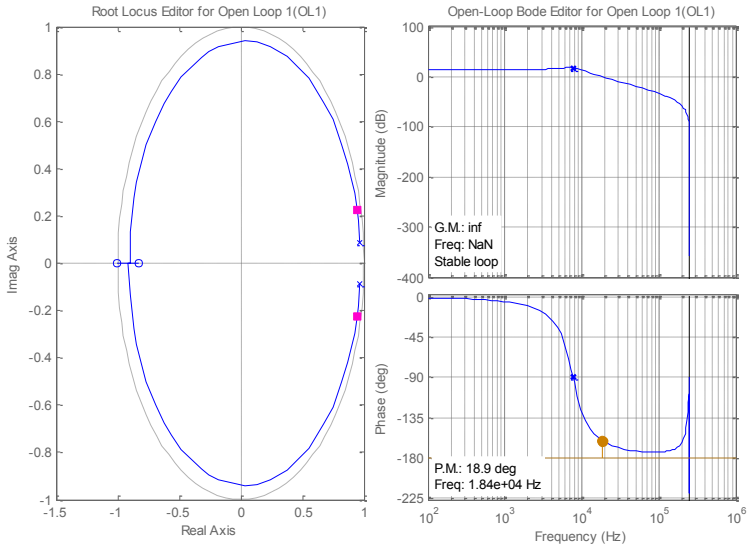


Figure 2.13 Root-Locus and Bode Plot of the Plant

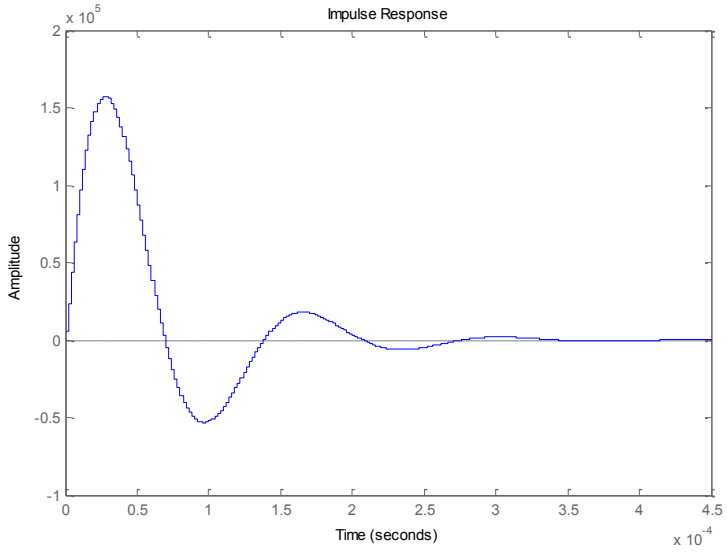


Figure 2.14 Impulse Response of the Plant

The compensator editor is used to add the poles and zeros to the plant transfer function to achieve desired closed-loop characteristics. Intuitive design and trial-and-error is used to arrive at a good choice for the compensator. Once the desired characteristics are obtained the compensator can be exported to the MATLAB command

window in the zpk form and the transfer function can be determined using the "*tf(x)*" command.

The zpk model of the compensator (F1) chosen and its transfer function are given in the MATLAB command window snippet in Figure 2.15. The bode plot and root-locus of the system are shown in Figure 2.16. The closed-loop impulse response is shown in Figure 2.17. The control-to-output transfer function of the closed loop system with unity feedback and the zpk model is shown in the MATLAB result snippet in Figure 2.18.

```
>> Comp

Comp =

  8.5256 (z^2 - 1.944z + 0.9517)
  ------------------------------
             z (z-1)

Sample time: 2e-06 seconds
Discrete-time zero/pole/gain model.

>> tf(Comp)

ans =

  8.526 z^2 - 16.57 z + 8.114
  ---------------------------
            z^2 - z

Sample time: 2e-06 seconds
Discrete-time transfer function.
```

Figure 2.15 Transfer Function of Compensator F1 Exported from MATLAB SISO Tool

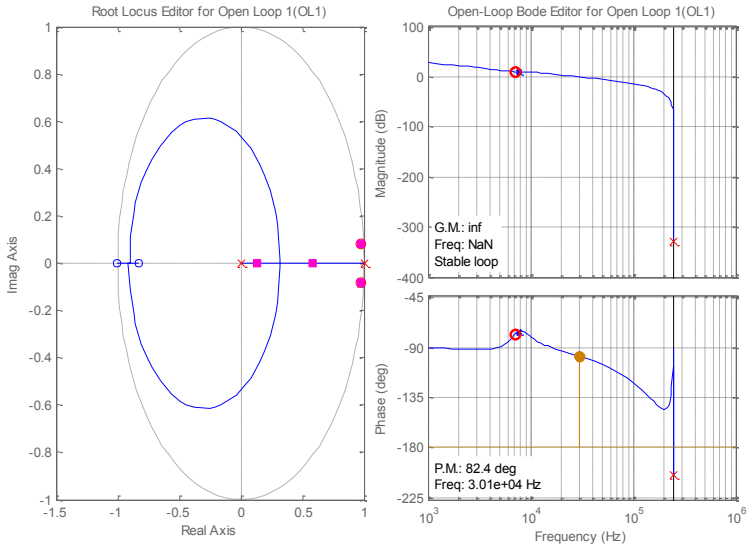


Figure 2.16 Root-Locus and Bode plot with Compensator F1

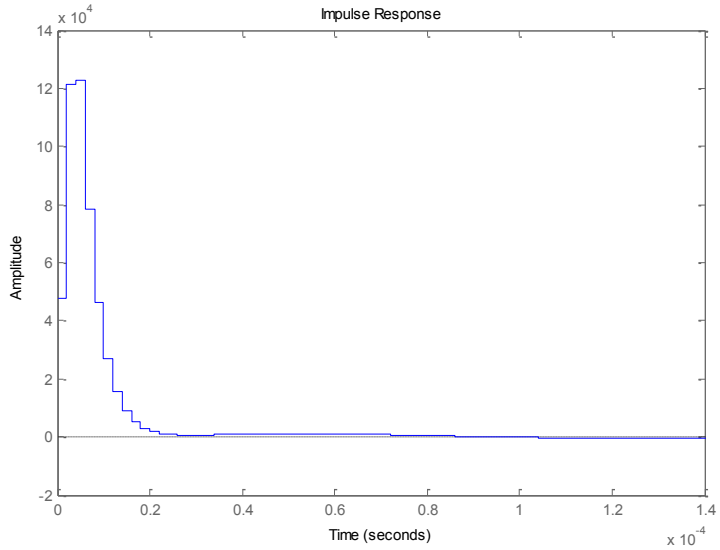


Figure 2.17 Impulse Response with Compensator F1

The chosen compensator (F1) significantly increases the PM of the plant and slightly increases the bandwidth. F1 retains the asymptotic stability of the plant thus making the closed-loop stable. Figure 2.19 shows the comparison of output voltage with and without the compensator. The compensator (F1) eliminates the steady-state error and

regulates the output at the set reference even during step variations in input voltage and output load. With these observations, it can be concluded that compensator F1 is a good choice for the plant.

```
Gvd_Z =

  0.01238 z^2 + 0.02263 z + 0.01025
  ---------------------------------
        z^2 - 1.93 z + 0.9388

Sample time: 2e-06 seconds
Discrete-time transfer function.


Sys_cl =

  0.1056 - 0.01227 z^-1 - 0.1873 z^-2 + 0.01368 z^-3 + 0.08319 z^-4
  ----------------------------------------------------------------
    1.118 - 2.932 z^-1 + 2.669 z^-2 - 0.9354 z^-3 + 0.08319 z^-4

Sample time: 2e-06 seconds
Discrete-time transfer function.

>> zpk(Sys_cl)

ans =

  0.094422 (1+z^-1) (1+0.8282z^-1) (1 - 1.944z^-1 + 0.9517z^-2)
  -------------------------------------------------------------
   (1-0.9454z^-1) (1-0.1315z^-1) (1 - 1.546z^-1 + 0.5985z^-2)

Sample time: 2e-06 seconds
Discrete-time zero/pole/gain model.

>> roots([1, -1.564, 0.5985])

ans =

    0.8961
    0.6679

``
```

Figure 2.18 Closed-Loop Transfer Function and Zero/Pole/Gain Representation with Compensator F1

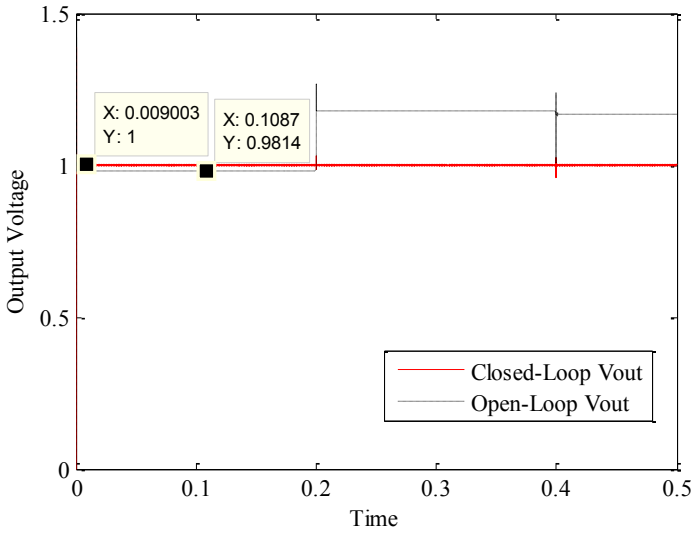


Figure 2.19 Output Voltage of Converter with Compensator F1

Another compensator (F2) which reduces the closed-loop bandwidth of the system is designed. Two compensators (F1 and F2) with different bandwidths are chosen to study the effect of bandwidth on the output spectral density. The zpk model of the compensator (F2) chosen and its transfer function is given in the MATLAB command window snippet in Figure 2.20. The bode plot and root-locus of the system are shown in Figure 2.21. The closed-loop impulse response is shown in Figure 2.22. The control-to-output transfer function of the closed loop system with unity feedback and the zpk model are shown in the MATLAB result snippet in Figure 2.24.

The chosen compensator (F2) significantly increases the PM of the plant and slightly increases the bandwidth. F2 retains the asymptotic stability of the plant thus making the closed-loop stable. Figure 2.23 shows the comparison of output voltage with and without the compensator. The compensator (F2) eliminates the steady-state error and regulates the output at the set reference even during step variations in input voltage and output load.

```
>> Comp

Comp =

  0.74105 (z-0.9966) (z-0.8401)
  -----------------------------
        (z-1) (z-0.0741)

Sample time: 2e-06 seconds
Discrete-time zero/pole/gain model.

>> tf(Comp)

ans =

  0.741 z^2 - 1.361 z + 0.6204
  ----------------------------
     z^2 - 1.074 z + 0.0741

Sample time: 2e-06 seconds
Discrete-time transfer function.
```

Figure 2.20 Transfer Function of Compensator F2 Exported from MATLAB SISO Tool

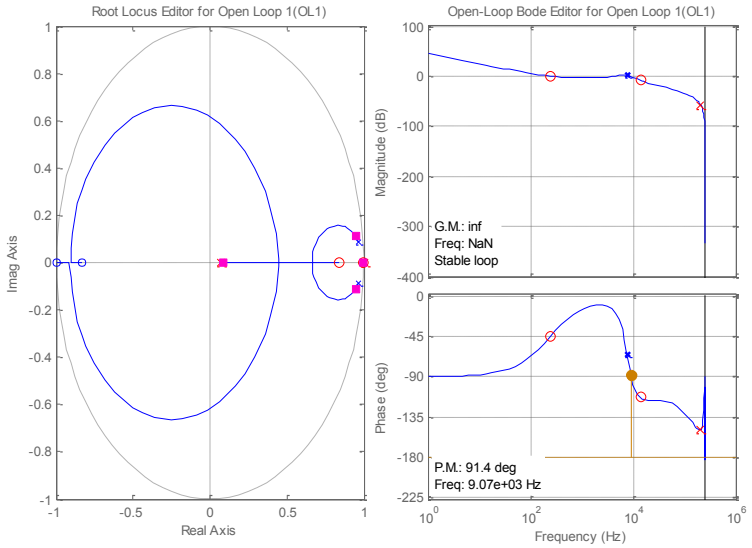


Figure 2.21 Root-Locus and Bode Plot with Compensator F2

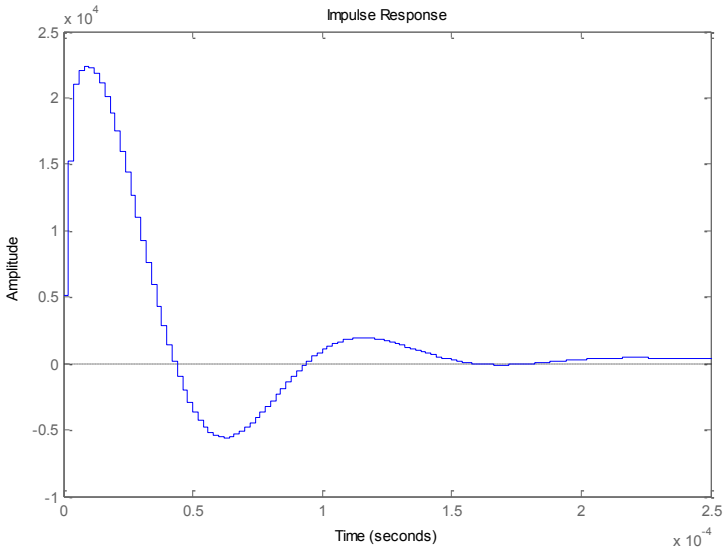


Figure 2.22 Impulse Response with Compensator F2

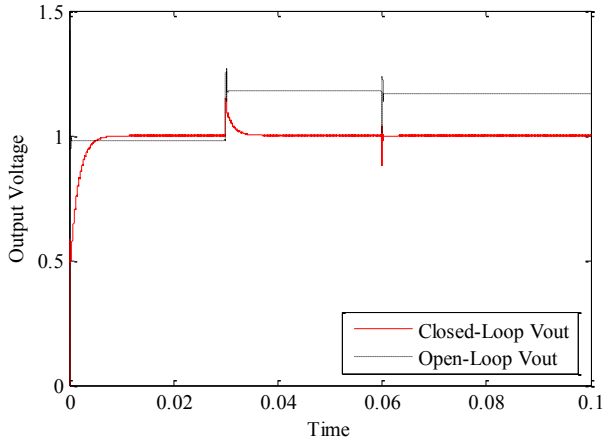


Figure 2.23 Output Voltage of Converter with Compensator F2

2.5. DIGITAL/ANALOG INTERFACE SYSTEMS

In a digitally controlled power converter system, the plant is an analog system whereas the controller is a digital system owing to the many advantages of digital controller. Though the plant is discretized to design the digital compensator, the real-time operation of the plant is in continuous-time domain. Therefore, an interface

system/device is required to establish a path for information exchange between the analog and digital devices.

```
Gvd_Z =

  0.01238 z^2 + 0.02263 z + 0.01025
  ---------------------------------
        z^2 - 1.93 z + 0.9388

Sample time: 2e-06 seconds
Discrete-time transfer function.


Sys_cl =

  0.009173 - 7.846e-05 z^-1 - 0.01552 z^-2 + 8.751e-05 z^-3 + 0.00636 z^-4
  ------------------------------------------------------------------------
         1.022 - 2.995 z^-1 + 3.057 z^-2 - 1.161 z^-3 + 0.0767 z^-4

Sample time: 2e-06 seconds
Discrete-time transfer function.

>> zpk(Sys_cl)

ans =

  0.0089795 (1-0.9966z^-1) (1-0.8401z^-1) (1+z^-1) (1+0.8282z^-1)
  ---------------------------------------------------------------
       (1-z^-1) (1-0.08267z^-1) (1 - 1.849z^-1 + 0.9086z^-2)

Sample time: 2e-06 seconds
Discrete-time zero/pole/gain model.

>> roots([1, -1.849, 0.9086])

ans =

   0.9245 + 0.2322i
   0.9245 - 0.2322i
```

Figure 2.24 Closed-Loop Transfer Function and Zero/Pole/Gain Representation with Compensator F2

Two such devices are the Analog-to-Digital Converter (ADC) and the Digital-to-Analog Converter (DAC) equivalent to Digital Pulse Width Modulator (DPWM) in power converter systems. In this section, these devices are introduced and an important design specification of these devices, called resolution, is explained.

2.5.1. Analog-to-Digital Converter (ADC). ADC is a mixed-signal device which is used to digitally represent the amplitude of an analog input. The analog input is usually the input or reference voltage. The corresponding output word is used by the digital compensator to calculate the error. ADCs which are used to digitize the analog signals consist of Quantizers and Sample-and-Hold(S/H) devices. S/H devices sample the analog signal at regular intervals (called sampling interval) and hold the value for a minimal period of time. This process is known as discretization. The quantizer then rounds-off or maps the amplitude of these samples to corresponding digital words. This process is called quantization. Digitization ensures that the output of the ADC matches the discrete-time and digital format of the compensator.

An ADC can also be thought of as a divider. A practical ADC has a reference analog voltage which is used as the basis for comparison to determine the output corresponding to a particular input. The output then tells us what fraction of analog reference is the analog input. The accuracy of the output depends on the precision with which this fraction is represented.

2.5.1.1. Resolution of ADC. The length or number of bits in the output digital word is determined by the resolution of the ADC. For example, a one-bit comparator can be thought of as a simple ADC. The input to the comparator is compared against a reference value and if the input is greater than the reference then the output is set to a

high state else it is set to a low state. Thus for a 1-bit comparator or ADC two output states are possible '0' or '1'. Thus, the length of the output digital word is 1 bit (which is sufficient to represent the states). Here, the resolution of the ADC is one. Similarly, for a 3-bit ADC, the resolution of ADC is 3, 8 output states are possible and the length of output word is 3. In general for an n-bit ADC 2^n states are possible and length of the output word is n. Figure 2.25 shows the output of ADCs with different bit resolutions for an input sine wave.

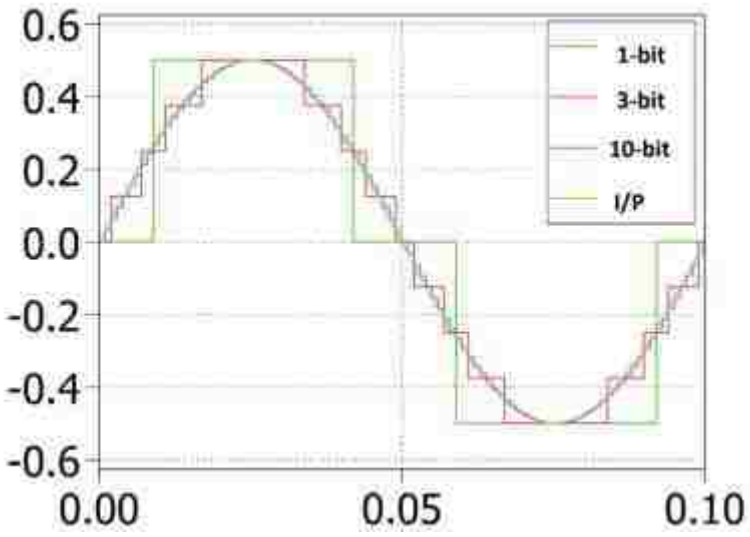


Figure 2.25 ADC Output for 3 Different Bit Resolutions

Resolution can also be specified in terms of the size of the Least Significant Bit (LSB) when the reference voltage is known. As the resolution of ADC (n) increases, the number of states in which the input can be resolved as output increases by a factor of 2^n. Consequently, the weight/size of each bit decreases by the same factor. For example, for a 1-bit ADC, in Figure 2.25, for a reference of 0.5V, the size of LSB is given by equation (49) and is equal to 0.25V.

$$Size\ of\ LSB = \frac{Vref}{2^n} \tag{49}$$

As long as the input is less than 0.25 V, output is represented by the code ‘0’ and when the input is greater than 0.25V, output is represented by code ‘1’ irrespective of the actual value of the input. Thus, the smallest increment this converter can resolve is 0.25V and is called the resolution of the converter.

2.5.1.2. Quantization and quantization error. The input-output characteristic curve of a 3-bit quantizer is shown in Figure 2.26. The x-axis denotes the input voltage as a fraction of the reference voltage of 0.5 V (equal to size of LSB). The y-axis denotes the output code. Initially, the input is zero therefore the ADC outputs “000” as the corresponding output code. As the input increases, the output remains at “000” as long as the input is less than 0.0625V irrespective of the actual value of the input. As the input voltage becomes greater than 0.0625V, the output jumps to code “001” and continues to stay there till the next change in LSB and so on. This process is called quantization and is effectively a round-off behavior.

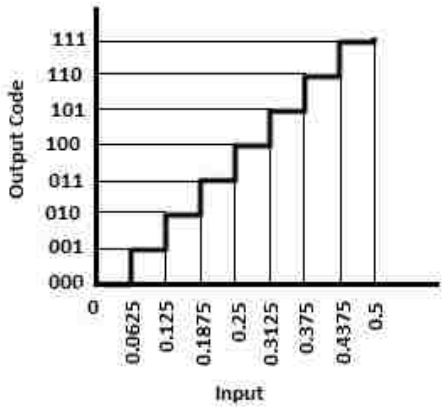


Figure 2.26 Input-Output Characteristic of a 3-bit Quantizer

Quantization introduces an error called quantization or round-off error in the output. When the input is 0, the output is “000” and the error is zero. As the input increases, the output remains at “000” till the input reaches 0.0625V irrespective of the value of input and the error increases correspondingly. When the input is equal to 0.0625V, the output code jumps to “001” which is the correct value and thus the error becomes zero. The plot of quantization error, Figure 2.27, is a saw-tooth waveform whose magnitude ranges from 0 to 1 LSB.

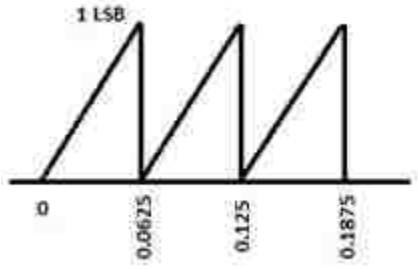


Figure 2.27 Quantization Error of a 3-bit Quantizer

2.5.1.3. Quantization modeling. The quantization process creates a noise called the quantization noise. The amplitude of the quantization noise is proportional to the maximum quantization error and thus at higher resolutions the quantization noise decreases due to the reduction in maximum error. Two approaches are commonly used to model quantization. A quantizer may be considered as a nonlinear gain element, as in [16]. Far from the origin, the gain approaches unity. However, near the origin, extreme nonlinearity may be observed, as shown in Figure 2.28.

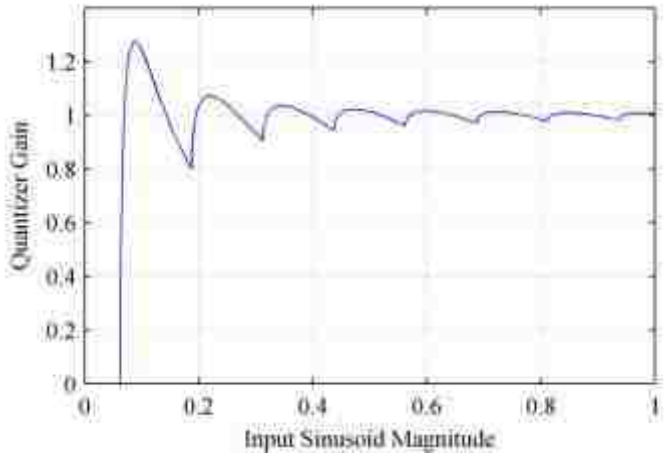


Figure 2.28 Describing Function (nonlinear gain) for 3-bit Quantizer

From an initial gain of zero, the quantizer's gain increases to a maximum of $(4/\pi)$ before settling near unity. If there is an offset (that is, if the signal is not centered on a precise bin value), the gain may be even larger [16]. This extreme gain effect may be avoided by ensuring that the voltage reference is exactly equal to one of the possible ADC output values. Still, the quantizer's gain may lie within $[0,(4/\pi)]$. Another approach is to model the quantizer (with "LSB" bits to the right of the radix) as adding some noise, as shown in Figure 2.29. The output y is equal to the input u plus some error e.

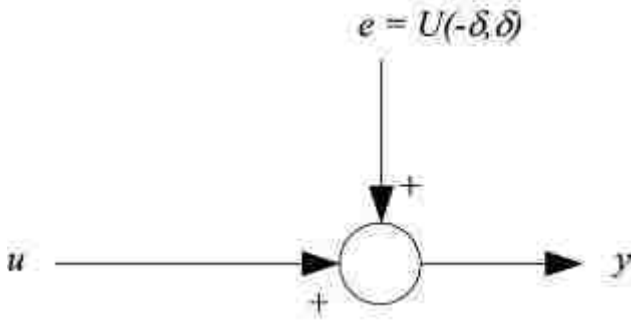


Figure 2.29 Quantizer Modeled with Additive White Noise

The error may be modeled as a uniform random variable, whose magnitude lies within $[-\delta,\delta]$, where

$$\sigma^2 = \frac{1}{3}\,\delta^2 \tag{50}$$

A sampled random variable with variance σ^2 is white noise with a flat power spectral density (PSD) equal to σ^2 at all frequencies, up to the Nyquist frequency.

An example of the PSD of a 12 bit ADC used to convert the reference voltage of 1V, sampling at 500 kHz is shown in Figure 2.30. The PSD of the ADC, shown in red, is a flat as expected. The plant Bode is superimposed on the PSD to show the attenuation by plant at higher frequencies. In the next section, an over-sampling modulator called Σ-Δ modulator is discussed which is used to shape the quantization noise to decrease its magnitude in the frequency of interest while increasing its magnitude at higher frequencies which will be attenuated by the filter components. The net result is a reduction in the noise magnitude than without the modulator.

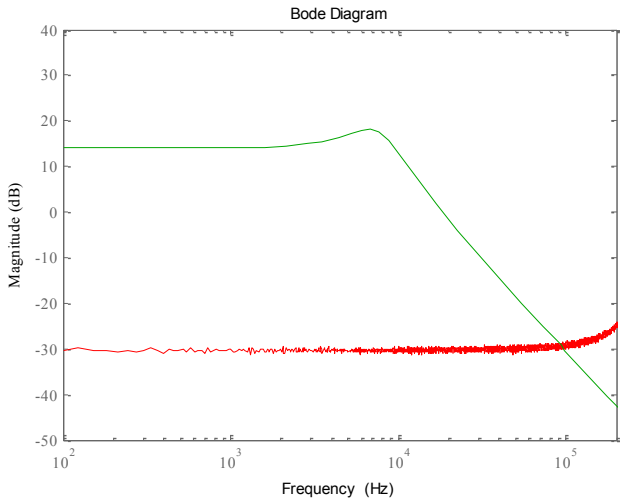


Figure 2.30 Attenuation of the Quantization Noise by the Plant's Filter at Higher Frequencies

2.5.1.4. Accuracy. Accuracy of output or lower quantization error can be achieved by using smaller LSB. This can be realized by either increasing the number of bits or by using smaller reference voltage or by doing both. In power converter application, the number of bits of the ADC is controlled to obtain desired output voltage accuracy. To meet the output voltage specifications, the error due to the the resolution of ADC should be lower than the allowed ripple in the output voltage. The ADC resolution which can satisfy this criterion is given in [31] as:

$$N_{ADC} = log_2 \frac{Vmax\ a/d}{V_{ref}} . \frac{V_{out}}{\Delta V_{out}} \tag{51}$$

Another relationship between the desired accuracy of the output voltage, ΔV_{out}, and resolution of ADC (N_{ADC}) is given in [6] as

$$N_{ADC} = \log_2 \frac{V_{in}}{\Delta V_{out}} \tag{52}$$

Figure 2.31 depicts the effect of ADC resolution on output voltage of the VMC synchronous buck converter. The DPWM resolution is assumed to be 12-bits and the impact of ADC resolution on the closed-loop system is observed. The switching frequency is 500 KHz and the reference voltage for regulation is 1V. The reference value for the ADC is 3.3 V. The green curve is the plot for ADC resolution of 3 bits. The high gain of the ADC affects the closed-loop gain margin and makes the closed-loop marginally stable creating oscillations in the output voltage. The red curve, a plot for 6-bit ADC, shows better stability than the 3-bit ADC but exhibits periodic behavior known as tones.

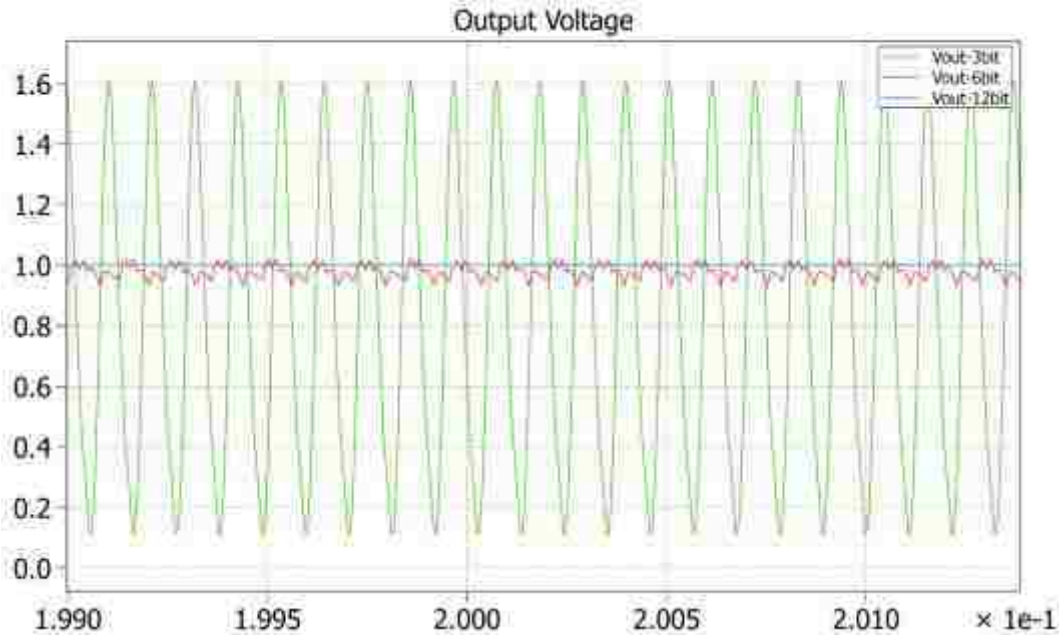


Figure 2.31 Output Voltage of Closed-Loop Synchronous Buck for 3 Different ADC Bit Resolutions

Low bit resolution of ADC decreases the duty cycle resolution of the converter which in turn affects the accuracy of output voltage. Table 2.4 summarizes the mean values of duty cycle and output voltage for various ADCs for a required output voltage regulation of 1V.

Table 2.4 Table Summarizing the effect of ADC Bit Resolution on Accuracy of Output Voltage

ADC	Mean of duty	Mean of output voltage
3-bit	0.1789	0.808
6-bit	0.2083	0.982
12-bit	0.2202	0.999

The resolution of an ADC is an important factor which contributes to the total output noise of a closed-loop converter. Due to its quantization function, the ADC

introduces a random noise in the system which depends on the bit resolution. Also, the closed-loop stability may be affected by the gain of ADC which again depends on the bit resolution of ADC. Also, the interaction of the ADC resolution with the DPWM resolution, given by equation (3), may introduce limit-cycles. It is important to analyze the system for various cases of the ADC resolution and thus it is used as a study variable in this work.

2.5.2 Digital Pulse Width Modulator (DPWM). In a digital control system the control signal output of the compensator is used to achieve the desired control action. In power converter system, the control action is to modulate the duty cycle of the power converter to achieve output regulation. Though the control signal inherently carries the duty ratio information, an interface is required to translate the control signal to the ON-OFF pulse signal understood by the gate driver. This translation is achieved by using a Digital Pulse Width modulator in power converter systems.

DPWM module helps retain the reliable and efficient Pulse Width Modulation (PWM) switching technique [2], used in analog switching power supplies for regulating voltage, in the digital control system. DPWM can be interpreted as a Digital-to-Analog Converter (DAC), which can be a simple PWM or an oversampling DAC like the Sigma-Delta Modulator (ΣΔ modulator). Effective implementation of DPWM is critical to ensure optimal system performance. An important characteristic which has direct impact on the quality of the regulated output voltage is the resolution of the DPWM.

2.5.2.1. Resolution of DPWM. In a DPWM the carrier signal used in analog PWM is replaced with a counter signal which counts up from zero to a maximum value (which depends on the number of bits) and then either counts back to zero or goes to zero

depending on whether it is implemented as a up counter or up-down counter. The actual implementation is discussed in detail in Chapter 5. The data from the compensator is stored in a register and compared against the counter signal to obtain the pulse-width modulated signals similar to an analog PWM. The time period of the pulse-width depends on the switching frequency, its duty ratio depends on the reference value and the time at which the next comparison occurs depends on the system clock. Thus, if the switching frequency and the system clock frequency are known the DPWM resolution can be determined using:

$$N_{DPWM} = \log_2 \frac{f_{clk}}{f_{sw}} \tag{53}$$

Like the ADC, the DPWM also exhibits quantization which results in a discrete value at its output. Due to this limited resolution of the duty cycle, the output voltage also has limited resolution and is given in [32] as:

$$\Delta Vout = Vin\, \Delta d \tag{54}$$

Figure 2.32 shows the effect of various DPWM resolutions on the accuracy or ripple of the output voltage for the closed-loop converter. The switching frequency is 500 KHz and the reference voltage for regulation is 1V. The resolution of the ADC is kept constant at 12 bits.

Figure 2.32 Output Voltage of Closed-Loop Synchronous Buck for 3 Different DPWM Bit Resolutions

The green curve is the output for a 3-bit DPWM, the red curve is for a 6-bit DPWM and the blue curve is for a 12 bit DPWM. The ripple in these curves is actually the switching ripple imposed upon the steady-state oscillations of the converter. This effect is clearly visible in the red curve in Figure 2.33. These oscillations are called limit cycles [15] and are discussed in detail in the next sub-section.

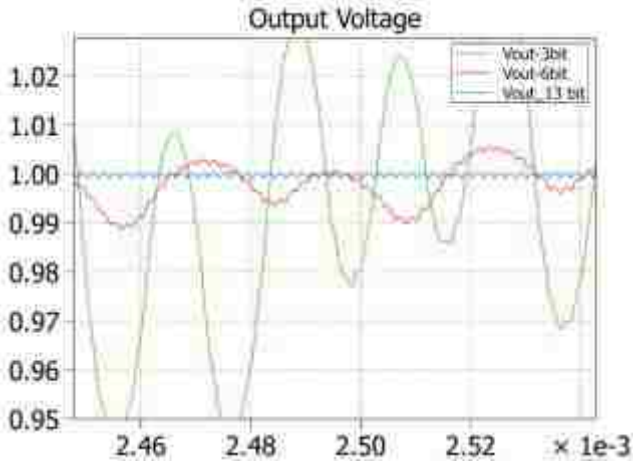


Figure 2.33 Steady-State Oscillations in the Low Resolution DPWM Module

2.5.2.2. Limit-cycle oscillations. The presence of two quantizers (ADC and DPWM) in a closed-loop digital control system leads to a steady state oscillations in the output voltage of the converter. These oscillations are known as limit-cycle oscillations (LCO) [15]. Limit cycle oscillations do not occur due to the PWM switching activity. Figure 2.34 shows a clear distinction between the switching ripple and the limit cycle oscillations for a closed-loop converter with 12 bit ADC resolution and 11 bit DPWM resolution. The PWM switching signal is included to show the scale of the switching ripple.

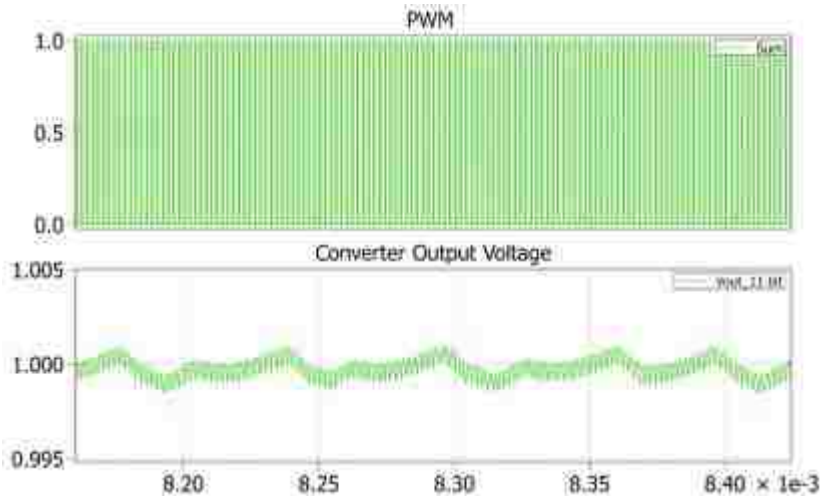


Figure 2.34 Limit-Cycle Oscillations in a 12-bit ADC, 11-bit DPWM Converter System

The accuracy of the output depends on the resolution of both the ADC and the DPWM. The LSB of steps in which the ADC quantizes the input voltage is given in [15] by:

$$\Delta V_{ADC} = \frac{V_{in}}{2^{N_{adc}}} \tag{55}$$

The LSB of steps in which the DPWM quantizes is given in [15] by:

$$\Delta V_{PWM} = \frac{V_{in}}{2^{N_{pwm}}} \tag{56}$$

For a reference input value of 1 V, and for the ADC reference voltage of 3.3 V, the ADC can output a digital code equal to 1241 (round-off error in ADC from 1241.2121 to 1241) to represent 0.998 V. This level is called the zero-error bin [15] because at steady-state the compensator will try to drive the output to this bin to achieve zero-error. But as the DPWM's resolution is 1 bit lower than the ADC, there is no exact DPWM bin which will correspond to the zero-error bin of the ADC. So, the DPWM outputs oscillate between 1.00088 V and 0.998623 V to average out 0.9998 V eventually. This causes the steady-state oscillations or limit-cycle oscillations in a power converter as depicted in Figure 2.35.

Elimination of limit cycles is essential because they cause output voltage oscillations whose frequency is lower than the switching frequency. Also, due to their unpredictable nature, the amplitude and frequency of such oscillations cannot be determined making it difficult to estimate the output noise and electro-magnetic interference (EMI) produced by the converter. To eliminate limit cycles, three conditions are proposed in [15]. First, the DPWM resolution should be at least one bit greater than the ADC resolution. Second, the control law should have an integral term. Third, the loop gain which includes the effective gain of the ADC should not be zero. Figure 2.35 shows the result when the DPWM resolution is one-bit greater than the ADC resolution.

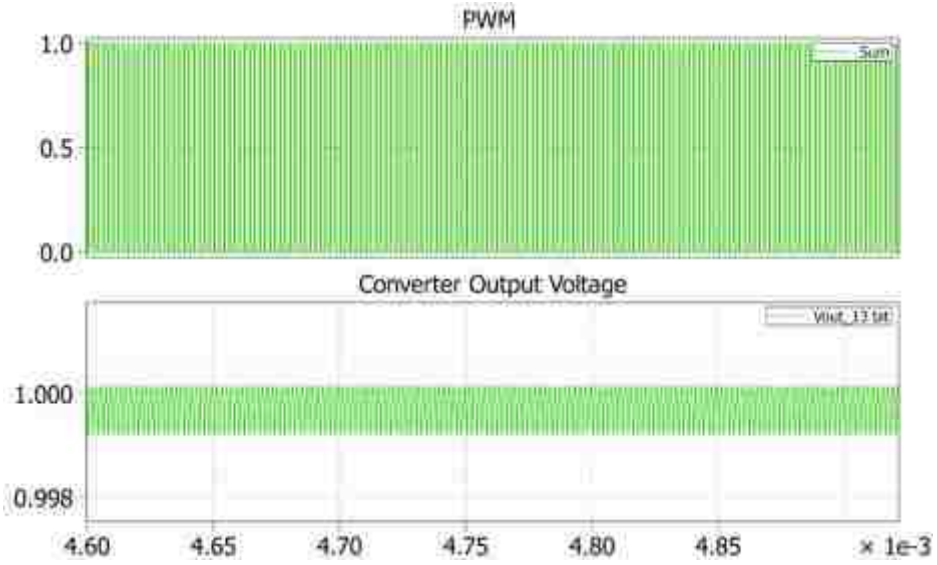


Figure 2.35 Elimination of Limit Cycles in a 12-bit ADC, 13-bit DPWM Converter System

As the system clock frequency is directly proportional to the resolution of DPWM, implementing a high resolution DPWM requires a very high frequency clock which may be not be possible with today's technological advances. One solution that has been proposed is to increase the effective resolution of the DPWM so as to be able to implement a high resolution DPWM with reasonable clock frequency. A few approaches have been discussed in the literature review section. Among those the popular technique of using the Σ-Δ modulator as a pre-processor block is used in the present work to increase the effective resolution of a 3-bit DPWM.

2.6. SIGMA-DELTA (Σ-Δ) MODULATION

The Σ-Δ modulator is a pre-processor block which increases the resolution of the input signal by varying its value over a period of time. By averaging this time varying signal a high resolution output signal can be obtained. Usually, a Σ-Δ modulator-DPWM

combination is used in a power converter system and the averaging process is done by the plant's filter components without the need for additional averaging circuitry.

In Digital Signal Processing terms, the Σ-Δ modulator can be described as an oversampling modulator sampling at rates much higher than the Nyquist rate. Unlike the Nyquist rate converters, oversampling converters use memory elements to be able to generate output using all the input values [22]. They sacrifice the one-to-one relation between the input and output samples to achieve over 20 Effective Number of Bits (ENOB) resolution at high speeds [22].

The effect of using a Σ-Δ modulator is to filter the quantization noise with some transfer function called the Noise Transfer Function (NTF):

$$NTF = (1 - z^{-1})^n \tag{57}$$

Here, z^{-1} represents a unit delay (in a discrete-time process) and n represents the order of the Σ-Δ modulator. The general structure of a Σ-Δ modulator is shown in Figure 2.37. Figure 2.37 has been adapted from [22] to explain the concept of noise-shaping.

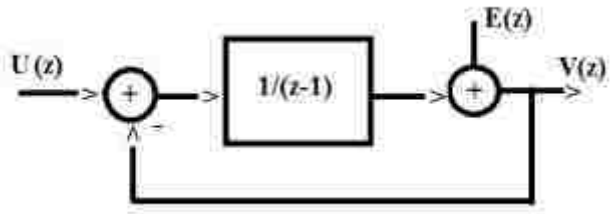


Figure 2.36 General Structure of a Σ-Δ Modulator
Courtesy: Reference [22]

In the above figure, U (z) represents the input to the modulator in discrete-time. E (z) is the quantization noise modeled as additive white noise. V (z) is the output of the modulator. The relationship between the input, additive noise and output can be given as [22]:

$$v(k) = u(k-1) + e(k) - e(k-1) \tag{58}$$

for a sampling rate (f_s) of 1Hz at time k. In words the above equation can be described as the output is equal to delayed input signal and a differentiated version of the error [22].

Thus, the differentiation of the error or noise attenuates it at frequencies which are less than the sampling frequency. This process is called noise shaping. When combined with the power stage of a power converter, this noise shaping modulator drastically reduces the quantization noise of the quantizers in the digital control loop of the system by pushing the noise in the lower frequencies to higher frequencies where it is filtered by the converter filter components. The noise shaping concept is also widely used in the ADCs and DACs. Depending on the value of n in the NTF equation, the modulator can be recognized as a 1st- order modulator, a 2nd – order modulator or a higher order modulator.

2.6.1 First-Order Σ-Δ Modulator. The error feedback form implementation of the 1st-order modulator is shown in Figure 2.37. In this form the feedback signal can be sized only for the resolution of the error than for the resolution of the input. In the Figure 2.37, d is the high resolution input signal from the compensator. d_{LR} is the input to the PWM process which had DPWM bits of resolution.

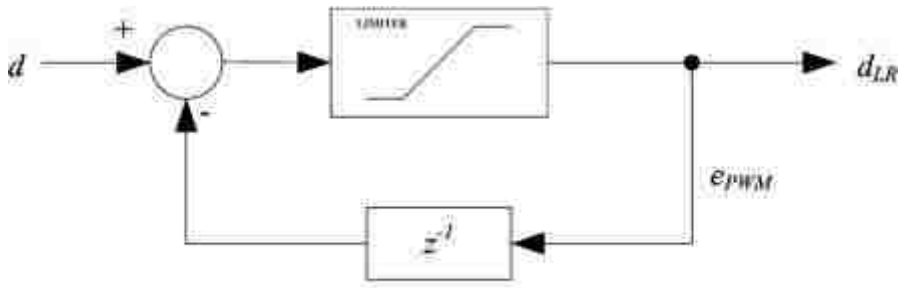


Figure 2.37 Error Feedback Form Implementation of 1st-order Modulator

Then the error e_{PWM} has d - d_{LR} bits of useful resolution which is passed through the filter with transfer function equivalent to NTF. The function of the limiter is to limit the error within the limits in which the actual PWM process is valid. For this modulator the NTF is $(1- z^{-1})$. The Power Spectral Density (PSD) of the output noise can be found by using the equation in [22] :

$$psd = (2\sin(\pi f T))^2 \, \sigma^2 \tag{59}$$

were σ^2 is the PSD of the quantization noise given by (50). T is the sampling period. The expected PSD of a 1st-order modulator for a sampling period of 2e-6 s and $\delta = 2^{-nPWM-1}$ for a 3-bit DPWM is shown in Figure 2.38.

The 1st – order modulators suffer from low-frequency noise which is introduced by the low-frequency periodic behavior known as tones [25]. These idle tones may be become evident as spikes in the PSD of the modulator. Some irregularity in the output also appears as spikes in the PSD of the modulator as shown in Figure 2.39 for a 0.1896973 duty input.

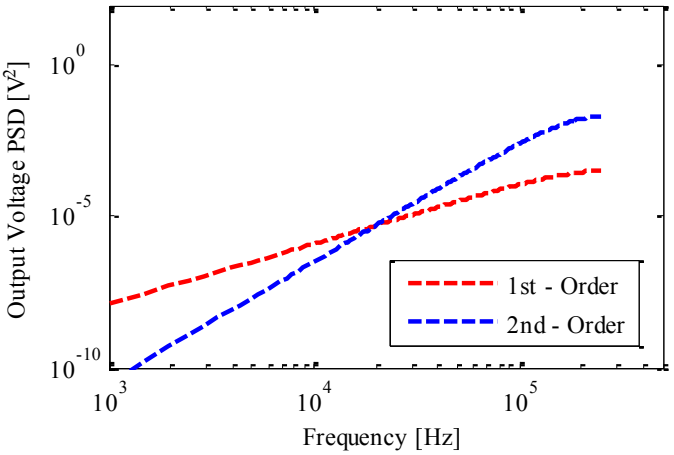


Figure 2.38 Expected PSDs of 1st-order and 2nd-order Σ-Δ Modulators

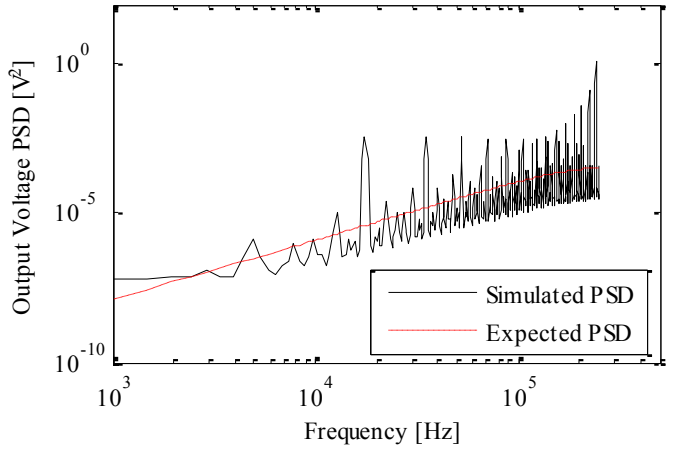


Figure 2.39 PSD of 1st-order Modulator Depicting the Spikes due to Idle Tones/Irregularities

2.6.2. Second-Order Σ-Δ Modulator. The error feedback form implementation of the 2nd -order modulator is shown in Figure 2.40. In this form the feedback signal can be sized only for the resolution of the error than for the resolution of the input. In the Figure 2.40, *d* is the high resolution input signal from the compensator. d_{LR} is the input to the PWM process which had DPWM bits of resolution.

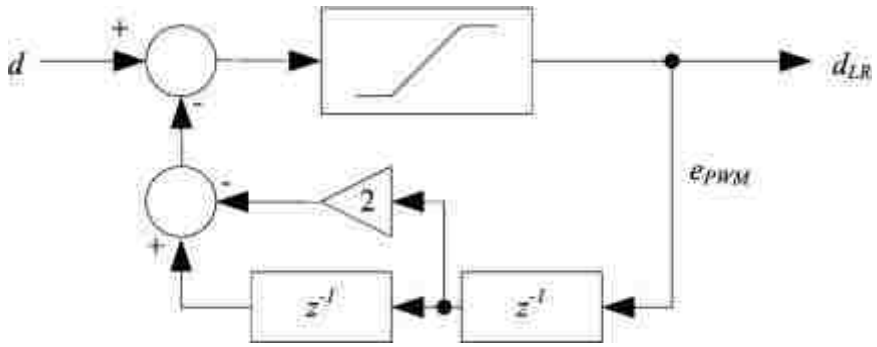


Figure 2.40 Error Feedback Form Implementation of 2nd-order Modulator

Then the error e_{PWM} has d - d_{LR} bits of useful resolution which is passed through the filter with transfer function equivalent to NTF. The function of the limiter is to limit the error within the limits in which the actual PWM process is valid. For this modulator the NTF is $(1- z^{-1})^2$. The Power Spectral Density (PSD) of the output noise can be found by using the equation in [22] :

$$psd = (2\sin(\pi f T))^4 \, \sigma^2 \tag{60}$$

Where σ^2 is the PSD of the quantization noise given by (50). T is the sampling period. The expected PSD of a 2nd -order modulator for a sampling period of 2e-6 s and $\delta = 2^{-nPWM-1}$ for a 3-bit DPWM is shown in Figure 2.38. The 2nd –order modulators suppress the low-frequency tones. The simulated PSD of a 2nd – order modulator for 0.1896973 duty input is shown in Figure 2.41. When compared to the 1st-order modulator with same duty input the 2nd – order modulator has a smooth curve and matches very well with the expected PSDwhen compared to the 1st – order modulator.

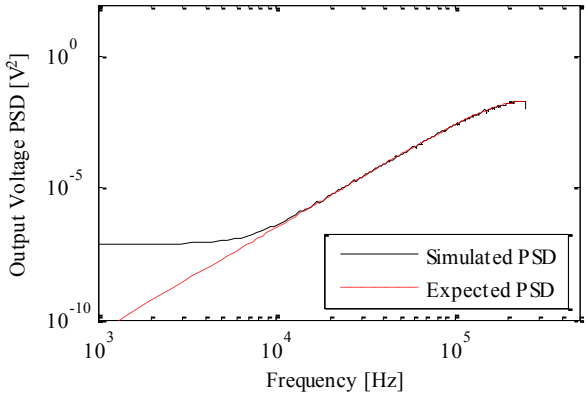


Figure 2.41 PSD of 2nd-order Modulator Depicting a Smooth Curve due to the Suppression of the Low-Frequency Tones/Irregularities

2.6.3. Higher Order Σ-Δ Modulator. Higher order modulators can be developed using the generic structures of the 1st – order and 2nd –order modulators. The general structure of a modulator used in [22] is given in Figure 2.42.

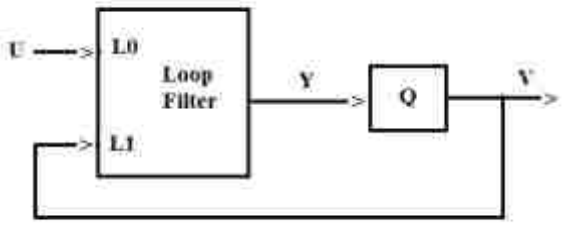


Figure 2.42 General Structure of a Σ-Δ Modulator
Courtesy: reference [22]

The NTF in terms of the loop filter input L1 is [22]:

$$NTF(Z) = \frac{1}{1-L_1(z)} \; ; L_1(z) = 1-(1-z^{-1})^N \tag{61}$$

Here, *N* is the number of times the quantization noise is differentiated by the NTF which is also equal to the order of the modulator and the number of poles of L_1 that lie on the unit circle. In the present work, the loop filter has only one input which is the difference of *u (n) – v (n)*. The general structure for this case is given in Figure 2.44. In this case, $L_1 = -L$ and the NTF is given by:

$$NTF(z) = \frac{1}{1+L(z)} \tag{62}$$

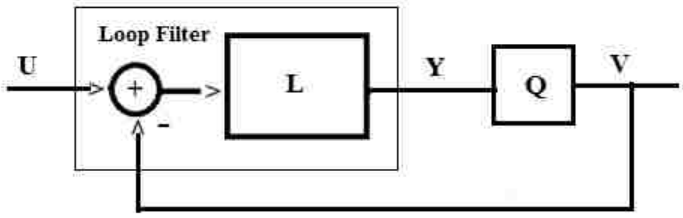


Figure 2.43 General Structure for Error Feedback Topology
Courtesy: Reference [22]

2.6.4. Stability of Σ-Δ Modulators. The 1st – order modulator is stable at all frequencies with inputs less than or equal to 1[22]. Due to the presence of the second integrator in a 2nd – order modulator, it is less stable than 1st – order modulator. The input should be limited to 0.8 or 0.9 to prevent large state of the second integrator [22]. The stability conditions for higher order modulator involve an additional variable, the quantizer bit size. Special considerations have to be made for stability while using higher order modulators and extensive simulations are needed to understand their stability criteria before implementing them.

2.6.5. Advantages of Σ-Δ Modulators. The noise shaping characteristic, demonstrated using an example in Figures 2.39 and 2.41, is a major advantage of using the modulators. The modulator shapes the flat PSD of quantizer noise depicted in Figure 2.30 such that the noise is increased at higher frequencies where it is attenuated by the plant's LC filter while decreasing the noise in the lower frequencies.

When used along with a DPWM, the modulator helps increase the effective resolution of the DPWM thus enabling the implementation of high resolution DPWM at reasonable clock frequencies.

The Equivalent Number of Bits (ENOB) can be calculated by using the Signal-to-Noise (SNR) method discussed in [24]. For a 1st – order modulator the SNR is given in [24] as:

$$SNR = 5.62 + 20\log_{10} N + 30\log_{10}\left(\frac{f_s}{2f_b}\right) \tag{63}$$

Where N is the core DPWM resolution in use, f_s is the sampling frequency and f_b is the bandwidth of the plant's LC filter. Using the above SNR value, the ENOB can be calculated as

$$ENOB = \frac{SNR-1.76}{6.02} \tag{64}$$

For the system used in this work, the resonant frequency of the plant is 7.3413 kHz. The actual Q factor which is the parallel combination of the series and parallel Q factor is 0.1918. The bandwidth is

$$f_b = \frac{Resonant\ Frequency}{Q\ Factor} = 38.3\ kHz \tag{65}$$

The sampling frequency is 500 kHz and N is 3. The SNR value is equal to 39.6124 and ENOB is 6 bits. To implement a DPWM of 6-bits, the system clock frequency required is 32 MHz according to (53). But with a Σ-Δ modulator pre-processor block, a 3-bit DPWM can be implemented with a system clock frequency of 4 MHz to achieve 6-bit precision. Based on the limit-cycle argument, the effective DPWM resolution is still less than the ADC resolution. As a result limit cycles do occur in the output voltage as shown in Figure 2.44 which is similar to the 6-bit curve in Figure 2.32. However, for the present work the noise shaping property of the modulator is of greater interest than the no limit cycle solution.

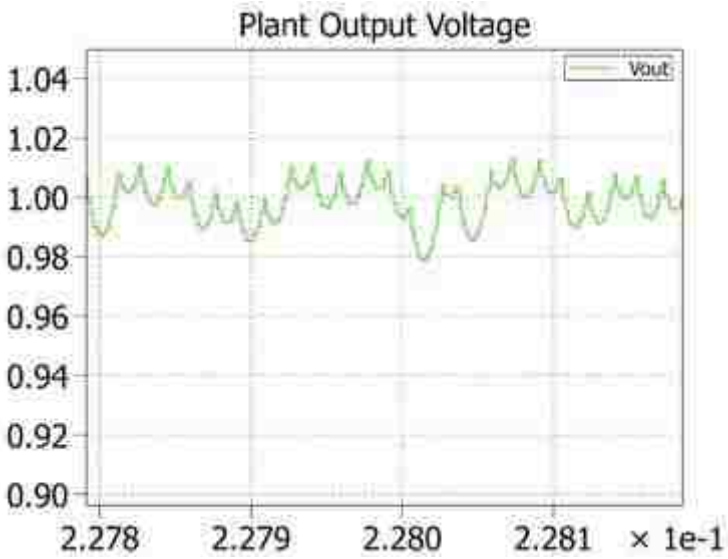


Figure 2.44 Limit Cycles in the System Output Voltage due to Low DPWM Resolution

2.6.6. Order of Σ-Δ Modulator as a Potential Study Variable. The order of the Σ-Δ modulator (N) is an important characteristic which determines the amount of quantization noise allowed in the output. The 1st – order modulator allows more noise

and tones in the output whereas the 2^{nd} – order modulator suppresses the tones and exhibits better noise shaping characteristic than the 1^{st} – order modulator. Also, the stability criteria to be considered vary with the order of modulator. While implementing the modulator in a closed-loop with two quantizers, ADC and DPWM, the order of the modulator is important when the DPWM noise dominates. Thus, N is chosen as a potential study variable to determine its effect on output noise under various conditions.

2.7. SECTION SUMMARY

In this section the theoretical concepts involved in the design of digital control of power converters were reviewed. The plant was modeled and the small-signal control-to-output transfer function was discretized to serve as the mathematical model for direct control design. The digital compensator was designed using MATLAB SISO tool. Two suitable compensators, F1 and F2, with different closed-loop bandwidths were designed. Both of the compensators satisfied the stability criterion and regulated the output voltage at the set reference under input voltage and load variations. The importance of resolution of the two quantizers ADC and the DPWM was discussed and the implementation of Σ-Δ modulator as a quantizer noise-shaping block was discussed. In the next section the theoretical analysis is developed and the system characteristic curves are generated to be used as a basis for comparison in the following sections.

3. NOISE ANALYSIS OF A DIGITAL VMC SYNCHRONOUS BUCK CONVERTER SYSTEM

3.1. INTRODUCTION

Most previous work has considered the Σ-Δ modulator in isolation. In [24], the results determine actuation spectra with and without Σ-Δ modulation, but do not include the remainder of the system. Since both Σ-Δ modulation and the voltage controller include integral feedback, and there is quantization in both the PWM module and the ADC, the spectral characteristics of the complete system are more complex.

Consider a common voltage-mode feedback control system shown in Figure 3.1. Output voltage y is measured with an ADC, which has n_{ADC} bits of resolution. The measured value y_{ADC} is subtracted from a reference V_{REF}. The resulting error e_V is fed to a compensator with transfer function F to determine the desired duty ratio d. A first- or second-order Σ-Δ modulator converts d into d_{LR}, which now has n_{PWM} bits of resolution. A PWM module drives the plant, which has a transfer function G. In this study, we would like to determine the spectral characteristics of y, specifically, the contribution of the digital-analog interface systems to the output noise.

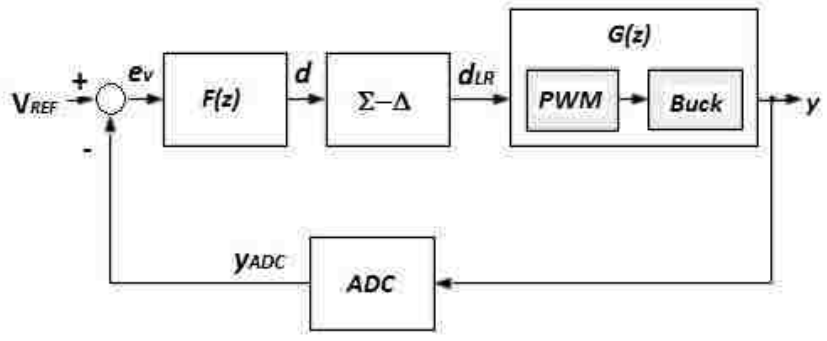


Figure 3.1 Voltage-Mode Feedback Control System

3.2. NOISE SOURCES AND THEIR CONTRIBUTION TO OUTPUT PSD

If the system were ideal with no quantizers (ADC, PWM) and using sufficiently long words in the computation, the closed-loop transfer function from V_{REF} to y would be

$$Sys_{Cl} = \frac{FG}{1+FG} = \frac{FG}{Loop\ Gain} \tag{66}$$

3.2.1. Contribution of ADC to Output Noise. With the addition of ADC to the system, the nonlinear gain of the ADC, denoted k_1, may lie in the range [0, (4/π)] and modifies the loop gain, so that

$$loopGain = 1 + k_1 FG \tag{67}$$

For large values of n_{ADC}, k_1 is approximately equal to 1. However, as the value of n_{ADC} decreases (typically around 8 bits or less), k_1 first rises to its maximum value, and then vanishes. The designer must consider this effect when designing the compensator F. If the loop gain, with the effect of k_1, has poor gain margin, idle tones may develop [16].

The ADC introduces quantization noise $e_{ADC} = y - y_{ADC}$, which is white noise with variance σ^2_{ADC}. This noise source is filtered by the closed-loop transfer function because it enters before the compensator. Therefore, its contribution to the output noise PSD is

$$P_{ADC} = \sigma^2_{ADC} \left| \frac{FG}{1+k_1 FG} \right|^2 \tag{68}$$

The expected spectral density curve of ADC quantization noise for the closed-loop system is shown in Figure 3.2. The curve is plotted for n_{ADC} of 12 bits and $k_1 = 1$. The ADC contribution has a constant white noise characteristic with attenuation at cut-off frequency of the closed-loop system.

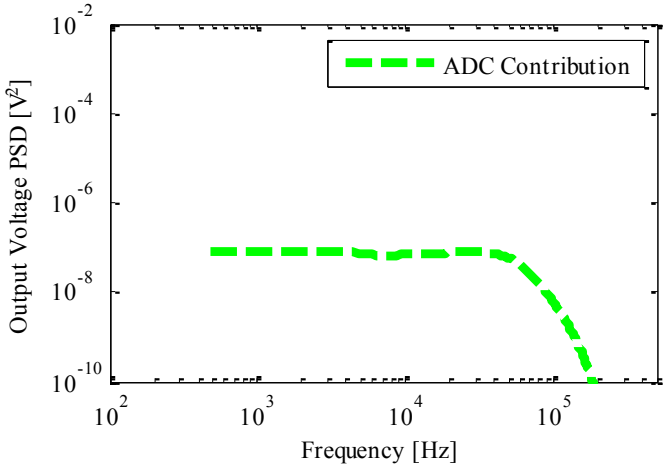


Figure 3.2 ADC Contribution to Output Noise PSD

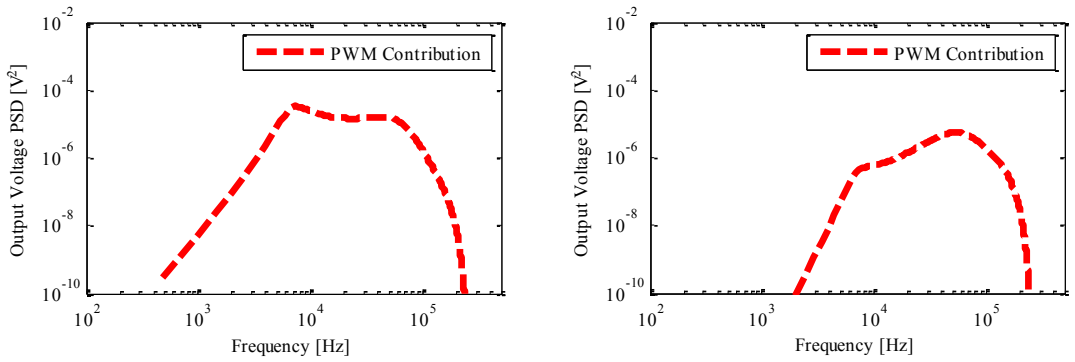
3.2.2. Contribution of PWM to Output Noise. The PWM module introduces quantization noise e_{PWM}, which is white noise with variance σ^2_{PWM}. This noise is filtered by the Σ-Δ process, and then applied to the plant. Because e_{PWM} enters after the compensator but before the noise-shaping filter, its contribution to the output voltage PSD is

$$P_{PWM} = \sigma^2_{PWM} \left| \frac{GN}{1+k_2FG} \right|^2 \tag{69}$$

A second quantizer gain k2 ∈ [0, (4/π)] has been introduced. Depending on the relative values of n_{ADC} and n_{PWM}, as well as the characteristics of G and F, k1 and k2 may be

different. For most design choices, k1 = k2 = 1, but these gains may take on other values for extremely low resolution systems.

The expected spectral density curve of PWM quantization noise for the closed-loop system is shown in Figure 3.3. Figure 3.3 (a) shows the expected curve for a first-order ∑- Δ modulator and Figure 3.3 (b) shows the expected curve for a second order modulator. Both the curves have been plotted for n_{ADC} of 12 bits and $k_2 = 1$. The constant white noise characteristic of the PWM quantizer has been shaped by the noise shaping transfer function of the ∑- Δ modulator pushing the low frequency noise to higher frequency where it is attenuated at cut-off frequency of the closed-loop system.



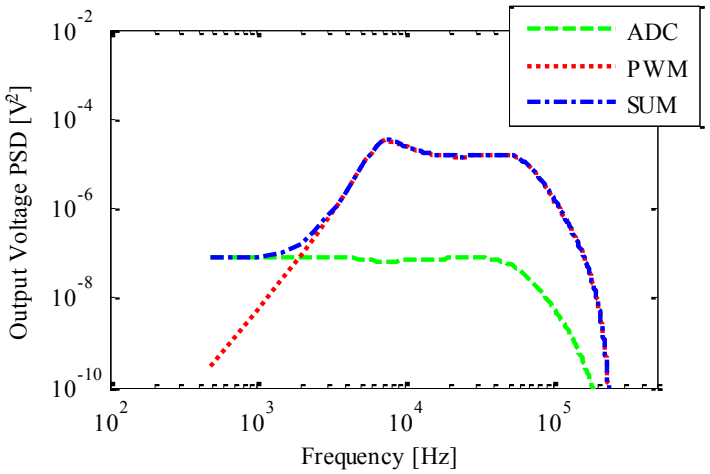
(a) (b)

Figure 3.3 PWM Contribution to Output Noise PSD; (a) With Noise Shaped by 1st – order modulator, (b) With Noise Shaped by 2nd – order modulator

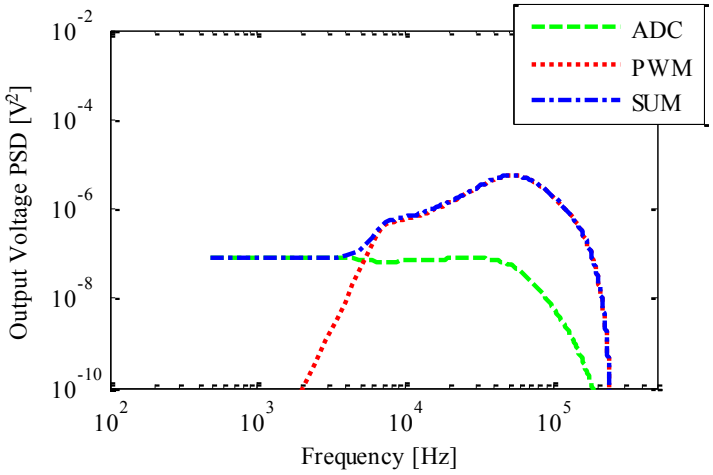
3.3. EXPECTED TOTAL SYSTEM POWER SPECTRAL DENSITY

The expected output PSD of the closed-loop system with the quantizers and the ∑- Δ modulator is shown in Figure 3.4. Figure 3.4 (a) is for 1st -order ∑- Δ modulator and Figure 3.4 (b) is for 2nd -order ∑- Δ modulator. The code used to generate these curves is

given in Appendix A. The blue curve obtained is the sum of the contributions of ADC and the noise shaped PWM.



(a)



(b)

Figure 3.4 Expected Total PSD of a System with *nADC* = 12 and *nPWM* = 3 for (a) 1st – order Modulator, (b) 2nd – order Modulator

The PSD equations (68) and (69) can be used to develop the expected PSD curves for systems with different ADC and PWM bit resolutions. By changing the values of ADC

resolution, DPWM resolution, and the compensator in the m-file, curves for various combinations of the study variables can be generated. These curves can then be used as guidelines and can be compared with the simulation and experimental analysis results to validate the theoretical analysis. In Section 4 the system is simulated and the simulation results are compared with the expected curves. In Section 5, the experimental results are compared with the simulation and expected curves.

4. SIMULATION STUDIES

4.1. INTRODUCTION

In this section, system-level simulation model using Simulink and circuit-level simulation model using PLECS are developed and analyzed. The results obtained are used to validate the theoretical analysis and to provide a more practical basis of comparison for the experimental analysis. First, the model is simulated with a word length of 64-bits out of which 32 are fractional bits. This word length has almost infinite precision and is expected to match precisely with the expected curves developed in Section 3. An attempt has been made to explain any deviations. Next, the model is simulated using PLECS for the plant and Simulink for outer control loop to imitate the real-time model more closely. The word length used in Simulink is equal to the word length used in the experiments. An attempt has been made to explain the differences among the models, if any.

Preliminary simulation studies were conducted by Dr. Jonathan Kimball and the system-level simulation models presented in this chapter have been adopted from those studies. The finite precision model and its analysis is an extension of the same model.

4.2. SYSTEM-LEVEL SIMULATION USING SIMULINK

4.2.1. Block Diagram and Model Description. The simulation model used for system-level simulation is presented in its block diagram format in Figure 4.2. The model represents digital voltage-mode control of the synchronous buck converter. The converter is modeled using discrete state-space. The reference voltage is 1V. The buck converter is operated with a 5V input at a switching frequency of 500 kHz with the duty controlled at

around 20% to obtain the desired 1V output. These parameters are specified in the m-file given in Appendix A and the state-space coefficient matrices are calculated which are specified as input arguments. The system is simulated for 500ms with a sampling frequency equal to the switching frequency of 500 kHz. A variable-step solver is used to allow Simulink to optimize the simulation time. A set up file, which contains the parameter values required by each block is used to allow for ease of modifications and parameter value changes for different cases. The set-up m-file is given in Appendix A. Similarly, another m-file is used to process the results collected during the simulation. The simulation and post-processing results are valid only after the simulation is complete. The m-file used for processing the results is given in Appendix A. This file calculates and plots the PSD using the welch method. Each block constitutes the sub-system of the various modules used in the digital control of the synchronous buck converter. The contents of each block are discussed in the following section.

4.2.1.1 PID compensator. The PID compensator is implemented in its Controller Canonical Form (CCF) as shown in Figure 4.1. The compensators F1 and F2 designed in section 2 are rewritten in Z^{-1} notation to allow the use of unit delay blocks and gain blocks to design the compensator in Z-domain. An additional delay block is added to model the actuation delay in real-time control systems. The input to the compensator block is the error between the reference voltage and the actual output voltage. The output control signal represents the converter duty signal in this system. Also, as the duty signal is bounded between 0 and 100%, the compensator output is bounded between 0 and 1 with the help of the saturation block. The output signal is sent to the ΣΔ modulator block for pre-processing.

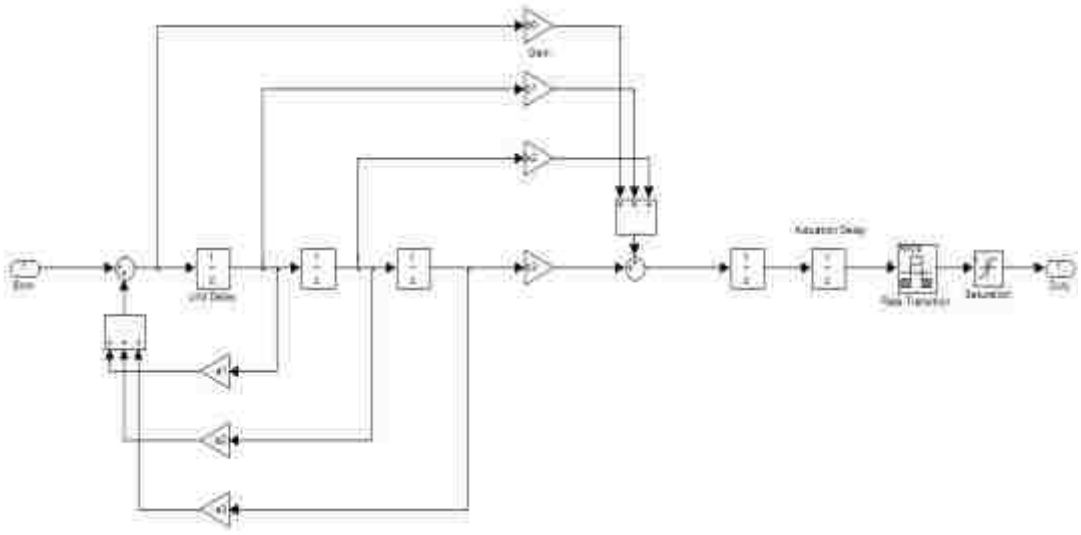


Figure 4.1 PID in Controller Canonical Form

4.2.1.2. Σ-Δ modulator. The Σ-Δ modulator is implemented in its error feedback form. In this form only the LSB bits of the duty which represent the error are fed back to be accumulated and processed thereby reducing the number of bits required in real-time. The models for the 1st order and 2nd order Σ-Δ modulator are shown in Figure 4.3. The feedback loop with the delay and shift block models the noise shaping transfer function of the modulator. The saturation block limits the duty between 0 and 1. The input is the high resolution duty signal of the compensator and output is the pre-processed low resolution duty to the PWM.

4.2.1.3. Discrete state space block. The synchronous buck converter is represented as a State Space model in the discrete domain. The block requires only the coefficient matrices of the state and the output equations as input parameters. The control-to-output transfer function is converted to a state space model using MATLAB command in the set-up m-file.

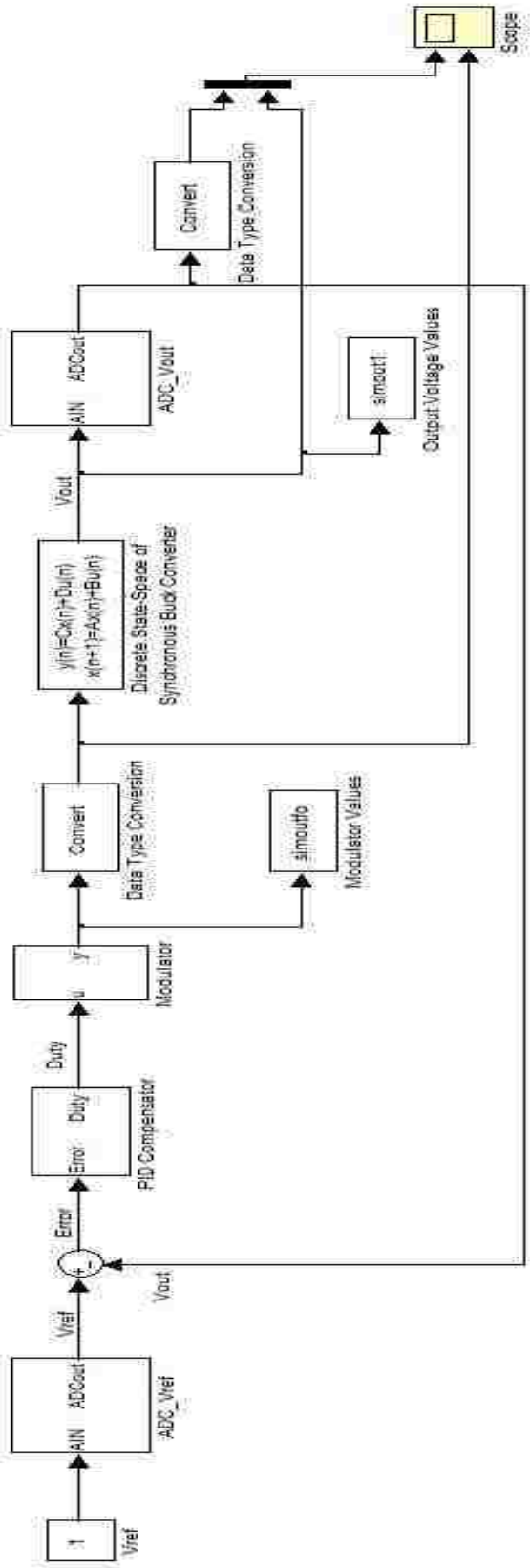
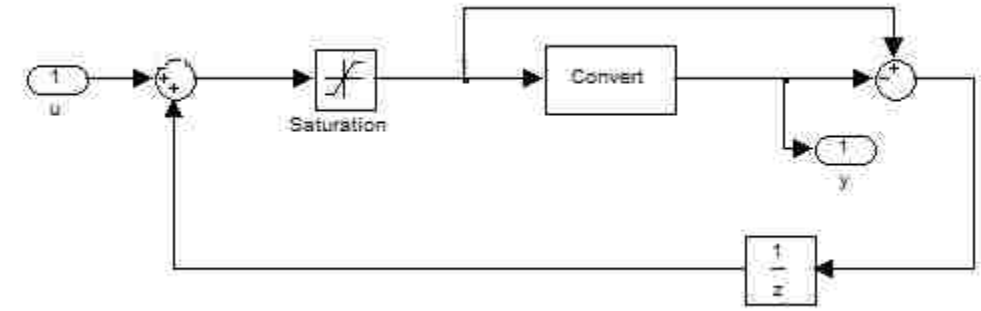
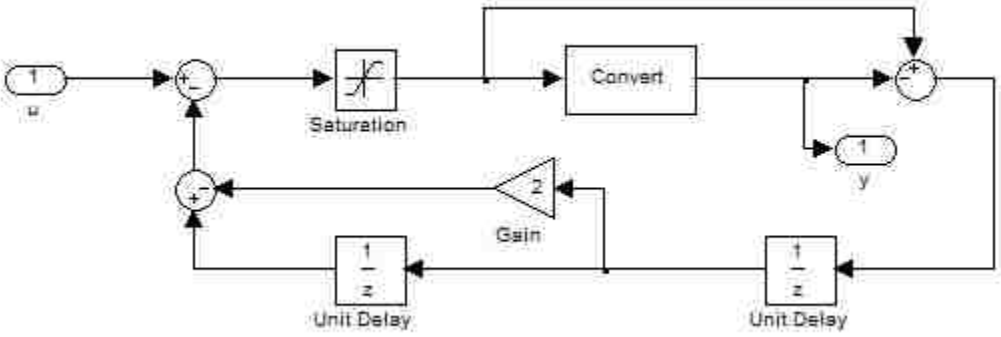


Figure 4.2 Block Diagram of Simulink Model



(a)



(b)

Figure 4.3 Error Feedback Implementation of Σ-Δ Modulator; (a) 1st-order Σ-Δ Modulator, (b) 2nd-order Σ-Δ Modulator

The coefficient matrices are then determined using another MATLAB command and the variables used to represent these coefficient matrices are then entered as input arguments inside the discrete block.The input signal is the duty from the modulator. The output is the actual output voltage which is the parameter of interest. To regulate the output, it is feedback via an ADC.

4.2.1.4. Analog-to-Digital Converter (ADC). The ADC is implemented using a quantizer and a Zero Order Hold (ZOH) as shown in Figure 4.4. The reference value for the ADC is 3.3V. The number of bit of ADC or bit-resolution depends on the case under

study and the value is assigned to the block from the set-up m-file. The ZOH block has a sample time of 2e-6. The ADC introduces a gain in the loop which is modeled using a gain block and its value is equal to the LSB value (49). The ADC block is used for conversion of reference voltage and for the conversion of the output voltage of the plant.

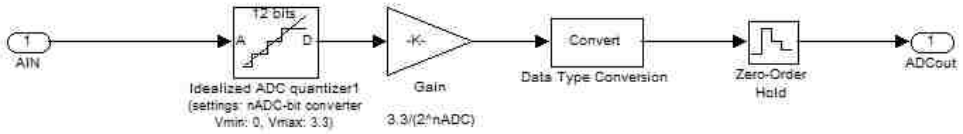


Figure 4.4 Analog-to-Digital Converter Implementation

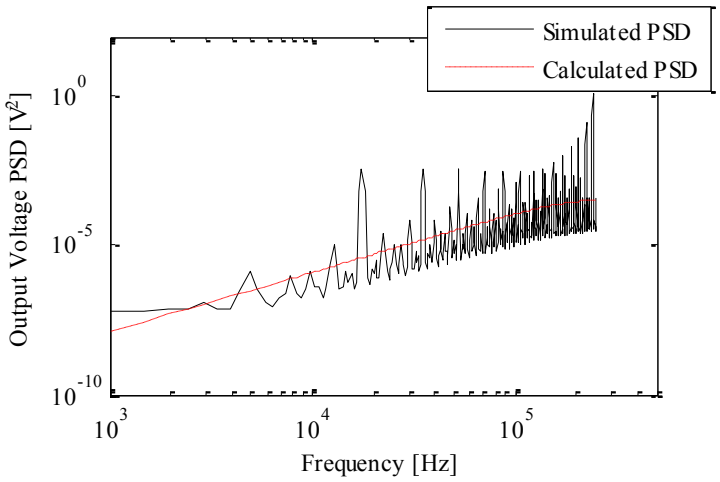
The blocks are integrated to develop the closed-loop model VMC model. To maintain data integrity throughout the loop, data conversion blocks or rate transitions blocks are used. These blocks convert the output values from the preceding block to a value consistent with the input to the next block. The output voltage value from the plant is stored as array in the MATLAB workspace using the simout block. This value is then processed using a post-process m-file to obtain the PSD of output voltage.

4.2.2. Very High Precision Analysis. The simulation analysis is carried out in steps starting with the simple Σ-Δ modulator block and gradually proceeding towards the more complex closed-loop analysis for various design choices with the open-loop plant and Σ-Δ modulator in between. A number of simulations have been performed to explore the impact of various design choices on output noise. Potential variables include: n_{ADC}, n_{PWM}, F, G, and N (1st-order or 2nd-order). For ease of comparison, a uniform scale was used for all plots, as well as a uniform trace legend: black solid for actual PSD, red

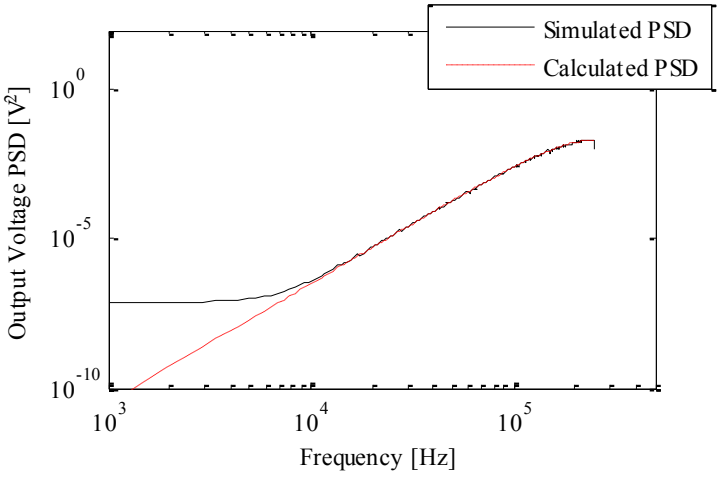
dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum. In each switching cycle, a new ADC value is obtained, used to create a new d value, which in turn creates a new d_{LR} value. The plot of open-loop 1st-order and 2nd-order Σ-Δ modulators is shown in Figure 4.5. The input to the modulators u is a constant block with a duty value of 0.1896973. The input word length for this case is 64 bits with a fractional length of 32 bits. The output word length is 3 bits with a fractional length of 3 bits. This means that the highest output PWM duty value that can be represented by this modulator is 0.875 with a precision of 0.125. It also means that with this modulator, 100 % duty is not achievable. Since the plant is a buck converter and the desired duty is only about 0.2, this loss is of no interest.

In this analysis there is no ADC or DPWM quantization noise source. But round-off occurs at the output end to limit the output resolution to the desired PWM resolution. This introduces an error which is fed back to be accumulated and resolved over a period of time. This loss of resolution due to round-off or truncation can be represented using a white noise model [33] having the same spectral characteristics of the quantization noise. Further, to be able to produce this white noise, the duty should have a fractional part.

The results depict the noise shaping characteristics of the Σ-Δ modulators. The 1st-order Σ-Δ modulator exhibits more deviation from the expected PSD. However, the model is still a reasonable fit. The PSD when using the 2nd-order Σ-Δ modulator almost exactly matches expectations. The 1^{st} –order modulator exhibits idle tones whereas the 2^{nd} – order modulator suppresses the idle tones demonstrating superior performance than the 1^{st} – order modulator as expected.



(a)

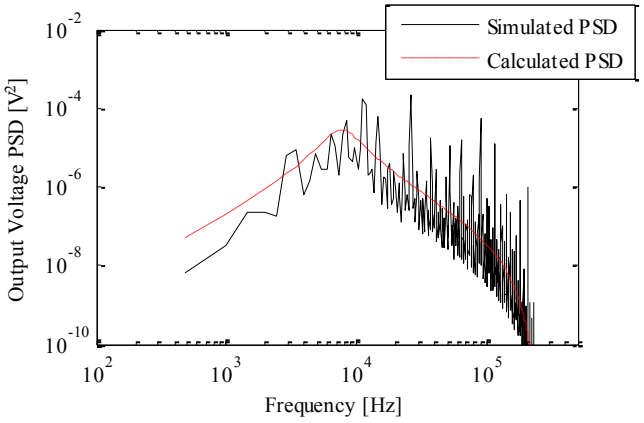


(b)

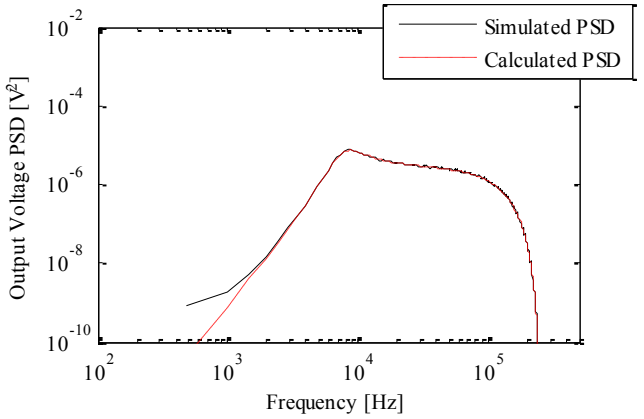
Figure 4.5 Simulated PSD of Open-Loop Σ-Δ Modulator for a Duty of 0.1896973 for 64-bit word length; (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

Next, the plant is introduced in the open-loop simulation and the PSD of the output voltage is plotted to understand the effect of the attenuation by the filter components of the plant. A duty of 0.1986973 is used to achieve a mean value of 0.99 V. The data points collected are limited to the last 249826 samples to eliminate the start-up transients' response from the actual response. The Σ-Δ modulator pushes the low

frequency noise to the higher frequency where it is attenuated by the plant's filter as shown in Figure 4.6.



(a)

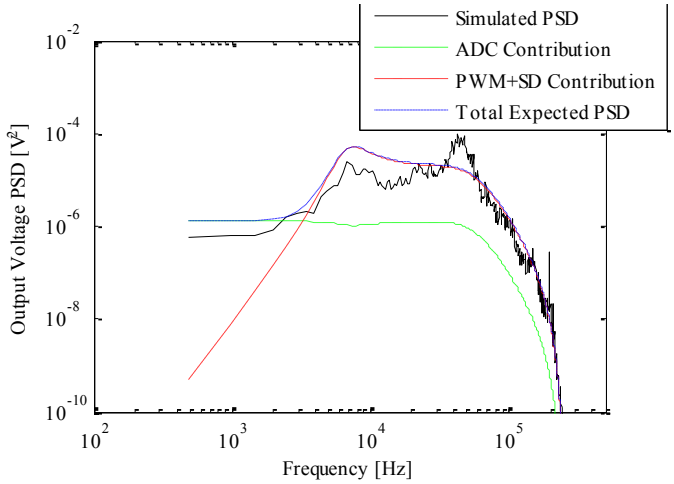


(b)

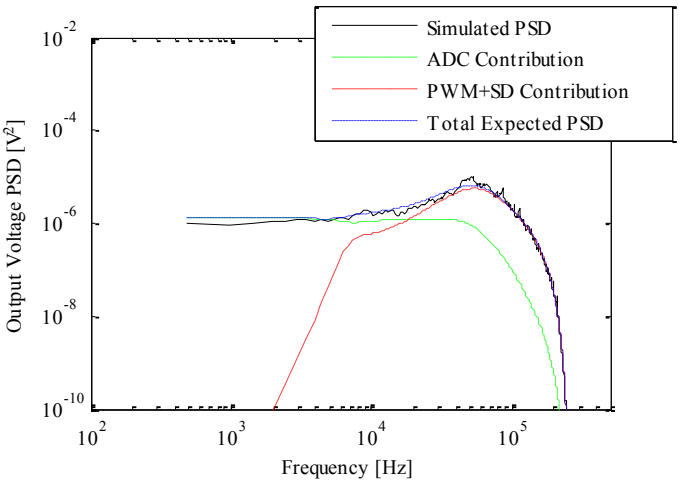
Figure 4.6 Simulink PSD of Plant Output Voltage for Open-Loop Σ-Δ Modulator and Plant for a Duty of 0.1896973 for 64-bit Word Length; (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

Next, the closed-loop simulations are performed for various design choices. The first choice is a 10-bit ADC with a 3-bit PWM and the compensator F1. Figure 4.7 (a) uses a 1st -order Σ-Δ modulator, whereas Figure 4.7 (b) uses a 2nd -order Σ-Δ modulator. The superimposed curves derive from the theoretical analysis of Section 3. The 1st-order

Σ-Δ modulator exhibits more deviation from the expected PSD. However, the model is still a reasonable fit. The PSD when using the 2nd-order Σ-Δ modulator almost exactly matches expectations.



(a)



(b)

Figure 4.7 Simulink PSD of Plant Output Voltage of the Closed-Loop System with 10-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st-order Σ-Δ Modulator (b) 2nd order Σ-Δ Modulator

A few conclusions may be drawn from Figure 4.7. First, the model given by (68) and (69) is a useful approximation. Second, the advantage of a 2nd -order Σ-Δ modulator is not as large as might be expected.

At low frequencies, ADC quantization noise is significant, and at high frequencies, both 1st - and 2nd -order Σ-Δ modulators benefit from the plant and compensator gains.

Figure 4.8 shows the output PSD with the same compensator F1, but with *nADC* = 12 and *nPWM* = 3. In this case because

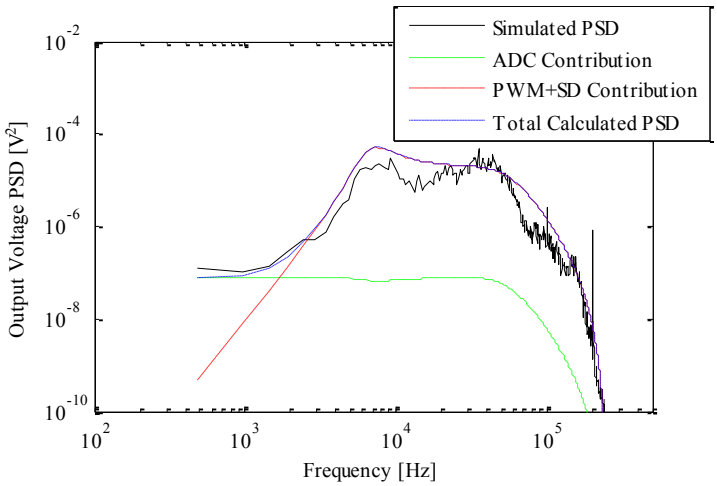
$$\sigma_{ADC}{}^2 < \sigma_{PWM}{}^2 \tag{70}$$

the 2nd-order modulator has a significant advantage over the 1st-order modulator. PWM noise dominates, so the effect of *N* is important. Figure 4.9 again uses the compensator F1, but with *nADC* = 8 and *nPWM* = 3. Because of the low resolution of the ADC, *k*2 = 0 in (68). Also, there is no quantization noise due to the ADC because the output voltage never crosses the boundary from one bin to another.

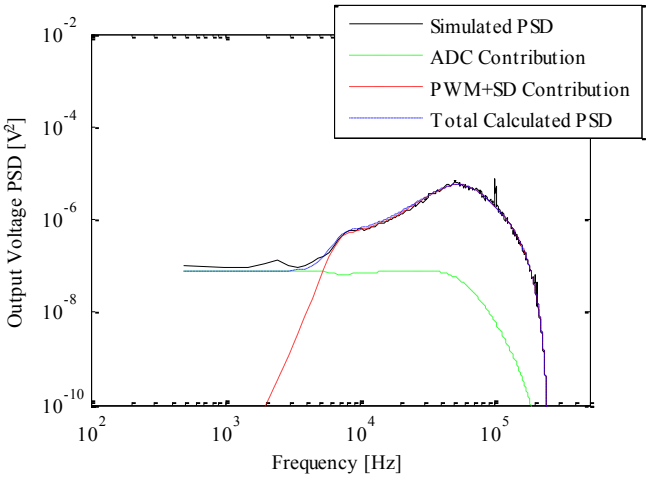
As a result, the total PSD is given by

$$\sigma^2|GN|^2 \tag{71}$$

The lack of feedback in the quantization noise transfer function and the imprecision of the ADC dramatically alter the shape of the PSD.



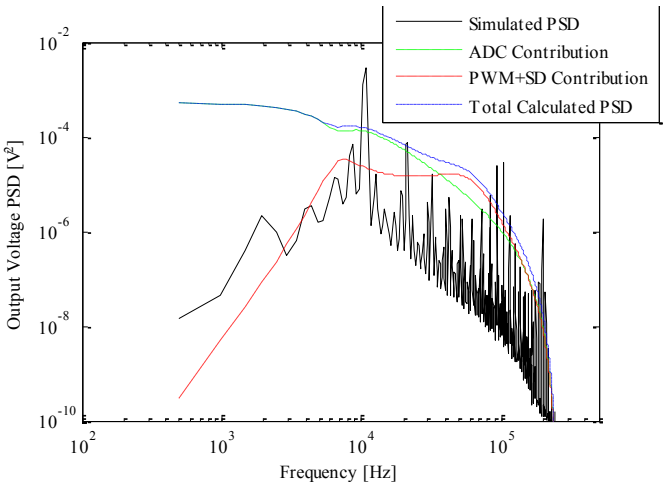
(a)



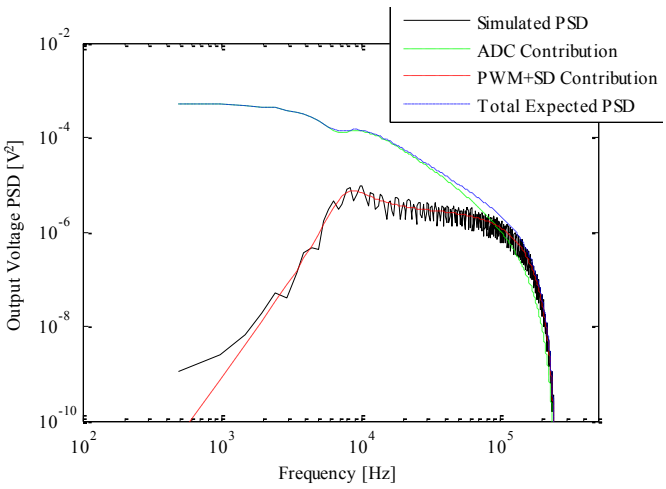
(b)

Figure 4.8 Simulink PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator (b) 2nd order Σ-Δ Modulator

Figure 4.10 shows the output PSD with *nADC* = 12 and *nPWM* = 3 but with compensator F2. This choice significantly reduces the closed-loop bandwidth of the system.



(a)



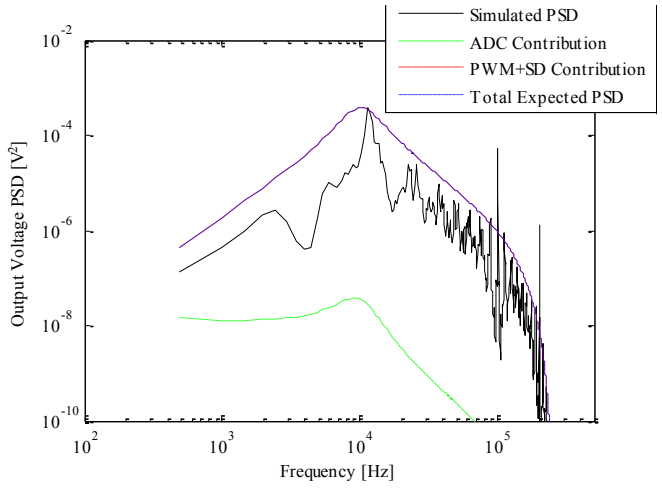
(b)

Figure 4.9 Simulink PSD of Plant Output Voltage of the Closed-Loop System with 8-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator (b) 2nd order Σ-Δ Modulator

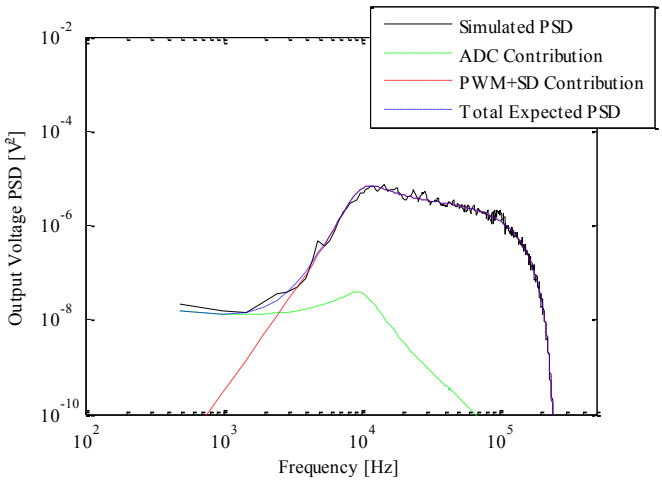
Again, results for both 1st- and 2nd-order Σ-Δ modulators are given. Because of the high ADC resolution and the low bandwidth, the contribution due to PWM quantization and Σ-Δ modulation dominates over a larger frequency range. Despite the significant change in the loop gain, the simulation results match expectations.

The conclusions drawn from the above analysis can be summarized as – For a system with reasonable ADC resolution, the 2nd – order modulator may not be more

advantageous than a 1st – order system except for the presence of weak idle tones at higher frequencies in the 1st – order modulator.



(a)



(b)

Figure 4.10 Simulink PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F2 with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

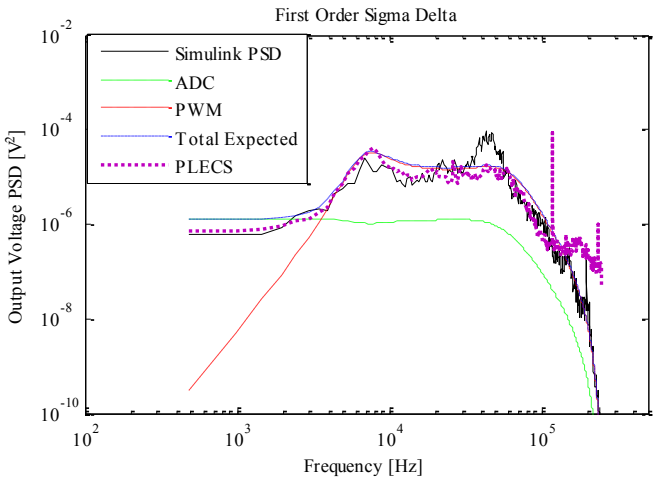
Trade-off has to be made between system cost and complexity and the presence of weak idle tones; for high resolution ADC, the PWM noise dominates so the effect of N is important i.e., the 2nd – order modulator performs better than the 1st – order modulator.

4.3. CIRCUIT-LEVEL SIMULATION USING PLECS

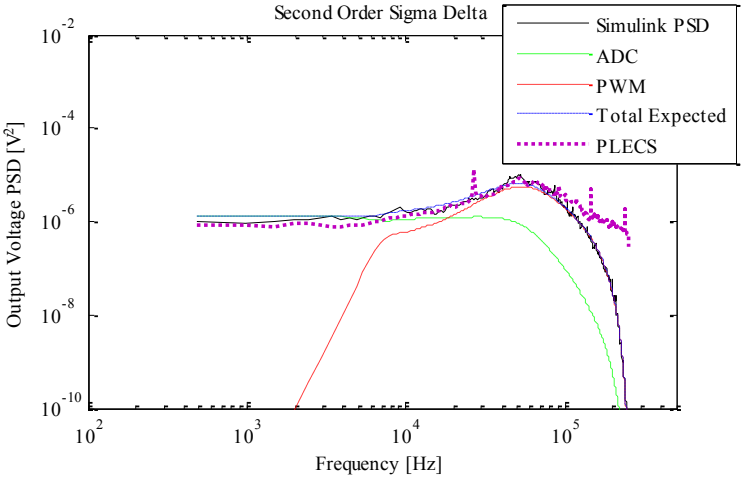
The circuit-level simulation using PLECS is presented in this section. The advantage of using a circuit-level model is that it allows modeling the plant using real-time component parameters. This facilitates the addition of circuit patristic resistances and capacitances found in the real-time system and also facilitates the addition of the input filter components and the actual DPWM switching. Also, dead time can be included between the gate signals. All the parameters used in the system-level simulation are retained in this model. The results obtained are compared with the expected curves and the Simulink curves.

The results are collected using a "To File" block with sample time 2μs. To process the results, an m-file, given in Appendix A, is used which calculates the PSD. To eliminate the response due to transients, the first 250 sample points are deleted. For ease of comparison, a uniform scale was used for all plots, as well as a uniform trace legend: magenta dot for PLECS PSD, black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum. In all the figures below, the Simulink curves are same as those obtained in the previous section and the other curves are same as those obtained in Section 3.

Figure 4.11 shows the output PSD with $nADC = 10$ and $nPWM = 3$ and with compensator F1. Figure 4.11 (a) uses a 1^{st} -order Σ-Δ modulator, whereas Figure 4.11 (b) uses a 2^{nd} -order Σ-Δ modulator. The response matches the expected PSD curve and has a reasonable fit with the Simulink curve. The abrupt cut-off near the plant cut-off frequency might be due to the sampling of the output voltage data. It is expected that the experimental results will demonstrate similar behavior.

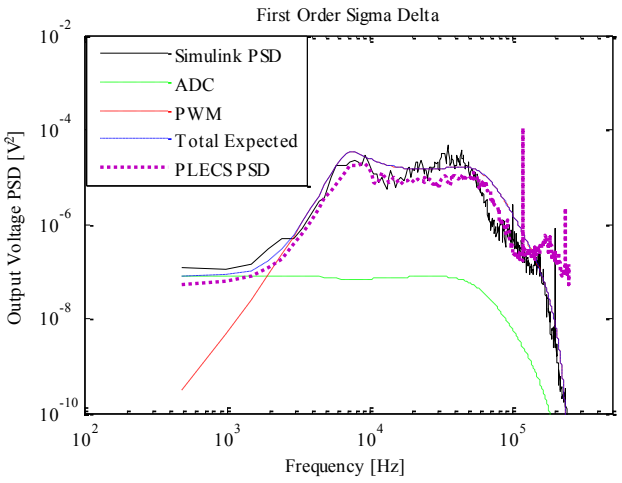


(a)

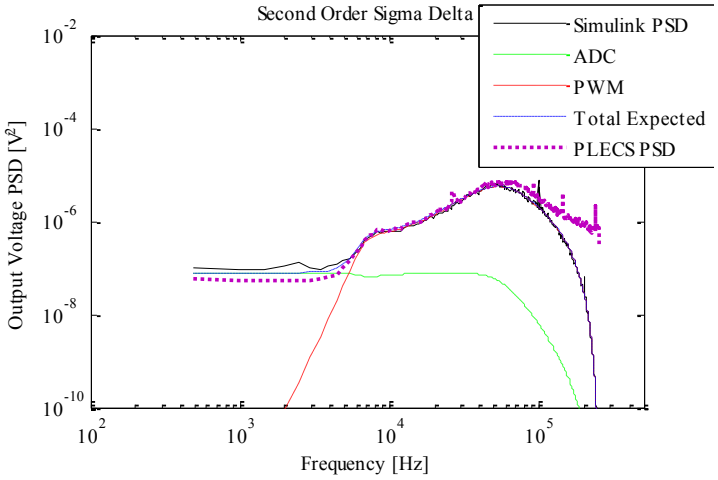


(b)

Figure 4.11 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 10-bit ADC, 3-bit PWM and Compensator F1with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator



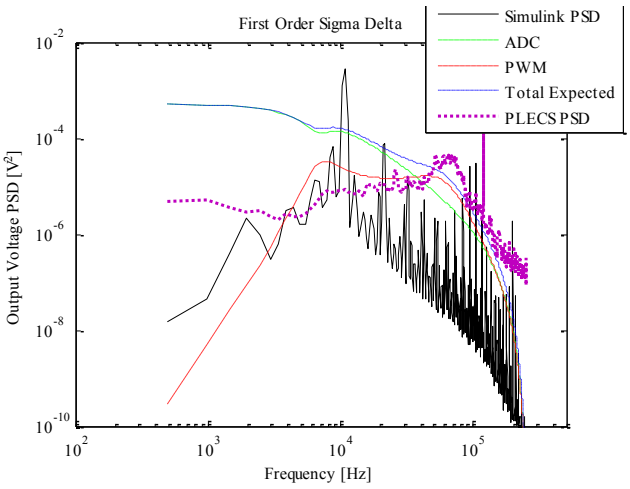
(a)



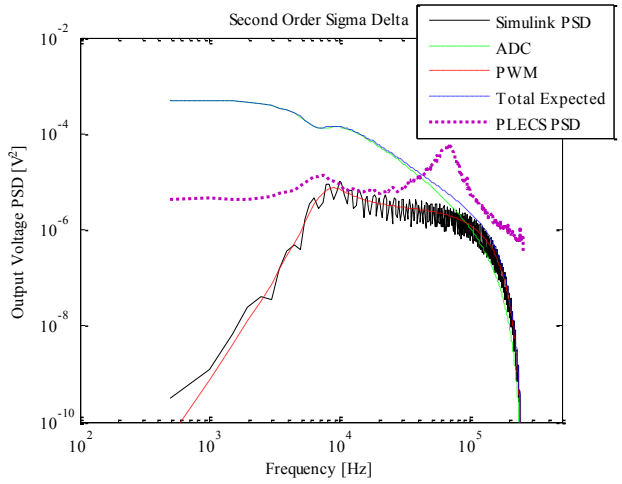
(b)

Figure 4.12 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

Figure 4.13 shows the output PSD with *nADC* = 8 and *nPWM* = 3 and with compensator F1. Figure 4.13 (a) uses a 1st -order Σ-Δ modulator, whereas Figure 4.13 (b) uses a 2nd -order Σ-Δ modulator. Low resolution of ADC alters the shape of the Simulink PSD and PLECS PSD. Also, in PLECS, an ADC has also been used for the reference voltage, as it will be for the experiment.



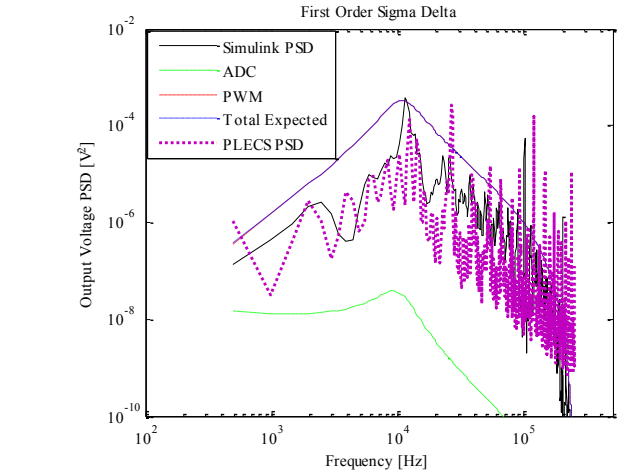
(a)



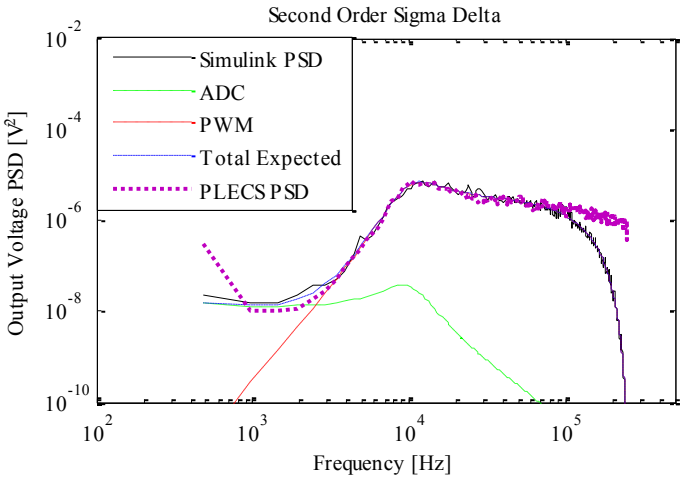
(b)

Figure 4.13 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 8-bit ADC, 3-bit PWM and Compensator F1with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

Figure 4.14 shows the output PSD with *nADC* = 8 and *nPWM* = 3 and with compensator F1. Figure 4.14 (a) uses a 1st -order Σ-Δ modulator, whereas Figure 4.14 (b) uses a 2nd -order Σ-Δ modulator. The low frequency spike like behavior is the response of the transients and can be eliminated by eliminating the transient data points.



(a)



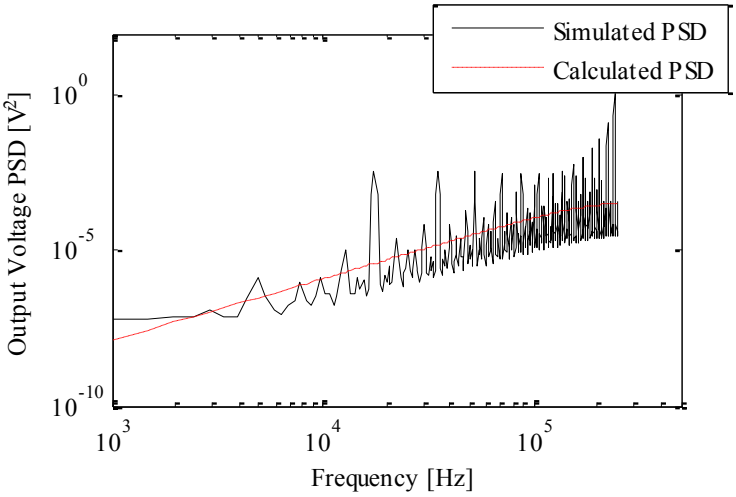
(b)

Figure 4.14 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F2 with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator

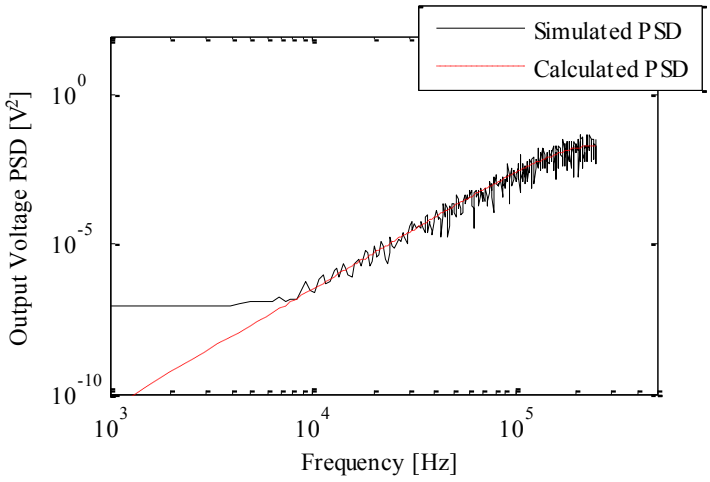
4.4. FINITE PRECISION ANALYSIS WITH PLECS PLANT AND SIMULINK CONTROL LOOP

The resolution of ADC and the clock frequency limit the resolution of a practical system. In this section, the simulation analysis is conducted using a limited resolution system. The reference and input voltages have ADC bits resolution with $n_{ADC} - 3$ bits of precision. The simulation result with limited resolution modulators is shown in

Figure 4.15.The input word length for this case is 12 bits with a fractional length of 9 bits. The output word length is 3 bits with no fractional bits. To be able to get some meaningful output value, the input duty used in the first case is multiplied by 8 to obtain a duty of 1.5175784. To obtain the same mean value as the output of first case, the output voltage is divided by 8. The saturation value in this case is increased to 7.99 which is the maximum value the block can represent for the given word and fractional lengths.



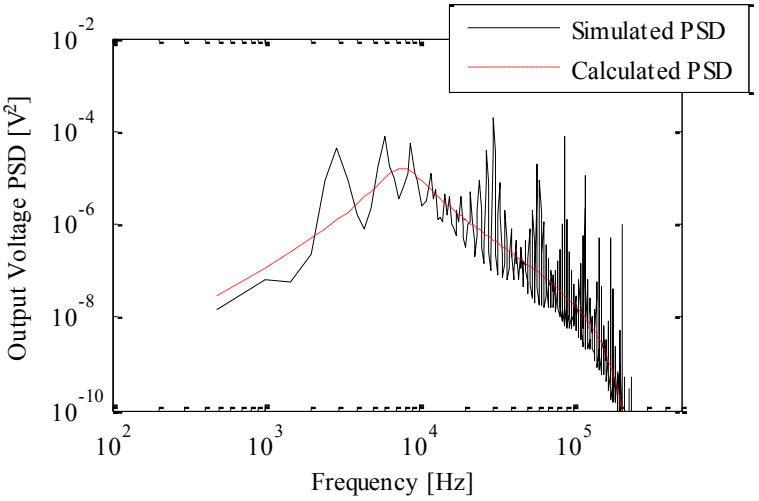
(a)



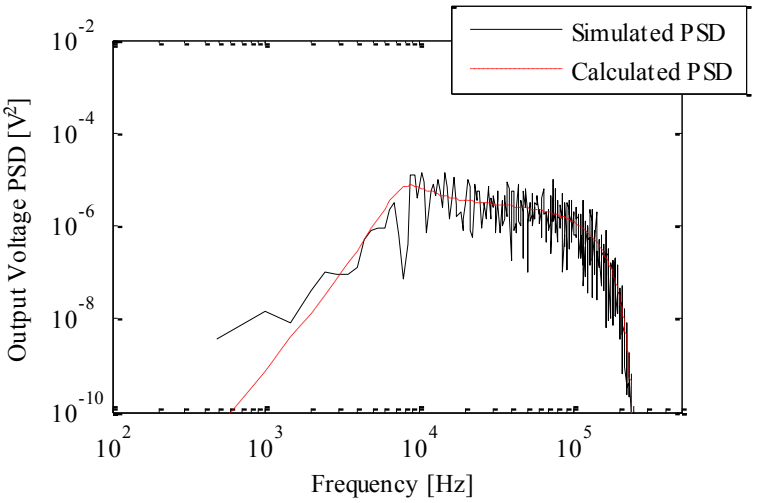
(b)

Figure 4.15 Simulated PSD of Open-Loop Σ-Δ Modulator for a Duty of 0.1896973 for 12-bit Word Length (a) 1st order Σ-Δ Modulator (b) 2nd order Σ-Δ Modulator

In this case, the 2nd – order modulator performs better than the 1st – order modulator but when compared to the high precision performance the modulator exhibits tones in its output. Figure 4.16 shows the PSD of the output voltage of the open-loop converter.



(a)



(b)

Figure 4.16 Simulated PSD of Plant Output Voltage for Open-Loop Σ-Δ Modulator and Plant for a Duty of 0.1896973 for 12-bit Word Length (a) 1st order Σ-Δ Modulator (b) 2nd order Σ-Δ Modulator

When the resolution is limited, the output voltage PSD with 2^{nd} – order modulator resembles the low resolution ADC case analyzed in the previous sections with the modulator exhibiting idle tones. However, the PSD matches the expected PSD.

The simulation model of the closed-loop 1^{st}- order modulator system is shown in Figure 4.17. The Discrete State-Space model is replaced with the PLECS switching circuit. The PLECS block is used as discrete-state space with sample time of 2e-6. The reference voltage and the input voltage are converter using two ADC blocks with n_{ADC} bits word length n_{ADC} – 4 bits precision. The figure shows the word lengths of each block for a 12-bit ADC and 3-bit DPWM resolutions. The results are shown in Figures 4.18 through 4.20. These results will be used as the basis of comparison for the experimental analysis.

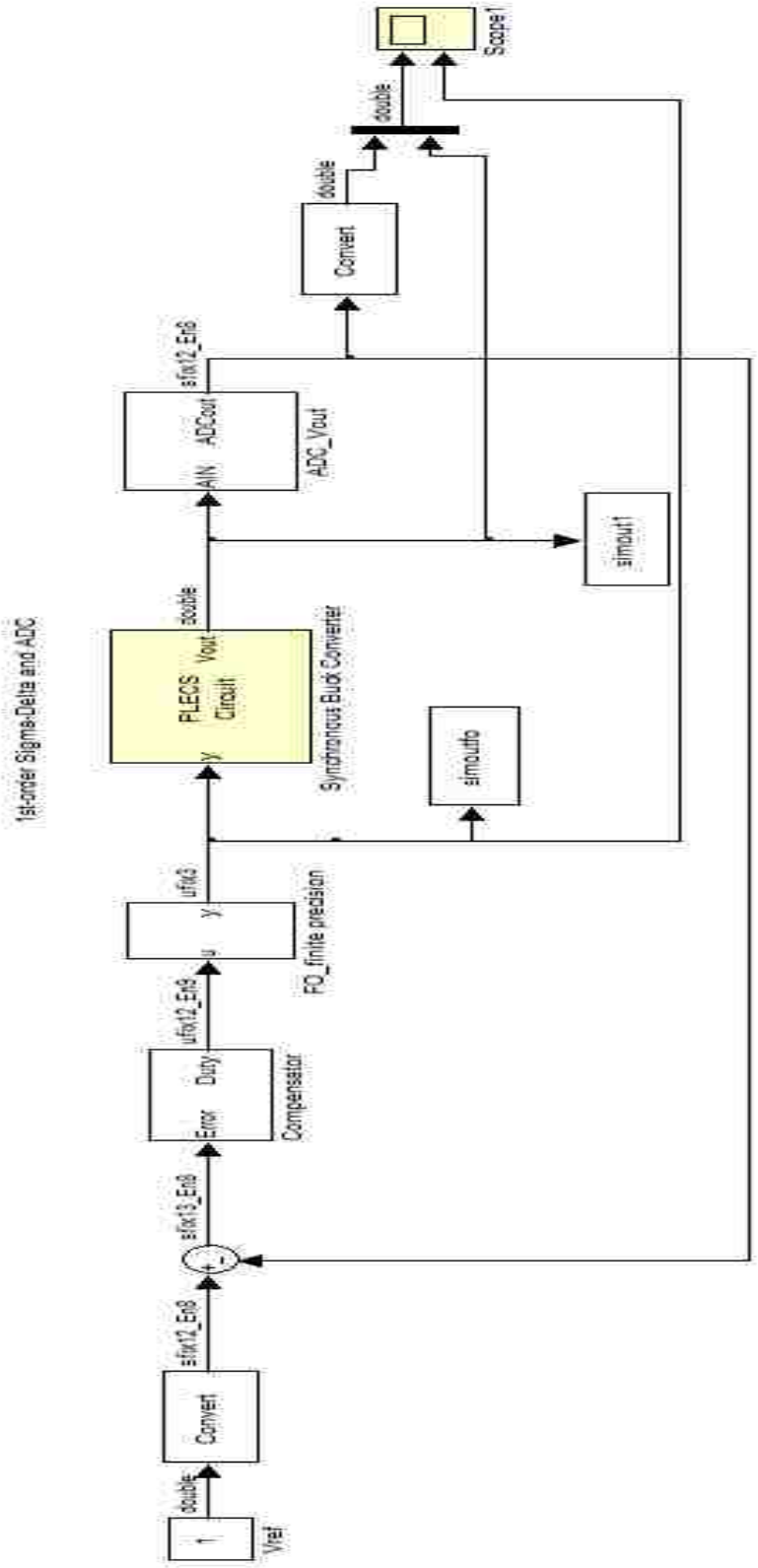
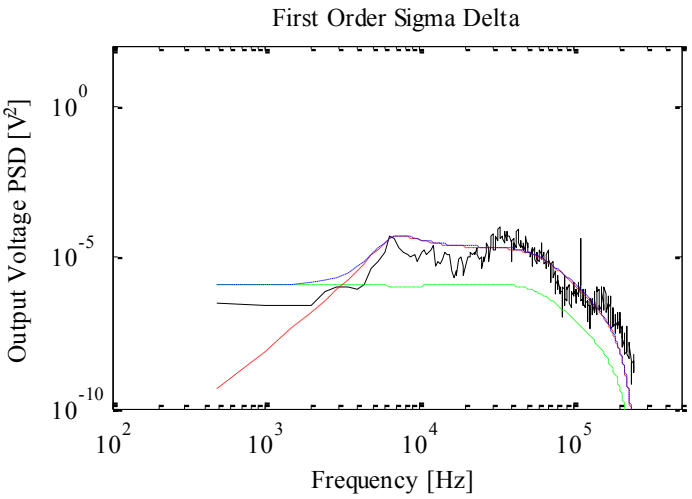
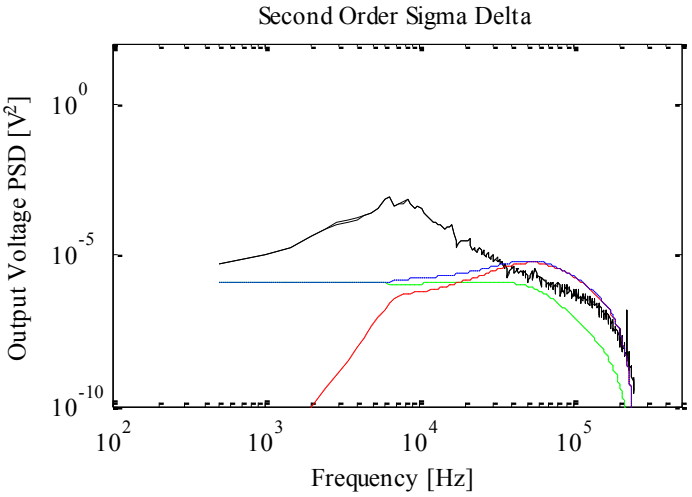


Figure 4.17 Simulation Model for Finite Precision PLECS Model Plant System

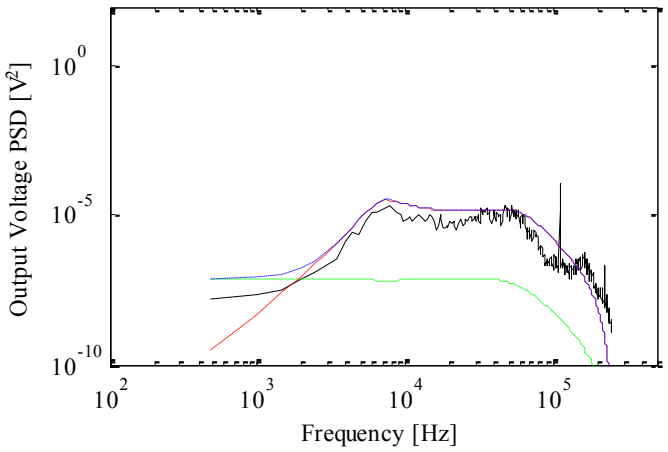


(a)

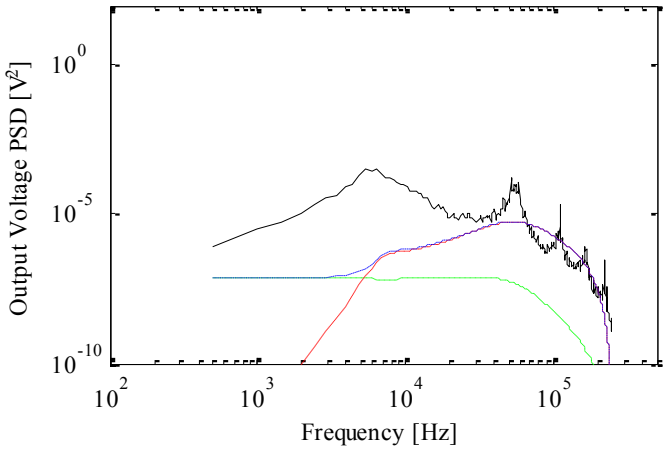


(b)

Figure 4.18 PSD of Plant Output Voltage of the Closed-Loop System with 10-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st -order Σ-Δ Modulator, (b) 2nd-order Σ-Δ Modulator. Trace legend: black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum

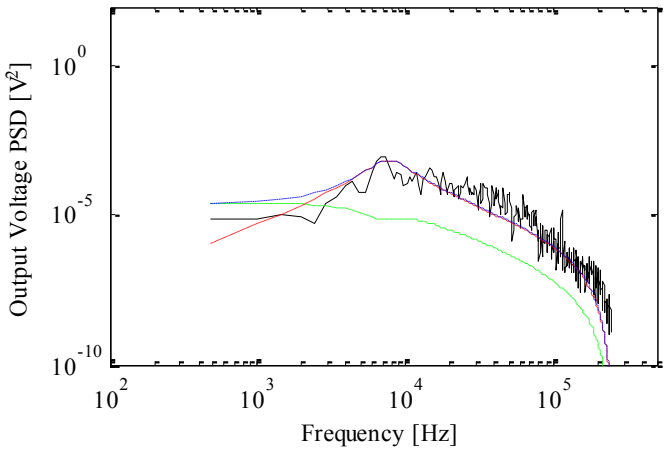


(a)

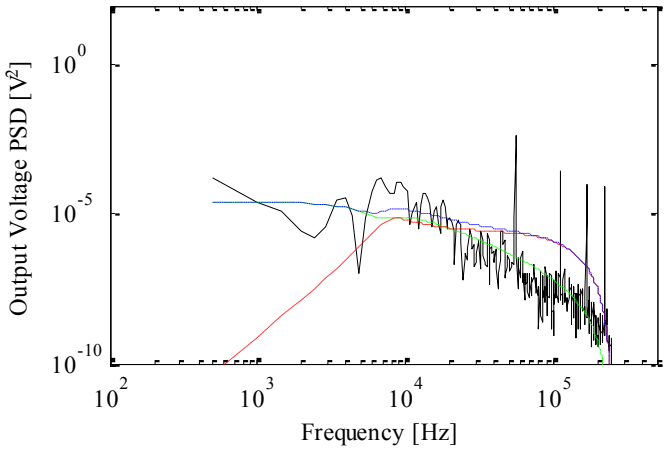


(b)

Figure 4.19 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator. Trace legend: black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum



(a)



(b)

Figure 4.20 PLECS PSD of Plant Output Voltage of the Closed-Loop System with 8-bit ADC, 3-bit PWM and Compensator F1with (a) 1st-order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator. Trace legend: black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum

5. EXPERIMENTAL STUDIES

5.1. INTRODUCTION

This section discusses the experimental analyses conducted to validate the simulation and theoretical results developed in the previous sections. The procedure to set up the test bench is discussed in detail with the hardware set up and connection discussed in subsection 2 and the software programming discussed in subsection 3. The test bench set-up and procedure to check are discussed in subsection 4. The experimental PSD curves obtained for various design choices are presented in subsection 5.

5.2. HARDWARE SETUP

To implement digital control of a synchronous buck converter three Printed Circuit Boards (PCBs) PCB0006 Rev A, PCB0012 Rev A and PCB0027 Rev A are used. PCB0027, as shown in Figure 5.1 is the synchronous buck converter board. The schematic used to design and develop this board is given in Appendix B.

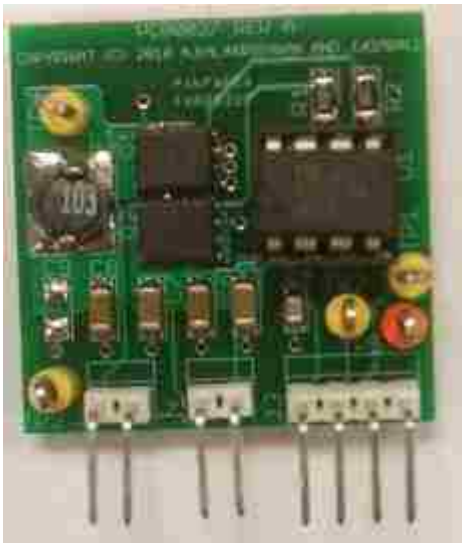


Figure 5.1 Synchronous Buck Converter Power Stage PCB (PCB0027)

The PCB was designed using Eagle[3] PCB Design Software. Some generic components found in the library have been used. For the MOSFETs (Q1 and Q2) and the Inductor (L1) the library component was developed using the pad patterns provided by the manufacturers. The part numbers of the components used along with the part number on the board are listed in Table 5.1. An external load of 1Ω is created by connecting 5 5Ω, 5W resistors in parallel on a bread board.

Table 5.1 List of Components Used in PCB0027 Design

Component Description	Part Number on Board	Component Used
MOSFET(s)	Q1, Q2	SI7160DP
MOSFET DRIVER	U1	MIC4424
Inductor	L1	XPL7030-102ML
Capacitors	C5, C6, C7, C8	JMK316BJ476ML-T
Capacitor	C4	0805YG105ZAT2A
Resistors	R1, R2	CRCW120610R0JNEA
Connectors (2-pin)	J1, J2	A30884-ND
Connector (4-pin)	J3	A30886-ND

This board is seated on another PCB, PCB0012, shown in Figure 5.2. PCB0012 can carry 8 such converter boards and was developed to be used in multi-phase converter experiments.

[3] Eagle is the trademark of CadSoft.

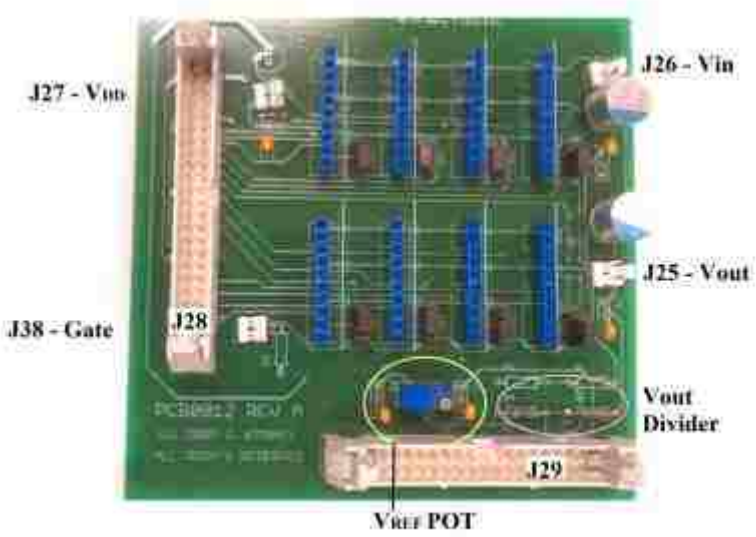


Figure 5.2 Interface Board (PCB0012) with Multi-Converter Operation Capability

On the PCB0012, connector J26 is used to supply the input voltage (5V) to the buck converter. J27 is used to supply the VDD (9V) to the gate driver. J25 carries the V_{out} of the converter and J38 is used to give the gate signal to the converter. When using a digital control system, J29 is used to send the V_{ref} and V_{out} signals to the controller and J28 is used to bring back the gate signals from the controller to the converter. The potentiometer V_{ref} POT is used to set the reference voltage. The V_{out} Divider is used to half the output voltage before passing it onto the digital controller to limit the signal during initial transients. The signal is multiplied by two inside the digital controller to set it back to its original value. A jumper (like J31) is used to route the appropriate gate signal (either analog from J38 or digital from J28) to the converter. A position like that of J31 indicates that the gate signal is from J28. A position like that of J32 indicates that the gate signal is from J38.

PCB0012 with PCB0027 mounted on it is connected to PCB0006 via the connectors J28 and J29. PCB0006 is a multi-purpose digital control board with a number of digital I/O ports and a variety of peripheral units as shown in Figure 5.3.

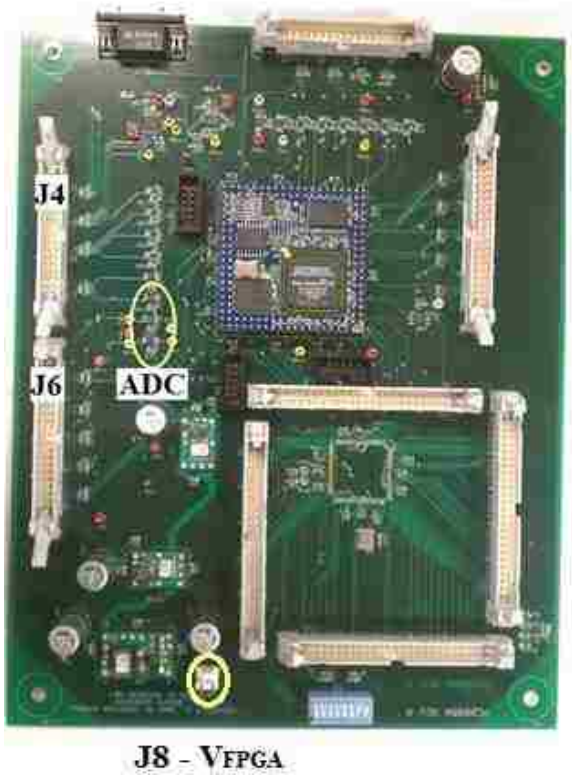


Figure 5.3 Digital Control Board PCB0006

In this experiment, only the digital I/O ports J4 and J6 are used. J4 is connected to J29 on PCB0012 via a data bus. J6 is connected to J28 on PCB0012 via another data bus. AD7276, a 12-bit ADC, is used to convert the Vref and Vout from PCB0012 to digital values which is then passed on to the FPGA Altera Cyclone II for further processing. Connector J8 is used to supply V_{FPGA} of 9V. The three boards are connected together as shown in Figure 5.4.

Figure 5.4 Connecting PCB0006, PCB0012 and PCB0027

5.3. FPGA PROGRAMMING

A Field Programmable Gate Array (FPGA) is implemented as a control and gate drive module. The digital compensator, ΣΔ modulator and the DPWM are developed using VHDL and programmed into the FPGA using Quartus II [4] design software. The VHDL codes for the digital compensator, 1st – order ΣΔ modulator, 2nd – order ΣΔ modulator and DPWM are given in Appendix C. The Quartus program developed for a 1st-order modulator is shown in Figure 5.5.

The PLL block is used to generate the clock and reset signals. A crystal oscillator which has a maximum frequency of 24MHz supplies the input to the PLL block. To maintain the same clock frequency at the output, a ratio of 1/1 is used. The duty ratio of the clock signal is 50%. The clock signal is used to synchronize the system. The locked signal from the PLL is used as the RESET signal which is used to reset the system to its initial values. The reset signal is active when it is low.

---

[4] Quartus II is a trademark of Altera.

The Start Pulse block is used to trigger the circuit in to operation mode when the clock signal is available. The inputs to this block are the clock signal, reset signal and a constant pulse width vector of length 4 bits. The constant signal is fed from another block called constant_sigdelta and its value is 0001. For a pulse width value of *n* the block outputs a pulse signal at every n^{th} clock pulse. This pulse is transmitted throughout the circuit in a serial fashion to ensure that one block completes its computation before passing on the data to the other block. Upon completion of one cycle, the done pulse from the final block in the series is 'OR'ed with the start pulse to be used as the start pulse in subsequent cycles. The start pulse frequency which is also equal to the done pulse is 500 kHz.

The mst_AD7276 block functions as the device driver for ADC AD7276. This block receives the analog-to-digital converted data from AD7276 as single bits through iSDO, strips the leading zeros to convert a 14-bit data into a 12-bit data and sends it out through oValOut. oSCK represents the serial clock and oLD is the chip select. The iClkdiv input is a constant 4-bit vector can be used to change the frequency at which this block operates. The constant value is fed using the constant_sigdelta block and its value is equal to 0001. Two such device driver blocks are used for Vref and Vout values of the buck converter.

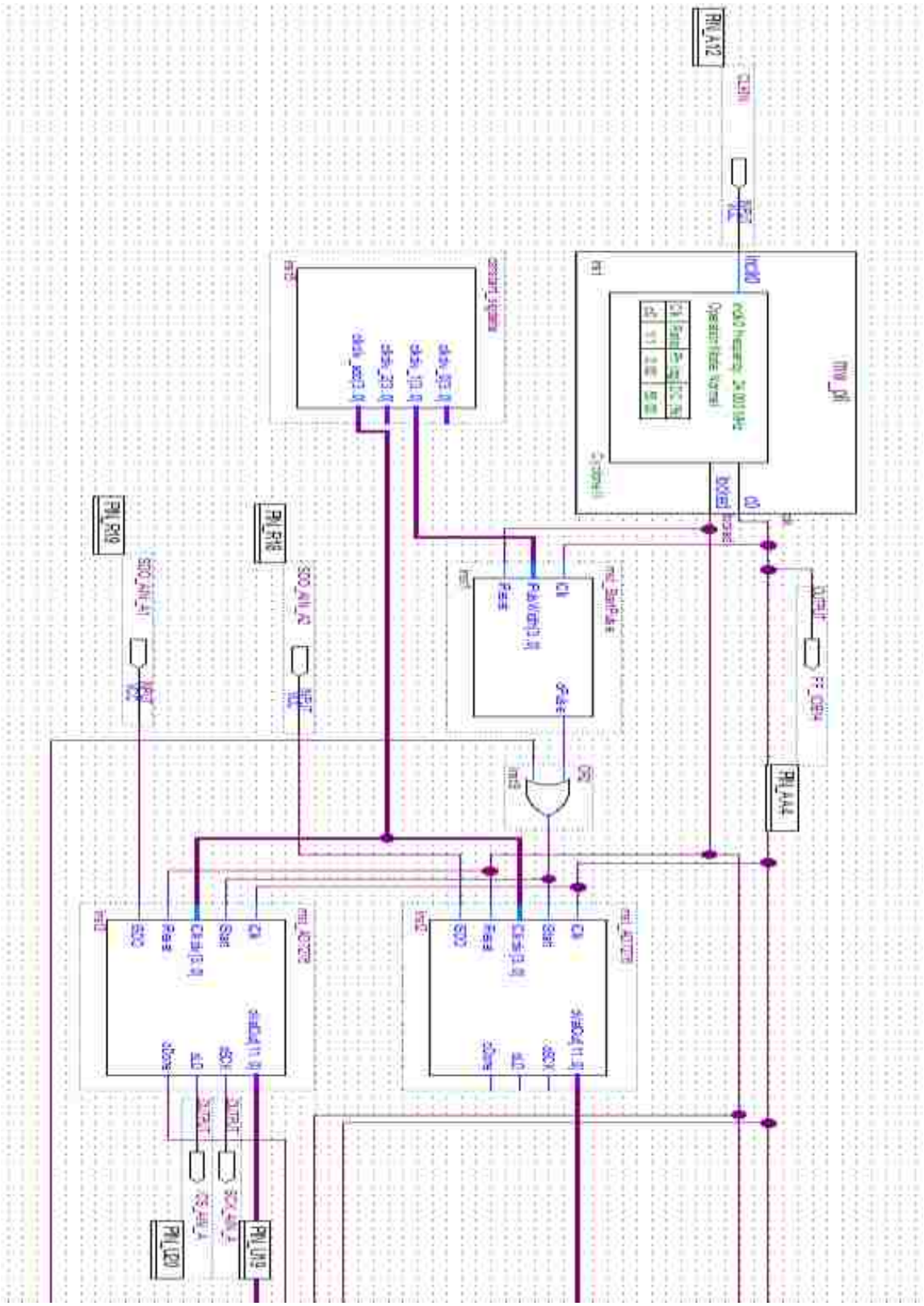


Figure 5.5 Quartus Program for 1st – Order ΣΔ Modulator, 12-bit ADC and 3-bit PWM

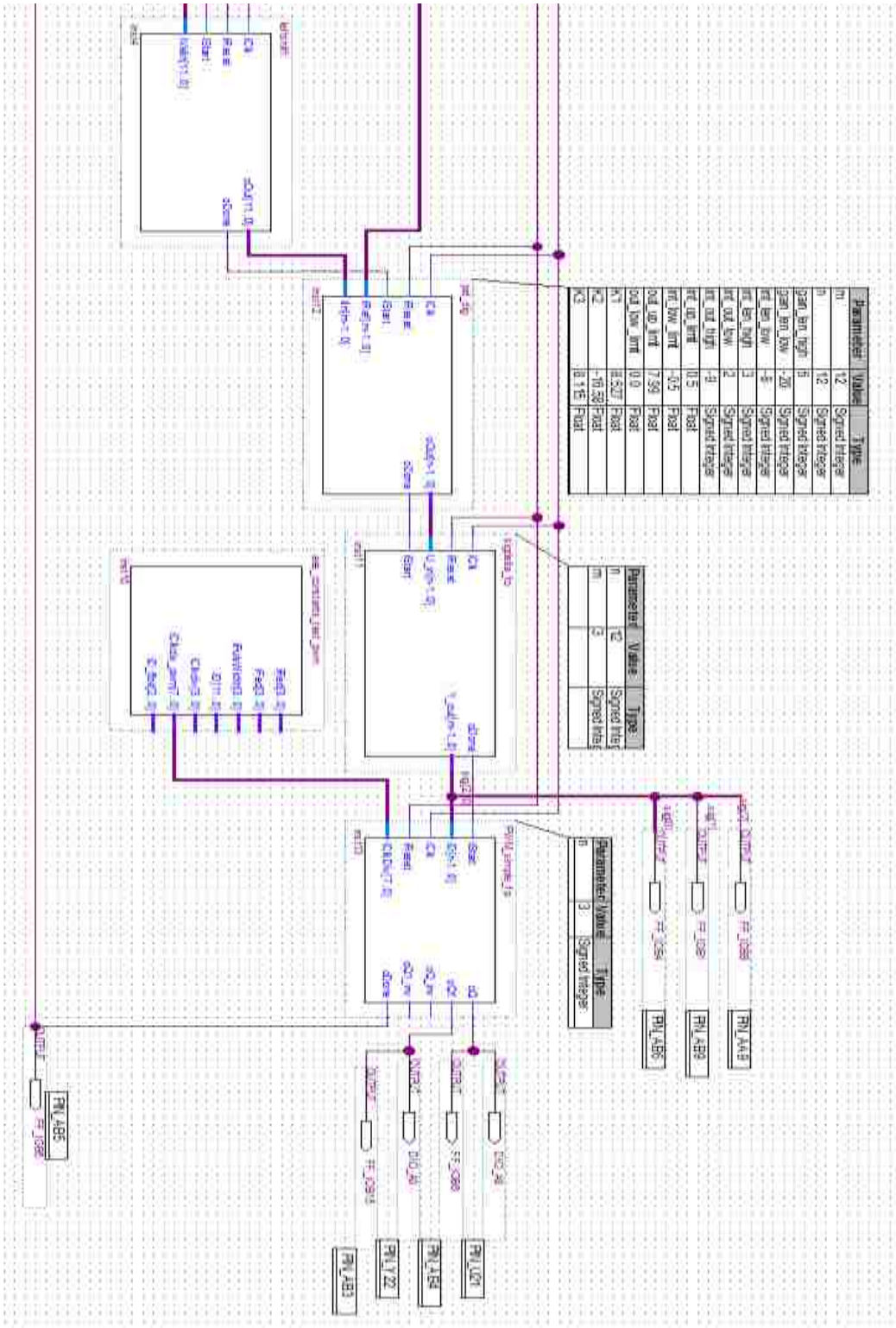


Figure 5.5 Quartus Program for 1st – Order Σ-Δ Modulator, 12-bit ADC and 3-bit PWM (cont.)

The leftshift block is used to multiply the Vout signal from PCB0012 by 2. The binary equivalent of an arithmetic multiplication by 2 is a left shift by 1 bit. So, this block receives the 12-bit digital equivalent of Vout/2 and left shifts it by 1 bit to make it equivalent to Vout. The code for this block is given in Appendix C.

Pid_dig block is the digital PID compensator implemented in the Controller Cannonical Form (CCF). It receives the 12-bit reference and output voltage values from the two ADC device drivers block, calculates the error and computes the duty signal and sends it out as a 12-bit vector value. The gains are defined as constants inside the block. The code for this block is given in Appendix C.

The sigdelta_fo and sigdelta_so are the 1^{st} – order and 2^{nd} – order ΣΔ modulators implemented in the error feedback form as discussed in Chapter 2. The block receives the high resolution (12-bit) vector duty signal from the PID block and converts it into a low resolution 3-bit vector signal. The error due to this conversion is accumulated and averaged over a period of time to achieve high effective resolution. The code for these block are given in Appendix C.

The PWM_simple block is a simple version of the digital PWM implementation which uses an up-down counter and the input value to generate four pulse signals at the output. The signals oQ and oQ1 are used as the gate signals for the buck converter which is sent to the digital I/O port J6 on PCB0012.

5.4. TEST BENCH

5.4.1. Test Bench Set-up. The test bench set up is shown in Figure 5.6. The buck converter (PCB0027) is seated on PCB0012 at J1, J2 and J3. The input (5V) and gate (9V) power supplies to the buck converter are supplied using two switching

mode power supplies. The output terminal on PCB0012 is connected across an external load of 1Ω.

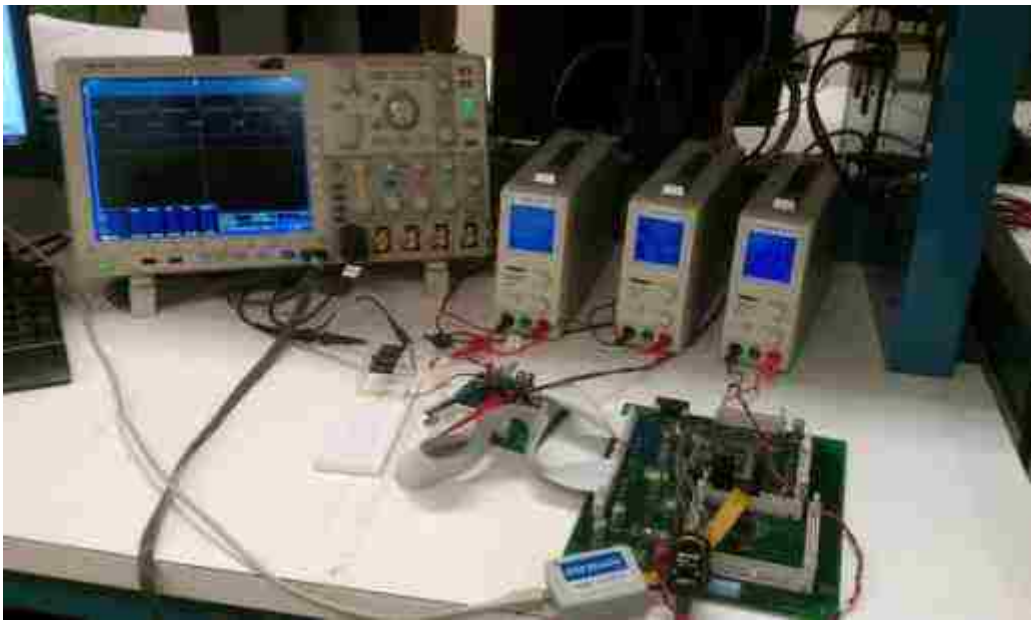


Figure 5.6 Test Bench Setup

The program developed using Quartus II is programmed into the FPGA via the USB Blaster hardware device. The output voltage and the digital signals are captured using a mixed signal oscilloscope. To achieve best results the scope is set to 40.0ms, 2.5MS/s with 1M record length. The timing resolution for digital signals is 400ns. For the analog signal 10mV/div AC Coupling is used. The acquire mode used is Hi Res.

5.4.2. Procedure to Check. To enable programming the FPGA ensure that the FPGA power supply is turned On. Check for successful completion of programming on the Quartus Programmer window. However, while conducting closed-loop tests the gate supply of the converter should be turned on first before programming the FPGA. The input voltage to the buck converter should be turned on after the programming is

complete and the transients have settled down. A minimum of 120 ns dead-time should be maintained between the two gate signals of the converter. The dead time used is illustrated in Figure 5.7. Channel 0 carries the gate signal for the primary switch and Channel 1 carries the gate signal for the secondary switch. Channel 2 is the done pulse and Channel 3 is the clock signal.

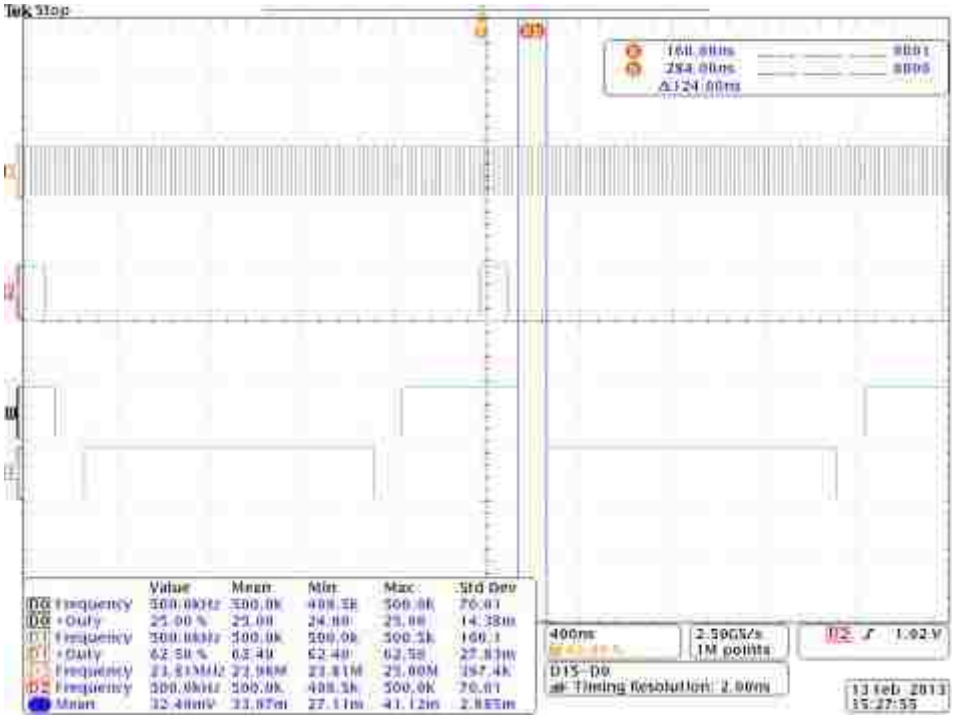


Figure 5.7 Dead Time of 124ns between the Two Gate Signals

5.5. RESULTS AND DISCUSSION

The experimental analysis is carried out in steps starting with the simple Σ-Δ modulator block and gradually proceeding towards the more complex closed-loop analysis for various design choices with the open-loop plant and Σ-Δ modulator in between. A number of experiments have been performed to explore the impact of various

design choices on output noise. For ease of comparison, a uniform scale was used for all plots, as well as a uniform trace legend: red solid for experimental PSD, black solid for finite precision analysis PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum.

Figure 5.8 shows the scope shot for with 1st – order modulator. Channel 0 carries the gate signal for the primary switch and Channel 1 carries the gate signal for the secondary switch. Channel 2 is the done pulse and Channel 3 is the clock signal. The bus B2 is the 3-bit 1st– order modulator output.

The output PSD plot of open-loop 1st-order Σ-Δ modulator is shown in Figure 5.9 (a). The input to the modulator *u* is a constant block with a duty value of 0.1896973. Figure 5.9 (b) shows the output voltage PSD plot of open-loop 1st-order Σ-Δ modulator with plant for the same duty value.

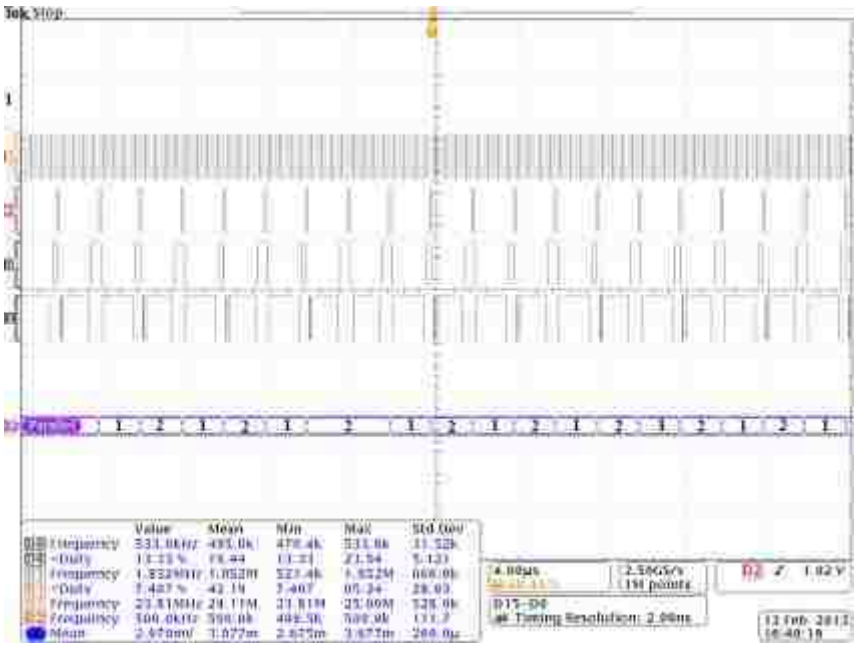
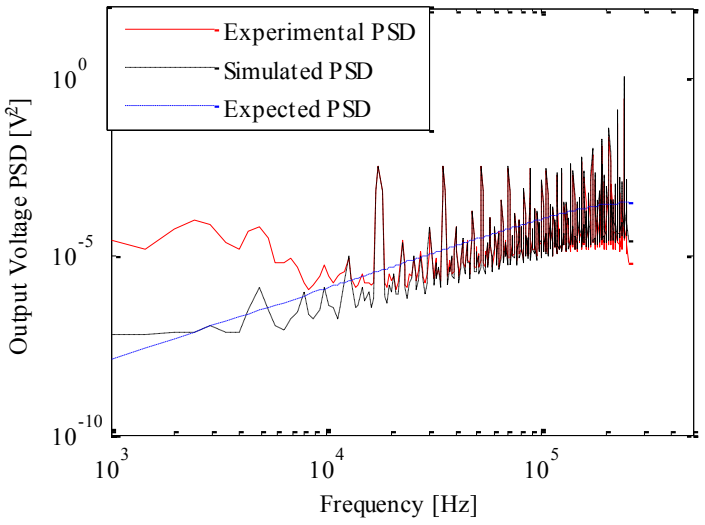
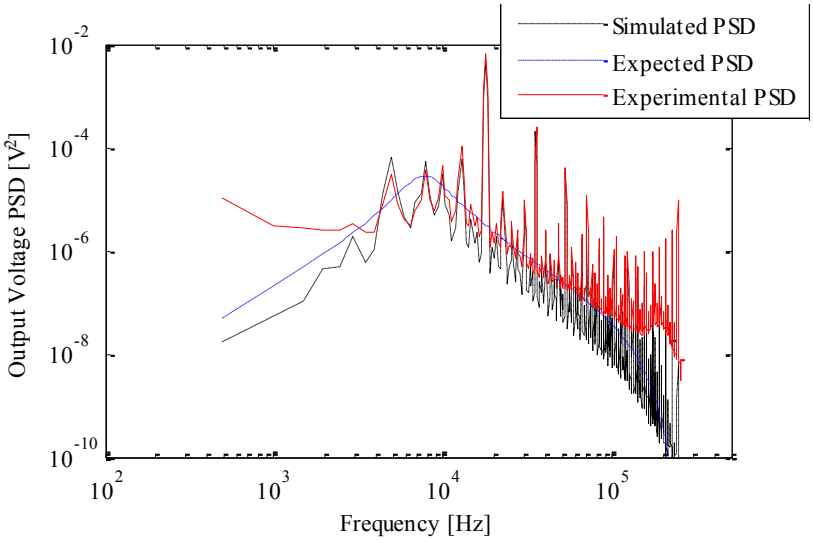


Figure 5.8 Scope Shot of the 3-bit 1st – order Σ-Δ Modulator Bus Depicting Variations between Two Levels for a Given Input



(a)



(b)

Figure 5.9 Experimental PSD for a 1st- order Σ-Δ Modulator with Duty Input of 1.5175784 for (a) Modulator Output (b) Plant Output Voltage

The experimental results obtained for the modulator output and plant output match the simulation results. This shows that the 1st – order modulator module developed

using VHDL is accurate. The strong spike at 1.7e4 is due to the limit cycle in the output waveform with a period of 5.8e-5s.

Similarly, Figure 5.10 shows the scope shot for with 2^{nd} – order modulator. Channel 0 carries the gate signal for the primary switch and Channel 1 carries the gate signal for the secondary switch. Channel 2 is the done pulse and Channel 3 is the clock signal. The bus B2 is the 3-bit 2^{nd} – order modulator output.

The output PSD plot of open-loop 2^{nd}-order Σ-Δ modulator is shown in Figure 5.11 (a). The input to the modulator *u* is a constant block with a duty value of 0.1896973. Figure 5.11 (b) shows the output voltage PSD plot of open-loop 2^{nd}-order Σ-Δ modulator with plant for the same duty value.

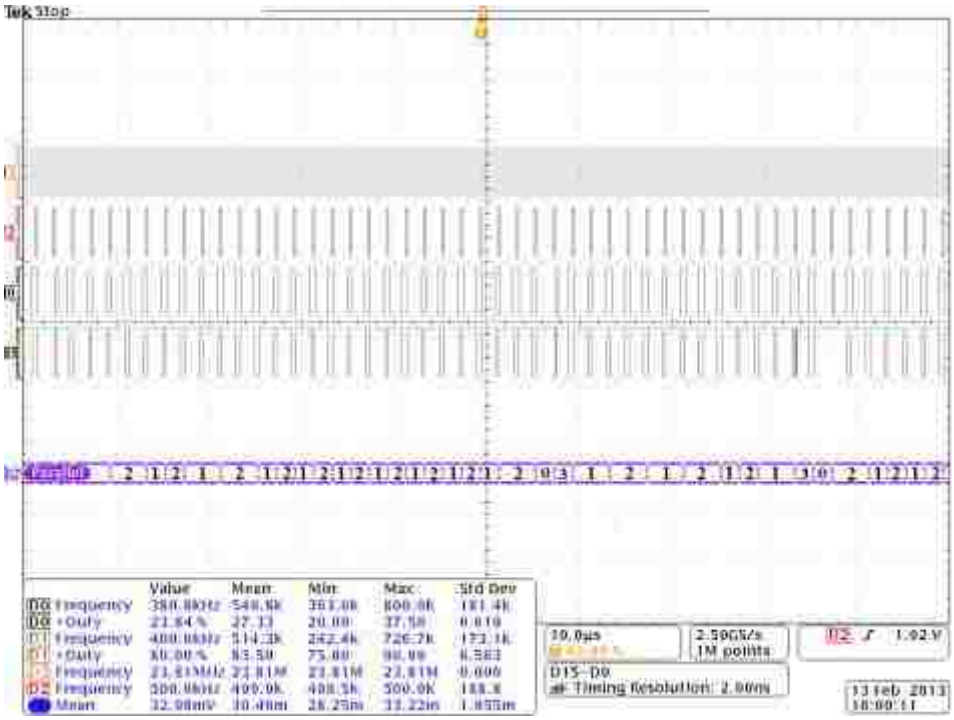
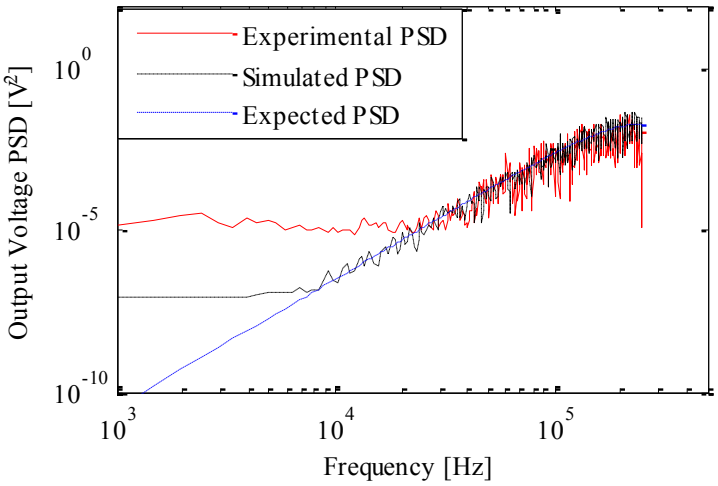
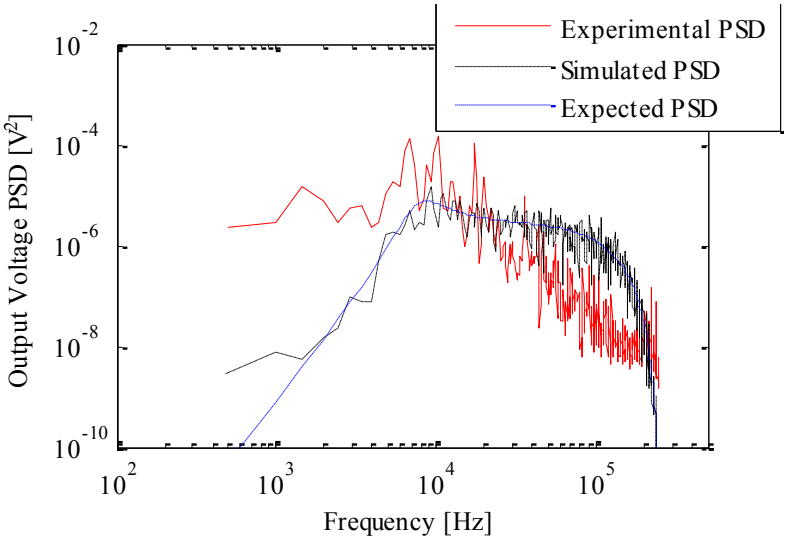


Figure 5.10 Scope Shot of the 3-bit 2^{nd} – order Σ-Δ Modulator Bus Depicting Variations between Four Levels for a given Input



(a)



(b)

Figure 5.11 Experimental PSD for a 2nd - order Σ-Δ Modulator with Duty Input of 1.5175784 for (a) Modulator Output (b) Plant Output Voltage

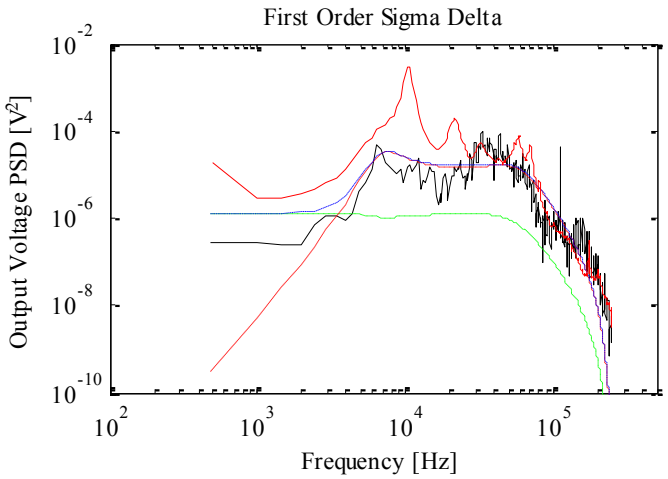
The 2nd – order Σ-Δ modulator output PSD matches the simulation and expected PSD at higher frequencies. The experimental noise floor is higher than the simulation noise floor as can be seen at lower frequencies in Figure 5.11 (a).

Next, the closed-loop experiments have been performed for the various design choices. For ease of comparison, a uniform scale was used for all plots, as well as a uniform trace legend: red solid for experimental PSD, black solid for actual PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum. The experimental PSD obtained is compared with the best choice simulation PSD obtained in the previous chapter.

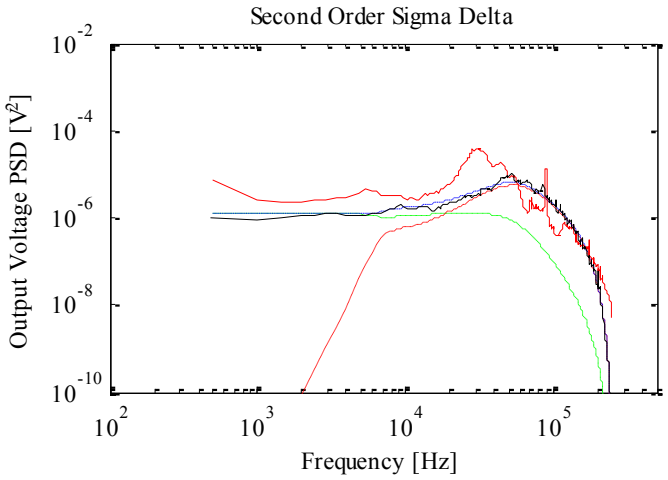
Figure 5.12 shows the output PSD with *nADC* = 10 and *nPWM* = 3 and with compensator F1. Figure 5.12 (a) uses a 1^{st} -order Σ-Δ modulator, whereas Figure 5.12 (b) uses a 2^{nd} -order Σ-Δ modulator. The 1^{st} – order modulator case has the best fit the finite precision analysis PSD whereas the 2^{nd}- order modulator has the best fit with very high precision analysis PSD.

Figure 5.13 shows the output voltage ripple of the converter system with first order modulator same as Figure 5.12. The frequency of the waveform is equal to 1e4 Hz which is same as the frequency where the spike occurs in the 1^{st}- order spectral characteristic in Figure 5.12 (a). Regularity in the output voltage reflects as spikes in the spectral characteristic curve.

Figure 5.14 shows the output PSD with *nADC* = 12 and *nPWM* = 3 and with compensator F1. Figure 5.14 (a) uses a 1^{st} -order Σ-Δ modulator, whereas Figure 5.14 (b) uses a 2^{nd} -order Σ-Δ modulator. The 1^{st} – order modulator case has the best fit with the very high precision analysis PSD whereas the 2^{nd}- order modulator has the best fit with finite precision analysis PSD. Figure 5.15 shows the output voltage ripple for the converter with 1^{st} – order modulator. The regularity in the output voltage waveform is reflected as spike in the spectral characteristic in Figure 5.14 (a).



(a)



(b)

Figure 5.12 Experimental PSD of Plant Output Voltage of the Closed-Loop System with 10-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator, (b) 2nd order Σ-Δ Modulator. Trace legend: red solid for experimental PSD, black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum

Figure 5.16 shows the output PSD with *nADC* = 8 and *nPWM* = 3 and with compensator F1. Figure 5.16 (a) uses a 1st -order Σ-Δ modulator, whereas Figure 5.16 (b) uses a 2nd -order Σ-Δ modulator. In this case both the 1st – order modulator and the 2nd-order modulator has the best fit with finite precision analysis PSD.

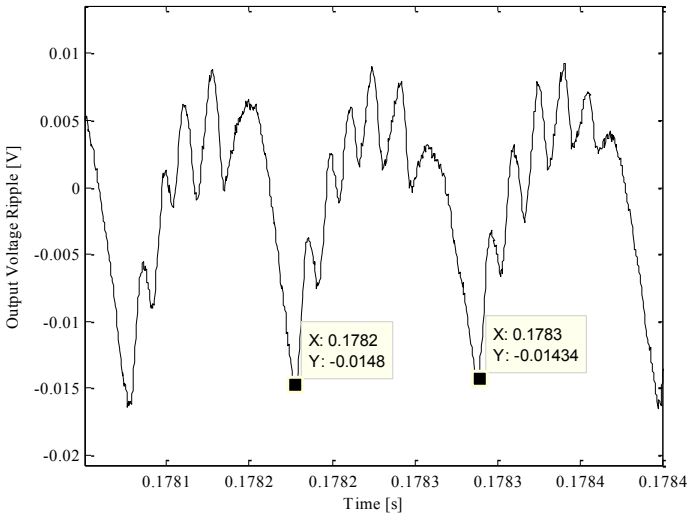
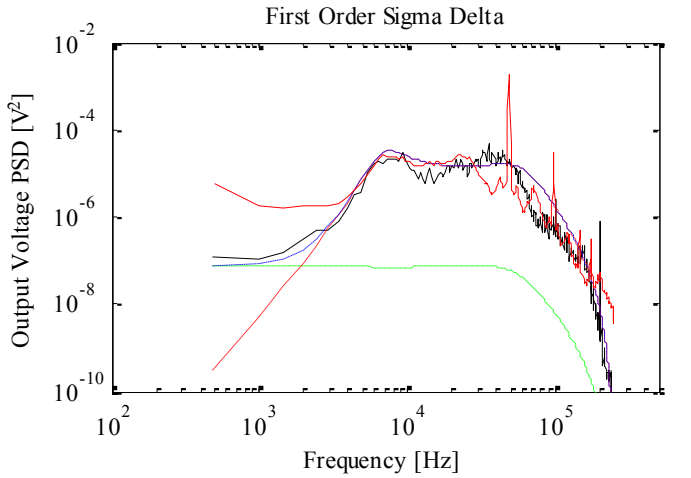
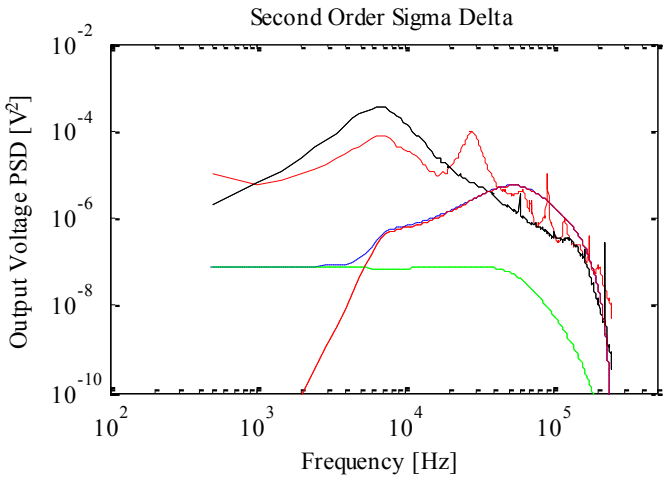


Figure 5.13 Output Voltage Ripple of the Closed-Loop System with 10-bit ADC, 3-bit PWM and Compensator F1 with 1st-order Σ-Δ Modulator



(a)

Figure 5.14 Experimental PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator. Trace legend: red solid for experimental PSD, black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum



(b)

Figure 5.14 Experimental PSD of Plant Output Voltage of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1 with (b) 2nd order Σ-Δ Modulator (cont.). Trace legend: red solid for experimental PSD, black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum

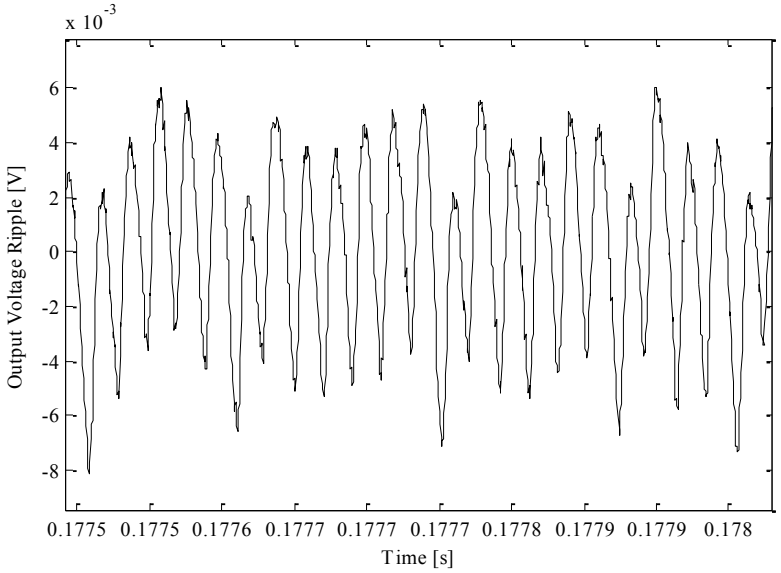
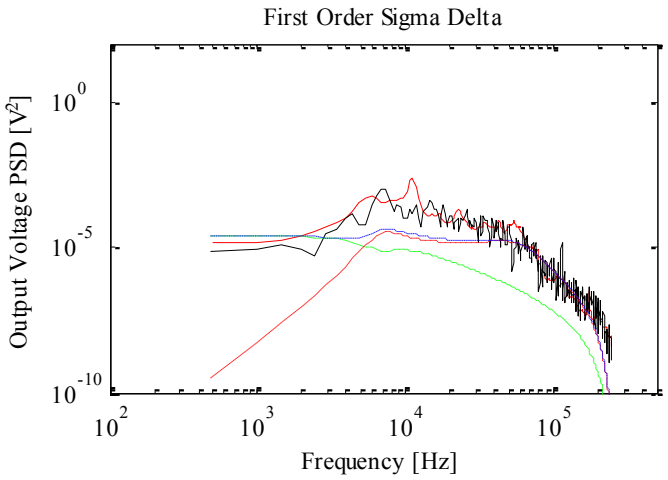
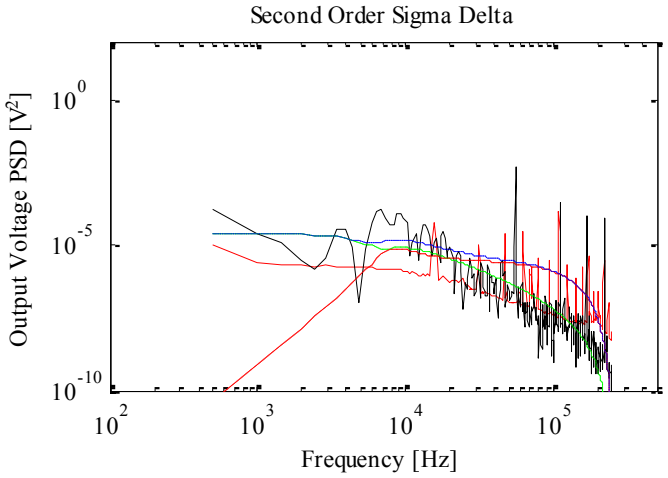


Figure 5.15 Output Voltage Ripple of the Closed-Loop System with 12-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator



(a)



(b)

Figure 5.16 Experimental PSD of Plant Output Voltage of the Closed-Loop System with 8-bit ADC, 3-bit PWM and Compensator F1 with (a) 1st order Σ-Δ Modulator (b) 2nd-order Σ-Δ Modulator (cont.). Trace legend: red solid for experimental PSD, black solid for Simulink PSD, red dashed for the Σ-Δ contribution, green dashed for the ADC contribution and blue dash-dot for the sum

The PSD curves derived from very high precision simulation models validate the theoretical model developed. To understand the impact of quantization noise due to finite computation word length on the output noise, a finite precision model was developed. It was observed that the results from the finite precision model were significantly different

from the very high precision model. Also, the 2nd – order curves for the finite precision model did not fit the expected curves suggesting that another noise source which includes the effect of finite computation word should be added to the model. A switching model was developed using PLECS to understand the impact of sampling on a continuous time output and its effect on the curve shape. Finally, to emulate the experimental set-up more accurately, the finite precision model with PLECS plant was chosen as the basis for comparison with the experimental results.

The PSD of the output of the open-loop 1st –order modulator and open-loop 1st – order modulator with plant for a duty of 1.57178 matched the expected and simulated curves. This validates the experimental module developed using VHDL. The PSD of the output of the open-loop 2nd –order modulator and for a duty of 1.57178 matched the expected and simulated curves suggesting that the modulator model is accurate. The PSD of open-loop 1st –order modulator with plant matched the expected curve.

The closed-loop output PSD with *nADC* = 10, *nPWM* = 3 and with compensator F1 for a 1st –order modulator exhibits the overall expected shape. The nature of the strong spike exhibited at 1e4 frequency, which is in the order of mV, may be due to the periodicity of the output voltage waveform. The closed-loop output PSD for both *nADC* = 12 and 8, *nPWM* = 3 and with compensator F1 for a 1st –order modulator match very well with the expected curve. Again, the strong spike in the spectral curve may be due to the regularity in the output voltage waveform.

The closed-loop output PSD with *nADC* = 10, *nPWM* = 3 and with compensator F1 for a 2nd –order modulator exhibits the overall expected shape but the nature of the hump with a peak at 2e4, which is in the order of mV, is unknown. The closed-loop

output PSD for both *nADC* = 12 and 8, *nPWM* = 3 and with compensator F1 for a 2^{nd}-order modulator did not meet the expectations. This may be due to errors in implementation of the experimental closed-loop or due to the error in data collection or due to any external noise that may have infiltrated the circuit and remained unaccounted for in the theoretical analysis. Further analysis is required to thoroughly understand the interaction of the 2^{nd} – order modulator with the closed-loop system.

6. CONCLUSIONS AND FUTURE SCOPE

The closed-loop output voltage spectral characteristic of a voltage-mode controlled synchronous buck converter has been analyzed. A theoretical model was developed to determine the contribution of the two quantizers, ADC and DPWM, considering the effects of noise-shaping by the Σ-Δ modulator and feedback. A number of simulations have been performed to explore the impact of various design choices on output noise. The study variables included the order of the Σ-Δ modulator, resolution of ADC, resolution of DPWM, the plant and the compensator. The simulation results of the system-level simulation model developed with high precision computational word length validated the theoretical model developed for all the design choices considered. A circuit-level simulation model was developed to be able to incorporate the parasitic resistances of real components and switching in the analysis. Another system-level model with finite precision computational word length was developed to analyze the effect of white noise introduced due to finite precision. The simulation results obtained from this model for the 2^{nd}- order modulator did not match the expected and the high precision simulation results. This may be due to the implementation error of the 2^{nd} – order modulator system. Additional work is necessary to identify the error. Also, the contribution of white noise due to finite precision to the output voltage PSD should be added to the present theoretical model.

The theoretical analysis and the high precision simulation results for the 1^{st}- order modulator system have been validated using experimental results. A PCB was designed to implement the synchronous buck converter power stage. The digital control algorithm was implemented using a FPGA. The experimental results for the 2^{nd} - order modulator

system did not match the expected and the high precision simulation curves. The spectral characteristics of the output voltage of the open-loop plant with 2^{nd}- order modulator showed deviations from the expected curves while the spectral characteristic of the modulator alone matched the expected curve. This suggests that the interaction of the 2^{nd}- order modulator with the plant and the closed-loop system is more complex than expected. This mismatch may be due to errors in implementation of the experimental closed-loop or due to the error in data collection or due to any external noise that may have infiltrated the circuit and remained unaccounted for. Further analysis is required to thoroughly understand the interaction of the 2^{nd} – order modulator with the closed-loop system. Also, in general there is a slight difference in the voltage levels between the simulated and the experimental curves. This difference is in the order of mV and is due to the high noise floor in the experimental set up.

Future scope includes validating the 2^{nd}- order modulator system, generalizing the analysis to other plant and feedback models, extending this work to multi-phase converter systems.

APPENDIX A

MATLAB FILES

```
%-----------------------------------------------------------
%File to generate expected total system PSD curves
%-----------------------------------------------------------

window = 1000;

%-----------------------------------------------------------
% Small Signal Model for Synchronous Buck Converter
% Transfer function of output w.r.t duty

s = tf('s');

Gvd = ((Vout/Dmax)*(1+
      (s*(Cout*Rcout))))/(1+(s*((L/(Rload+Rsw1+RL))+
      ((Cout*Rload*(Rsw1+RL))/(Rload+Rsw1+RL))+
      (Cout*Rcout)))+(s*s*
      (L*Cout*((Rload+Rcout)/(Rload+Rsw1  +RL)))));

% Discretizing the Small Signal Model
% Transfer Function in Discrete Time

Gvd_Z = c2d(Gvd, 1/fsw,'tustin');

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

%Compensator 'comp' - two choices F1 and F2

% F1
Cnum = [8.527, -16.58, 8.115];
Cden = [1,-1,0];

%F2
%     Cnum = [0.741, -1.361, 0.6204];
%     Cden = [1,-1.07412,0.07412];

comp = tf(Cnum, Cden, 1/500e3, 'Variable','z^-1');
%-----------------------------------------------------------

myZ = tf(1,[1,0],1/500e3);

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

nADC = 12; % Three choices 8, 10, 12

% Gain Values for ADC = 8
%     k1 = 0.2;
%     k2_s = 0;
%     k2_f = 4/pi;

% Gain Values for ADC = 12, 10
k1 = 1;
k2_f = 4/pi;
k2_s = 1;
```

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

nPWM = 3;

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

%NTF of Sigma-Delta Modulators(SDM)

% For First Order Modulator
sd1 = tf([1,-1],[1],1/500e3,'Variable','z^-1');

% For Second Order Modulator
sd2 = tf([1,-2,1],[1],1/500e3,'Variable','z^-1');
%-----------------------------------------------------------

% Transfer Function for Open-loop Plant and SDM+PWM

% For First Order Modulator
sd1nofb = Gvd_Z*sd1*myZ*k2_f;

% For Second Order Modulator
sd2nofb = Gvd_Z*sd2*myZ*k2_s;

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

% Transfer Function for Closed-loop Comp, Plant and SDM+PWM

% For First Order Modulator
sd1fb = Gvd_Z*sd1*feedback(1,k2_f*Gvd_Z*comp*myZ);

% For Second Order Modulator
sd2fb = Gvd_Z*sd2*feedback(1,k2_s*Gvd_Z*comp*myZ);

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

% Transfer Function of ADc Contribution

adc = feedback(Gvd_Z*comp*myZ,k1);

%-----------------------------------------------------------
% Variance Calculation of Quantizers ADC and DPWM

delta1 = 2^(-nADC-1);
delta2 = 2^(-nPWM-1);
sigmasqrd1 = (1/3)*delta1^2;
sigmasqrd2 = (1/3)*delta2^2;
%-----------------------------------------------------------
% ADC and PWM Contribution Calculation

myfilt1 = adc;
myfilt2 = sd2fb;
myfilt4 = sd1fb;
```

```
[mag1,phase1] = bode(myfilt1,w1*fsw);
mymag1 = zeros(size(w1));
mymag1(:) = mag1(1,1,:);
[mag2,phase2] = bode(myfilt2,w1*fsw);
mymag2 = zeros(size(w1));
mymag2(:) = mag2(1,1,:);
[mag4,phase4] = bode(myfilt4,w1*fsw);
mymag4 = zeros(size(w1));
mymag4(:) = mag4(1,1,:);


expected_psd1 = mymag1 .^2 * (sigmasqrd1);
expected_psd2 = mymag2 .^ 2 * (sigmasqrd2);
expected_psd3 = expected_psd1 + expected_psd2;
expected_psd4 = mymag4 .^2 * (sigmasqrd2);
expected_psd5 = expected_psd1 + expected_psd4;

%-------------------------------------------------------------
%Plot Curves

% For First Order SDM

figure(1)
loglog(f1,expected_psd1,'g--',f1,expected_psd4,
            'r--',f1,expected_psd5,'b-.');
set (gcf, 'Position', [200 200 3.45*96 2.25*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
myaxis = [1e2,5e5,1e-10,1e-2];
axis(myaxis)

% For Second Order SDM

figure(2)
loglog(f2,expected_psd1,'g--',f2,expected_psd2,
            'r--',f2,expected_psd3,'b-.');
set (gcf, 'Position', [300 300 3.45*96 2.25*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
axis(myaxis);

%-------------------------------------------------------
% End of file generate expected total system PSD curves
%-------------------------------------------------------
```

```
%-----------------------------------------------------------
% Set-up File for Simulations
%-----------------------------------------------------------

% Synchronous Buck Converter Component Specifications

Cin=141e-6;      % Input Capacitor
Rcin = 0.00067; % ESR of Input Capacitor

Rsw1 = 0.01;     % RDSon of MOSFET Q1
Rsw2 = 0.01;     % RDSon of MOSFET Q2

L = 10e-6;       % Inductor
RL = 0.092;      % DCR of Inductor

Cout = 47e-6;    % Output Capacitor
Rcout= 0.002;    % ESR of Output Capacitor

Rload = 1.0;     % External Load

Vin = 5;         % Input Voltage
Vout = 1;        % Desired Output Voltage

Dmax = Vout/Vin; % Duty Ratio

fsw = 500e3;      % Switching Frequency

%-----------------------------------------------------------
% Setup PWM, sigma-delta
nPWM = 3;
nWord = 64; % Word length for very-high precision analysis
nFrac = 32; % essentially infinite precision
nADC = 12;  % Three choices 12, 10, 8
%-----------------------------------------------------------

%Buck converter discrete-time model (like AVM)
A = [-RL/L, -1/L; 1/Cout, -1/(Rload*Cout)];
B = [Vin/L; 0];
buckavm = ss(A, B, [0 1], 0);
buckZ = c2d(buckavm, 1/fsw);
[buckA, buckB, buckC, buckD, buckT] = ssdata(buckZ);

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

% Small-Signal Model for Synchronous Buck Converter
% Transfer function of output w.r.t duty

s = tf('s');

Gvd = ((Vout/Dmax)*(1+(s*
      (Cout*Rcout))))/(1+(s*((L/(Rload+Rsw1+RL))+
      ((Cout*Rload*(Rsw1+RL))/(Rload+Rsw1+RL))+
      (Cout*Rcout)))+(s*s*(L*Cout*
      ((Rload+Rcout)/(Rload+Rsw1+RL)))));
```

```
zpk_Gvd = zpk(Gvd);

Gnum = [0, Cout*Rcout*Vout/Dmax, 1];

Gden = [(L*Cout*((Rload+Rcout)/(Rload+Rsw1+RL))),
         ((L/(Rload+Rsw1+RL))+
         ((Cout*Rload*(Rsw1+RL))/(Rload+Rsw1+RL))
         + (Cout*Rcout)), 1];

[As, Bs, Cs, Ds] = tf2ss(Gnum, Gden);

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

% Another Small-Signal Model for Gvd in Terms of wo and Q

s = tf('s');

sz1 = 1/(Rcout*Cout);

wo = ((1/(sqrt(L*Cout)))*
      (sqrt(((1+(RL/Rload))/((Rcout/Rload)+1)))));

Q = ((1/wo)*(1/((L/(RL+Rload))+
(Cout*(Rcout+((RL*Rload)/(RL+Rload))))))));

Gvd1 = (Vin*(1+(s/sz1)))/(1+(s*(1/(wo*Q)))+(s^2*(1/wo^2)));

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

% Discretizing the Small Signal Model
% Transfer Function in Discrete Time

Gvd_Z = c2d(Gvd, 1/fsw,'tustin');

[Az, Bz, Cz, Dz, buckT] = ssdata(Gvd_Z);

controlT = buckT;

zpk_Gvd_z = zpk(Gvd_Z);

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

% Compensator Selection
%F1
Cnum = [8.527, -16.58, 8.115];
Cden = [1,-1,0];

%F2
%      Cnum = [0.741, -1.361, 0.6204];
%      Cden = [1,-1.07412,0.07412];

comp = tf(Cnum, Cden, 1/500e3, 'Variable','z^-1');
myZ = tf(1,[1,0],1/500e3);
Sys_cl = feedback(comp*Gvd_Z, 1;
```

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

% Split out coefficients for control canonical form
b0 = Cnum(1);
b1 = Cnum(2);
b2 = Cnum(3);
b3 = 0;

a1 = -Cden(2);
a2 = -Cden(3);
a3 = 0;

closed = feedback(Gvd_Z*comp*myZ,1);

%-------------------------------------------------------------
% End of file Set-up for Simulations
%-------------------------------------------------------------



%-------------------------------------------------------------
% File to process the simulation results. This file can be
used to process both Simulink and PLECS files. This file can
also generate the expected curves thus making comparison
easier. To superimpose PLECS result on Simulink result, run
either of them first, hold the figure and then run the second
one. The '.csv' files obtained from simulations should be
stripped of their header information first and saved in
the '.csv' format in the same path.
%-------------------------------------------------------------

window = 1000;
fsw = 500e3;

%-------------------------------------------------------------
% Read output voltage information about the system with
First Order Modulator stored in file named simout1

time1 = simout1.time;
sigs1 = simout1.signals.values;
voltage1 = sigs1 (:,1);
mean_v1 = mean (voltage1)
rms1 = rms(voltage1 - mean_v1)

%-------------------------------------------------------------
% Compute Vwelch1 using 'pwelch' function in MATLAB

N1 = length (time1);
[Vwelch1temp,w1] = pwelch(voltage1-mean_v1,window);
Vwelch1 = Vwelch1temp * pi;
f1 = w1 * fsw / (2*pi);
```

```
%-------------------------------------------------------------
% Read output voltage information about the system with
Second Order Modulator stored in file named simout2

time2 = simout2.time;
sigs2 = simout2.signals.values;
voltage2 = sigs2(:,1);
mean_v2 = mean(voltage2)
rms2 = rms(voltage2 - mean_v2)

%-------------------------------------------------------------
% Compute Vwelch1 using 'pwelch' function in MATLAB

N2 = length(time2);
[Vwelch2temp,w2] = pwelch(voltage2-mean_v2,window);
Vwelch2 = Vwelch2temp * pi;
f2 = w2 * fsw / (2*pi);


%-------------------------------------------------------------------
% To plot the expected curves in the same plot
%-------------------------------------------------------------------

% Small Signal Model for Synchronous Buck Converter
% Transfer function of output w.r.t duty

s = tf('s');

Gvd = ((Vout/Dmax)*(1+(s*(Cout*Rcout))))/(1+(s*((L/(
Rload+Rsw1+RL))+ ((Cout*Rload*(Rsw1+RL))/(Rload+Rsw1+RL))+
(Cout*Rcout)))+(s*s*(L*Cout*((Rload+Rcout)/(Rload+Rsw1+RL)))));

% Discretizing the Small Signal Model
% Transfer Function in Discrete Time

Gvd_Z = c2d(Gvd, 1/fsw,'tustin');

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

%Compensator 'comp' - two choices F1 and F2

% F1
Cnum = [8.527, -16.58, 8.115];
Cden = [1,-1,0];

%F2
%     Cnum = [0.741, -1.361, 0.6204];
%     Cden = [1,-1.07412,0.07412];

comp = tf(Cnum, Cden, 1/500e3, 'Variable','z^-1');
```

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

myZ = tf(1,[1,0],1/500e3);

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

nADC = 12; % Three choices 8, 10, 12

% Gain Values for ADC = 8
%      k1 = 0.2;
%      k2_s = 0;
%      k2_f = 4/pi;

% Gain Values for ADC = 12, 10
k1 = 1;
k2_f = 4/pi;
k2_s = 1;


%-----------------------------------------------------------
nPWM = 3;
%-----------------------------------------------------------

%NTF of Sigma-Delta Modulators (SDM)

% For First Order Modulator
sd1 = tf([1,-1],[1],1/500e3,'Variable','z^-1');

% For Second Order Modulator
sd2 = tf([1,-2,1],[1],1/500e3,'Variable','z^-1');
%-------------------------------------------------------------------

% Transfer Function for Open-loop Plant and SDM+PWM

% For First Order Modulator
sd1nofb = Gvd_Z*sd1*myZ*k2_f;

% For Second Order Modulator
sd2nofb = Gvd_Z*sd2*myZ*k2_s;

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

% Transfer Function for Closed-loop Comp, Plant and SDM+PWM

% For First Order Modulator
sd1fb = Gvd_Z*sd1*feedback(1,k2_f*Gvd_Z*comp*myZ);

% For Second Order Modulator
      sd2fb = Gvd_Z*sd2*feedback(1,k2_s*Gvd_Z*comp*myZ);
```

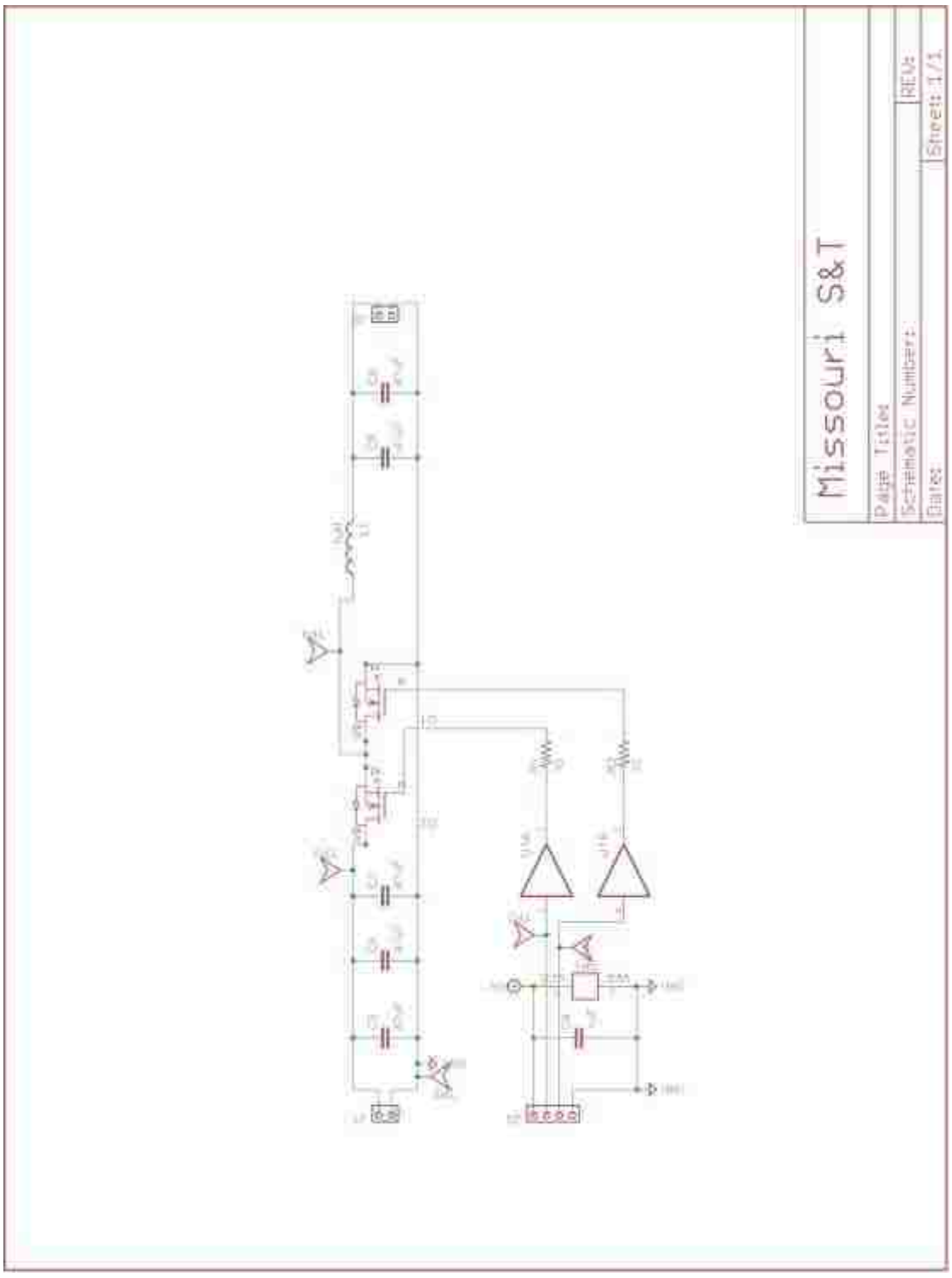
```
%-------------------------------------------------------

% Transfer Function of ADc Contribution

adc = feedback(Gvd_Z*comp*myZ,k1);

%-------------------------------------------------------
% Variance Calculation of Quantizers ADC and DPWM

delta1 = 2^(-nADC-1);
delta2 = 2^(-nPWM-1);
sigmasqrd1 = (1/3)*delta1^2;
sigmasqrd2 = (1/3)*delta2^2;
%-------------------------------------------------------
% ADC and PWM Contribution Calculation

myfilt1 = adc;
myfilt2 = sd2fb;
myfilt4 = sd1fb;

[mag1,phase1] = bode(myfilt1,w1*fsw);
mymag1 = zeros(size(w1));
mymag1(:) = mag1(1,1,:);
[mag2,phase2] = bode(myfilt2,w1*fsw);
mymag2 = zeros(size(w1));
mymag2(:) = mag2(1,1,:);
[mag4,phase4] = bode(myfilt4,w1*fsw);
mymag4 = zeros(size(w1));
mymag4(:) = mag4(1,1,:);


expected_psd1 = mymag1 .^2 * (sigmasqrd1);
expected_psd2 = mymag2 .^ 2 * (sigmasqrd2);
expected_psd3 = expected_psd1 + expected_psd2;
expected_psd4 = mymag4 .^2 * (sigmasqrd2);
expected_psd5 = expected_psd1 + expected_psd4;

%-------------------------------------------------------
%Plot Curves
%-------------------------------------------------------

figure(1)
loglog(f1,Vwelch1,'k-',f1,expected_psd1,'g--',
       f1,expected_psd4,'r--',f1,expected_psd5,'b-.');
set (gcf, 'Position', [200 200 3.45*96 2.25*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
myaxis = [1e2,5e5,1e-10,1e-2];
axis(myaxis)
```

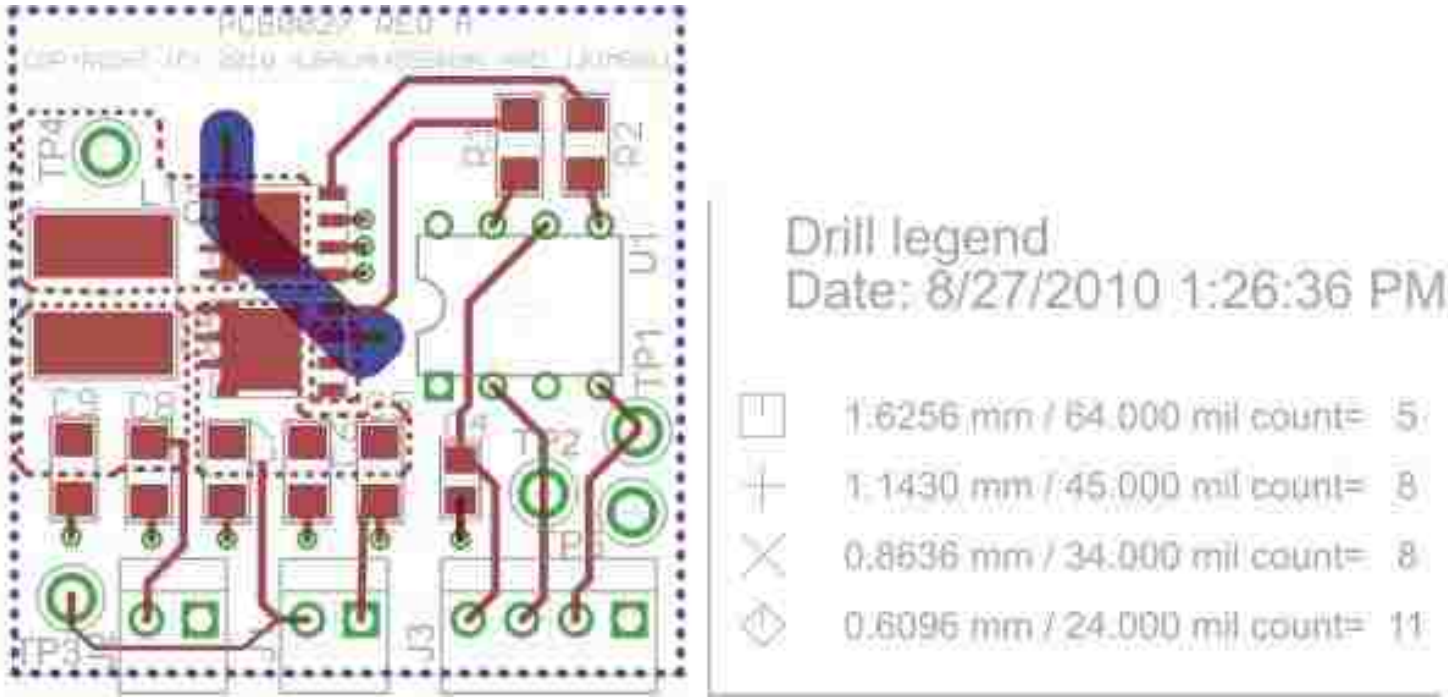
```
figure(2)
loglog(f2,Vwelch2,'k-',f2,expected_psd1,'g--',
       f2,expected_psd2,'r--',f2,expected_psd3,'b-.');
set (gcf, 'Position', [300 300 3.45*96 2.25*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
axis(myaxis);


%---------------------------------------------------------------
% End of file to process simulation results
%---------------------------------------------------------------

%---------------------------------------------------------------
% File to process experimental results
%---------------------------------------------------------------

window = 1000;

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

% First Order CSV file read and extraction

x1       = csvread('filename.csv');
time1    = x1(:,1);
voltage1 = x1(:,2);

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

%- Resample data to match the simulation sample rate

voltage1_resamp = resample(voltage1,1,5);
time1_resamp    = resample(time1, 1, 5);
mean_v1         = mean(voltage1_resamp)

%---------------------------------------------------------------
% First Order pwelch

N1 = length(time1_resamp);
[Vwelch1temp,w1] = pwelch((voltage1_resamp-mean_v1),window);
Vwelch1 = Vwelch1temp*pi;
dt1 = time1_resamp(38) - time1_resamp(37);
fsamp1 = (1/dt1);
f1 = w1 * fsamp1 / (2*pi);
```

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

%First Order PSD Plot
figure(1)
loglog(f1,Vwelch1,'r')
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
title('First Order Sigma Delta')


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

% Second Order CSV file read and extraction
x2 = csvread('filename.csv');
time2 = x2(:,1);
voltage2 = x2(:,2);

%-------------------------------------------------------------
%- Resample data to match the simulation sample rate

voltage2_resamp = resample(voltage2,1,5);
time2_resamp = resample(time2, 1, 5);
mean_v2 = mean(voltage2_resamp)

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

% Second Order pwelch

N2 = length(time2_resamp);
[Vwelch2temp,w2] = pwelch(voltage2_resamp-mean_v2,window);
Vwelch2 = Vwelch2temp * pi;
dt2 = time2_resamp(38)-time2_resamp(37);
fsamp2 = 1/dt2;
f2 = w2 * fsamp2 / (2*pi);

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

%Second Order PSD Plot
figure(2)
loglog(f2,Vwelch2,'r')
set (gcf, 'Position', [200 200 3.45*96 2.25*96])
set(gca,'FontName','Times')
set(gca,'FontSize',[8])
xlabel('Frequency [Hz]')
ylabel('Output Voltage PSD [V^2]')
title('Second Order Sigma Delta')

%-------------------------------------------------------------------------
% End of file to process experimental results
%-------------------------------------------------------------
```

APPENDIX B

SYNCHRONOUS BUCK CONVERTER SCHEMATIC


Missouri S&T
Page Title:
Schematic Number:
REV:
Date:
Sheet: 1/1


Drill legend
Date: 8/27/2010 1:26:36 PM
1:6256 mm / 64.000 mil count= 5
1:1430 mm / 45.000 mil count= 8
0.8636 mm / 34.000 mil count= 8
0.6096 mm / 24.000 mil count= 11

APPENDIX C

VHDL CODES

```
----------------------------------------------------------------

     met
   leftshi
     ft

   Descrp
    tion:
       Implementing left shift on 12-bit ADC input for
multiplicatio
    n by 2.
       The code uses fixed-point format.

   Inputs:
           iReset          - Active Low Reset.
           iClk            - Input clock.
           iValin          - 12 bit input.
           iStart          - Send pulse to begin.  iReset must
be 1.
   Outputs:
           oDone           - Pulsed when done.
           oOut            - 12 bit output left shifted by 1
bit and padded with 0.
----------------------------------------------------------------

Library ieee;
use ieee.std_logic_1164.all;
use  ieee.std_logic_unsigned.all;
use  ieee.fixed_pkg.all;
use ieee.fixed_float_types.all;
use  ieee.float_pkg.all;

entity met leftshift is
port (

     iClk, iReset   in std_logic;
     iStart         in std_logic;
     iValin         in std_logic_vector  (11 downto 0);
     oOut           out std_logic_vector  (11 downto 0);
     oDone          out std_logic

    );

end mst_leftshift;

architecture behav of mst_leftshift is
signal  delay  : std_logic;
begin
process (iClk, iReset, iStart, iValin ) is
begin
       if  (iReset =  '0') then
```

```
            oOut <=  (others => '0');

        elsif rising_edge(iClk)then

            if  (iStart = '1') then

                oout <= iValin  (10 downto 0)  &  '0';

            end if;

        oDone <= iStart;

        end if;

end process;
end behav;
```

```
   mst_truncate
-:- Description:
      Implementing truncation on 12-bit ADC input to obtain
8-bit or 10-bit ADC.
      The code uses fixed-point  format.
   Inputs:
           iReset          - Active Low Reset.
           iClk            - Input clock.
           iValin          - 12 bit input.
           iStart          - Send pulse to begin.  iReset must
be 1.
   Outputs:
           oDone           - Pulsed when done.
           oOut            - m bit output.
----------------------------------------------------------------

Library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.fixed_pkg.all;
use ieee.fixed_float_types.all;
use ieee.float_pkg.all;

entity mst_truncate is
generic  (

  n   integer := 12          variable input bit length
  m   integer .= 8           variable output bit length

        );
port
   (
     iClk, iReset   in std_logic;
     iStart         in std_logic;
     iValin         in std_logic_vector  (n-1 downto 0);
     oOut           out std_logic_vector  (m-1 downto 0);
     oDone          out std_logic

    );
end mst_truncate;

architecture behav of mst_truncate is
begin
process (iClk, iReset, iStart, iValin   is
begin
       if (iReset    '0') then
```

```
            oOut <=  (others => '0');
            oDone <= '0';

        elsif rising_edge(iClk)then

                  if  (iStart = '1') then
                      oOut <= iValin(n-1 downto n-m);
                  end if;

        oDone <= iStart;
        end if;

end process;
end behav;
```

```
----------------------------------------------------------------

   mst_pid

   Description:
      A PID controller in Controller Cannonical Form for 12 bit
input and 12 bit output.
      To obtain different output bit resolution change the
int_out_low and int_out_high values.
      The code uses fixed-point format.

   Inputs:

           iClk            - Input clock.
           iReset          - Active Low Reset.
           iStart          - Send fUlse to begin.  iReset must
be 1.
           iRef           - Reference Voltage from ADC of
length m bits
           iin             - Output Voltage of converter from
ADC of length m bits
   Outputs:
           oDone           - Pulsed when done.
           Oout           - Output from the modulator of
length n bits in vector format.
----------------------------------------------------------------

-- 10/18/2012 - v001
-- Initial Version
----------------------------------------------------------------

-- Copyright (c) 2011 by Jonathan Kimball and Anupama
Balakrishnan. All rights reserved.
----------------------------------------------------------------


library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.a:)..1;
use ieee.numeric_std.all;
use ieee.fixed_pkg.all;
use ieee.fixed_float_types.all;
use ieee.float_pkg.all;

entity mst_pid is

generic (
```

```
    m    natural  := 12;        variable input bit length
    n    natural  := 12;        variable output bit length

    gain_len_high    integer  := 6;      gain decimal bits in sfixed
    gain_len_low     integer  := -20;    gain fractional bits in
sfixed
    int_len_low      integer  := -8;     input and ref fractional
bit in sfixed
    int_len_high     integer  := 3;      input and ref decimal bit
in sfixed
    int_out_low      integer  := 2;      can be used to adjust the
ouput
    int_out_high     integer  := 9

    constant  int_up_limit      real     := 0.5;       Anti-windup
integral upper limit
    constant  int_low_limit     real     := -0.5;      Anti-windup
integral lower limit
    constant  out_up_limit      real     := 7.99;      Output Upper
limit
    constant  out_low_limit     real     := 0.0        Output Lower
limit
    constant K1                 real     := 8.527;     bo
    constant K2                 real     := -16.58;    b1
    constant K3                 real     := 8.115      b2

);

port

    iClk             in  std_logic;
    iReset           in  std_logic;
    iStart           in  std_logic;
    iRef             in  std_logic_vector   (m-1 downto 0);
    iin              in  std_logic_vector   (m-1 downto 0);
    oOut             out  std_logic_vector(n-1 downto  0);
    oDone            out  std_logic

);


end entity mst_pid;

architecture  behavioral  of mst_pid  is

    signal  1Err_prev1    sfixed  (int_len_high+1  downto
int_len_low);
    signal  1Err_prev2    sfixed  (int_len_high+1  downto
```

```
int_len_low);
    signal lOut_prev      sfixed  (14  downto  -14);
    signal lRef_temp      sfixed  (int_len_high   downto
int_len_low);
    signal lin_temp       sfixed  (int_len_high   downto
int_len_low);
    signal delay          std_logic;
    signal lout_high      sfixed  (14 downto  -14);
    signal lout_low       sfixed  (14  downto  -14);
    signal lK1_dis        sfixed  (gain_len_high downto
gain_len_low);
    signal lK2_dis        sfixed  (gain_len_high downto
gain_len_low);
    signal lK3_dis        sfixed  (gain_len_high downto
gain_len_low);

begin

process( iClk, iReset, iStart, iRef, iin)is

      variable lKp          sfixed  (gain_len_high  downto
gain_len_low);
      variable lKi          sfixed  (gain_len_high  downto
gain_len_low);
      variable lKd          sfixed  (gain_len_high  downto €
gain_len_low);
      variable lRef         sfixed  (int len_high  downto
int_len_low);
      variable l_In         sfixed  (int_len_high   downto
int_len_low);
      variable lErr         sfixed  (int_len_high+1 downto
int_len_low);
      variable lK1          sfixed  (gain_len_high downto
gain len low);
      variable lK2          sfixed  (gain_len_high  downto
gain_len_low);
      variable lK3          sfixed  (gain_len_high  downto
gain_len_low);
      variable lOut         sfixed  (14 downto  -14);
      variable lOut_temp    sfixed  (int_out_low  downto
int_out_high);
      variable l0ut_temp1   ufixed  (int out_low  downto
int_out_high);

begin

 if iReset    '0' then
```

```
            lRef        := (others => '0');
            lin         := (others => '0');
            lOut        := (others => '0');
            lOut_prev   <= (others => '0');
            oOut        <= (others => '0');
            1Err_prev1  <= (others => '0');
            1Err_prev2  <= (others => '0');
            oDone       <= '0';
            delay       <= '0';
            lout_high   <= to sfixed(out_up_limit  , lout_high);
            lout low    <= to_sfixed(out_low limit   lout_low);
            1K1         := to_sfixed(K1,  lK1);
            1K2         := to_sfixed(K2,  1K2);
            1K3         := to_sfixed(K3,  lK3);
            1K3 dis     <= 1K3;
            1K2 dis     <= lK2;
            1K1 dis     <= 1K1;


elsif rising_edge(iClk)  then

            if (iStart   '1' and delay = '0')then

                  lRef        := to_sfixed(iRef,  lRef);
                  lin         := to sfixed  (iin, lin);
                  lRef_temp   <= lRef;
                  lin_temp    <= lin;
                  lErr        := lRef - lin;
                  lErr_pre¥2  <= 1Err_prev1;
                  1Err_prev1  <= lErr;
                  lOut        .- resize  (lOut_prev + (1K1*1Err) +
(1K2*1Err_prev1)  + (lK3*1Err_prev2),  lOut);
                  lOut_prev   <= resize  (lOut, lOut_prev);


                  if lOut < lout low then

                     l0ut_temp1  := (others => '0');

                  elsif lOut > lout_high  then

                     10ut_temp1  := ( others => '1');

                  else

                     lOut_temp  := resize  (lOut, lOut_temp);
                     10ut_temp1  := ufixed  (lOut_temp);
```

```
                end if;

            end if;


            delay <= iStart;
            oDone <= delay;

 end if;

oOut <= to_slv(lOut_templ);

end process;
end behavioral;
```

```
----------------------------------------------------------------

    mst_sigdelta_fo

    Description:
       A generic first order Sigma-Delta Modulator. The number
  of bits of input and output can be chosen
    using the values of n amd m respectively. The code uses
  fixed-point format.

    Inputs:
            iReset     - Active Low Reset.
            iClk       - Input clock.
            u_in       - Input to the modulator of length n bits
  in ufixed format.
            iStart     - Send pulse to begin.  iReset must be 1.
    Outputs:
            oDone      - Pulsed when done.
            Y_out      - Output from the modulator of length m
  bits in ufixed format.
  ----------------------------------------------------------------
  ---------------------------------
  -- 10/8/2011 - v001
  -- Initial Version
  ----------------------------------------------------------------

  -- Copyright (c) 2011 by Jonathan Kimball and Anupama
  Balakrishnan. All rights reserved.
  ----------------------------------------------------------------
  -------------    ---------- -- ---------

  library ieee;
  use ieee.std_logic_1164.all;
  use ieee.std_logic_unsigned.all;
  use ieee.fixed_pkg.all;
  use ieee.fixed_float_types.all;
  use ieee.float_pkg.all;

  ----Entity Declaration-----

  entity sigdelta_fo is

  generic  ( n   integer .- 12         variable input bit length
             m   integer .- 3          variable output bit length
           );

  port  (  iClk,iReset   in  std_logic;
              u_in       in  std_logic_vector(n-1 downto 0);
              iStart     in  std_logic;
```

```
            oDone     out std_logic;
            Y out     out std_logic_vector(m-1 downto 0)
        );

end sigdelta_fo;

----Architecture----

architecture behav of sigdelta_fo is

signal          iu in   ufixed(n-1 downto 0);
signal            iu    sfixed(m downto m-n);
signal            ix    sfixed(0 downto m-n);
signal            ie    ufixed(m-1 downto m-n);
signal       ie unsat   sfixed(m+1 downto m-n);
signal       ie trunc   sfixed(0 downto m-n);
signal       ix trunc   ufixed(m-1 downto 0);
signal ie_trunc_prev    sfixed(0 downto m-n);
signal     Y_out_temp   ufixed(m-1 downto 0);
signal          delay   std_logic;
signal       ie min s   ufixed(m-1 downto m-n):=(others => '0');
signal       ie max s   ufixed(m-1 downto m-n):= (others => '1');



begin

  ie_unsat            <= iu + ix;
  ie(m-1 downto m-n)  <= ie_min_s when (ie_unsat(m+1) = '1') elsee
 ie_max_s when (ie_unsat( ) = '1') else ufixed(ie_unsat(m-1 e
downto m-n));
  ie_trunc            <= sfixed('9' & ie(-1 downto m-n));
  ix_trunc            <= ie(m-1 downto 0);
  ix                  <= ie_trunc_prev;
  Y_out_temp          <= ix trunc;
  Y_out               <= to_slv(Y_out_temp);


process  (iClk,iReset,ie_trunc,ie_trunc_prev, U_in, iStart) is

variable  iutemp   ufixed(m-1 downto m-n);

begin

if (iReset ='0') then

    ie_trunc_prev   <= (others =>'0');
    iu              <= (others =>'0');
```

```
    iu_in          <=  (others => '0');


elsif (rising_edge(iClk)) then

    if (iStart ='1' and delay  '0') then

         iutemp iu      .- to_ufixed(U_in, iutemp);
         ie_trunc_prev  <= to_sfixed(iutemp);
                        <= ie_trunc;
    end if;

    delay  <= iStart;
    oDone  <= delay;

end if;
end process;
end behav;
```

```
----------------------------------------------------------------

   mst_sigdelta_so

   Description:
      A generic second order Sigma-Delta Modulator. The number
of bits of input and output can be chosen
    using the values of n amd m respectively. The code uses
fixed-point format.

   Inputs:
           iReset     - Active Low Reset.
           iClk       - Input clock.
           U in       - Input to the modulator of length n bits
in ufixed format.
           iStart     - Send pulse to begin.  iReset must be 1.
   Outputs:
           oDone     - Pulse when done.
         Y out       - Output from the modulator of length m
bits in ufixed format.
----------------------------------------------------------------

-- 10/8/2011 - v001
-- Initial Version
----------------------------------------------------------------

-- Copyright (c) 2011 by Jonathan Kimball and Anupama
Balakrishnan. All rights reserved.
----------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.fixed_pkg.all;
use ieee.fixed_float_types.all;
use ieee.float_pkg.all;

----Entity Declaration----

entity mst_sigdelta_so is

generic  (
          n   integer := 12          variable input bit length
          m   integer := 3           variable output bit length

        );

port  (
```

```
        iReset        in  std_logic;
        U in          in std_logic_vector(n-1 downto 0);
        iStart        in std_logic;
        oDone         out std_logis;
        Y_out         out std_logic_vector(m-1 downto 0)

    );
end mst_sigdelta_so;

----Architecture----

architecture behav of mst_sigdelta_so  is

signal            iu    sfixed(m downto m-n);
signal            ix    sfixed(m-1 downto m-n);
signal            ie    ufixed(m-1 downto m-n);
signal       ie unsat   sfixed(m+1 downto m-n);
signal       ie_trunc   sfixed(m downto m-n);
signal       ix_trunc   ufixed(m-1 downto 0);
signal ie_trunc_prev1   sfixed(m downto m-n);
signal ie_trunc_shift   sfixed(m+1 downto m-n);
signal ie_trunc_prev2   sfixed (m downto m-n);
signal     Y out_temp   ufixed (m-1 downto 0);
signal          delay   std_logic;
signal       ie min s   ufixed(m-1 downto m-n):=(others  => '0');
signal       ie max s   ufixed(m-1 downto m-n):=( others => '11');


begin

  ie unsat           <= iu - ix;
  ie(m-1 downto m-n) <= ie_min_s when  (ie unsat(m+1) = '1') else
 ie_max_s when  (ie_unsat(m) = '1') else ufixed(ie_unsat(m-1
downto m-n));
  ie_trunc           <=  sfixed(ie-ix_trunc);
  ix_trunc           <= ie(m-1 downto 0);
  ie trunc shift     <= scalb(ie_trunc_prev1,1)& '0';
  ix                 <= resize(ie_trunc_prev2  ie_trunc_shift,
ix);
  y_out_temp         <= ix_trunc;
  Y_out              <= to_slv(Y_out_temp);


process (iClk,iReset,ie_trunc,ie_trunc_prev1, U_in, iStart) is

variable iutemp       ufixed(m-1 downto m-n);
```

```
begin

if (iReset ='0') then

      ie_trunc_prev1   <=  (others =>'0');
      ie_trunc_prev2   <=  (others =>'0');
      oDone            <= '0';
      delay            <= '0';
      iu               <=  (others =>'0');

elsif  (rising_edge(iClk)) then

      if (iStart ='1' and delay=  '0') then

          iutemp iu       := to_ufixed(U_in,iutemp);
          ie_trunc_prev2  <= to_sfixed(iutemp);
          ie_trunc_prev1  <= ie_trunc_prev1;
                          <= ie_trunc;
      end if;

      delay <= iStart;
      oDone <= delay;

end if;
end process;
end behav;
```

```
----------------------------------------------------------------

   mst_pwm_deadtime

   Description:
     A generic Pulse Width Modulator with i cycle dead time
between the two output signals.
     The number of bits of input can be chosen using the values
of n.
     The code uses fixed-point format.

   Inputs:
           iReset     - Active Low Reset.
           iClk       - Input clock.
           iD         - Input  to the modulator of length n bits
in slv format.
           iStart       Send pulse to begin.  iReset must be 1.
           iClkDiv    - Clock Divider to set frequency of gate
signals.
   Outputs:
           oDone      - Pulsed when done.
           oQ oQ1     - Output pulse 1.
           oQ_inv     - Output pulse 2.
           oQ1_inv    - Output pulse 3, inverse of output 1.
                      - Output pulse 4, inverse of output 2..
----------------------------------------------------------------

----------------------------------------------------------------

-- Copyright (c) 2012 by  Anupama Balakrishnan. All rights
reserved.
----------------------------------------------------------------


library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.fixed_pkg.all;
use ieee.fixed_float_types.all;
use ieee.float_pkg.all;

ENTITY mst_pwm_deadtime is

generic (
  n: natural :=3 -- n can be changed as needed
        );

port(
```

```
    iStart     in std_logic; --pulse  at start
    iD         in std_logic_vector(n-1 downto  0 );
    iClk       in std_logic;
    iReset     in std_logic;
    iClkDiv    in std_logic_vector(7 downto  0); --clock divider
for PWM

    oQ  oQ1    out  std_logic;
    oQ_inv     out  std_logic;
    oQ1_inv    out  std_logic;
    oDone      out  std_logic;
    );         out std_logic

end mst_pwm_deadtime;

Architecture  Behavioral  of mst_pwm_deadtime  is

signal lDIR                std_logic;
signal !Dreg               ufixed(-1 downto  -n);
signal lDreg_max           ufixed(-1 downto  -n);
signal lDreg_min           ufixed(-1 downto  -n);
signal !Count              ufixed(-1 downto  -n);
signal !Count num          std_logic_vector  (3 downto  0);
signal !Count_div          std_logic_vector (7 downto  0);
signal lQint               std_logic;
signal 1Q1                 std_logic;
signal lDreg_count         std_logic_vector(n-1 downto  0);
signal lDreg_max_count     std_logic_vector(n-1 downto  0);
signal lDreg_min_count     std_logic_vector(n-1 downto  0);
signal D max               ufixed(-1 downto  -n);
Begin

process  (iClk,iReset,iStart,lDIR,lQint,lDreg,lCount,lQ1)

Constant  C_min   ufixed(-1 downto  -n):=(others=>  '0');
Constant  C max   ufixed(-1 downto  -n):=  (others=>' ');
variable  liD     ufixed(-1 downto  -n );

Begin

if  (iReset   '0') then

    !Count      <=  (others=>'0');
    lDIR        <= '0';
    lDreg_count<=   (others=>'0');
    lDreg_max   <=  (others=>'0');
```

```
    !Dreg_min  <=  (others=>'0');
    lQint      <='0';
    lQ1        <='0';
    !Count_div <=  (otherS=>'0');
    lCount_num <=  (others=>'0');

elsif  ( rising_edge(iClk)  ) then

    D_max      <= to_ufixed(0.125,  D_max);
    !Dreg_max  <= resize(lDreg  + D_max,  lDreg_max);
    !Dreg_min  <= resize(lDreg  - D_max,  lDreg_min);

    if  (lCount_div = iClkDiv)  then

        lCount_div  <=  (others=>'0');

        if  (lDIR =  '1') then

            if  (!Count    C_max and !Count_num    "0111") then

                !Count      <= C_max;
                !Count_num  <= lCount_num+1;

            elsif  (!Count= C_max and !Count_num  /=  "0111") then

                lDIR        <= '0';
                oDone       <= '0';
                !Count      <= resize(lCount-D_max,   !Count);
                !Count_num  <= !Count_num+1;

            else

                !Count      <= resize(lCount+D_max,lCount);
                !Count_num  <= lCount_num+1;
                oDone       <= '0';

            end if;

        else

            if  (!Count= c_min  and !Count_num    "1111") then

                !Count      <= c_min;
                lCount_num  <= "0000";

            elsif  (!Count= c_min  and lCount_num  /=  "1111") then

                lDIR        <='1';
```

```
                oDone      <= '1';
                !Count     <=resize(lCount+D_max,lCount);
                lCount_num <= lCount_num+1;

            else

                oDone      <= '0';
                !Count     <=resize(lCount-D_max, !Count);
                lCount_num <= lCount_num+1;

            end if ;

        end if ;

    else

    !Count div <= lCount_div + 1;

    end if;
-------------------------------------------------------------------

    if( !Dreg = C_max) then
       lQint <= '0';
    else
       if (!Count >= lDreg_max) then
           lQint <='1';
       else
           lQint <='0';
       end if;
    end if;
-------------------------------------------------------------------

    if( !Dreg= C_min) then
       lQ1 <= '0';
    else
       if ( !Count <= lDreg_min )then
           lQ1 <= '1';
       else
           lQ1 <='0';
       end if;
    end if;
-------------------------------------------------------------------

    if ( iStart ='1') then
       liD   := to_ufixed(iD, liD);
       !Dreg <= liD;
    end if;
-------------------------------------------------------------------
```

```
end if;
end process  ;
oQ      <= lQint;
oQl     <= lQl;
oQ_inv  <= not  (lQint);
        oQl_inv <= not  (lQl);
          end Behavioral
```

BIBLIOGRAPHY

[1] C.H.Houpis and G.B.Lamont. “Introduction,” in Digital Control Systems, S.W.Director, New York: McGraw-Hill, 1985, pp.1-26.

[2] K.M.Moudgalya. “Introduction,” in Digital Control, Great Britain: John Wiley & Sons, Ltd, 2007, pp.1-

[3] K.M.Moudgalya. “Modeling of Sampled Data Systems,” in Digital Control, Great Britain: John Wiley & Sons, Ltd, 2007, pp.14-16.

[4] B.W.Bequette. Process Control: Modeling, Design and Simulation. Prentice-Hall, Inc., Upper Saddle River, NJ, 2003.

[5] J.C.Proakis and D.G.Manolakis. “The Z-transform,” in Digital Signal Processing: Principles, Algorithms and Applications, Prentice-Hall, Inc., Upper Saddle River, NJ, 3rd edition, 1996, pp.80-110.

[6] Y.F.Liu and P.C. Sen. “Digital control of switching power converters,” in Proc. IEEE Conference on Control Applications, 2005, pp. 635-640.

[7] Dunlap, S.; Fiez, T. "Design of a delta-sigma modulated switching power supply," *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on* , vol.1, no., pp.189-192 vol.1, 31 May-3 Jun 1998

[8] Yan-Fei Liu; Xiaodong Liu; , "Recent developments in digital control strategies for DC/DC switching power converters," *Power Electronics and Motion Control Conference, 2009. IPEMC '09. IEEE 6th International* , vol., no., pp.307-314, 17-20 May 2009

[9] Martin, T.W.; Ang, S.S.; , "Digital control for switching converters," *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on* , vol.2, no., pp.480-484 vol.2, 10-14 Jul1995

[10] Duan, Y.; Jin, H.; , "Digital controller design for switchmode power converters," *Applied Power Electronics Conference and Exposition, 1999. APEC '99. Fourteenth Annual* , vol.2, no., pp.967-973 vol.2, 14-18 Mar 1999

[11] Al-Atrash, H.; Batarseh, I.; , "Digital Controller Design for a Practicing Power Electronics Engineer," *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE* , vol., no., pp.34-41, Feb. 25 2007-March 1 2007

[12] C.B. Bettin. “Effects of quantization on the stability of feedback control systems.” M.S.E.E. thesis, University of Washington, Seattle, 1961.

[13] S.A.White. “Quantizer-Induced Digital Controller Limit Cycles.” IEEE Transactions on automatic Control, pp.430-432,Aug. 1969.

[14] Wu, A.M.; Jinwen Xiao; Markovic, D.; Sanders, S.R.; , "Digital PWM control: application in voltage regulation modules," *Power Electronics Specialists Conference, 1999. PESC 99. 30th Annual IEEE* , vol.1, no., pp.77-83 vol.1, Aug 1999

[15] Peterchev, A.V.; Sanders, S.R.; , "Quantization resolution and limit cycling in digitally controlled PWM converters," *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual* , vol.2, no., pp.465-471 vol.2, 2001

[16] Peng, H.; Prodic, A.; Alarcon, E.; Maksimovic, D.; , "Modeling of Quantization Effects in Digitally Controlled DC–DC Converters," *Power Electronics, IEEE Transactions on* , vol.22, no.1, pp.208-215, Jan. 2007

[17] Gu-Yeon Wei; Horowitz, M. "A fully digital, energy-efficient, adaptive power-supply regulator ," *Solid-State Circuits, IEEE Journal of* , vol.34, no.4, pp.520-528, Apr 1999

[18] Peterchev, A.V.; Jinwen Xiao; Sanders, S.R.; , "Architecture and IC implementation of a digital VRM controller," *Power Electronics, IEEE Transactions on* , vol.18, no.1, pp. 356- 364, Jan 2003

[19] Dancy, A.P.; Chandrakasan, A.P.; , "Ultra low power control circuits for PWM converters," *Power Electronics Specialists Conference, 1997. PESC '97 Record., 28th Annual IEEE* , vol.1, no., pp.21-27 vol.1, 22-27 Jun 1997

[20] Zhengyu Lu; Zhaoming Qian; Yong Zeng; Wenxi Yao; Guozhu Chen; Yousheng Wang; , "Reduction of digital PWM limit ring with novel control algorithm ," *Applied Power Electronics Conference and Exposition, 2001. APEC 2001. Sixteenth Annual IEEE* , vol.1, no., pp.521-525 vol.1, 2001

[21] Magrath, A.J.; Sandler, M.B. "Hybrid pulse width modulation/sigma-delta modulation power digital-to-analogue converter," *Circuits, Devices and Systems, IEE Proceedings* - , vol.143, no.3, pp.149-156, Jun 1996

[22] R. Schreirar and G. C. Temes, Understanding Delta-Sigma Data Converters.New York: Wiley, 2005.

[23] Z. Lukic, K. Wang, and A. Prodic, “High-frequency digital controller for dc–dc converters based on multi-bit sigma–delta pulse-width modulation,”in Proc. IEEE APEC Conf., 2005, pp. 35–40.

[24] A. Kelly and K. Rinne, “High resolution DPWM in a dc–dc converter application using digital sigma–delta techniques,” in Proc. IEEE PESC Conf., 2005, pp. 1458–1463.

[25] Lukic, Z.; Rahman, N.; Prodic, A.; , "Multibit Σ–Δ PWM Digital Controller IC for DC–DC Converters Operating at Switching Frequencies Beyond 10 MHz," *Power Electronics, IEEE Transactions on* , vol.22, no.5, pp.1693-1707, Sept. 2007

[26] Dunlap, S.K.; Fiez, T.S.; , "A noise-shaped switching power supply using a delta-sigma modulator," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.51, no.6, pp. 1051- 1061, June 2004

[27] Paramesh, J.; von Jouanne, A.; , "Use of sigma-delta modulation to control EMI from switch-mode power supplies," *Industrial Electronics, IEEE Transactions on* , vol.48, no.1, pp.111-117, Feb 2001

[28] Kheraluwala, M.H.; Divan, D.M.; , "Delta modulation strategies for resonant link inverters," *Power Electronics, IEEE Transactions on* , vol.5, no.2, pp.220-228, Apr 1990

[29] Manjrekar, M.; Venkataramanan, G.; , "Advanced topologies and modulation strategies for multilevel inverters," *Power Electronics Specialists Conference, 1996. PESC '96 Record., 27th Annual IEEE* , vol.2, no., pp.1013-1018 vol.2, 23-27 Jun 1996

[30] Dunlap, S.; Fiez, T.; , "Design of a delta-sigma modulated switching power supply," *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on* , vol.1, no., pp.189-192 vol.1, 31 May-3 Jun 1998

[31] Prodic, A.; Maksimovic, D.; Erickson, R.W.; , "Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter," *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE* , vol.2, no., pp.893-898 vol.2, 2001

[32] Qiu, Yang; Li, Jian; Xu, Ming; Ha, Dong S.; Lee, Fred C.; , "Proposed DPWM Scheme with Improved Resolution for Switching Power Converters," *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE* , vol., no., pp.1588-1593, Feb. 25 2007-March 1 2007

[33] C.H.Houpis and G.B.Lamont. Digital Control Systems, S.W.Director, New York: McGraw-Hill, 1985.

[34] R.W.Erickson and D. Maksimovic. Fundamentals of Power Electronics, New York:Kulwer Academic Publishers, 2001.

[35] P.T. Krein. Elements of Power electronics, New York: Oxford University Press Inc., 1998.

VITA

Anupama Balakrishnan completed her Bachelor of Technology in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in 2009. In May of 2013, she received her Master of Science degree in Electrical Engineering with specialization in Power Electronics from Missouri University of Science and Technology, Rolla, MO. After graduation she joined Yaskawa America, Inc., as Product Engineer in the Product Development group.