POLITECNICO DI TORINO

Department of Electronics and Telecommunications Master's Degree in Electronic Engineering

Master's degree thesis

Limit-cycle Free Digitally Controlled Power Converter

Supervisor:
Prof. Musolino Francesco
Co-Supervisor:
Prof. Crovetti Paolo Stefano

Author:
Abdullah Ahmed

ACADEMIC YEAR 2019/2020

Acknowledgements

First, I am thankful to prof. Francesco Musolino and prof. Paolo S. Crovetti for their great supervision, technical support, encouragement, motivation and for providing necessary guidance concerning thesis. They helped me a lot and whenever I feel any kind of problem, they are always tried their best to provide solution.

I am also grateful to Higher Education Commission (HEC) of Pakistan and Politecnico di Torino, Italy for provision of expertise, technical support and financial support. Without their support, the thesis would lack in quality of outcomes, and thus their support has been essential.

I would like to express my sincere thanks towards thesis commission who devoted their time and give me a great support and hope.

Nevertheless, I express my gratitude toward my family and colleagues for their kind co-operation and encouragement which help me in completion of this thesis.

Abstract

Control of switch-mode power converters, which has been traditionally carried out using analog circuits, is nowadays preferably performed digitally due to greater flexibility, higher integrability, less sensitivity to thermal changes, lower costs and possibility to implement advanced control techniques and, for these reasons, it has become a major research focus in power electronics field.

Despite the mentioned advantages digitally controlled power converters could cause the onset of non-linear issues due to inherent quantization effect of analog to digital converter (ADC) and digital pulse width modulator (DPWM) which converts the controller output from digital into analog signals. Such quantization effects cause steady-state limit-cycle oscillations (LCOs) that are a major concern. The mitigation or the reduction of these problems have been a topic of extensive research in the field of power electronics and design guidelines for LCO-free operation have been formulated. More specifically, it has been shown that the resolution of DPWM should be higher than that of ADC for LCO-free operation. Meeting the above guideline, unfortunately, results either in a limited DC accuracy and/or in increased cost and complexity especially for converters operating at high switching frequency (MHz range) taking advantage of emerging semiconductor device technology (GaN, SiC power transistors).

In this thesis, an innovative technique intended to increase the resolution of the DPWM for LCO-free operation of switch-mode power converters is analyzed and experimentally evaluated. More precisely, the novel Dyadic Digital PWM (DDPWM) is adopted as a systematic approach that generalizes and extends the standard PWM dithering techniques to achieve accurate, LCO-free operation in a digital converter at negligible cost and design effort and without any detrimental effect on the output ripple voltage.

A digitally controlled DC-DC Boost converter is considered to validate the approach proposed in the thesis. This choice is justified by the growing importance of the Boost converter in a variety of practical applications concerning renewable energy technologies, power efficiency and power quality. Among the various applications, the

boost is employed as one of the main building blocks in photovoltaic power systems, in battery powered and portable device applications, in the regenerative braking of motors and in power factor correction (PFC) circuits to enhance the power quality of power supplies.

In this thesis, the converter is designed to operate efficiently over a range of the input voltages and to supply a constant current to the external load, and the effectiveness of the DDPWM in mitigating the onset of the LCOs is verified versus different operating conditions and digital control parameters. More specifically, the converter is designed to be operated in continuous-conduction-mode (CCM) with a switching frequency in the MHz range while voltage-mode digital control algorithm is considered by referring to the implementation form of the PID compensator. Simulation tools, in these phases, are employed to optimize the design of the power stage and the controller with the aim to analyze both the static and the dynamic performance of the converters.

Contents

Page

Acknowledgements 2

Abstract 3

List of Figures 7

List of Tables 8

1. Introduction 9
   1.1. Problem Statement 9
   1.2. Objectives 11
   1.3. Thesis Outline 12
2. Limit-cycles Oscillation 13
   2.1. Onset of Limit-cycle Oscillations 13
   2.2. LCOs free design guidelines 14
   2.3. Amplitude Quantization 17
3. Dyadic Digital Pulse Width Modulation (DDPWM) 20
   3.1. Dyadic Digital Pulse Modulator (DDPM) 21
   3.2. Dyadic digital pulse width modulation (DDPWM) 22
4. Implementation & Approach 24
   4.1. Boost converter Continuous Time Modeling 24
      4.1.1. PID Controller Modeling 26
   4.2. Boost Converter Discrete Time Modeling 29
      4.2.1. Digital PID compensator 30
   4.3. Digital PID compensator Implementation 32
      4.3.1. PID Coefficients Scaling and Quantization 32
      4.3.2. Fixed Point PID controller implementation in HDL 34
      4.3.3. DDPWM Implementation in HDL 37
   4.4. Simulation-based Experimental Methodology 38
      4.4.1. MATLAB-Simulink for Power Converter's Simulation 38
      4.4.2. Modelsim PE for Digital PID compensator and DDPWM realizations 39
      4.4.3. Co-simulation of Simulink and Modelsim 39
   4.5. Hardware Implementation Setup 42
      4.5.1. ADC Input & Output Interfacing 43
      4.5.2. Boost converter PCB designing and connectivity with EPC2007 Board 44

4.5.3. FPGA Cyclone-IV DE2-115 for HDL Synthesis 46
5. Result & Analysis 47
5.1. Simulation-based Results 47
5.1.1. DPWM controller 47
5.1.2. DDPWM controller 50
6. Conclusion 54
Appendices 55
A. Matlab code for continuous-time boost converter and PID controller 55
B. Matlab code for discrete-time boost converter and PID controller 57
C. Matlab code to find hardware-dynamic ranges of all signals involved in digital PID controller 59
D. Verilog code for digital PID controller 60
References 64

List of Figures

FIGURE 1.1: DIGITALLY CONTROLLED SYNCHRONOUS BOOST CONVERTER WITH PID COMPENSATOR....10
FIGURE 2.1: QUALITATIVE BEHAVIOR OF Vo (A) $NDPWM$ IS LOWER THAN $NADC$ (B) $NDPWM$ IS HIGHER THAN $NADC$ [1]....15
FIGURE 2.2: QUALITATIVE BEHAVIOR OF Vo (A) INTEGRAL TERM $Ki = 0$ (B) INTEGRAL TERM $Ki \neq 0$ IN CONTROL LAW [1]....16
FIGURE 2.3: DESCRIBING FUNCTION FOR SINUSOIDAL SIGNAL WITH ZERO OFFSET [1] 17
FIGURE 2.4: DPWM-INDUCED QUANTIZATION ON OUTPUT VOLTAGE [4] 19
FIGURE 3.1: DYADIC DIGITAL PULSE MODULATION (DDPM) PATTERN....20
FIGURE 3.2: DYADIC DIGITAL PULSE MODULATOR HARDWARE 22
FIGURE 3.3: DDPM PATTERN FOR D=293/512 WITH N=5 AND M=4 23
FIGURE 3.4: PROPOSED DDPWM ARCHITECTURE 23
FIGURE 4.1: BOOST CONVERTER CIRCUIT TOPOLOGY....24
FIGURE 4.2: BODE PLOT OF CONTINUOUS TIME TRANSFER FUNCTION OF BOOST CONVERTER 26
FIGURE 4.3: CONTINUOUS-TIME PID CONTROLLER 27
FIGURE 4.4: BODE PLOT OF POWER CONVERTER, PID CONTROLLER AND OPEN LOOP SYSTEM 29
FIGURE 4.5: TRAILING EDGE MODULATOR 30
FIGURE 4.6: BODE PLOT OF DISCRETE-TIME POWER CONVERTER, DISCRETE-TIME PID CONTROLLER AND OPEN LOOP SYSTEM....32
FIGURE 4.7: DIGITAL PID COMPENSATOR DESIGN....34
FIGURE 4.8: BLOCK DIAGRAM FOR HDL IMPLEMENTATION OF PARALLEL PID COMPENSATOR REALIZATION....37
FIGURE 4.9: DDPWM ARCHITECTURE....37
FIGURE 4.10: BOOST CONVERTER SIMULINK DESIGN 38
FIGURE 4.11: DATAFLOW DIAGRAM OF DIGITAL CONTROLLER IN MODELSIM....39
FIGURE 4.12: MATLAB CO-SIMULATION WIZARD 40
FIGURE 4.13: MODELSIM SIMULATOR BLOCK IN SIMULINK WINDOW....41
FIGURE 4.14: SIMULINK DESIGN OF COMPLETE SYSTEM....42
FIGURE 4.15: ADC INPUT INTERFACING CIRCUIT....43
FIGURE 4.16: SCHEMATIC DIAGRAM OF PCB DESIGN ON KICAD 44
FIGURE 4.17: PCB LAYOUT OF BOOST CONVERTER WITH EPC9006C BOARD....45
FIGURE 4.18: COMPLETE HARDWARE SETUP....46
FIGURE 5.1: BOOST CONVERTER OUTPUT VOLTAGE vo WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 0$....48
FIGURE 5.2: BOOST CONVERTER OUTPUT CURRENT io WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 0$....48
FIGURE 5.3: BOOST CONVERTER INDUCTOR CURRENT iL WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 0$....49
FIGURE 5.4: TIMING DIAGRAM OF DIGITAL CONTROLLER PARAMETERS WITH $NADC = 5, NDPWM = 4, NDDPM = 0$....49
FIGURE 5.5: BOOST CONVERTER OUTPUT VOLTAGE vo WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 4$....50
FIGURE 5.6: BOOST CONVERTER OUTPUT CURRENT io WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 4$....51
FIGURE 5.7: BOOST CONVERTER INDUCTOR CURRENT iL WITH $NADC = 5, NDPWM = 4$ AND $NDDPM = 4$....51
FIGURE 5.8: TIMING DIAGRAM OF DIGITAL CONTROLLER PARAMETERS WITH $NADC = 5, NDPWM = 4, NDDPM = 4$....52
FIGURE 5.9: DDPM RIPPLES AT OUTPUT VOLTAGE vo....53

List of Tables

TABLE 4.1: DESIGN SPECIFICATIONS OF BOOST CONVERTER 38
TABLE 4.2: INPUT-OUTPUT OF TOP MODULE 39
TABLE 4.3: DIGITAL CONTROLLER DESIGN SPECIFICATIONS 41
TABLE 4.4: ADA PORT AND PINS 44
TABLE 4.5: FINALIZED COMPONENTS FOR PCB LAYOUT 45
TABLE 4.6: PIN ASSIGNMENTS FOR FPGA 46

Chapter 1

1. Introduction

Switch-mode power supplies are most used and the best choice for DC-DC power conversion. These supplies provide a marked benefits and advantages in comparison with other techniques of DC-DC power conversion. For many years, switch-mode power converters were mainly controlled by using analog control circuits. Despite of having less sensitivity and technology advancement, analog controllers provide lack of flexibility, difficulty in adjusting and low reliability in comparison with digital controllers. This has made digital control more fascinating to substitute analog control. Moreover, digital controllers are more flexible, programmable, more reliable, less susceptible to aging, faster and changing a controller doesn't require any change in circuit. That's why, digital control become a major research focus in power electronics field.

1.1. Problem Statement

Despite the mentioned advantages, digitally controlled power converters could cause some major concerns and problems. Some of these problems are listed here;

The major concern in digital controllers is the onset of non-linear issues due to inherent quantization effect of analog to digital converter (ADC) which converts the control variable from analog to digital signal and digital pulse width modulator (DPWM) which converts control variable from digital to analog signal as shown in Figure 1. Such quantization effects cause low-frequency steady state limit-cycles oscillations (LCOs) that are major concerns [1]. The block diagram of synchronous boost DC-DC converter operated at $\mathrm{fs} = \frac{1}{\mathrm{Ts}}$ switching frequency with digital Proportional Integrative Derivate (PID) compensator is shown in Figure 1.1.

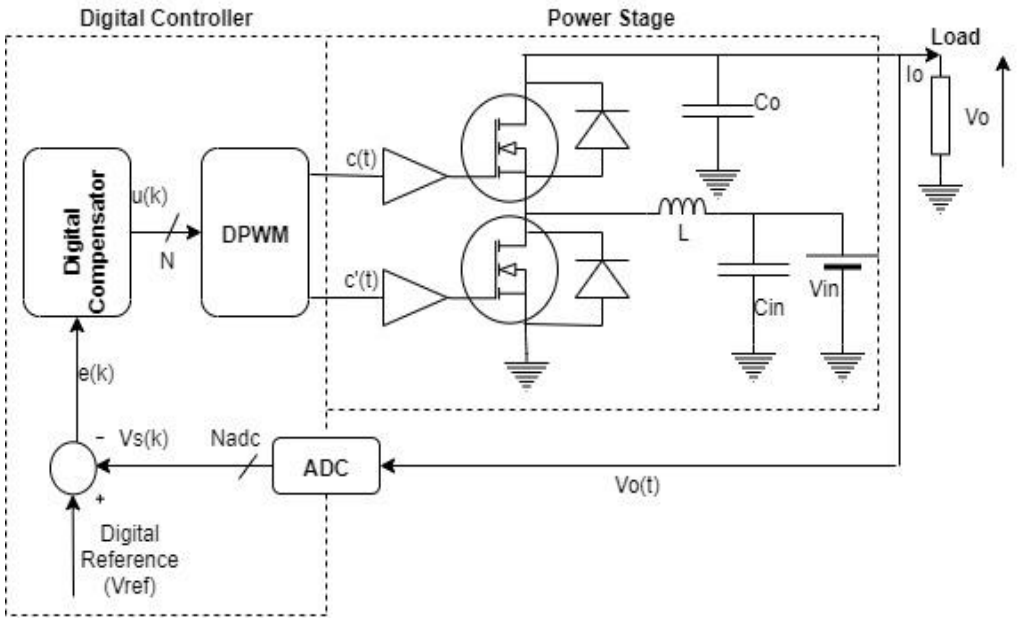


Figure 1.1: Digitally controlled synchronous boost converter with PID compensator

The LCO's issues are in broader spectrum related to limited resolution of DPWM. Since, standard counter-based DPWM converts digital signal to analog signal by comparing digital counter output with latched digital control command. So, n_{dpwm}-bit resolution DPWM requires clock frequency of $f_{clk} = 2^{n_{dpwm}} f_{sw}$. As an example, for $f_{sw} = 3MHz$ and 8-bit DPWM, clock frequency of about $750MHz$ is required which is very high. So, we must use counter clock frequency for DPWM which results in limited resolution and more power consumption for high switching frequency (MHz range).

In order to get effective high-resolution, LCO free DPWMs, specific techniques have been suggested at hardware level and is the major research focus. Most of proposes solutions, however, results either in a limited DC accuracy and /or in an increased cost and complexity.

Delay-line and ring oscillators can provide high frequency clock by using set of delay elements, but it requires large silicon area and is less immune to thermal variations, supply voltages and fabrication process [1].

Hybrid architectures of delay-line and counter-clock approaches provides tradeoff among DPWM resolution and occupied silicon area but is still sensitive to environmental

and manufacturing parameters. There are many other hardware-based techniques for the improvement of DPWM resolution like multiphase PWM and interleaved converters. Digital dithering technique can be used to control average duty cycle by varying duty cycle of sub-LSB over pre-defined dithering pattern. This way we can increase effective DPWM resolution without increasing clock frequency. Digital dithering is easy to implement and highly controllable, but it may introduce harmonic distortion due to noise at switching frequency sub-harmonics which cannot be filtered out properly, hence causes ripples at output voltage [1].

Sigma-delta (∑Δ) another dithering technique, is found to be more effective to enhance DPWM resolution but its conversion rate is slow and feedback loop is non-linear which provides additional ripples at output voltage [1].

1.2. Objectives

The main objective is to build power converter with digital controller to have no limit-cycle oscillations at output. It can be achieved by Dyadic Digital Pulse Width Modulation (DDPWM), an innovative technique intended to increase the resolution of the DPWM for LCO-free operation of switch-mode power converters. More precisely, the novel Dyadic Digital PWM (DDPWM) is adopted as a systematic approach that generalizes and extends the standard PWM dithering techniques to achieve accurate, LCO-free operation in a digital converter at negligible cost and design effort and without any detrimental effect on the output ripple voltage.

A digitally controlled DC-DC Boost converter is considered to validate the approach proposed in the thesis. This choice is justified by the growing importance of the boost converter in a variety of practical applications concerning renewable energy technologies, power efficiency and power quality. Among the various applications, the boost is employed as one of the main building blocks in photovoltaic power systems, in battery powered and portable device applications, in the regenerative braking of motors and in power factor correction (PFC) circuits to enhance the power quality of power supplies.

The converter is designed to operate efficiently over a range of the input voltages and to supply a constant current to the external load, and the effectiveness of the DDPWM in mitigating the onset of the LCOs is verified versus different operating conditions and digital control parameters. More specifically, the converter is designed to be operated in continuous-conduction-mode (CCM) with a switching frequency in the MHz range while the voltage-mode digital control algorithms is considered by referring to the implementation form of the PID compensator. Simulation tools, in these phases, are employed to optimize the design of the power stage and the controller with the aim to analyze both the static and the dynamic performance of the converters.

A printed circuit board (PCB) is designed to accommodate the power stage employing EPC2007C Half-Bridge with onboard gate drive for enhancement (eGAN) field-effect transistor while the digital control algorithm has been prototyped and implemented on a FPGA interfaced to the power stage by adopting state-of-the art ADC. The use of programmable logic allows to easily test different control strategies without major hardware changes with the aim to set the converter at different ADC and DPWM resolutions to validate the proposed method.

1.3. Thesis Outline

In chapter 2, detailed analysis about onset of LCO's in digitally controlled power converters is described and the necessary design guidelines to have LCO-free output is proposed. In chapter 3, the high-resolution DDPWM technique is proposed.

Chapter 4 describes implementation and approach which have been followed throughout thesis. Mathematical modeling, coding and designing of both power stage and digital controller have been discussed. Experimental methodology for both simulation and hardware implementation have been described.

Chapter 5 discusses the analysis of results obtained through both simulation and practical implementation. The outputs obtained through both are compared with errored output respectively and discussed how much improvement has been done.

Chapter 6 is the conclusion and future recommendation, which describes the benefits, applications, usage and future work related to project.

Chapter 2

2. Limit-cycles Oscillation

For the system with closed loop digital controller, there must be a signal amplitude quantizers like the ADC and DAC modules in the feedback mechanism. Due to the quantization error and quantization level mismatch among the signal amplitude quantizers, low-frequency steady-state oscillations are generated at the output voltage and at other system variables. These steady-state oscillations are basically the limit-cycles oscillation whose frequency is less than the converter switching frequency f_{sw}. These limit-cycles oscillations are unacceptable since it leads to large output voltage distortion. Moreover, it is difficult to discriminate between limit-cycles and electromagnetic interference (EMI), since amplitude and frequency of limit cycles are difficult to measure.

2.1. Onset of Limit-cycle Oscillations

In order to discuss creation of quantization-incited limit-cycles oscillation, consider a digitally controlled synchronous DC-DC boost converter operating at switching frequency $f_{sw} = \frac{1}{T_{sw}}$. The digital PID compensator is employed as a digital controller. The whole block diagram is reported in Figure 1.1. At the end, the proposed design guidelines for LCO-free operation are suggested.

As shown in Figure 1.1, ADC is used to convert converter's output voltage $v_o(t)$ to digital voltage $v_s[k]$ at switching frequency f_{sw}, i.e.,

$$v_s[k] = v_o(kT_{sw})$$

The quantization level of ADC is given as;

$$q_{ADC} = \frac{V_{in\_adc}}{2^{N_{ADC}}} \qquad (2.1)$$

Where V_{in_adc} is ADC input range and N_{ADC} is no. of bits of ADC.

The digital voltage $v_s[k]$ at output of ADC is then compared with constant reference digital voltage to get digital error signal $e[k]$,i.e.,

$$e[k] = v_{ref} - v_s[k]$$

The $e[k]$ is then given to digital PID controller which perform all the corrective measure to compensate for error and resulting duty cycle is produced. The N_{DPWM}-bit DPWM modulator is then used to generate PWM signal of required duty cycle for driving converter switches. DPWM modulator is operating at $f_{clk} = 1/T_{clk}$ and since there are fixed quantization intervals of DPWM modulator, the active period Ton is restricted to be an integer multiple of T_{clk} resulting in a quantized duty cycle $D = T_{on}/T_s$. As a result, there is non-equilibrium behavior at the output voltage, such as steady state limit cycling. [2]

2.2. LCOs free design guidelines

There are some design guidelines which must be followed in order to achieve LCO-free operation, which are;

1) The resolution of DPWM must be greater than the resolution of ADC, i.e., [1]

$$N_{DPWM} > N_{ADC}$$

Each time controller is trying to drive output to zero-error bin, but if above condition is not respected, there is no DPWM quantization interval lies into the zero-error bin, therefore output is oscillating around the zero-error bin at DPWM levels as shown in Figure 2.1(a) [1].

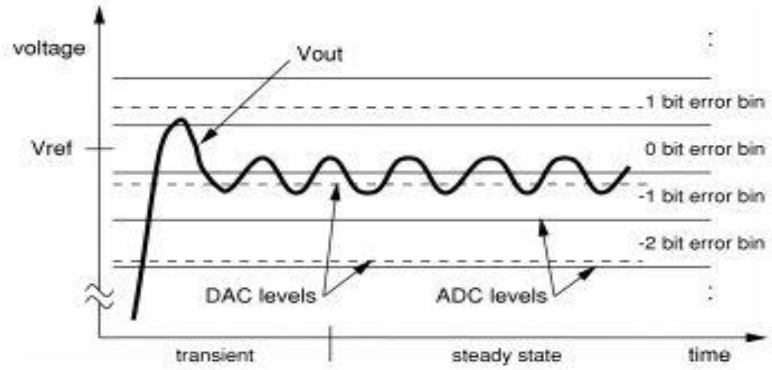
Above condition means that ADC quantization interval (q_{ADC}) is higher than the DPWM quantization (q_{DPWM}). Even if we have $N_{DPWM} = N_{ADC} + 1$, it is guaranteed that one of the DPWM level lies into zero-error bin.

But meeting above condition requires a tradeoff among f_{clk}, f_{sw} and N_{ADC}. The lower N_{ADC} results in DC inaccuracy of output voltage. For example, for a clock frequency $f_{clk} = 10MHz$, $f_{sw} = 150kHz$, The N_{DPWM} are found to be;

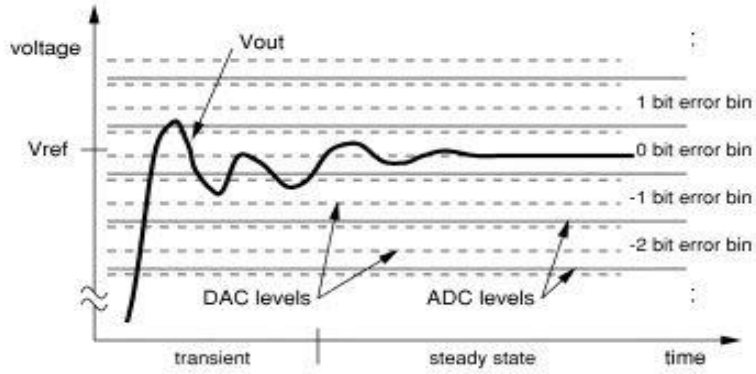
$$2^{N_{DPWM}} = \frac{f_{clk}}{f_{sw}}$$

$$N_{DPWM} = 7$$

Meeting (1) requires $N_{ADC} = 6$ bits or less, which is very low for most practical applications. Alternatively, for high DC accuracy $N_{ADC} = 9$ bits and switching frequency $f_{sw} = 5Mhz$ requires $f_{clk} = 5Ghz$ at least which is again impractical high clock frequency.



(a)



(b)

Figure 2.1: Qualitative behavior of V_o (a) N_{DPWM} is lower than N_{ADC} (b) N_{DPWM} is higher than N_{ADC} [1]

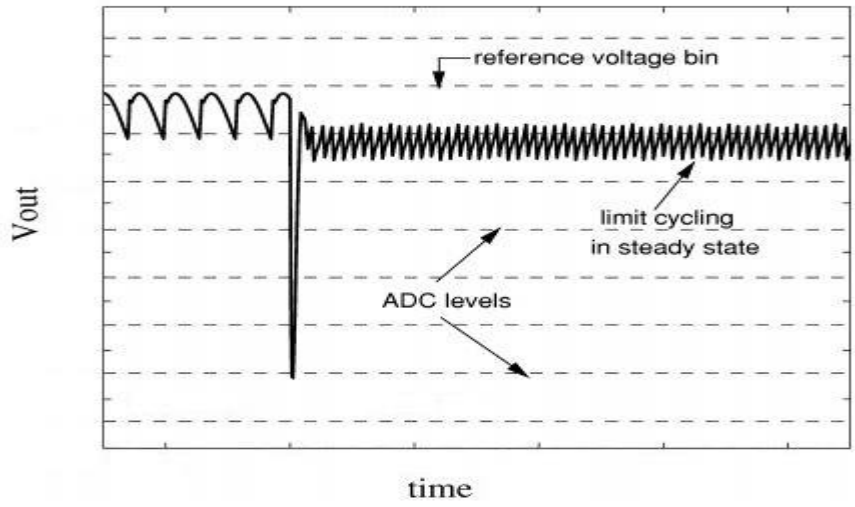
2) There must be an integral gain K_i in the PID control law, i.e.,[1]

$$0 < K_i \leq 1$$

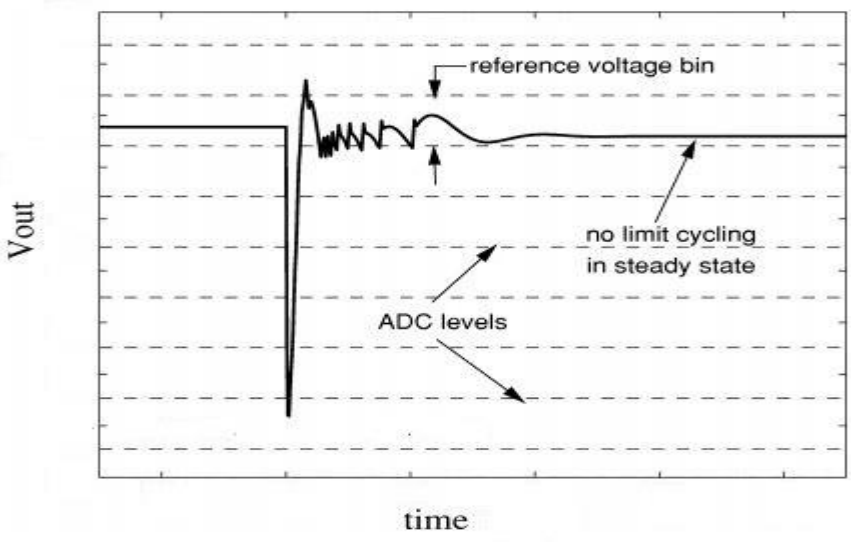
In case of $K_i = 0$ the controller drives output towards zero-error bin if there is non-zero error signal. When output reaches zero-error bin, the error signal becomes zero and again output comes out of zero-error bin. This cycle continues again and again resulting in the limit-cycle oscillations at output as shown in Figure 2.2(a) [1].

But having an integral term K_i in the controller, the integrator output will gradually be approaching zero-error bin and will remain there as long as error signal is zero. The

Figure 2.2(b) [1] shows the transient response of PWM converter having $K_i \neq 0$ in digital control law.



(a)



(b)

Figure 2.2: Qualitative behavior of V_o (a) Integral term $K_i = 0$ (b) Integral term $K_i \neq 0$ in control law [1]

3) Nyquist stability criterion should be followed for high loop gains, i.e.,[1]

$$1 + L(j\omega)N(A) \neq 0$$

where $N(A)$ is the describing function for non-linear quantizers, A is the AC amplitude for ADC and $L(j\omega)$ is loop gain.

While dealing with high loop gains, the first two requirements are not more sufficient for LCO free operation. The non-linearity and high gain distortion of the both ADC and DPWM in the closed loop causes limit-cycling [21].

To eliminate limit cycles while dealing high loop gains, one must determine maximum allowable loop gain by properly studying describing function of ADC. The describing function can be determined by using non-linear system analysis tools. Figure 2.3 [1] shows the describing function of ADC for sinusoidal signal with zero offset.

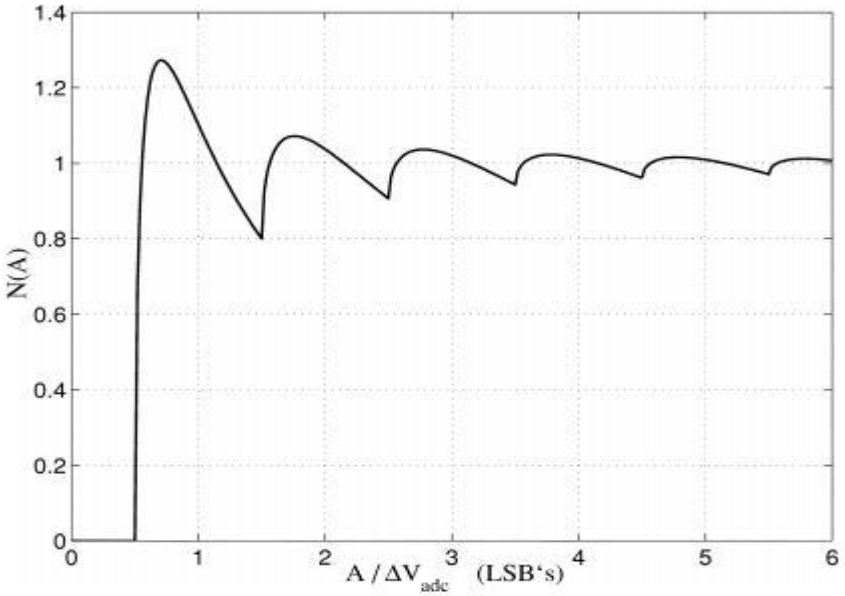


Figure 2.3: Describing function for sinusoidal signal with zero offset [21]

2.3. Amplitude Quantization

As we know that, there is a zero-error bin which represents the particular quantization level of sensed signal v_s at controller digital set point v_{ref}. At $v_s = v_{ref}$, we have steady-state point, i.e., $e = 0$, which means that the sampled version of the sensed analog signal lies inside the zero-error bin. [4]

Suppose H_0 is the dc-gain of conditioning circuitry, so the quantization interval of v_o is,

$$q_{v_o}^{ADC} = \frac{q_{v_o}^{ADC}}{H_0} = \frac{V_{FS,o}}{2^{N_{ADC}}}$$

Where as $V_{FS,o} = \frac{V_{FS}}{H_o}$ is the equivalent ADC conversion range on output voltage.

Suppose that the regulation bin q_{v_o} must be less than $\in$ % of the nominal output voltage v_{ref}/H_o, so from above eq. we get,

$$N_{ADC} > log_2\left(\frac{100}{\varepsilon}\right) + log_2(\frac{V_{FS}}{v_{ref}})$$

For instance, $\epsilon = 1$ and $\frac{V_{FS}}{V_{ref}} = 2$, $n_{ADC} = 8$ bits of resolution are required.

Moving toward the DPWM, it only produces quantized duty cycle which can be equivalently modeled as a quantization on the control command $u[k]$. Given that q_d is the smallest duty cycle variation and q_u is the smallest command variation,

$$q_u = q_D N_r = \frac{N_r}{2^{N_{dpwm}}}$$

The control command $u[k]$ can be modeled as,

$$u[k] = Q_{DPWM}\big[u[k]\big] = N_r Q_D \left[\frac{u[k]}{N_r}\right]$$

As we know that,

$$M(D) = \frac{v_o}{v_{in}}$$

$$v_o = M(D)\, v_{in}$$

At the steady state, the smallest variation $q_D$produces a variation q_{v_o} which is approximately equal to,

$$q_{v_o}^{DPWM} = \frac{\partial M}{\partial D}_{D^o}\, q_D v_{in} \qquad (2.2)$$

In our case of boost converter, we know that

$$M(D) = \frac{1}{1-D} = \frac{\partial M}{\partial D}_{D^o} = \frac{1}{(1-D^o)^2}$$

So, eq. 2.2 becomes,

$$q_{v_o}^{DPWM} = \frac{1}{(1-D^o)^2} q_D v_{in}$$

More compactly it can be expressed in term of DC gain G_{vd_0}, i.e.,

$$\frac{\partial M}{\partial D} v_{in} = G_{vd_0}(s=0) = N_r G_{vu}(z=1)$$

In general, DPWM-induced quantization on output voltage for boost converter shown in Figure 2.4 can be written as,

$$q_{v_o}^{DPWM} = G_{vd_0}(s=0) q_D = G_{vu}(z=1) q_u \qquad (2.3)$$

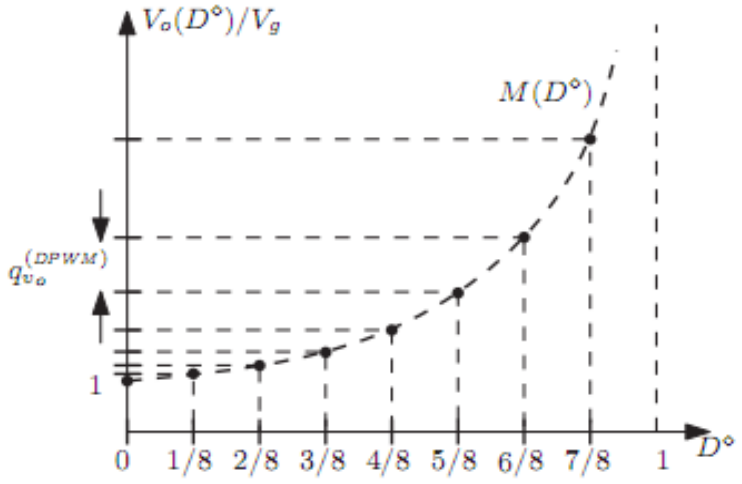


Figure 2.4: DPWM-induced quantization on output voltage [4]

Chapter 3

3. Dyadic Digital Pulse Width Modulation (DDPWM)

DDPM of M-bit binary number 'm' is a long digital stream $\sum_m(t)$ of "one" digital pulses equal to the binary number 'm', i.e.,

$$\sum\nolimits_m(t) = \sum_{i=0}^{M-1} b_i S_i(t) \qquad 0 < t < T \qquad (3.1)$$

Where $b_i \in \{0,1\}$ is the Boolean value, $i = 0,1, \dots, M$

$S_i(t)$ are the dyadic basis signals which contains "one" pulses of 2^i and "zero" pulses of $2^{M-i} - 1$. For all $i = 0,1, \dots, M$, $S_i(t)$ are made orthogonal to each other, i.e., $(S_i(t).S_k(t) = 0, \forall i \neq k)$. With reference to Figure 3.1, DDPM pattern for $M = 4$ are drawn and digital stream for $m = 11$ is shown.

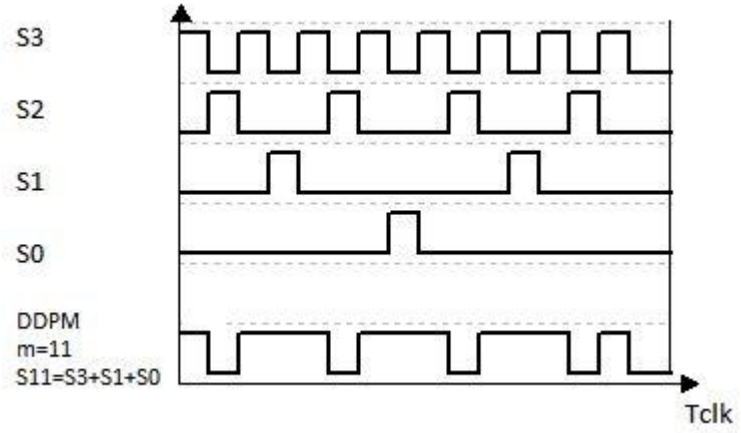


Figure 3.1: Dyadic Digital Pulse Modulation (DDPM) pattern

The expression for the frequency domain of DDPM is,

$$X_m^{DDPM}(f) = \sum_{k=-\infty}^{+\infty} a_k C_{k.m} \delta(f - k f_0) \qquad (3.2)$$

Where $f_o = 1/T_o$

$$a_k = \frac{1}{2^M} sinc\left(\frac{k}{2^M}\right) e^{-\frac{jk\pi}{2^M}} \quad (3.3)$$

$$C_{k,m} = \sum_{i=0}^{M-1} b_{i,m} 2^i \sum_{p=0}^{2^{M-i}-1} (-1)^p \delta\left[k - 2^i p\right] \quad (3.4)$$

$\delta[.]$ is the Kronecker function defined as

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (3.5)$$

The spectral component found from (3) for different m is

$$S(kf_0) = \max_m |X_m^{DDPM}(kf_0)| \quad (3.6)$$

From above equations the qualitative analysis of spectral component has been carried out. The AC spectra's of DDPM are intended at the high frequencies where they can be easily filtered out by employing a first order filter of cut-off frequency $f_c = f_0/\sqrt{3}$.

3.1. Dyadic Digital Pulse Modulator (DDPM)

The above whole discussion can be implemented on hardware by using priority multiplexer and simple binary counter. The multiplexer has $D_{M-1} \dots D_0$ data inputs, $S_{M-1} \dots S_0$ selection inputs and one output $'O'$. The following Boolean function is implemented by mux, [2]

$$OUT = \sum_{i=0}^{M-1} D_{m-i-1}.S_i.\prod_{k=0}^{i-1} \overline{S_k} \quad (3.7)$$

The data inputs of mux are fed with M-bit data register sampled at f_o while the selection inputs are driven by M-bit counter clocked at $2^M f_o$. Each time new counter value arrives at selection inputs which is checked for 'one' starting from LSB where k is the index of first one (i.e. $k = 1$ for LSB up to $K = M$ for MSB). The mux priority is arranged in such a way that mux output 'O' takes the value of D_{M-k} of data input. The output 'O' is kept zero in case when all selection inputs are zero. The DDPM modulator hardware circuit diagram is shown in Figure 3.2.

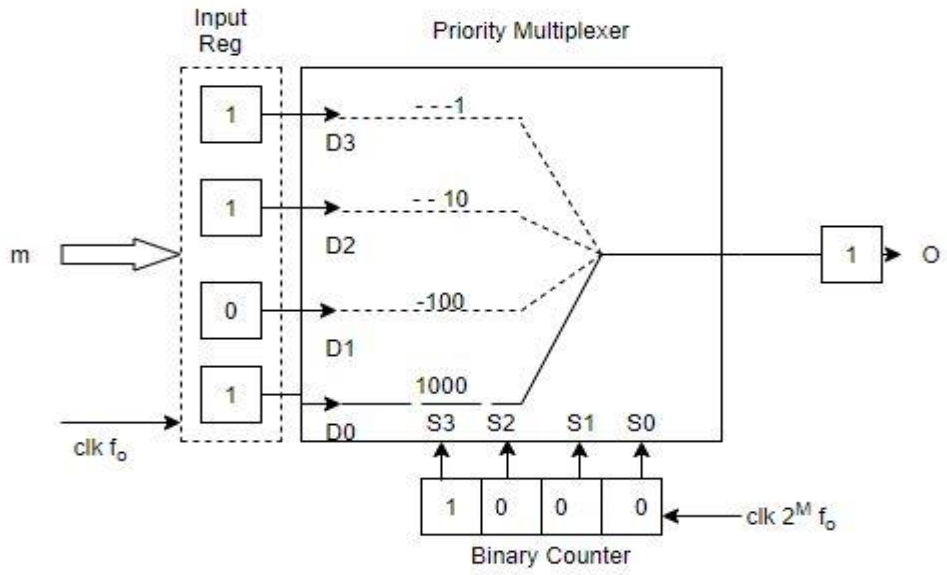


Figure 3.2: Dyadic Digital Pulse Modulator Hardware

3.2. Dyadic digital pulse width modulation (DDPWM)

As we discussed in the previous chapter, there is a tradeoff among f_{clk}, f_{sw} and N_{DPWM} so DC accuracy. For having high resolution of DPWM, we need very high clock frequency (Ghz range) which is impractical in most applications. In this thesis, the effective resolution of N-bit DPWM modulator is increased to N+M bit by employing DDPWM technique to achieve LCO's free operation without trading off among DC accuracy and clock frequency.

[3] The value obtained from PID compensator is separated in n and m, where n is the value represented in first N-bits while m is the value represented in last M-bits. According to the technique, duty cycle is varying among two adjacent quantization levels, i.e.,

$$D_o = \frac{n}{2^n}$$

$$D_1 = \frac{n+1}{2^n}$$

So, the average duty cycle over period $2^M T_{clk}$ is calculated as,

$$D_o = \frac{n}{2^n}.\frac{2^M - m}{2^M} + \frac{n+1}{2^N}.\frac{m}{2^M} = \frac{n.2^M + m}{2^{N+M}} \qquad (3.8)$$

As an example, for $D = 593/512, N = 5$ and $M = 4$, the whole process is explained in Figure 3.3 below;

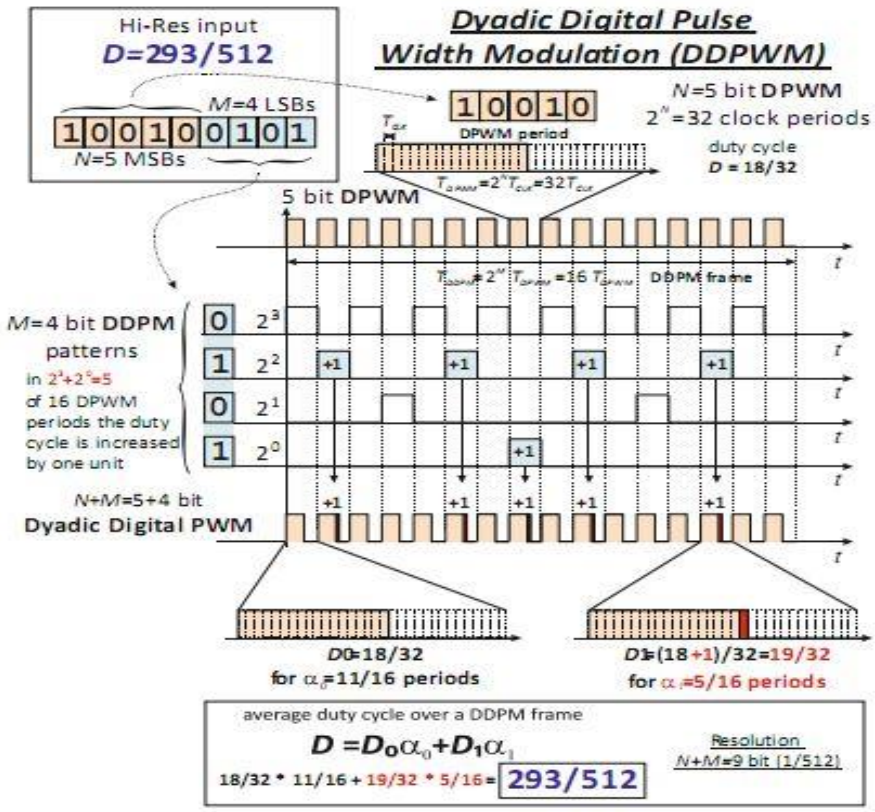


Figure 3.3: DDPM pattern for D=293/512 with N=5 and M=4

DDPWM technique can be practically implemented by adding DDPM modulator along with DPWM as shown in Figure 3.4. This way we can increase the effective resolution of DPWM so the ADC resolution and hence achieve the LCO's free operation condition.

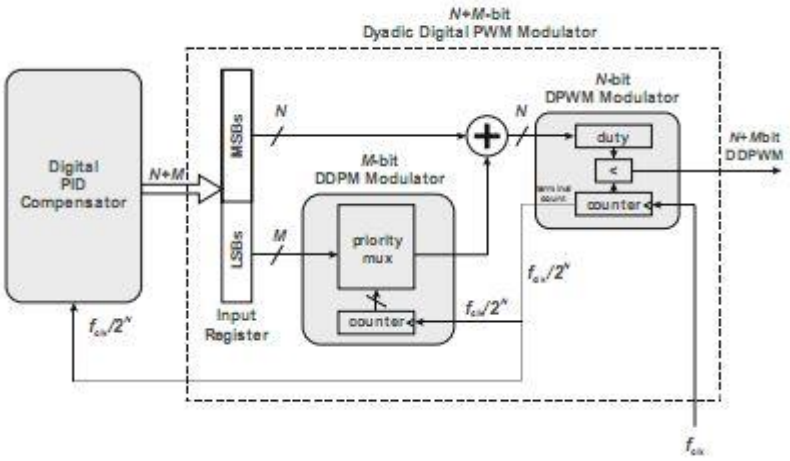


Figure 3.4: Proposed DDPWM Architecture

Chapter 4

4. Implementation & Approach

A digitally controlled synchronous DC-DC Boost converter is considered to validate the approach proposed in the thesis. This choice is justified by the growing importance of the boost converter in a variety of practical applications concerning renewable energy technologies, power efficiency and power quality. Among the various applications, the boost is employed as one of the main building blocks in photovoltaic power systems, in battery powered and portable device applications, in the regenerative braking of motors and in power factor correction (PFC) circuits to enhance the power quality of power supplies.

4.1. Boost converter Continuous Time Modeling

Let's start the discussion with the continuous time modeling of boost converter designed to be operated in continuous-conduction-mode (CCM) with a switching frequency in the MHz range while the voltage-mode (VM) digital control algorithm is considered.

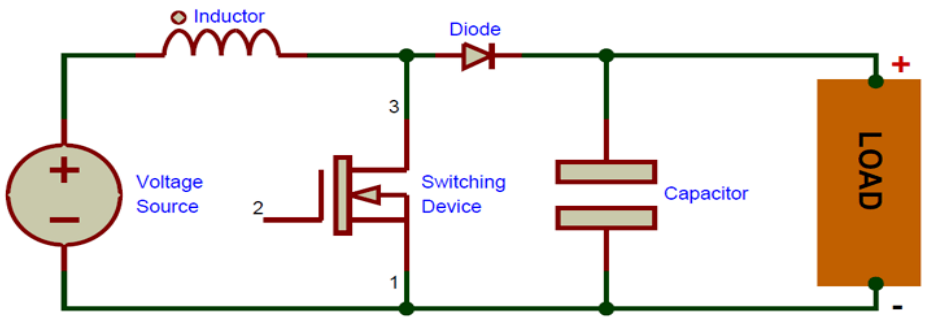


Figure 4.1: Boost converter circuit topology

The Transfer function of power stage for boost converter as shown in Figure 4.1 is [5],

$$\frac{v_o}{d} = \frac{v_{in}}{(1-D)^2} \frac{(1 - \frac{l}{R(1-D)^2}\ S)(1 + s.ESR.C_o)}{1 + \frac{L}{R}\frac{1}{(1-D)^2}S + \frac{LC_o}{(1-D)^2}S^2} \qquad (4.1)$$

Where,

There are double moving poles due to LC_o resonance and their frequency is,

$$f_{p_{1,2}} = f_{LC-resonant} = \frac{1 - D_{max}}{2\pi\sqrt{LC_o}}$$

During switch-off, the inductance is hanging so introduces right half plane (RHP) zero which makes control difficult. RHP zero is function of inductor (smaller is better) and load resistance, i.e.,

$$f_{RHP-z} = (1 - D_{max})^2 \frac{R_{min}}{L}$$

In terms of magnitude RHP zero has same effect as regular zero but in phase it introduces 90° delay instead of 90° boost.

The electrostatic series resistance (ESR) of output capacitor C_o introduces zero in transfer function whose frequency is,

$$f_{ESR-Z} = \frac{1}{2\pi\ ESRC_o}$$

The power gain of converter is,

$$Power\_gain\ = \frac{v_{in_{min}}}{(1 - D_{max})^2 V_{TR}}$$

The design constraints for cross-over frequency f_c is,

$$f_C < \frac{1}{10} f_{sw}$$

$$f_c < \frac{1}{4} f_{RHP-Z}$$

$$f_c \geq 2 f_{LC-resorant}$$

Design specifications of the converter:

Input Voltage $Vin = \ 7V - 10V$
Output Voltage $Vout = \ 12V$
Output capacitor $C_o = 3uF, ESR = 45m\Omega$
Switching Frequency $f_{sw} = 3MHz$
Load Resistor $R = \ 25\Omega - 30\Omega$

For inductance use the relation [3], $L_{max} = C\left(\frac{R_{min}}{M}.\frac{Vin_{min}}{Vo}\right)^2 = 900nH$

Cross-over frequency $f_c = \frac{f_{sw}}{10} = 300KHz$

Referring to MATLAB code in Appendix A, transfer function of power converter has been calculated for above specifications by strictly following all the design constraints and is reported in Figure 4.2.

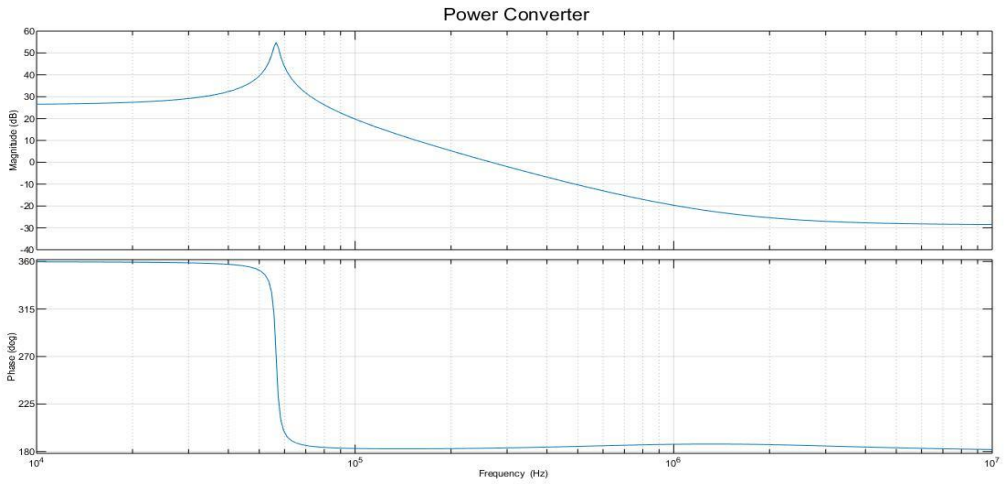


Figure 4.2: Bode plot of continuous time transfer function of boost converter

$$\frac{V_o(s)}{d(s)} = \frac{-2.938 * 10^{-13}\, s^2 + \ 6.008 * 10^{-7} s \ + \ 20.57}{7.935 * 10^{-12}\, s^2 + \ 1.058 * 10^{-7}\, s \ + \ 1} \qquad (4.2)$$

$$f_{RHP-z} \approx 1.4299\ MHz$$

$$f_{LC-res} \approx 60\ KHz$$

$$f_{RHP-z} \approx 1.15 MHz$$

4.1.1. PID Controller Modeling

Next consider the PID controller (shown in Figure 4.3) transfer function to control CCM VM boost converter,

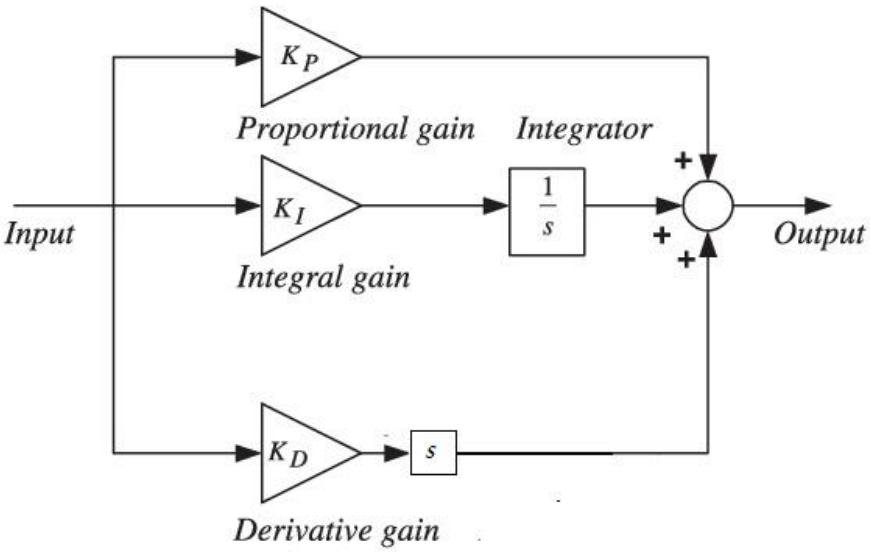


Figure 4.3: Continuous-time PID controller

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d\, s}{(1 + \frac{s}{N})}$$

We also must add one pole 'N' with differentiator in order to avoid high frequency noise

Rearranging we get,

$$C(s) = \frac{\left(\frac{K_p}{N} + K_d\right)s^2 + \left(K_p + \frac{K_i}{N}\right)s + K_i}{s(1 + \frac{s}{N})} \qquad (i)$$

Also consider the controller having proportional-integral (PI) cascaded with proportional-derivative (PD),

$$C(s) = K\frac{(1 + T_1 s)}{T_1 s}\frac{1 + T_2 s}{(1 + \frac{s}{N})}$$

It can also be written as;

$$C(s) = \frac{(K\,T_2)s^2 + \frac{K(T_1 + T_2)}{T_1}s + \frac{K}{T_1}}{s(1 + \frac{s}{N})} \qquad (ii)$$

By comparing above two equations for controller, we get;

$$K_i = \frac{K}{T_1}, \qquad K_p = \frac{K(T_1 + T_2)}{T_1} - \frac{K_i}{N}\ , \quad K_d = KT_2 - \frac{K_p}{N}$$

At ω_c, the loop gain must be unity, i.e.,

$$K\,G(j\omega_c) = 1 \qquad (iii)$$

With reference to Eq. 4.2 and Figure 4.2, at target crossover frequency $f_c = 300kHz$ ($\omega_c = 1.884\,Mrad/sec$), the magnitude $G(j\omega_c) = 0.818$ and Phase Margin $PM = 4^o$.

From (iii), we get $K = 1.23$

Further, to have Phase margin $PM = 60^o$, we must adjust PD controller parameters, i.e.,

$$\Delta Mf = PM - PM(\omega_c)$$
$$\Delta Mf = 60 - 4 = 56$$
$$\omega_c T_2 = \tan^{-1}(\Delta Mf)$$
$$T_2 = 4.72 * 10^{-5}$$

This PD also effects ω_c, so again find K

$$K + 20\log(\omega_c\, T_2) = -|G(j\omega_c)|$$
$$K = 0.0138$$

For adding PI-controller part, choose T_1 such that $\frac{1}{T1} \in (\frac{\omega_c}{30}, \frac{\omega_c}{10})$

$$\frac{1}{T_1} = \frac{\omega_c}{15} => T_1 = 7.96 * 10^{-6}$$

Now, we can easily find $K_p, K_d,\ K_i\ and\ N$. For N choose frequency at RHP-zero frequency.

$$N = 7.536 * 10^6\ rad/sec$$
$$Ki = 1733.6,\ \ Kp = 0.0953,\ \ \ Kd = 6.38 * 10^{-7}$$

So, the transfer function of PID controller is;

$$C(s) = \frac{4.828\,s^2 + 7.199 * 10^5\,s + 1.306 * 10^{10}}{s^2 + 7.536 * 10^6\,s} \qquad (4.3)$$

The bode plot of power converter, PID controller and open loop system is shown in Figure 4.4 below, (Appendix A)

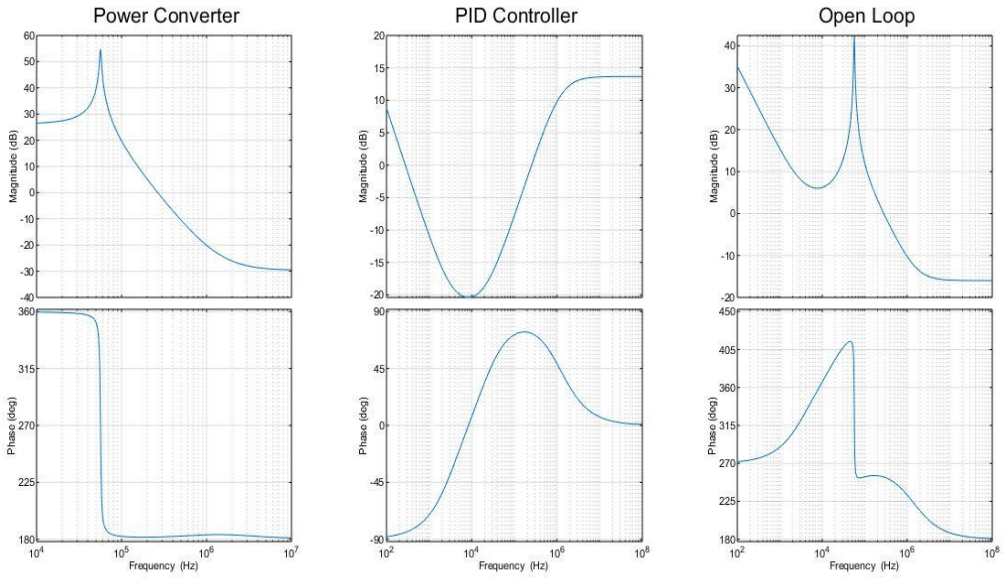


Figure 4.4: Bode plot of power converter, PID controller and open loop system

4.2. Boost Converter Discrete Time Modeling

[4] Consider now the discrete version of boost converter VM in CCM. Determine first the discrete-time small-signal dynamics of the control-to-output voltage, described by transfer function $G_{vu}(z)$. The state space matrices of boost converter during switch-on S_1 and switch-off S_2 are,

$$A_1 = \begin{bmatrix} -\frac{r_L}{L} & 0 \\ 0 & -\frac{1}{RC_o} \end{bmatrix}, \qquad A_0 = \begin{bmatrix} -\frac{r_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC_o} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, \qquad B_1 = B_0 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & \frac{1}{\frac{1+r_C}{R}} \end{bmatrix}, \qquad C_0 = \begin{bmatrix} r_{par} & \frac{1}{\frac{1+r_C}{R}} \end{bmatrix}$$

As shown in Figure 4.5, trailing-edge modulator has been employed for modulation, whose small signal parameters are,

$$phi \ '\varphi' = e^{A_0(T_s - t_d)} e^{A_1 D T_s} e^{A_0(t_d - D T_s)}$$

$$gamma\ 'γ' = \frac{T_s}{N_r} e^{A_0(T_s - t_d)} F$$

$$delta\ 'δ' = C_0$$

$$F = (A_1 X + B_1 V) - (A_0 X + B_0 V)$$

$$X = \left(I - e^{(A_1 D T_s)} e^{(A_0 D' T_s)}\right)^{-1} \left[-e^{A_1 D T_s} A_0^{-1} \left(I - e^{(A_0 D' T_s)}\right) B_0 - A_1^{-1} \left(I - e^{(A_1 D T_s)}\right) B_1\right] V$$

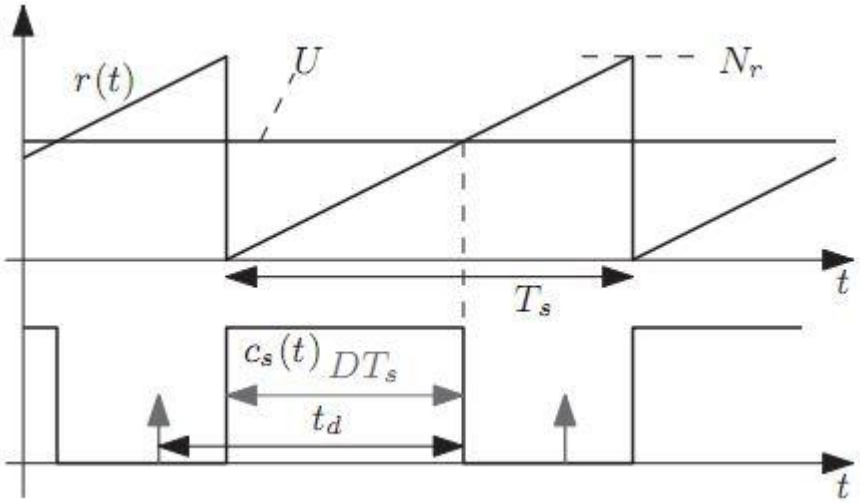


Figure 4.5: Trailing Edge Modulator

Writing all the equations in MATLAB and using command $ss(Phi, gamma, delta, 0, Ts)$ we can evaluate the transfer function of discrete-time VM CCM boost converter. The MATLAB code is given in Appendix. B, bode plot of discrete-time transfer function is shown in Figure 14.

$$G_{vu}(z) = \frac{0.4206\ z\ -\ 0.01327}{\mathrm{z}^2 -\ 1.96\ \mathrm{z}\ +\ 0.9882} \qquad (4.4)$$

4.2.1. Digital PID compensator

The parallel structure PID realization is considered here because of its inherent simplicity, which is governed by the following equations,

$$u_p[k] = K_p e[k]$$

$$u_i[k] = u_i[k-1] + K_i e[k]$$

$$u_d[k] = K_d(e[k] - e[k-1])$$

$$u[k] = u_p[k] + u_d[k] + u_i[k]$$

Taking Z-transform yields the standard additive form of digital PID transfer function,

$$G_{PID}(z) = K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \qquad (4.5)$$

The above transfer function has two poles at $z = 0$ and $z = 1$ and three zeros determined by three coefficients $K_p, K_i \; and \; K_d$.

In p-domain eq. 4.5 can be written as,

$$G'_{PID}(p) = K_p + \frac{K_i}{T_s}\frac{\left(1 + \frac{p}{\omega_p}\right)}{p} + K_d T_s \frac{p}{\left(1 + \frac{p}{\omega_p}\right)}$$

Where,

$$\omega_p = \frac{2}{T_s} = \frac{\omega_s}{\pi}$$

In multiplicative form, it can be written as

$$G'_{PID}(p) = G'_{PI\infty}\left(1 + \frac{\omega_{PI}}{p}\right).\frac{G'_{PD0}\left(1 + \frac{p}{\omega_{PD}}\right)}{1 + \frac{p}{\omega_p}} \qquad (4.6)$$

Once we determine p-domain parameters ($G'_{PI\infty}, G'_{PD0}, \omega_{PI} \; and \; \omega_{PD}$), the z-domain parameters can be calculated as,

$$K_p = G'_{PI\infty}G'_{PD0}\left(1 + \frac{\omega_{PI}}{\omega_{PD}} - \left(\frac{2\omega_{PI}}{\omega_P}\right)\right) \qquad (4.7a)$$

$$K_i = 2G'_{PI\infty}G'_{PD0}\left(\frac{\omega_{PI}}{\omega_p}\right) \qquad (4.7b)$$

$$K_d = \frac{G'_{PI\infty}G'_{PD0}}{2}\left(1 - \frac{\omega_{PI}}{\omega_p}\right)\left(\frac{\omega_p}{\omega_{PD}} - 1\right) \qquad (4.7c)$$

Using MATLAB code in Appendix. B to calculate above coefficients, we get

$$K_p = 0.26843$$

$$K_i = 0.00927$$

$$K_p = 1.2079$$

The z-domain transfer function of PID compensator is,

$$G_{PID}(z) = \frac{1.486\, z^2 - \; 2.684\, z \; + \; 1.208}{\mathrm{z}^2 - \; \mathrm{z}} \qquad (4.8)$$

The bode plot of PID controller and open loop system is shown in Figure 4.6.

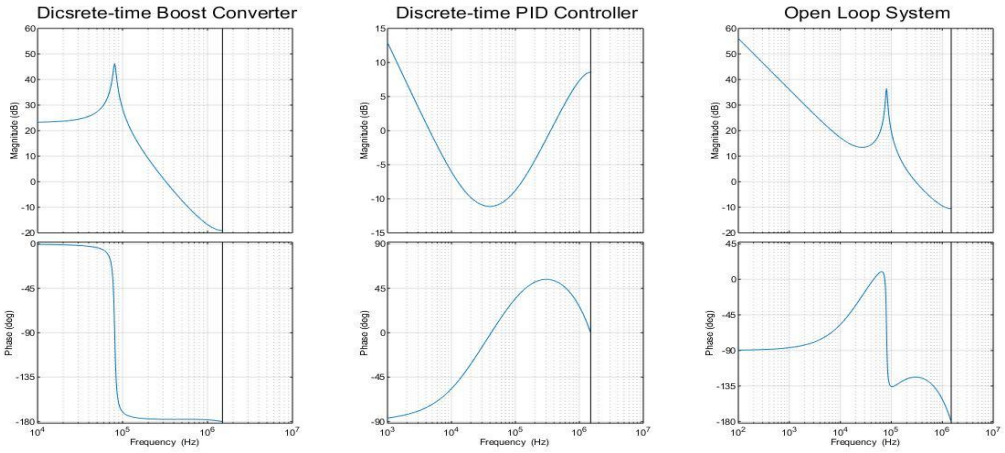


Figure 4.6: Bode plot of discrete-time power converter, discrete-time PID controller and open loop system

4.3. Digital PID compensator Implementation

As anticipated in chapter 3, the DPWM can only produce pulses of quantized duty cycle. Such quantization can be equivalently modeled as a quantization on the control command $u[k]$. [4]

4.3.1. PID Coefficients Scaling and Quantization

Scaling and quantization are the two steps necessary to bring the compensator from its system level formulation to a form suitable for implementation.

As we know, our ADC specifications are,

ADC resolution $'N_{ADC}' = 5$
ADC input range $'V_{pp}' = 2v$
ADC quantization interval $'q'_{ADC} = 0.0625$
DPWM resolution $'N_{DPWM}' = 8$
$N_r = 2^8 - 1 = 255$
Therefore, the compensator scaling factor is

$\lambda = q_{ADC} N_r = 15.93$

Applying λ to PID coefficients computed before, the new scaled coefficients are,

$$\widetilde{K_p} = \lambda K_p = 0.4277$$

$$\widetilde{K_d} = \lambda K_d = 20.24$$

$$\widetilde{K_i} = \lambda K_i = 0.1477$$

Coefficient quantization may affect low frequency gain and crossover frequency as well as phase margin. So, in order to minimize these effects, we must impose constraints so that system may not get unstable or have lower performance.

Suppose $\tilde{T}(z)$ is the loop gain of PID compensator after scaling and quantization. We must impose constraint of ϵ_c and α_c on magnitude and phase respectively to have crossover frequency acceptable for my design, i.e.,

$$|\tilde{T}(z) - 1|_{\omega_c} < \epsilon_c \qquad (4.9a)$$

$$|\angle\tilde{T}(z) - \angle T(z)|_{\omega_c} < \alpha_c \qquad (4.9b)$$

Similarly, for low frequency performances (DC gain) we must impose constraint of ϵ_0, i.e.,

$$\left|\frac{\tilde{T}(z) - T(z)}{T(z)}\right|_0 < \epsilon_0 \qquad (4.10)$$

Considering the following constraints for our design,

Constraint-I:

$\epsilon_c = 0.1\%$

$\alpha_c = \sin^{-1}(\epsilon_c) = 0.057^0$

Constraint-II:

$\epsilon_0 = 10\%$

Following are the steps followed for the coefficient quantization

Transfer function now is; $G_{PID} = \widetilde{K_p} + \frac{\widetilde{K_i}}{1-z^{-1}} + \widetilde{K_d}(1 - z^{-1})$

Applying derivative; $dG_{PID} = \frac{\partial G_{PID}}{\partial k} d\widetilde{K}$

We get; $dG_{PID} = dn_p\widetilde{K_p} + \frac{dn_i\widetilde{K_i}}{1-z^{-1}} + dn_d\widetilde{K_d}(1 - z^{-1})$

We know that, for low frequency constraint-II only K_i is responsible;

$$\partial G_{PID}|_{\omega=0} = \frac{|dn_i\widetilde{K_i}|}{\widetilde{K_i}}$$

$$\epsilon_0 = \frac{|dn_i\widetilde{K_i}|}{\widetilde{K_i}} => \epsilon_0\widetilde{K_i} = |dn_i\widetilde{K_i}|$$

$\widetilde{K_i} = 0.1477$ is the best-chosen value in constraint range, which is represented in binary form, i.e.,

$$\acute{K}_\iota = Q_n[\widetilde{K}_\iota]$$

$$[\boldsymbol{\acute{K}_\iota}]_{\mathbf{-12}}^{\mathbf{11}} = [\mathbf{01001011101}]$$

Moving to crossover frequency constraint-I, $dn_i\widetilde{K}_\iota$ has negligible effect on the constraint compared to $dn_p\widetilde{K_p}$ and $dn_d\widetilde{K_d}$. So, we use previous $\acute{K}_\iota$ and investigate ∂G_{PID} depending on n_p and n_d.

For this use MATLAB code (Appendix. C) for sensitivity function to find contours at different errors (0.1%, 0.2%, etc.) to get minimum no. of bits required for K_p and K_d and then choose best values in binary form.

$$\acute{K}_p = Q_n[\widetilde{K_p}] \quad => \ [\boldsymbol{\acute{K}_p}]_{\mathbf{-7}}^{\mathbf{7}} = [\mathbf{0110111}]$$

$$\acute{K}_d = Q_n[\widetilde{K_d}] => [\boldsymbol{\acute{K}_d}]_{\mathbf{-6}}^{\mathbf{12}} = [\mathbf{010100001111}]$$

4.3.2. Fixed Point PID controller implementation in HDL

In order to implement PID compensator on controller like FPGA, we need to write the Hardware Description Language (HDL) of PID design. Here in this section the hardware dynamic range of all the parameters involved in PID design shown in Figure 4.7 are calculated. The effective dynamic range is, [4]

$$DR_{eff}[\breve{x}] = \ 20log10\left(\frac{\lceil \breve{x} \rceil}{\lfloor \breve{x} \rfloor}\right)$$

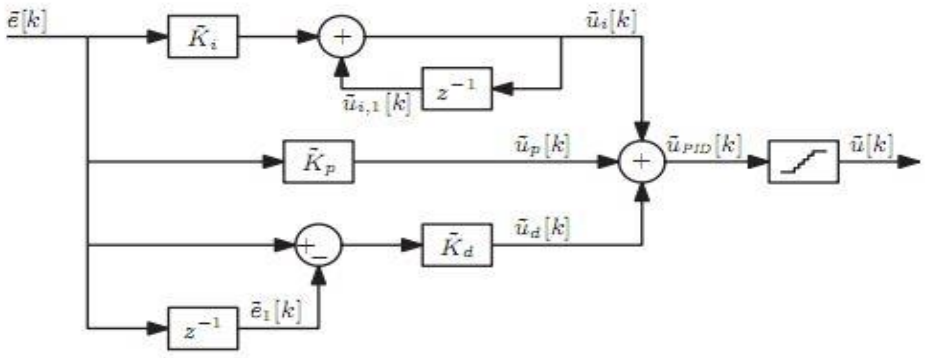


Figure 4.7: Digital PID compensator design

The hardware dynamic range is

$$DR_{hw}[\breve{x}] = \ 20log10\left(\frac{\lceil \breve{x} \rceil}{2^q}\right)$$

$$DR_{hw}[\breve{x}] = 20log10\left(\frac{2^{n-1}2^q}{2^q}\right) \qquad (4.11)$$

Then using following formula to calculate word length,

$$n = 1 + ceil\left(\frac{DR_{hw}[\breve{x}]}{20log2}\right) \qquad (4.12)$$

The PID output $'u'$ is biased signal ranging from $0\ to\ N_r - 1$, where $Nr = 2^8 - 1 = 255$ with granularity $'g'_u = 1\ (i.e, 2^0)$. Therefore, the hardware dynamic range of u is,

$$DR_{hw}[\breve{x}] = 20log10\left(\frac{255}{2^0}\right) = 72.24\ db$$

$$n = 1 + ceil\left(\frac{DR_{hw}[\breve{x}]}{20log2}\right) = 9\ bit$$

u is represented by 9-bit word with $scale = 2^0$. One extra bit is for binary 2's complement (B2C), but u is always positive, so we can discard MSB.

Next step is to find e_{max} that is given to PID controller as an input. Our $V_{ref} = 12\ V$ and maximum output voltage transient is $V_{o_{max}} = 24\ V$. For $N_{ADC} = 5$ and $V_{ADC\_PP} = 2\ V$, e_{max} will be,

$$e_{max} = 15\ LSB$$

Proportional term u_p:

$$\breve{u}_p[k] = \acute{K}_p e[k]$$

lower bound of $\breve{u}_p$ is quantization level 2^q of $\acute{K}_p, i.e.,\ \lfloor \breve{u}_p \rfloor = 2^{-7}$

And upper bound is, $\lceil \breve{u}_p \rceil = \acute{K}_p e_{max}$

After some numerical calculations using eq. 20 and 21, we get

$$\boldsymbol{n_{u_p} = 11\ bit}$$

Derivative term u_d:

$$\breve{u}_d[k] = \acute{K}_d(e[k] - e[k-1])$$

lower bound of $\breve{u}_d$ is quantization level 2^q of $\acute{K}_d, i.e.,\ \lfloor \breve{u}_d \rfloor = 2^{-6}$

And upper bound is, $\lceil \breve{u}_d \rceil = 2\acute{K}_d e_{max}$

After some numerical calculations using eq. 20 and 21, we get

$$\boldsymbol{n_{u_d} = 17\ bit}$$

Integral term u_i:

lower bound of $\breve{u}_i$ is quantization level 2^q of $\acute{K}_i, i.e.,\ \lfloor \breve{u}_i \rfloor = 2^{-12}$

And upper bound is, $\lceil \breve{u}_i \rceil = N_r - 1$

After some numerical calculations using eq. 20 and 21, we get

$$\boldsymbol{n_{u_i} = 21\ bit}$$

Also, for w_i, lower bound of $\breve{w}_i$ is quantization level $2^q$of $\acute{K}_\iota, i.e.,\ \lfloor \breve{u}_i \rfloor = 2^{-12}$

And upper bound is, $\lceil \breve{u}_i \rceil = \acute{K}_\iota e_{max}$

After some numerical calculations using eq. 20 and 21, we get

$$\boldsymbol{n_{w_i} = 15\ bit}$$

As shown in Figure 14, the overall high resolution PID signal is;

$$u_{PID} = \breve{u}_p + \breve{u}_d + \breve{u}_i$$

B2C addition requires alignment of both $\breve{u}_p$ and $\breve{u}_d$ to $\breve{u}_i$, because$\breve{u}_i$ has lowest quantization level, i.e., 2^{-12}

$$[\breve{u}_p]_{-7}^{11} \dot{\rightarrow} [\breve{u}_p]_{-12}^{16}$$

$$[\breve{u}_d]_{-6}^{17} \dot{\rightarrow} [\breve{u}_d]_{-12}^{23}$$

and $[\breve{u}_i]_{-12}^{21}$

Now addition will be;

$$[u_{PID}]_{-12}^{21} = [\breve{u}_p]_{-12}^{16} + [\breve{u}_d]_{-12}^{23} + [\breve{u}_i]_{-12}^{21}$$

transforming u_{PID} from $2^{-12}\ to\ 2^0$, i.e.,

$$[u_{PID}]_{-12}^{21} \dot{\rightarrow} [u_x]_0^9$$

The extra 9th bit is due to B2C. Discarding MSB, since control signal value can't be negative.

$$u_0^8 = \begin{cases} 255\ if\ u_x > 255 \\ 0\ if\ u_x \leq 0 \\ u_x\ otherwise \end{cases}$$

MATLAB code in Appendix. C is used to find the hardware dynamic ranges of all the required signal used for HDL implementation. The block diagram for HDL implementation of parallel PID compensator realization is shown in Figure 4.8. The Verilog code of complete PID compensator is attached in Appendix. D.

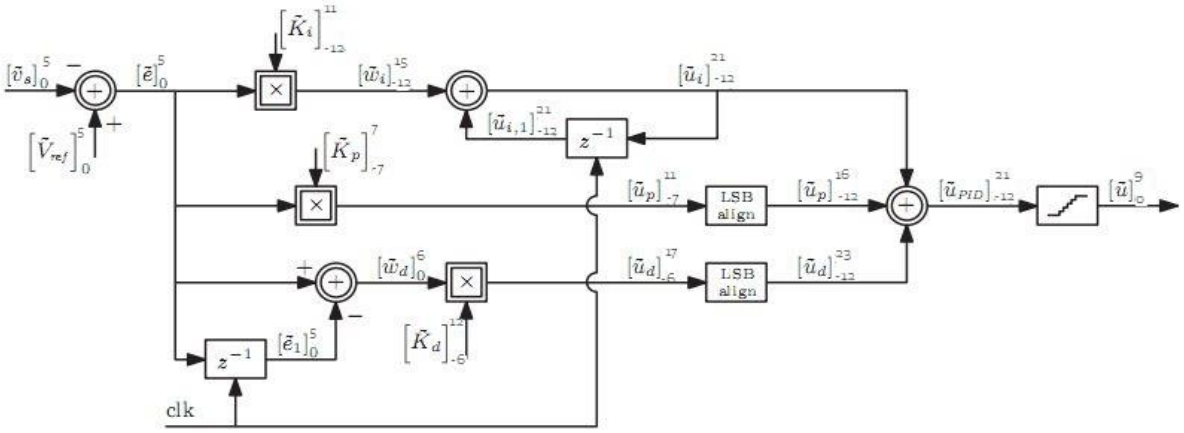


Figure 4.8: Block diagram for HDL implementation of parallel PID compensator realization

4.3.3. DDPWM Implementation in HDL

The block diagram for HDL implementation of DDPWM realization is shown in Figure 4.9 below,[2]

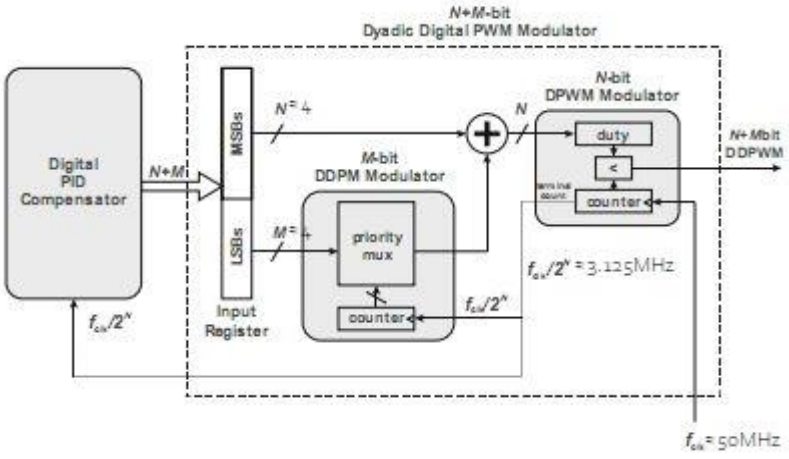


Figure 4.9: DDPWM Architecture

Our clock frequency is $f_{clk} = 50MHz$. The value $8 - bits$ long value $'u'$ obtained from PID compensator is sampled at rate of $f_{clk}/2^N$ in $N = 4$ and $M = 4$ bits input register. The $M -$ bits are given to $4 - bit$ DDPM modulator clocked at $f_{clk}/2^N$which results either $0\ or\ 1$ which is then added to value 'n'represented by $N - bits$. This added result is then given to $4 - bit$s DPWM modulator whose counter is running at clock frequency f_{clk}. The final DPWM output is the PWM signal which is used for converter switching

operation. The frequency of PWM signal is $\frac{f_{clk}}{2^N} = 3.125\ MHz$ which is the switching frequency of converter.[3]

The Verilog code of DDPWM modulator is attached in Appendix. D.

4.4. Simulation-based Experimental Methodology

In the complete system, we have mainly two stages, i.e., boost converter and digital PID compensator. MATLAB-Simulink, Modelsim and co-simulation of both Simulink and Modelsim are used in order to perform the simulation of complete system.

4.4.1. MATLAB-Simulink for Power Converter's Simulation

The synchronous DC-DC boost converter has been simulated in Simulink. The design specifications of converter adopted for the simulation are listed in Table 4.1,

Table 4.1: Design specifications of boost converter

Input Voltage V_{in}	$7 - 10\ Volts$
Input Capacitor C_{in}	$1uF, r_c = 2m\Omega$
Inductor L	$900nH, r_L = 8m\Omega$
MOSFET R_{on}	$24m\Omega$
Diode R_{on}	$24m\Omega$
Output Capacitor C_o	$3uF, r_c = 40m\Omega$
Load Resistor R_L	$25\Omega - 30\ \Omega$

The boost converter topology is shown in Figure 4.10 below,

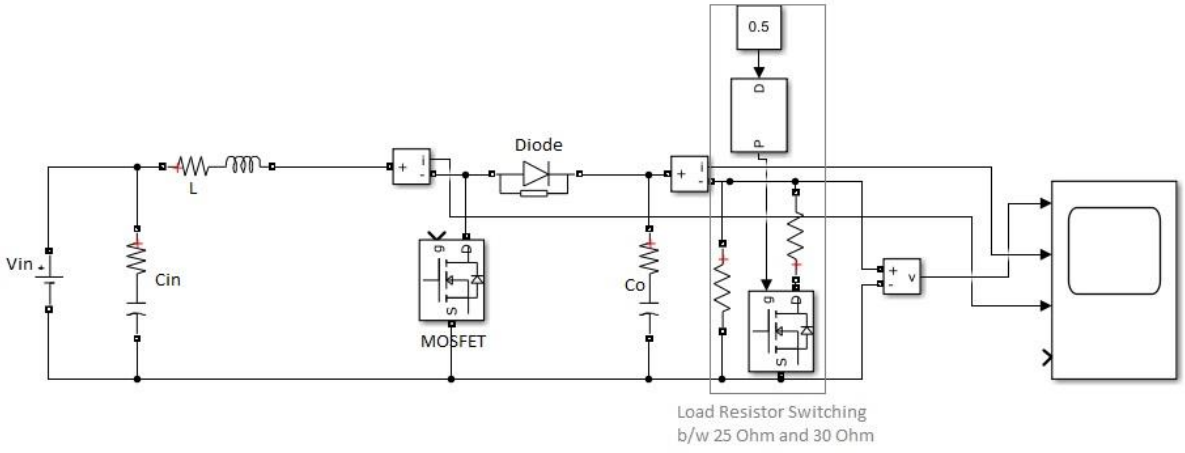


Figure 4.10: Boost converter Simulink design

As shown in Figure, load resistor switching block switches the load resistor to 25Ω or 30Ω during the simulation. This block is included to test the stability of circuit during load transients.

4.4.2. Modelsim PE for Digital PID compensator and DDPWM realizations

Since digital controller is implemented in HDL, Modelsim PE an HDL simulator has been used for its simulation. Both the HDL Verilog modules of digital PID compensator and DDPWM are instantiated in top module. The inputs and outputs of top module are listed in Table 4.2,

Table 4.2: Input-output of top module

Inputs		Output	
Clock	clk	PWM output signal	$'u'$
Reset	rst		
proportional gain	7-bits K_p		
integral gain	11-bits K_i		
derivative gain	12-bits K_d		
ADC resolution	5-bits v_{adc}		

The dataflow diagram of digital controller is shown in Figure 4.11 below,

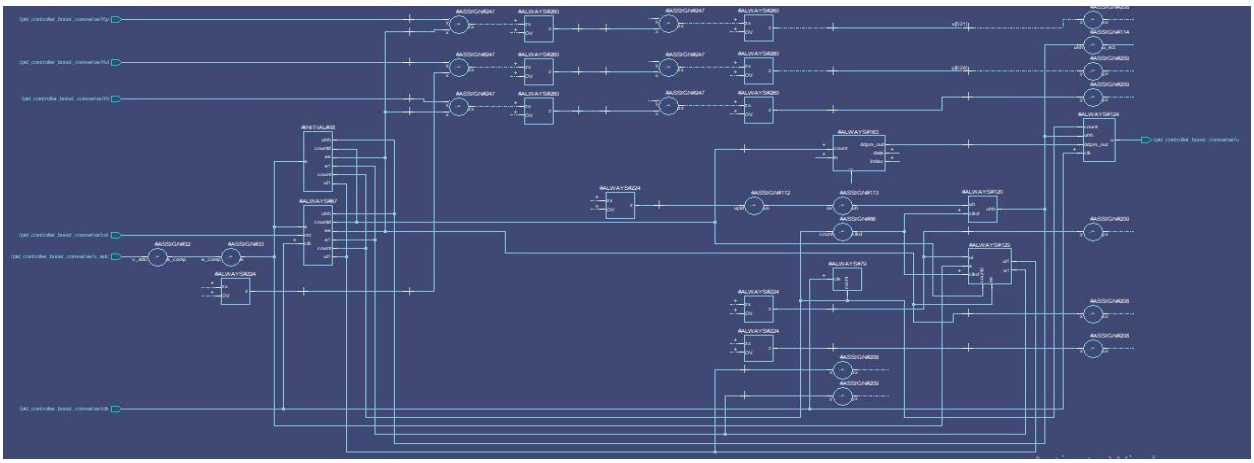


Figure 4.11: Dataflow diagram of digital controller in Modelsim

4.4.3. Co-simulation of Simulink and Modelsim

Since we are using two different tools, i.e., Simulink for power converter and Modelsim PE for digital controller, so for the simulation of complete design we must perform co-

simulation. Co-simulation has been done in MATLAB using cosim wizard shown in Figure 4.12 below,

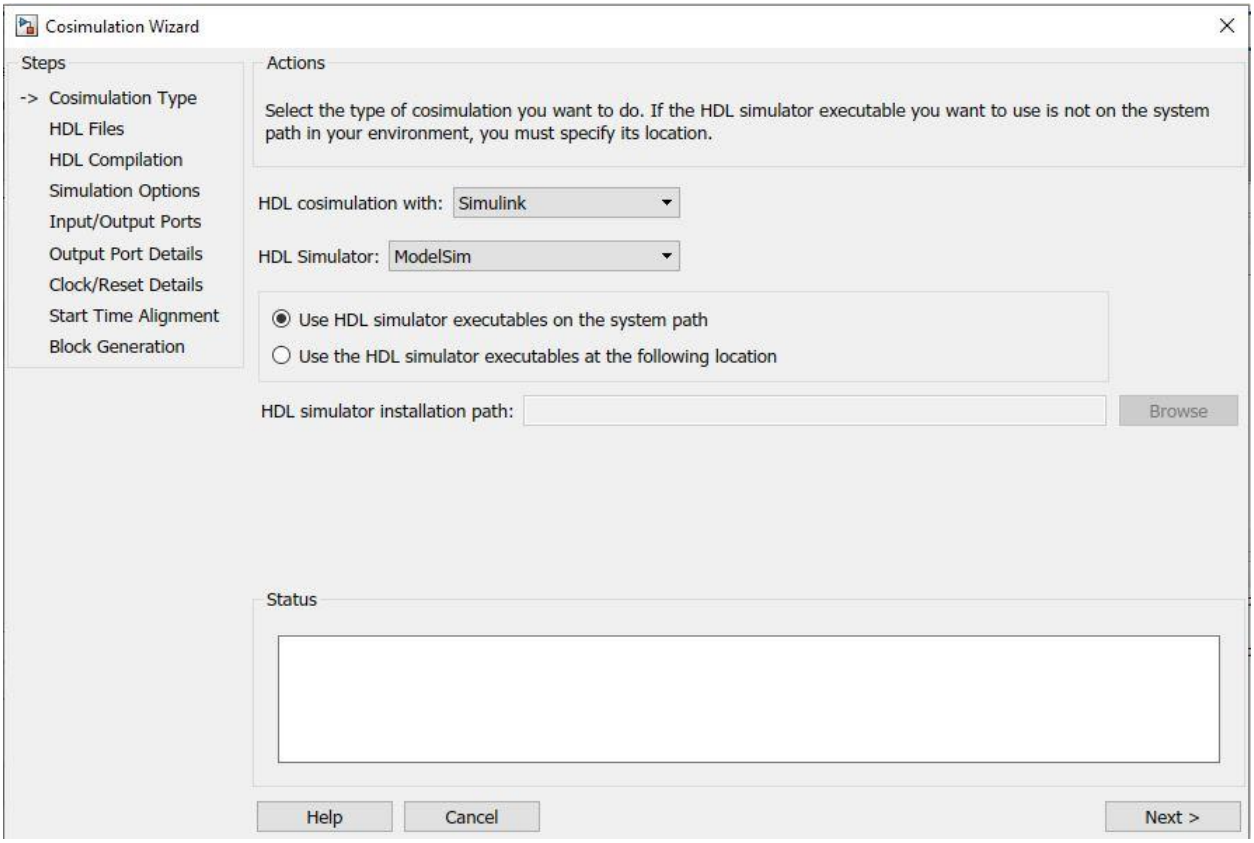


Figure 4.12: MATLAB co-simulation Wizard

After performing necessary steps, we are ended with Modelsim Simulator block within a Simulink window as shown in Figure 4.13. The block contains the HDL files for digital controller that are simulating in Modelsim.

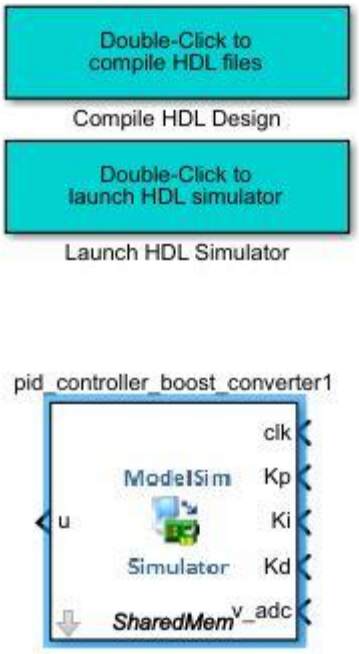


Figure 4.13: Modelsim Simulator block in Simulink window

Clock having frequency $50\,Mhz$ is provided externally, values of $K_p, K_i\ and\ K_d$ are assigned. The output voltage of converter $'v_o'$ is divided by 24 in order to interface with ADC input range (i.e., $2V_{pp}$). The output of ADC is connected to 'v_adc' and $'u'$ is connected to MOSFET gate. The Simulink design window for complete system simulation is shown in Figure 4.14.

Table 4.3: Digital controller design Specifications

$clock\ frequency\ f_{clk}$	$50\ MHz$
$switching\ frequency\ f_{sw}$	$3.125\ MHz$
K_p	[0110111]
K_i	[01001011101]
K_d	[010100001111]
N_{ADC}	5
$N_{DPWM} + N_{DDPM}$	$4 + 4$

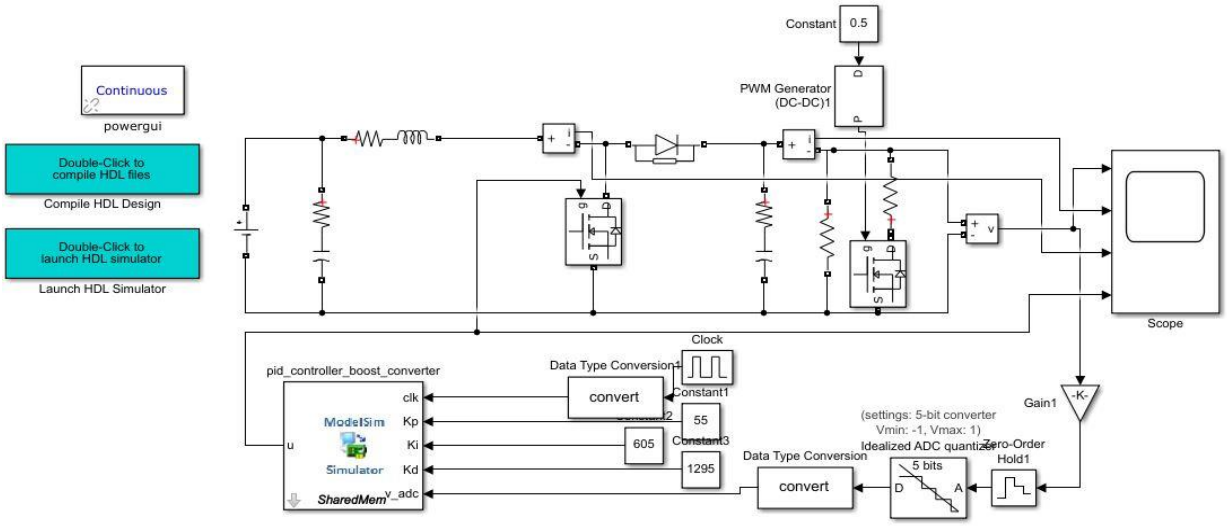


Figure 4.14: Simulink design of complete system

4.5. Hardware Implementation Setup

The hardware implementation setup has mainly four stages, that are,

- $EPC9006C$ development board is a $100\,V$ maximum device voltage, $7\,A$ maximum output current, half bridge with onboard gate drives, featuring the $EPC2007C$ enhancement mode ($eGaN$®) field effect transistor (FET). The purpose of this development board is to simplify the evaluation process of the $EPC2007C$ $eGaN$ FET by including all the critical components on a single board that can be easily connected into any existing converter
- Printed Circuit Board (PCB) designed for synchronous DC-DC boost converter which includes 12-pin female header used for its connectivity with $EPC9006C$ board and female SMA connector for ADC connection.
- High Speed Mezzanine Cards (HSMCs), is Data Conversion HSMC which contains two $AD9254, 14-bit$ 150 MSPS A/D converters. This device is designed for high speed and high-performance applications.
- Altera $DE2-115$ cyclone-IV FPGA board used for HDL implementation of digital controller which includes both digital PID compensator and DDPWM.

4.5.1. ADC Input & Output Interfacing

Data Conversion HSMC has two $AD9254, 14-bit$ 150 MSPS A/D converters which are represented by channel A and channel B.

Input Interfacing:

There is a female SMA connector used for input to ADC. The inputs to these A/D converters are transformer-coupled in order to create a balanced input. of the converter. As the maximum differential input voltage to the converter is $2\,V_{PP}$ and usable voltage input to the SMA connector is approximately $512\,mV$ when driven from a $50\,Ohm$ source, so in order to meet these requirements we have to interface it properly. Voltage divider circuit is used for ADC input interfacing as shown in Figure 4.15.

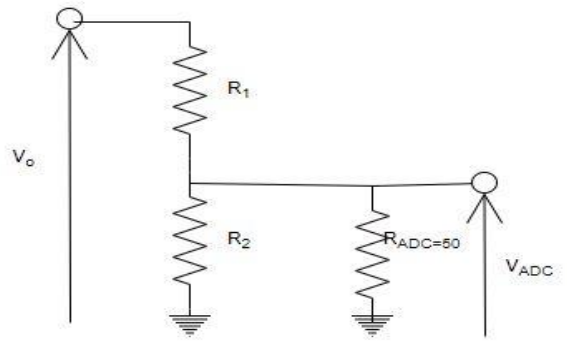


Figure 4.15: ADC input interfacing circuit

With reference to Figure 22, the $R_1\ and\ R_2$ can be calculated as,

$$V_{in_{ADC}} = V_o.\frac{R_2 \parallel 50}{R_1 + (R_2 \parallel 50)} \qquad (4.13a)$$

$$(R_1 \parallel R_2) = 50 \qquad (4.13b)$$

From simulation we know $V_{o_{max}} = 24V$ and taking $V_{in_{ADC-max}} = 512mV$, we get from $eq.(4.13a)$ and $(4.13b)$, we get

$$R_1 = 1.15K\Omega \ \ and\ R_2 = 50.1\Omega$$

Output Interfacing:

The HSMC connector is used to connect ADC with FPGA by proper interfacing through Verilog code. The following Table 4.4 shows ports and pins have been used to enable channel-A A/D, clock selection and data output enable.

Table 4.4: ADA Port and Pins

ADA_OR	out of range indicator
AD_SCLK	data format selection, i.e., offset binary or B2C form
AD_SDIO	duty cycle stabilizer effect
ADA_OE	ADA output enable
ADA_SPI_CS	disable serial port interfacing
FPGA_CLK_A_P	clock input

4.5.2. Boost converter PCB designing and connectivity with EPC2007 Board

The schematic diagram builds for PCB designing of synchronous DC-DC boost converter is shown in Figure 4.16. The tool used for PCB designing is KiCAD.

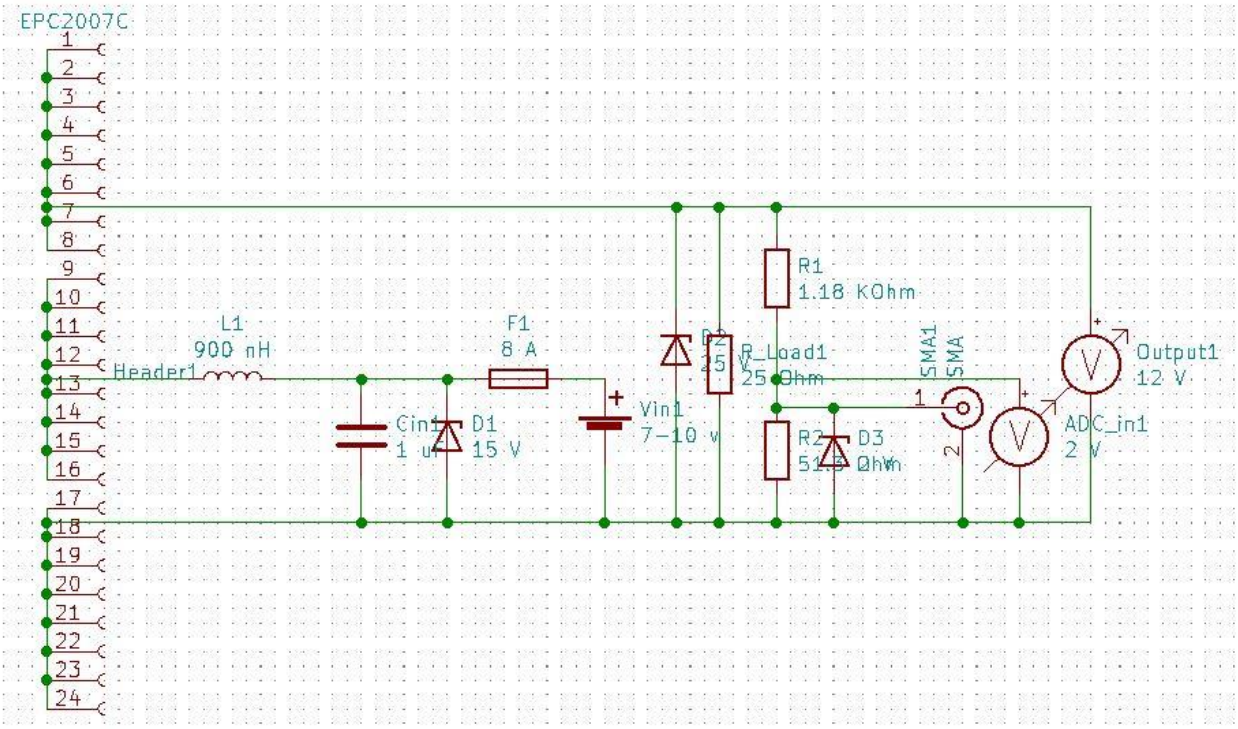


Figure 4.16: Schematic diagram of PCB design on KiCad

The list of components finalized for hardware implementation are shown in Table 4.5 below,

Table 4.5: Finalized components for PCB layout

Zener Diode for ADC input voltage protections	$2V$
Fuse	$5A$
Zener Diode for input voltage protections	$15\ V$
Input Ceramic Capacitor	$1uF, 25V$
Load resistor	$30\Omega, 30W$
Zener Diode for output voltage protections	$25\ V$
Inductor	$900nH, 8A$
R_1	$1.18K\Omega$
R_2	52.3Ω
ADC connection	Female SMA connector
Connection with $EPC9006C$ development board	12 −pin Female header

The final PCB layout is shown in Figure 4.17,

Figure 4.17: PCB layout of boost converter with EPC9006C board

4.5.3. FPGA Cyclone-IV DE2-115 for HDL Synthesis

The HDL-implementation of digital controller, i.e., digital PID compensator and DDPWM has been done in cyclone-IV DE2-115 using Quartus prime tool. The controller will use internal clock having frequency of $50\ MHZ$ of FPGA, which is divided by clock divider HDL code to $3.125\ MHz$ equivalent to switching frequency. The pin assignment used in our project is described in Table 4.6,

Table 4.6: Pin assignments for FPGA

$KEY[3]$	Reset
$SW[0]$	Data Format Select
$SW[1]$	Duty cycle stabilizer
$SW[2]$	DDPM enabling or disabling
$LEDG[3]$	ADC out of range indication
$GPIO[1]$	output PWM signal

The picture of complete hardware setup is shown in Figure 4.18 below,

Figure 4.18: Complete hardware setup

Chapter 5

5. Result & Analysis

5.1. Simulation-based Results

The impact of using DDPWM to remove limit-cycles oscillations was described in section 3.1. We compare the output voltage, output current and inductor current of boost converter with DPWM-based and DDPWM based digital controller in this subsection. The design specifications used for boost converter and digital controller was described in table 4.1 and 4.3 respectively (detailed in section 4.4).

5.1.1. DPWM controller

The design guidelines for LCO's free operation is to have $N_{DPWM} > N_{ADC}$ described in section 2.2. But in this case our digital controller has only DPWM module along with PID compensator whose resolution is $4 - bits$, while the ADC resolution is $5 - bits$. Figure 5.1, 5.2 and 5.3 shows the output voltage v_o, output current i_o and inductor current i_L of boost converter respectively.

Each time controller is trying to drive output to zero-error bin, but there is no DPWM quantization interval lies into the zero-error bin, therefore output is oscillating around the zero-error bin at DPWM levels so produces square-wave like low-frequency oscillations that are clearly seen at output voltage v_o and output current i_o in Figure 5.1 and 5.2 respectively. These unwanted low-frequency oscillations are the limit-cycles oscillation. The frequency of these oscillations is found to be around $16KHz$.

As discussed in section $4.4.1$, the load is switches from $25\ \Omega\ to\ 30\Omega$ at $0.5ms$. This effect can be seen in Figure 5.2, output current i_o is switches from $0.45\ A\ to\ 0.35\ A$ at load transient, while output voltage v_o stays constant at approximately $12\ V$.

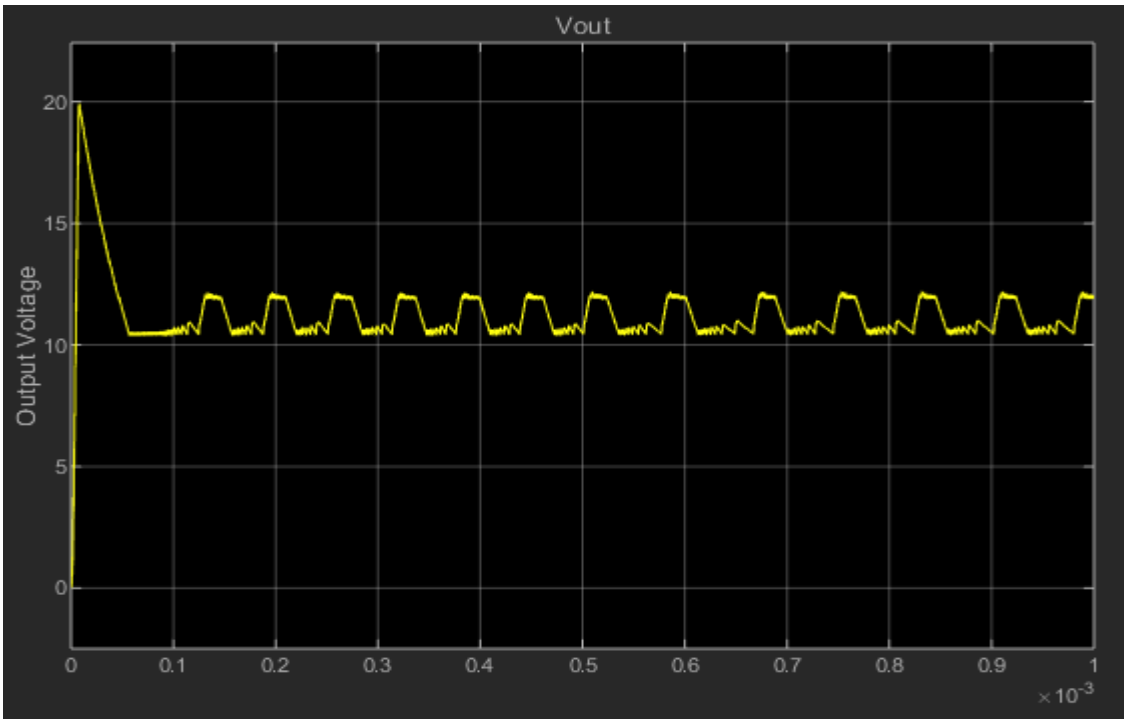


Figure 5.1: Boost converter output Voltage v_o with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 0$

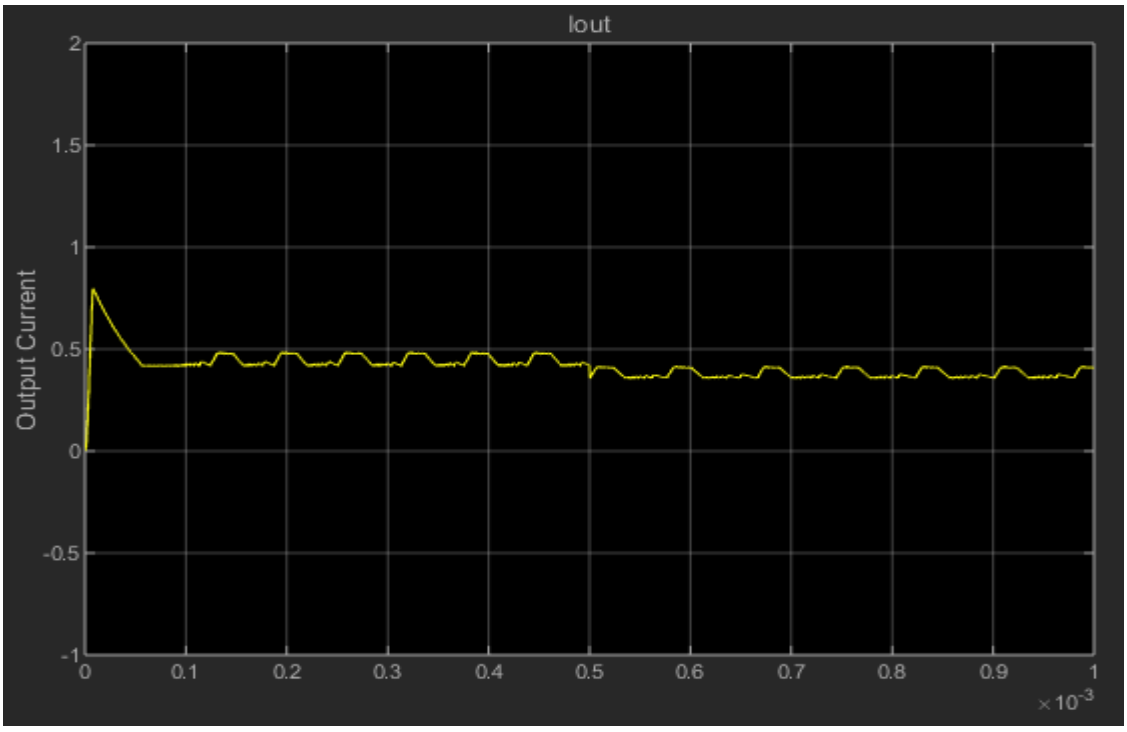


Figure 5.2: Boost converter output current i_o with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 0$

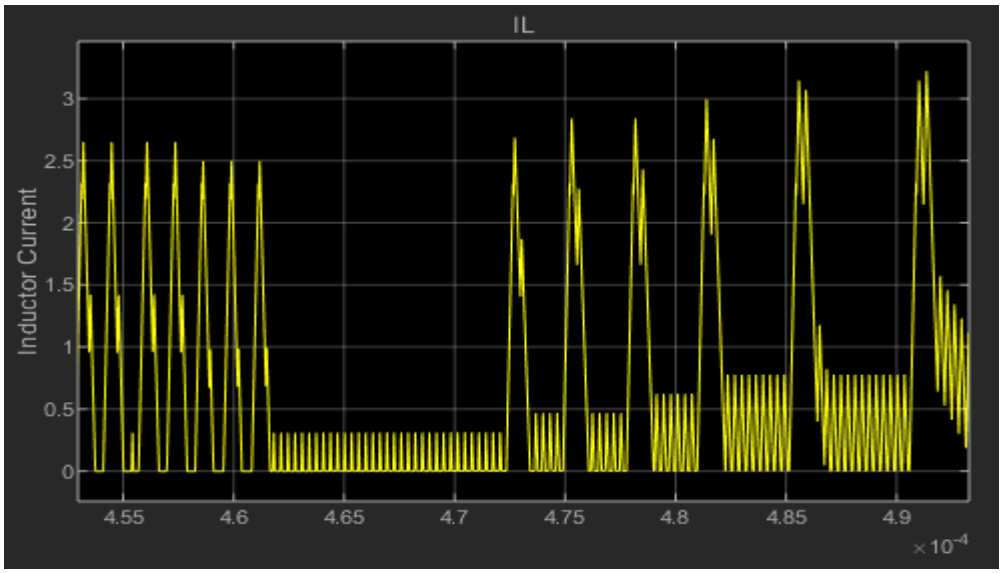


Figure 5.3: Boost converter inductor current i_L with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 0$

The Figure 5.3 shows the inductor current i_L of the boost converter. There is a large distortion and pulses of different magnitudes in i_L. It is seen that i_L also approaches to zero which is unacceptable in continuous conduction mode.

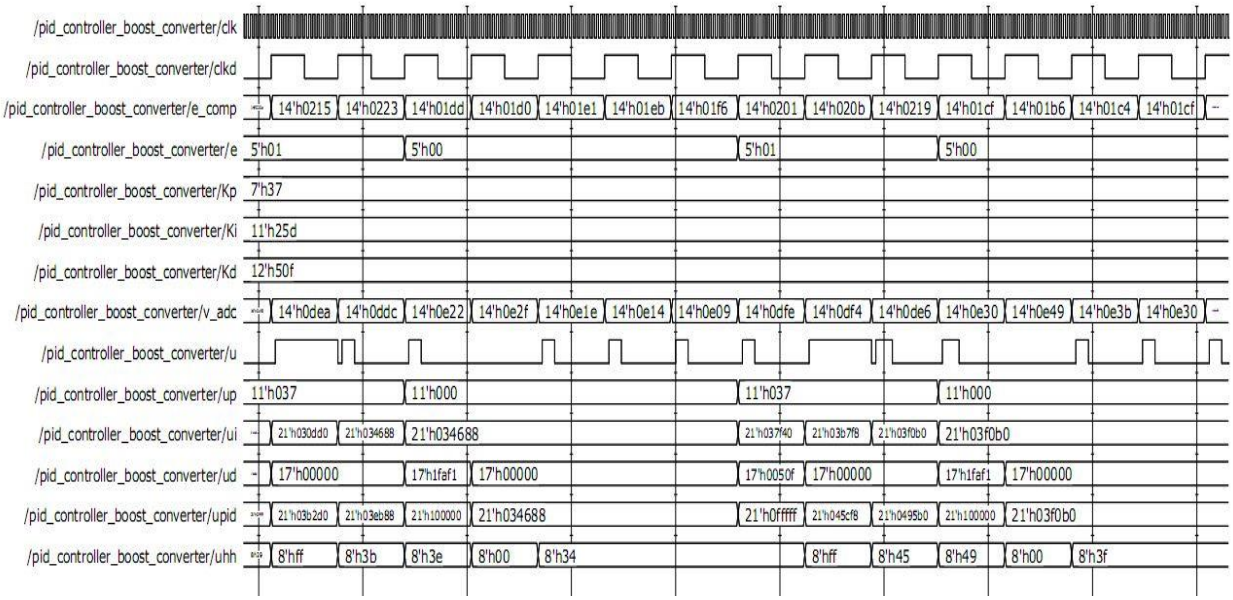


Figure 5.4: Timing diagram of digital controller parameters with $N_{ADC} = 5$, $N_{DPWM} = 4$, $N_{DDPM} = 0$

Figure 5.4 shows the Modelsim simulation results of timing diagrams of all the signals and wires used in digital controllers, (i.e. digital PID compensator and DPWM). The $'uhh'$ is the duty cycles generated by PID compensator which is not constant since controller is not able to drive output to zero-error bin.

5.1.2. DDPWM controller

In this case digital controller has both $4-bits\ DPWM$ and $4-bits\ DDPM$ module along with PID compensator. Following the design guideline for LCO's free operation, i.e., $N_{DPWM} > N_{ADC}$ now the effective resolution of DDPWM is $8-bits$, while the ADC resolution is $5-bits$. Figure 5.5, 5.6 and 5.7 shows the output voltage v_o, output current i_o and inductor current i_L of boost converter respectively. It is seen that, there is no limit-cycles oscillations at output voltage and current. Also, during the load switching ($i.e., 25\Omega\ to\ 30\Omega\ at\ 0.5ms$), the voltage remains stable at $12\ V$ and current switches from $0.48\ A\ to\ 0.4\ A$ after short transient.

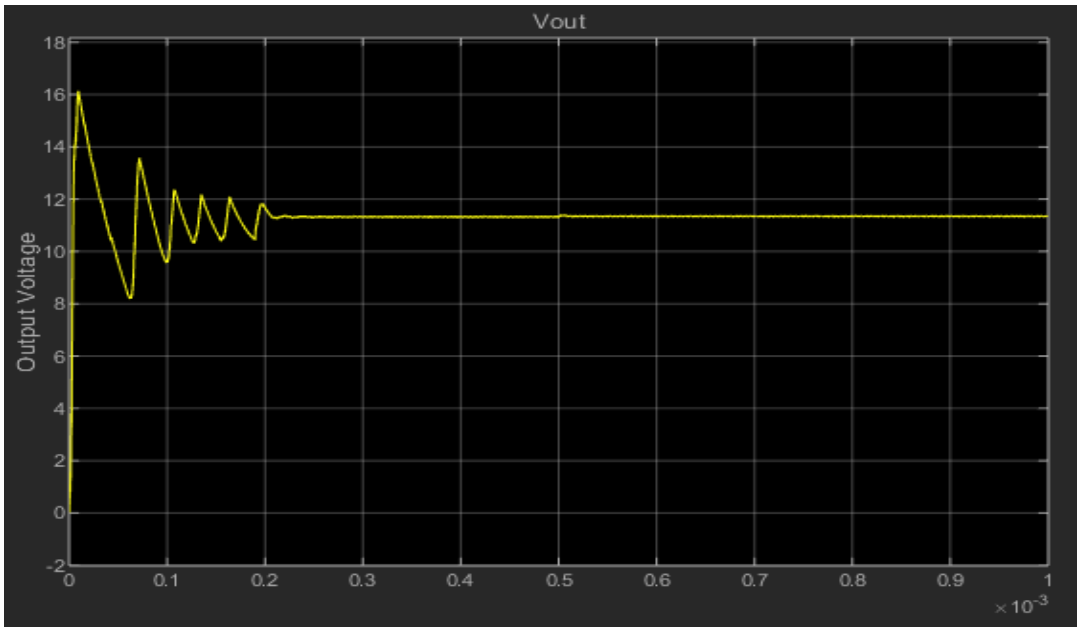


Figure 5.5: Boost converter output Voltage v_o with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 4$

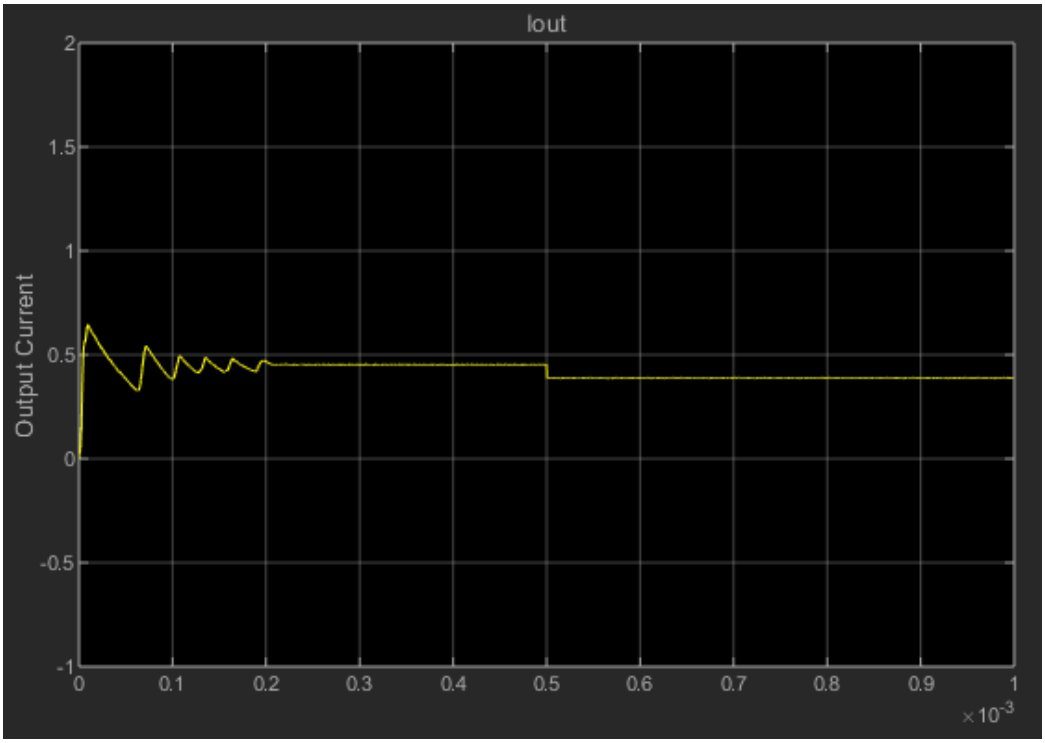


Figure 5.6: Boost converter output current i_o with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 4$

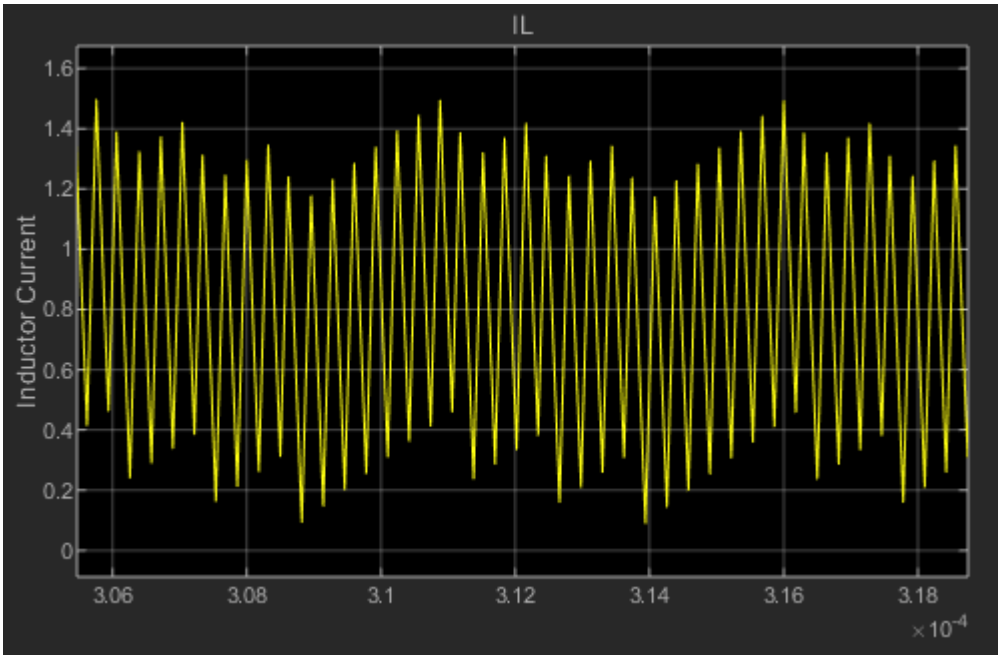


Figure 5.7: Boost converter inductor current i_L with $N_{ADC} = 5$, $N_{DPWM} = 4$ and $N_{DDPM} = 4$

The inductor current i_L has no distortion and it never approaches to zero which shows continuous conduction mode (CCM) of converter.

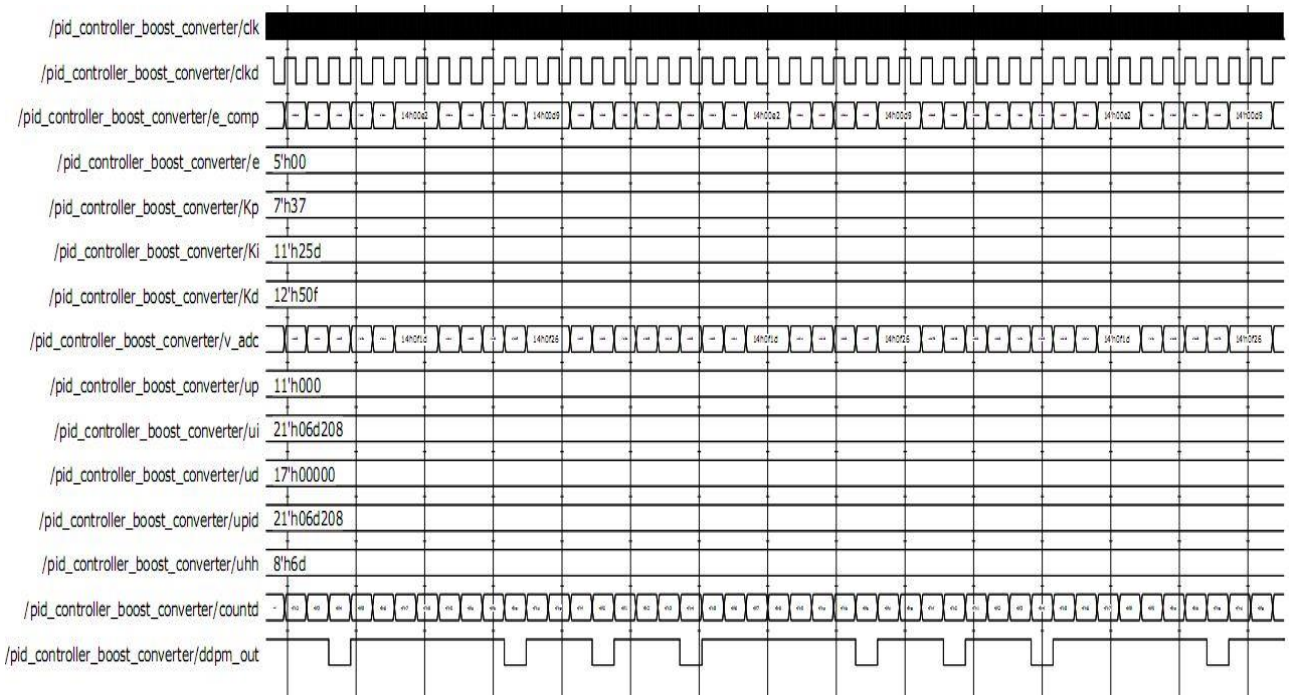


Figure 5.8: Timing diagram of digital controller parameters with $N_{ADC} = 5$, $N_{DPWM} = 4$, $N_{DDPM} = 4$

Figure 5.8 shows the Modelsim simulation results of timing diagrams of all the signals and wires used in digital controllers during steady state. The duty-cycle $8 - bit$ 'uhh' generated by PID compensator is constant and error 'e' to controller is zero since controller is now able to drive output to zero-error bin. The controller output 'u' is modulated between two adjacent quantized levels. The output of DDPM '$ddpm\_out$' is periodic over a period of $16T_{sw}$. Due to this periodicity, the output voltage contains DDPM ripples of frequency $\frac{f_{sw}}{2^{N_{DPWM}}} = \frac{3.125M}{16} \cong 195KHz$ as shown in Figure 5.9.

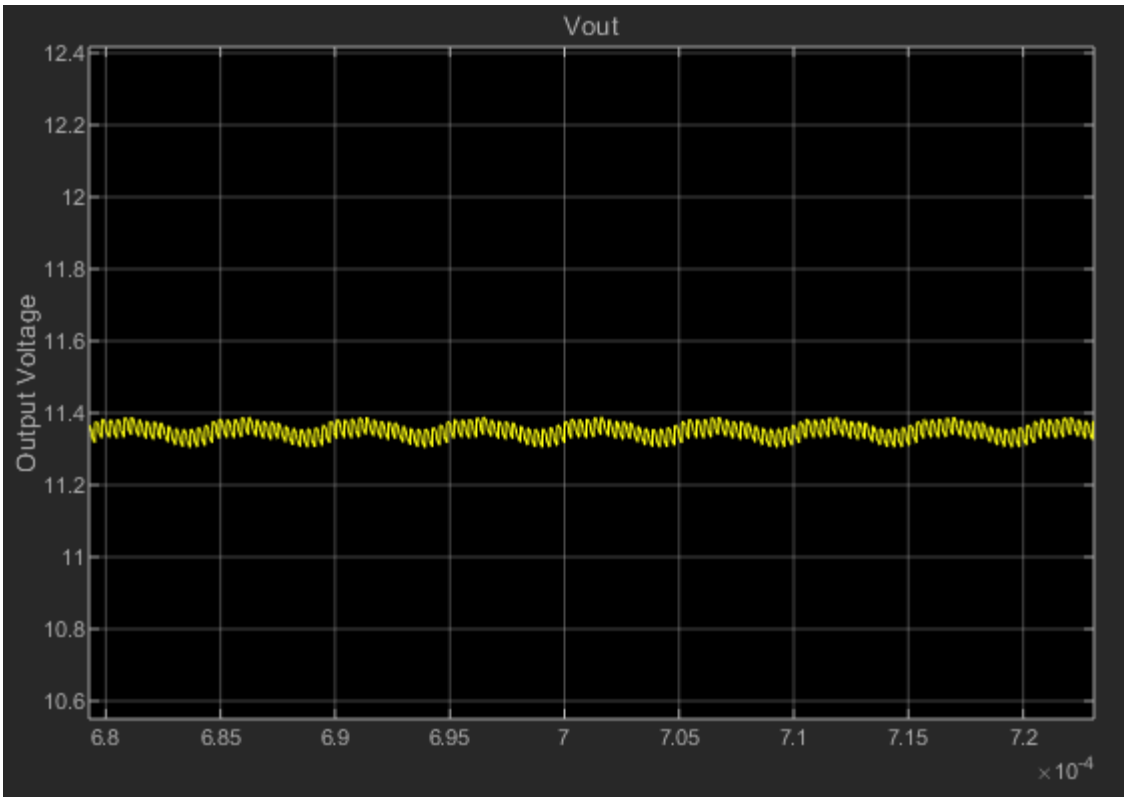


Figure 5.9: DDPM ripples at output voltage v_o

Chapter 6

6. Conclusion

In this thesis, an innovative technique intended to increase the resolution of the DPWM for LCO-free operation of switch-mode power converters is analyzed and experimentally evaluated. More precisely, the novel DDPWM is adopted as a systematic approach that generalizes and extends the standard PWM dithering techniques to achieve accurate, LCO-free operation in a digital converter at negligible cost and design effort and without any detrimental effect on the output ripple voltage. A digitally controlled DC-DC Boost converter is considered to validate the approach proposed in the thesis, which is designed to operate in continuous-conduction-mode with a switching frequency in the MHz range while voltage-mode digital control algorithm is considered. We have demonstrated our approach on Simulink and Modelsim simulation tools. The obtained results, i.e., output voltage, output current and inductor current for both DPWM-modulator and DDPWM-modulator are compared and the effectiveness of the DDPWM in mitigating the onset of the LCOs is verified versus different operating conditions and digital control parameters.

DDPWM is effective technique for LCOs free operation, but it introduces DDPM ripples at output voltage which may be undesirous for low-resolution DPWM applications. Future work could consider implementing DDPWM in additional benchmark, i.e., the mitigation of DDPM ripples at output voltage.

Appendices

A. Matlab code for continuous-time boost converter and PID controller

```
clear all
clc
format shorte
format compact

Vin_min=7; Vin_max=10;Esr=0.04;Vref=2.5;Vtr=1;
Vo=12;
C=3e-6; %fixed output Capacitor
fsw=3e6;

%Duty Cycles Calculations
Dmax = 1 - Vin_min/Vo
Dmin = 1 - Vin_max/Vo

%Frequencies Adjustment
fc = fsw/10;   %crossover frequency


R= input("Put any normalized value for load resistance: ");    % between L
and R we have very low range to adjust frhp and fres under control
Rmin = R- 5/100*R; Rmax= R+ 5/100*R ;  % 5% tolerance           % so we have
to choose carefully both L and R according to given formulas.
Iin_max = Vo/((1-Dmax)*Rmin)
I0_max = Vo/(Rmin)
Dccm_max = 1/3;   % Max value for expression D(1-D)^2 is at D=1/3 which is in
our range
Lccm = (1-Dccm_max)^2*(Dccm_max)*Rmax/(2*fsw)
M = 15;  % this is for ceramic capacitor (ref slva274a)
Lmax = C*(Rmin/M*Vin_min/Vo)^2    % reference from power book
%L=input("Enter normalized value of L greater than Lccm and less than Lmax:
")
L=900e-9;



D=1/2;   % Max value for expression D(1-D) is at D=1/2
delta_I_l = Vo*D*(1-D)/(L*fsw)   % Inductor Ripple Current


frhp_z = (Rmin*(1-Dmax)^2)/(2*3.14*L)   % Right Half plane Zero frequency due
to L
fres = (1-Dmax)/(2*pi*sqrt(L*C))    %resonant frequency
fesr_z = 1/(2*pi*C*Esr)   %ESR zero frequency

Iin_max = Vo/((1-Dmax)*Rmin)
Iin_min = Vo/((1-Dmin)*Rmax)
```

```
Imax = Iin_max + delta_I_l/2

dlta_I_l = Vo*Dmin*(1-Dmin)/(L*fsw);
Imin = Iin_min - dlta_I_l/2

if fres <= fc/3   %check for resonant freq (fc/3)
   fprintf("Resonance Frequency is okay\n");
else
    fprintf("Resonance Frequency is not okay\n");
end

if frhp_z >= 4*fc
    fprintf("RHP zero Frequency is okay\n");
else
    fprintf("RHP zero Frequency is not okay\n");
end
%%%%%%%%% Power Block %%%%%%%%%%%%%%%
s=tf('s');
kd=(Vin_min/(1-Dmax)^2)/Vtr;
sz1=1/(Esr*C);
sz2 = ((1-Dmax)^2)*R/L;
sp=(1-Dmax)/sqrt(L*C);
Q=R*(1-Dmax)*sqrt(C/L);
n1 = [1/sz1 1];
n2=[-1/sz2 1];
NUM= conv(n1,n2);
DEN = [1/(sp^2) 1/(sp*Q) 1];
poles_pow=roots(DEN)
sysPOW = kd*tf(NUM,DEN)
poles=roots(DEN);
PP=bodeoptions;
PP.FreqUnits ='Hz';
PP.Grid ='on';
subplot(131)
bode(sysPOW,PP)
%pidtool(sysPOW)
title("\fontsize{16} Power Converter")
%PID
kp=0.0953;
ki=1733;
kd=6.28e-7;
N=7.536e6;
sysPID=(kp+ki/s+(kd*s/(1+s/N)))
subplot(132)
bode(sysPID,PP)
title("\fontsize{16} PID Controller")
%closed loop
sysLOOP = sysPOW * sysPID
subplot(133)
bode(sysLOOP,PP)
title("\fontsize{16} Open Loop")
```

B. Matlab code for discrete-time boost converter and PID controller

```
Clc
clear all
Vg=10;
Rload=25;
Vload=12;
Iload = Vload/Rload;
D=1-Vg/Vload;
td=100e-9;
fsw=3e6;
Ts=1/fsw;
L=900e-9;
rL=0.02;
C=3e-6;
rC=0.04;
Dprime=1-D;
rpar=(rC)/(1+rC/Rload);
z=tf('z',Ts);

A1=[-rL/L 0;0 -1/Rload/C];
A0=[-rL/L -1/L;1/C -1/Rload/C];
b1=[1/L;0];
b0=b1;
c1=[0 1/(1+rC/Rload)];
c0=[rpar 1/(1+rC/Rload)];

A1i=A1^-1;
A0i=A0^-1;
Xdown=((eye(2)-expm(A1*D*Ts)*expm(A0*Dprime*Ts))^-1)*(-
expm(A1*D*Ts)*A0i*(eye(2)-expm(A0*Dprime*Ts))*b0-A1i*(eye(2)-
expm(A1*D*Ts))*b1)*[Vg];
Xup=expm(A0*Dprime*Ts)*Xdown-A0i*(eye(2)-expm(A0*Dprime*Ts))*b0*[Vg];
Fdown=(A1-A0)*Xdown+(b1-b0)*[Vg];
Fup=(A1-A0)*Xup+(b1-b0)*[Vg];

%Small-signalmodelmatrices

%Phi=expm(A0*Dprime*Ts/2)*expm(A1*D*Ts)*expm(A0*Dprime*Ts/2);   %sym edge
%gamma=(Ts/2)*expm(A0*Dprime*Ts/2)*(Fdown+expm(A1*D*Ts)*Fup);%sym edge

Phi=expm(A0*(Ts-td))*expm(A1*D*Ts)*expm(A0*(td-D*Ts));  %trailing edge
gamma = Ts*expm(A0*(Ts-td))*Fdown;   %trailing edge
delta=c0;
%Convertfromstate-spacetotransferfunctionobject
sys=ss(Phi,gamma,delta,0,Ts);
Wz=tf(sys)

%%%%%%%%%%%%PID%%%%%%%%%%%%%%%%%%%%%%%%%%
%Targetcrossoverfrequencyandphasemargin
wc=2*pi*300e3;
mphi=(pi/180)*60;%Inradians
%MagnitudeandphaseofTuzatthetargetcrossoverfrequency
[m,p]=bode(Wz,wc);
```

```
p=(pi/180)*p;
%Prewarpingonwc
wcp=(2/Ts)*tan(wc*Ts/2);
%PDDesign
wp=2*pi*fsw/pi;
pw=atan(wcp/wp);
wPD=(1/(tan(-pi+mphi-p+pw)/wcp));
GPD0=sqrt(1+(wcp/wp)^2)/(m*(sqrt(1+(wcp/wPD)^2)));
%PIzeroandhigh-frequencygain
wPI=wc/15;
GPIinf=1;
%Proportional,IntegralandDerivativeGains
Kp=GPIinf*GPD0*(1+wPI/wPD-2*wPI/wp)
%Kp=0.05
Ki=2*GPIinf*GPD0*wPI/wp
Kd=GPIinf*GPD0/2*(1-wPI/wp)*(wp/wPD-1)
%PIDTransferfunction
z=tf('z',Ts);
Gcz=Kp+Ki/(1-z^-1)+Kd*(1-z^-1)

PP=bodeoptions;
PP.FreqUnits ='Hz';
PP.Grid ='on';
subplot(131)
bode(Wz,PP)
title("\fontsize{16} Dicsrete-time Boost Converter")
subplot(132)
bode(Gcz,PP)
title("\fontsize{16} Discrete-time PID Controller")
subplot(133)
bode(Wz*Gcz,PP)
title("\fontsize{16} Open Loop System")
```

C. Matlab code to find hardware-dynamic ranges of all signals involved in digital PID controller

```
Clc
clear all
n_adc=input('Enter no. of bits of ADC: ');
dpwm_n=input('Enter no. of bits of DPWM: ');
ddpwm_n=input('Enter no. of bits of DDPWM: ');
%kp=0.0249;
%ki=0.0087;
%kd=1.083;
kp=0.02684;
ki=0.00927;
kd=1.27;
vmin=0;
vmax=2;
n_dpwm=dpwm_n+ddpwm_n;
ADC_q= (vmax-vmin)/2^n_adc;
Nr=2^n_dpwm-1;
sf=ADC_q*Nr;
kpp=sf*kp;
kii=ki*sf;
kdd=sf*kd;
kp_n=7;
kpq=Qn(kpp,kp_n);
kp_q=-kpq.q;
kd_n=12;
kdq=Qn(kdd,kd_n);
kd_q=-kdq.q;
ki_n=11;
kiq=Qn(kii,ki_n);
ki_q=-kiq.q;
emax=2^(n_adc-1)-1;
n_up=1+ceil(20*log(kpp*emax/2^kpq.q)/20/log(2));
n_upp=n_up+7-1;
n_ud=1+ceil(20*log(2*kdd*emax/2^kdq.q)/20/log(2));
n_udd=n_ud+7-1;
n_ui=1+ceil(20*log(Nr/2^kiq.q)/20/log(2));
n_wi=1+ceil(20*log(kii*emax/2^kiq.q)/20/log(2));
n_wii=n_wi+7-1;
upp_zeros=kp_q-ki_q;
udd_zeros=kd_q-ki_q;
emax;
fprintf('Parameters for controller\ne_n= %i\ndpwm_n= %i\nddpwm_n= %i\nkp_n=
%i\nkp_q= %i\nkd_n= %i\nkd_q= %i\nki_n= %i\nki_q= %i\nup_n= %i\nud_n=
%i\nui_n= %i\nwi_n= %i\n'...
    ,n_adc,dpwm_n,ddpwm_n,kp_n,kp_q,kd_n,kd_q,ki_n,ki_q,n_up,n_ud,n_ui,n_wi)
fprintf('vref= %i,   Kp= %i,   Ki= %i,   Kd= %i\n',emax,kpq.w,kiq.w,kdq.w)
```

D. Verilog code for digital PID controller

```
module pid_controller_boost_converter(rst,clk,Kp,Ki,Kd,e,u);

parameter e_n=5;
parameter dpwm_n= 4;
parameter ddpwm_n= 4;
parameter kp_n=7;
parameter kp_q=8;
parameter kd_n=12;
parameter kd_q=7;
parameter ki_n=11;
parameter ki_q=3;
parameter up_n=11;
parameter ud_n=17;
parameter ui_n=22;
parameter wi_n=15;

input clk;
wire clk12;
input rst;
reg signed [e_n-1:0] ee;
input signed [kp_n-1:0] Kp;  //[7,-13]
input signed [ki_n-1:0] Ki; //[7,-14]
input signed [kd_n-1:0] Kd; //[10,-10]
input signed [e_n-1:0] e; //[14,0]
output u;//[1,0]
reg signed [e_n-1:0] e1; //[14,0]
wire signed [up_n-1:0] up; //[20,-13]
wire signed [up_n-1+6:0] upp; //[26,-13]
wire signed[ui_n-1:0] ui; //[27,-14]
reg signed[ui_n-1:0] uii; //[27,-14]
reg signed [ui_n-1:0] ui1; //[27,-14]
wire signed [wi_n-1:0] wi; //[20,-14]
wire signed [wi_n-1+6:0] wii; //[26,-14]
wire signed [ud_n-1:0] ud; //[24,-10]
wire signed [ud_n-1+6:0] udd; //[30,-10]
wire signed [e_n:0] wd; //[15,0]
wire signed [ud_n-1+6+(ki_q-kd_q):0] upd; //[34,-14]
wire signed [ui_n-1:0] upid; //[34,-14]
wire signed [dpwm_n+ddpwm_n:0] ux; //[20,0]
wire unsigned [dpwm_n+ddpwm_n-1:0]uh;//[12,0]
reg [dpwm_n-1:0] count;
reg [ddpwm_n-1:0] countm;
reg ddpm_out;
wire [ddpwm_n-1:0] ff;
//**************************************
///Combinationalpart
//**************************************
initial
begin
ui1<={(ui_n){1'b0}};
count<=0;
```

```
countm<=0;
end
always@(negedge clk12)
begin
ee<=e;
uii<=ui;
end

always@(negedge clk)
if(rst)
begin
ui1<={(ui_n){1'b0}};
count<=0;
countm<=0;
end
saturated_multiplier Kp_mult (e,Kp,up,);
defparam Kp_mult.n=e_n,Kp_mult.p=kp_n,Kp_mult.m=up_n;
saturated_multiplier Kp_mult4 (up,6'b011000,upp,);
defparam Kp_mult4.n=up_n,Kp_mult4.p=6,Kp_mult4.m=up_n+6;
saturated_multiplier Ki_mult(e,Ki,wi,);
defparam Ki_mult.n=e_n,Ki_mult.p=ki_n,Ki_mult.m=wi_n;
saturated_multiplier Kp_mult5 (wi,6'b011000,wii,);
defparam Kp_mult5.n=wi_n,Kp_mult5.p=6,Kp_mult5.m=wi_n+6;
saturated_adder Ki_add(ui1,wii,ui,,1'b1);
defparam Ki_add.n=ui_n,Ki_add.p=wi_n+6,Ki_add.m=ui_n;
saturated_adder Kd_sub(e,e1,wd,,1'b0);
defparam Kd_sub.n=e_n,Kd_sub.p=e_n,Kd_sub.m=e_n+1;
saturated_multiplier Kd_mult(wd,Kd,ud,);
defparam Kd_mult.n=e_n+1,Kd_mult.p=kd_n,Kd_mult.m=ud_n;
saturated_multiplier Kp_mult6 (ud,6'b011000,udd,);
defparam Kp_mult6.n=ud_n,Kp_mult6.p=6,Kp_mult6.m=ud_n+6;
saturated_adder           upd_add({upp,{(ki_q-kp_q){1'b0}}},{udd,{(ki_q-
kd_q){1'b0}}},upd,,1'b1);
defparam            upd_add.n=up_n+6+(ki_q-kp_q),upd_add.p=ud_n+6+(ki_q-
kd_q),upd_add.m=ud_n+6+(ki_q-kd_q);
saturated_adder upid_add(upd,ui,upid,,1'b1);
defparam upid_add.n=ud_n+6+(ki_q-kd_q),upid_add.p=ui_n,upid_add.m=ui_n;
assign ux=upid[ui_n-1:ki_q];

assign uh=(ux[dpwm_n]==1'b1)?{(dpwm_n){1'b0}}:ux[dpwm_n-1:0];
assign u = (uh[dpwm_n-1:ddpwm_n]+ddpm_out)>count;
assign ff=uh[ddpwm_n-1:0];
assign clk12=~count[dpwm_n-1];
//ddpm uddp(ff,clk12,countm,u1,rst,clk,dataa5);
//**************************************
//Sequentialpart
//**************************************
always@(posedge clk12)
begin
e1<=ee;
ui1<=uii;
countm<=countm+1;
end
```

```
always@(posedge clk)
begin
count<=count+1;
end

always @ (*)
begin
if (countm[0]==1)
ddpm_out <= ff[3];
else if(countm[1]==1)
ddpm_out <= ff[2];
else if(countm[2]==1)
ddpm_out <= ff[1];
else if(countm[3]==1)
ddpm_out <= ff[0];
else
ddpm_out <= 0;
end

endmodule

module saturated_adder(x,y,z,OV,op);
function integer max;
input integer left,right;
if(left>right)
max=left;
else
max=right;
endfunction
parameter n;
parameter p;
parameter m;//Assumingm<=max(n,p)+1
parameter mx=max(n,p)+1;
input signed [n-1:0]x;
input signed [p-1:0]y;
output reg signed [m-1:0] z;
input op;
output reg OV;
wire signed[n:0]xx={x[n-1],x};
wire signed[p:0]yx={y[p-1],y};
wire signed[mx-1:0]zx;
assign zx=(op==1'b1)?xx+yx:xx-yx;
reg temp;
integer I;
always@(zx)
begin
temp=1'b0;
for(I=m;I<=mx-1;I=I+1)
begin
if((zx[I]^zx[m-1])==1'b1)
temp=1'b1;
end
OV=temp;
```

```
end
always@(OV,zx)
case(OV)
1'b0:z=zx[m-1:0];
1'b1:
begin
if(zx[mx-1]==1'b0)
z={1'b0,{(m-1){1'b1}}};
else
z={1'b1,{(m-1){1'b0}}};
end
endcase
endmodule


module saturated_multiplier(x,y,z,OV);
parameter n;
parameter p;
parameter m;//Assumingm<=n+p
input signed [n-1:0] x;//[n,q]
input signed[p-1:0]y;//[p,l]
output reg [m-1:0]z;//[m,q+l]
output reg OV;
wire signed [n+p-1:0]zx;
assign zx=x*y;
reg temp;
integer I;
always@(zx)
begin
temp=1'b0;
for(I=m;I<=n+p-1;I=I+1)
begin
if((zx[I]^zx[m-1])==1'b1)
temp=1'b1;
end
OV=temp;
end
always@(OV,zx)
case(OV)
1'b0:z=zx[m-1:0];
1'b1:
begin
if(zx[n+p-1]==1'b0)
z={1'b0,{(m-1){1'b1}}};
else
z={1'b1,{(m-1){1'b0}}};
end
endcase
endmodule
```

References

[1] Angel V. Peterchev, S. R. (2003). Quantization Resolution and Limit Cycling in Digitally Controlled PWM Converters. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 8.

[2] P.S. Crovetti, F. G. (n.d.). Dyadic Digital Pulse Width Modulation for Limit Cycle-Free Digitally Controlled Power Converters . 3. 480-484.

[3] Crovetti, P. S. (2017). All-Digital High Resolution D/A Conversion by Dyadic Digital Pulse Modulation. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I*, 12.

[4] Luca Corradini, D. M. (n.d.). *Digital Control of high-frequency Switched-mode Power Converters.* IEEE Press.

[5] Mohan, U. R. (n.d.). *Power Electronics: Converters, Applications and Design.* Addison Wesley.

[6] A. M. Wu, J. X. (1999). PWM control Application in voltage regulation modules. *Proc. IEEE Power Electronics*, 77-83.

[7] A. P. Dancy, R. A. (2000). High-efficiency multiple-output DC–DC conversion for low-voltage systems. *IEEE Trans. VLSI Syst.*, 252-263.

[8] Ang, T. W. (1995). Digital control of switching converters. *IEEE Int. Symp. Ind Electron*,

[9] Angel Vladimirov Peterchev, S. R. (2006). *Digital Control of PWM Converters: Analysis and Application to Voltage Regulation Modules.* University of California at Berkeley: Technical Report No. UCB/EECS-2006-146.

[10] B. J. Patella, A. P. (2003). High-frequency digital PWM controller IC for dc–dc converters. *IEEE Trans. Power Electronics*, 438-446.

[11] Chandrakasan, A. P. (1997). Ultra low power control circuits for PWM converters. *IEEE Power Electron.*, 21-27.

[12] (2008). *Data Conversion HSMC.* ALTERA.

[13] G. F. Franklin, J. D. (1998). Digital Control of Dynamic Systems. *Addison-Wesley*.

[14] G. Luckjiff, I. D. (1995). Interpolative sigma delta modulators for high frequency power electronic applications. *IEEE Power Electron*, 444-455.

[15] Galton, G. a. (2013). A reconfigurable mostly-digital delta-sigma ADC with a worst-case FOM of 160dB. *IEEE J. Solid-State Circuits*, 983-995.

[16] Hao Peng, A. P. (2007). Modeling of Quantization Effects in Digitally Controlled DC–DC Converters. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 8.

[17] Kartik Sharma, D. K. (2017). Design of Digital PID Controller for Voltage Mode Control of DC-DC Converters. (p. 6). Rajasthan, India: IEEE.

[18] Kurian, S. G. (2016). Reduction in limit cycle oscillation and conducted electromagnetic emissions by switching frequency adjustment in digitally controlled DC-DC converters. *EPE Journal*, 9.

[19] Lenk, S. C. (1997). Stabilized power converter having quantized duty cycle. *U.S. Patent 5 594 324*.

[20] P. Murphy, M. X. (n.d.). Study of Digital Vs Analog Control. *Center for Power Electronics Systems*, (p. 4). Blacksburg, VA 24061 USA .

[21] Peng H, M. D. (2004). Modelling of quantization efects in digitally controlled DC–DC converters. *IEEE Trans Power Electron*, 208-215.

[22] Sanders, A. P. (2003). Quantization resolution and limit cycling in digitally controlled PWM converters. *IEEE Trans. Power Electron*, 301-308.