Fast Transient Digitally Controlled Buck Regulator

With Inductor Current Slew Rate Boost

by

Ahmed Hashim

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2013 by the
Graduate Supervisory Committee:

Bertan Bakkaloglu, Chair
Sayfe Kiaei,
Sule Ozev

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Mobile electronic devices such as smart phones, netbooks and tablets have seen increasing demand in recent years, and so has the need for efficient, responsive and small power management solutions that are integrated into these devices. Every thing from the battery life to the screen brightness to how warm the device gets depends on the power management solution integrated within the device. Much of the future success of these mobile devices will depend on innovative, reliable and efficient power solutions. Perhaps this is one of the drivers behind the intense research activity seen in the power management field in recent years.

The demand for higher accuracy regulation and fast response in switching converters has led to the exploration of digital control techniques as a way to implement more advanced control architectures. In this thesis, a novel digitally controlled step-down (buck) switching converter architecture that makes use of switched capacitors to improve the transient response is presented. Using the proposed architecture, the transient response is improved by a factor of two or more in comparison to the theoretical limits that can be achieved with a basic step down converter control architecture. The architecture presented in this thesis is not limited to digitally controlled topologies but rather can also be used in analog topologies as well. Design and simulation results of a 1.8V, 15W, 1MHz digitally controlled step down converter with a 12mV Analog to Digital Converter (ADC) resolution and a 2ns DPWM (Digital Pulse Width Modulator) resolution are presented.

Dedicated to my mother: Mama Adoula

Mama; I deeply miss you. Every thing that I am today is because of you.
May you be showered with Allah's Mercy, Peace and Blessings.

ACKNOWLEDGMENTS

I would like to thank Dr. Bertan Bakkaloglu for his continued support and his motivating words but more importantly for his patience with me. I would also like to thank Dr. Sayfe Kiaei and Dr. Sule Ozev for their time and support.

I would like to thank my parents, especially my father for always guiding me and advising me in all that matters whether small or large. Finally I would like to thank my wife Samra and my children, Salmaan, Abdurahman and Omar for their patience with me as well as their understanding and support.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

A/D	Analog/Digital Converter
ADC	Analog-to-Digital Converter
D/A	Digital/Analog Converter
DAC	Digital-to-Analog Converter
DPWM	Digital Pulse Width Modulator
DSP	Digital Signal Processor
FET	Field Effect Transistor
FPGA	Field Programmable Gate Array
I/O	Input/Output
IC	Integrated Circuit
LSB	Least Significant Bit
N-FET	N-type Field Effect transistor
MSB	Most Significant Bit
NMOS	N-type MOSFET
MOSFET	Metal Oxide Field Effect Transistor
PID	Proportional Integral Derivative Controller
P-FET	P-type Field Effect Transistor
PMOS	P-type MOSFET
S/H	Sample and Hold
SMPS	Switch Mode Power Supply
VHDL	Very-high-speed-integrated-circuit Hardware Description Language

CHAPTER 1: INTRODUCTION

Switch-mode Power Supplies (SMPS) have always been popular due their high efficiency and flexibility. They are flexible in the sense that the output can be a voltage or a current, it can also be lower, higher, or even equal to the input voltage. Their inherent high efficient has become even more desirable as the need for portable battery powered electronic devices has been on the rise. On the other hand they are notorious for generating noise on the output due to their switching nature. They also tend to be large in size because of the need an inductor and large output capacitor. Also when compared to Low Drop Regulators (LDO), switching regulators tend to have a much lower bandwidth and hence a worse transient response.

Traditionally SMPS have been controlled using analog control loops that are generally basic in nature [1]. More advanced control schemes [2] do exist that might help solve some of the drawbacks of SMPS such as its poor transient response, however, these schemes are generally not easily implemented in analog approaches. This is due to the amount of circuits needed to implement these control schemes as well as the analog system susceptibility to noise and signal corruption.

Analog control of switch mode power supplies had been and remains today the workhorse when it comes to control architectures in industry today. While digital control has its advantages, its applications are somewhat niche and most applications are easily serviceable with basic analog control mechanisms. This might be the case today but as technology advances and specifications tighten Digital control will have its day in the spotlight.

A block diagram of a voltage mode analog controlled switching regulator is shown in Figure 1.1 below. A clock pulse turns on the top switch and current starts to flow from the input through the inductor and charges the output voltage. The output voltage is sensed and compared to a reference by an error amplifier. The output of the error amplifier is an error signal that dictates what the top switch duty cycle will be. This error signal is then compared to a ramp signal and the output of that comparator is the reset signal for the top switch. Once the top switch is turned off, the bottom switch is turned on and current in the inductor starts to decrease. Figure 1.2 shows typical voltage waveforms at the switching node and the typical inductor current. For a buck (step-down) regulator as is shown in Figure 1.1, the average of the switch node waveform sets the output voltage while the average inductor current is equal to the output current.

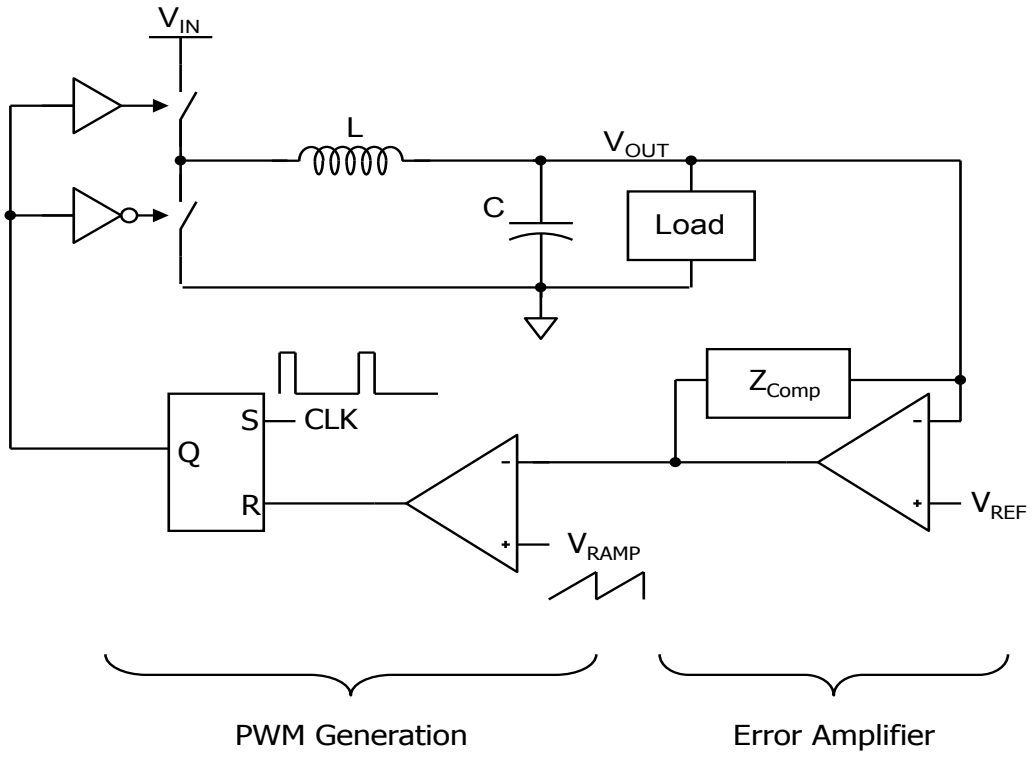


Figure 1.1: Analog Control Loop for Step-Down Regulator

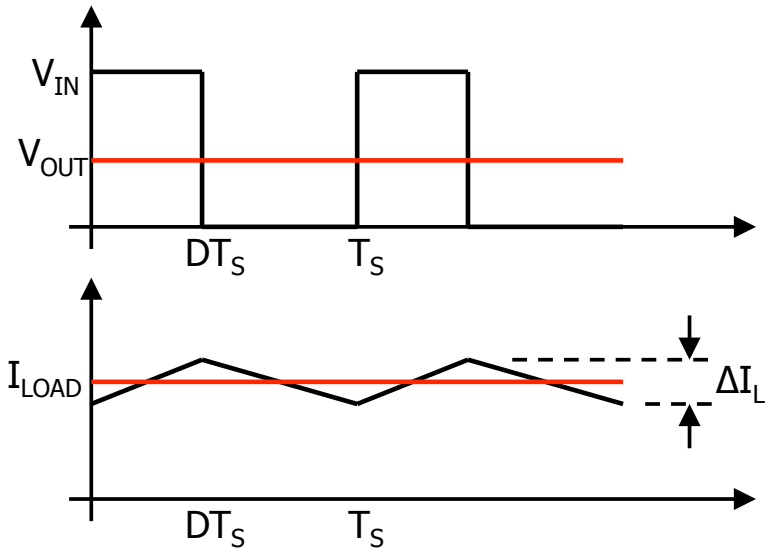


Figure 1.2: Step-Down Regulator Typical Waveforms

Although the type of control explained in the paragraph above is voltage mode, there are other control mechanisms that are not discussed such as “Peak Current Mode Control” [6], “Average Current Mode Control” [7], “Hysteretic Control” etc.

Since digital circuits do not suffer from the signal integrity issues that an analog equivalent circuit suffers from, it is natural to try to adapt a digital solution to the problems faced when trying to advance SMPS control architectures. Over that past decade a lot of research has been done in the area of digital control for SMPS and we are now seeing very innovative ideas that are targeted at not only solving SMPS drawbacks but also push the limits resulting in power solutions that are compatible with the future needs of electronic systems [3][4].

1.1 Research Motivation

As computer processors advance and clocking speeds for these processors increase into the multiple GHz more stringent specifications become required of the power management solutions used. Today's modern processors can have a current demand that goes from a few mili-amperes to hundreds of amperes in a few nano-seconds. More over, the power supply voltage must remain constant or follow a load line specification [17]. It is much harder to increase the switching frequency and bandwidth of switching power converters than it is to do so for a processor hence the output capacitor ends up doing most of the work during load transients which implies that very large capacitors must be used taking up precious board space.

All these factors together make it important for the community to research and propose new and non-traditional methods to improve the transient response of switching power converters. Which is exactly what the work in this thesis has set out to do.

1.2 Contributions

This thesis focuses on solutions that have been developed to improve SMPS transient response and the fundamentals that limit these solutions. A novel approach is introduced that improves SMPS transient response beyond the fundamental limits that constrain the state of the art approaches. Simulation results will be presented to prove the viability of the approach introduced and that it outperforms the state of the art approaches that can be found in the literature.

For the remainder of this thesis, the focus will be on Step-Down SMPS other wise known as "Buck Regulators" since about 90% of the SMPS market is of this nature [5], however the idea proposed in this thesis is by no means only limited to buck regulators.

1.3 Thesis Outline

CHAPTER 2 discusses the architecture behind Digitally Controlled Step-Down SMPS and its various building blocks. CHAPTER 2 also discusses the state of the art methods that are aimed at improving SMPS transient response. CHAPTER 3 proposes a novel approach to improving the transient response of a digitally controlled SMPS that goes beyond anything found in the literature today. CHAPTER 4 then discusses the design and simulations that validate the proposed approach and compares it to the state of the art. Finally, CHAPTER 5 concludes with the outcomes of this research and the proposed future work.

CHAPTER 2: STATE OF THE ART DIGITAL CONTROL

While digital control has not taken off yet as the standard control scheme in switch mode power supplies, as discussed in CHAPTER 1, the future is indeed bright as more advanced control architectures become necessary.

2.1 Digital Control Concepts

A basic digital controller is shown in Figure 2.1 below. The power stage remains identical to that seen when discussing analog control however, the output is sensed by an Analog to Digital Converter (ADC) that generates the error signal. The error signal is then fed into a Proportional Integral Derivative (PID) controller/compensator. The output of the PID controller/compensator is a digital code that is equivalent to the necessary duty cycle. A Digital Pulse Width modulator then converts the PID output code to a pulse at the desired duty cycle, which then is buffered by the driver and turns on/off the power switches.

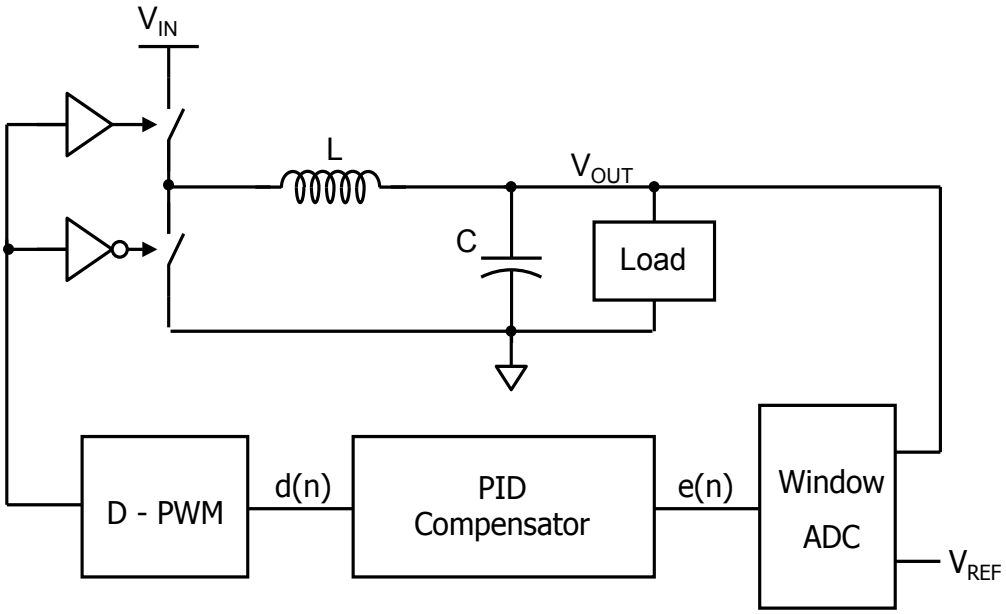


Figure 2.1: Digital Control Loop for Step-Down Regulator

2.1.1 Analog To Digital Converter

The ADC Least Significant Bit (LSB) dictates the resolution to which the output will be regulated. A window ADC is all that is needed for this type of control [8][9]. The Output is compared to the reference voltage and the delta between them is then converted to an error signal. Figure 2.2 below shows a sample of the ADC output bins.

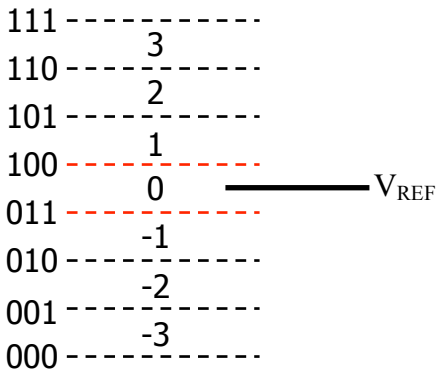


Figure 2.2: ADC Output Bins

A large number of bits is not needed. A few bits around the target output are sufficient for digital control applications. When the output is within 1/2LSB of the reference voltage, the output of the ADC is in the "0" bin which is a "0" error signal telling that PID controller/compensator that no changes to the duty cycle code is needed. If the output is higher than 1/2LSB but less than 1 ½ LSB, the ADC output is code of "1". On the other hand a lower output voltage will generate a negative error code. The ADC code is then processed by the PID to generate the duty cycle command for the next switching period. In the example shown in Figure 2.2 above, the two LSBs give the error code while the MSB acts as the sign bit.

2.1.2 PID Controller

The PID controller implements a second order controller function [8][9][13]. It does so by processing the present cycle error code from the ADC as well as the previous two cycles code. Equation 2.1 shows the relationship between the ADC error code and the duty cycle command.

$$d(n) = d(n-1) + a \cdot e(n) - b \cdot e(n-1) + c \cdot e(n-2) \tag{2.1}$$

In equation (2.1) above, *d(n)* is the present cycle calculated duty cycle, *d(n-1)* is the previous cycle duty cycle. While *e(n), e(n-1)* and *e(n-2)* are the current, previous and two previous cycles ADC error code. Finally *a, b* and *c* are the PID coefficients. Figure 2.3 shows a representation of the PID controller/compensator.

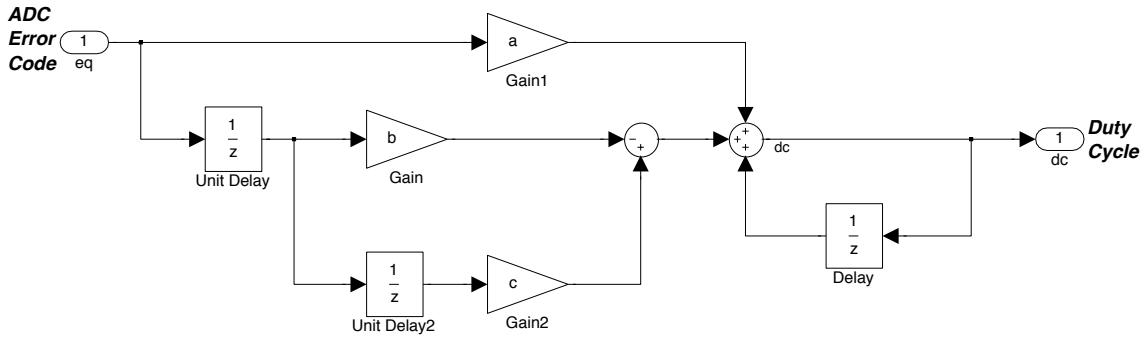


Figure 2.3: PID Block Diagram

Care must be taken when selecting the number of bits that are used to implement the PID multipliers and accumulators as they affect the accuracy of the coefficient representation. The z-domain transfer function of the duty cycle ($\tilde{d}$) to the error code ($\tilde{e}$) is represented in equation (2.2). It is a second order transfer function that implements two zeros, and two poles. One of these poles is at the origin (integrator pole) and the second at high frequency.

$$\frac{\tilde{d}}{\tilde{e}} = \frac{az^2 - bz + c}{z(z-1)} \tag{2.2}$$

2.1.3 Digital Pulse Width Modulator

The resolution of the Digital Pulse width Modulator is very critical to the operation of the controller as a whole. In a buck regulator, the output voltage is equal to the product of the duty cycle and the input voltage ($D \bullet V_{IN}$). In an analog system the duty cycle can change in a continuous fashion hence the output voltage can also be regulated anywhere between zero volts and V_{IN} in a continuous fashion. The DPWM on the other hand sets a duty cycle that is discrete in nature.

$$\Delta V_{OUT} = \frac{V_{IN}}{2^{n\_pwm}} \tag{2.3}$$

The output voltage in a digital system can be anywhere between zero volts and V_{IN} in discrete bins that are ΔV_{OUT} wide. In equation (2.3) above, n_pwm is the number of bits used in the DPWM. In order to avoid an instability phenomenon known as "limit cycle oscillations" [10] ΔV_{OUT} must be designed to be smaller than the ADC LSB. If this is not the case, then the ADC error code will never get to the "0".

There are a few different approaches in the literature that the DPWM is implemented [8][9]. In the coming subsections the most popular and practical of these approaches will be discussed.

The simplest approach to perform the DPWM function is to use a digital clock counter. The circuit counts between 0 and 2^{n_pwm}. Since the circuit has to count the full count every switching period, the counter has to be clocked at a very high frequency clock with a frequency given in equation (2.4).

$$F_{PWM\_clk} = 2^{n\_pwm} \cdot F_{SW} \tag{2.4}$$

Where F_{PWM_clk} is the clock frequency the DPWM counter is clocked at and F_{SW} is the switching frequency for the regulator. While this approach is simple, it requires a very high frequency clock.

Another simple approach to perform the DPWM function is to use a series of cascaded Delay Locked Loop (DLL) stages. Figure 2.4 shows a block diagram of a DLL based DPWM.

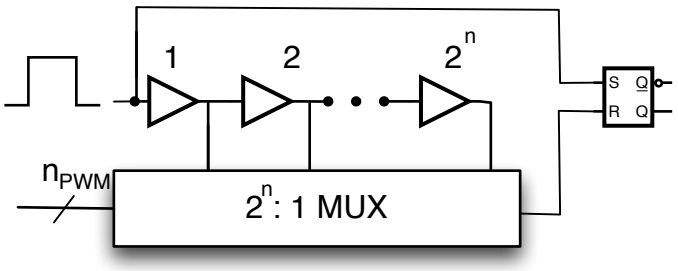


Figure 2.4: DLL based DPWM Block Diagram

A pulse at the regulator switching frequency travels through the DLL stages, as it travels through the first stage it sets the RS latch beginning the switching cycle. A multiplexer selects which stage is used to reset the RS latch based on the input bits that come form the PID controller. This approach relieves the need for a high frequency as in the clock counter discussed above but on the other hand it requires 2^n DLL stages, which can be very area intensive.

Both DPWM approached discussed in the are quite simple and easy to implement, however, they both have their drawbacks. A combination of the two approaches is perhaps the best way to over come the drawbacks while harnessing the simplicity of the approaches discussed. The combined approach splits the DPWM input bits into two sections, "*x*" MSB bits and "*y*" LSB bits. The MSB bits are resolved using the clock

counter approach to set the coarse duty cycle, while the LSB bits are resolved using the DLL approach hence setting the fine duty cycle [8][13].

2.1.4 Advantages of Digital Control

Digital Control brings many advantages to the table when compared to its analog counter part. These advantages are each discussed in the following sub-sections. There are many control architectures that solve various system limitations in very innovative but complex ways. This complexity that comes with advanced control architectures renders them difficult to implement on analog control systems. Thankfully the digital processing revolution has made it easy to implement many complex functions in Digital Signal Processors (DSPs). Perhaps this is one of the greatest advantages of digital control as it opens the door to unique approaches to solve SMPS limitations.

We see today that many electronic products go through multiple cycles each year as the consumer electronics market has become very cutthroat. Such a market needs the system designers to be very quick at implementing their next generation solutions. Digital control allows for rapid prototyping of the power solutions as an FPGA can be used to test and even debug the solution, hence the production products have a higher chance of success at the first pass while delivering solutions to the market in a very timely manner.

Chances are that the system being powered is it self a digital system, whether a micro-controller or a memory chip etc. Using digital control allows the system designer to integrate the power solution on the target IC that is being powered in the first place. This saves precious board space and overall cost.

In general the ADC needs to only sample the output once in every switching cycle. The rest of the cycle the ADC is idle awaiting the next switching cycle to sample

and convert the error to the digital domain. Similarly, the PID controller uses the error signal from the ADC to calculate the duty cycle then remains idle until the next switching period. It is a very logical step to multiplex the ADC and the PID such that they do not remain idle, instead they could be performing their function for a second, or third regulator etc. Again this opens the door to great area and cost savings that are unattainable in analog systems.

One of the nuances of analog control is the external components needed to compensate the regulator, a minimum of two passive components and up to five components are needed to compensate the regulator. All these components replaced with the internal PID coefficients that set the location of the necessary poles and zeros that compensate the controller.

The PID coefficients are certainly easily programmable which allows for a programmable system such that once system design can accommodate a multitude of applications by simply programming the PID coefficients to match with the intended application.

As a system ages or as temperature or voltage conditions change, the system response would no longer be optimum. Digital control again opens a door for the system to adapt the compensation network on the fly by adjusting the PID coefficients [11][12].

A conventional voltage mode analog compensator exhibits peaking in its open loop frequency response due to the LC filter complex poles. The transfer function implemented by the PID compensator allows for complex zeros which can be used to compensate for the complex poles caused by the LC filter resulting in an open loop gain response that is free of peaking. Please refer to Figure 2.5.

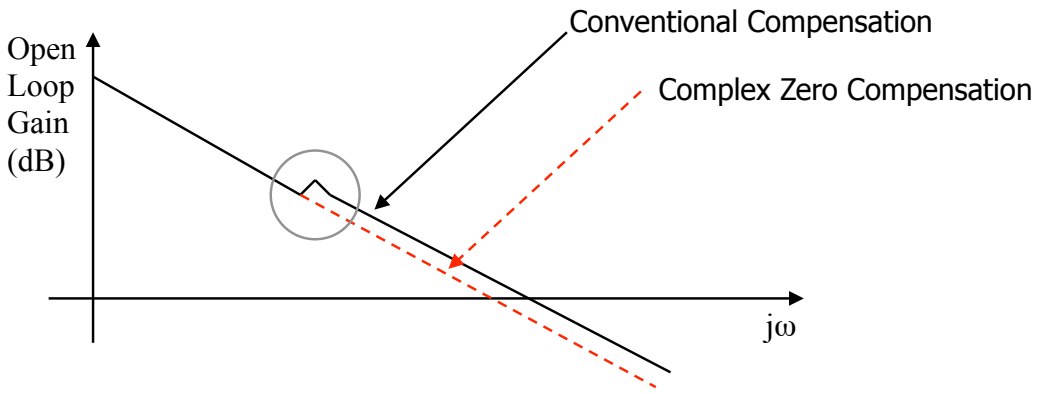


Figure 2.5: Complex Zero Digital Compensator

2.1.5 Digital Control Limitations

While digital control has many advantages it does not come with out limitations. However, for the most part these limitations are not showstoppers in any way. The following paragraphs discuss some of the main limitations of digital control and are followed by Table 2.1, which compares digital control to analog control.

For basic SMPS control, an analog control system is very simple and has withstood the test of time. Once the system is implemented in the digital domain, it becomes more complex in the sense that the ADC is perhaps more complicated than its analog equivalent (error amplifier) and the DPWM is also more complicated than a simple comparator used in analog control.

In a digital control implementation, the PID as well as the DPWM need a very high frequency clock that is not needed in an analog implementations. While this is listed as a limitation, it is not new to digital systems and is commonplace is such systems.

Having an ADC and all the logic gates switching at very high frequency could lead to a digital solution that is more power hungry than an analog equivalent. This becomes less of an issue multiplexing the controller between multiple regulators.

Replacing the analog system error amplifier with an ADC and the PWM comparator with a DPWM while adding a PID will surely result in an overall larger silicon area for the digital comparator when compared to the analog system. However, one must keep in mind that although the silicon area is larger, the over all solution size is smaller since the external compensation components are eliminated.

The regulation accuracy of an analog controller is limited by the reference accuracy but in the digital control we have the added limitation of the ADC resolution. The less the ADC resolution, the less accurately the system will regulate the output to a target value.

Table 2.1: Digital and Analog Control Comparison

Characteristic	*Digital Control*	*Analog Control*
Advanced Control Architectures	Yes	No
Rapid Prototyping	Yes	No
Integrated Power	Yes	No
Single Controller, Multiple Regulators	Yes	No
External Compensation Components	No	Yes
Programmable	Yes	No
Adaptive Control	Possible but	Difficult
Complex Zero Compensation	Possible	Not possible
Complexity	Complex	Simple
High Frequency Clock	Yes	No
Silicon Area	Larger	Smaller
Overall Solution Area	Smaller	Larger
Accuracy	Depends on ADC	Very accurate

2.2 State of the Art Dynamic Improvement Approaches

The demand for higher accuracy regulation and fast response in switching converters is increasing. This requirement stems from the lower output voltages required by sub 100nm processes, where processor clock frequencies exceed 1 GHz and their load currents can step to tens if not hundreds of amperes in a few nano-seconds. These types of conditions place stringent requirements on the regulators that power these processors, as they have to supply such transient currents with out loosing regulation of the output voltage.

A load transient happens when a regulator load current changes rapidity from one value to another. The regulator cannot react instantly due to its bandwidth. While the regulator slews the inductor current to the new load current, the load current is supplied by the output capacitor (in the case where the load current has increased). Since the capacitor is momentarily supplying the load current, the output voltage drops and continues to drop until the inductor current reaches the new load current, please refer to Figure 2.6. Even though the load current is now supplied by the regulator, the output voltage is lower than the target value hence the inductor current overshoots the load current in order to re-charge the capacitor to the target output voltage. Finally, the output voltage reaches the target value and the inductor current settles to the load current and a steady state condition is achieved.

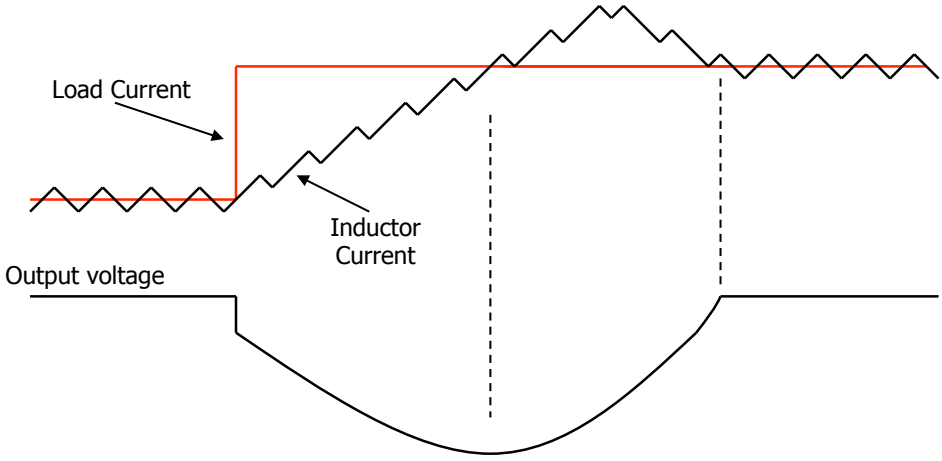


Figure 2.6: Typical Load Transient Event

A negative load transient in which the load current decreases from a high value to a lower one works in much the same way that a positive transient works except that the output voltage overshoots the target value rather than undershoots it.

Several papers have been published with ideas to improve converter transient response [2][3][4]. They propose numerous solutions that try to improve the dynamic response of digitally controlled regulators. Some of these solutions lend them selves well to digital control, as they would be very complicated to implement in analog control approaches. In this chapter several of these solutions along with their limitations are discussed and compared.

2.2.1 Hysteretic Control

Hysteretic control is very popular in application where transient response is to be optimized. It is easily adaptable to digital control where the ADC samples the output voltage or can even be asynchronous using a window comparator around the desired output voltage [14]. For a positive load transients load transient once the output drops below a set threshold the regulator turns on the top switch for a fixed on time. The

regulator then turns the top switch off until the next switching frequency clock pulse; it then turns the top switch on again for a fixed on time. This behavior continues until the output voltage reaches another set threshold above the target output voltage.

This type of architecture works well for low load current but is not practical for high output currents since the inductor operates in discontinuous mode. Also since the output is always hunting between the two set thresholds around the target output voltage, it is not the best topology for tight output regulation. Yet another limitation is the lack of a constant switching frequency but rather bursts of pulses at a given frequency followed by a period of no switching that depends on the load and output capacitor value. If the bursts of pulses happen at a low enough frequency, audible noise caused by the lower frequency and the ceramic output capacitor can be heard which is undesirable.

2.2.2 Non-Linear ADC

Another interesting approach that aims to solve the transient response issues is the use of an ADC that has non-linear output bins [15][16]. An example of the ADC output bins is shown in Figure 2.7. As the output voltage deviates from the target ("0" bin), the ADC code increases in a non-linear fashion effectively skipping codes.

This approach improves the transient response by effectively increasing the system gain as the output voltage deviates from the target value. The higher the ADC error code is, the more the PID will try to compensate by adjusting the duty cycle code in order to correct for the output voltage deviation quicker.

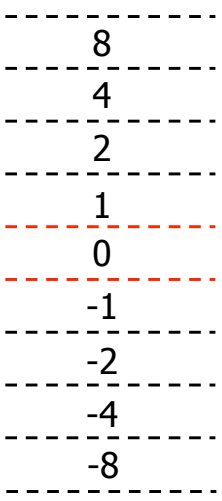


Figure 2.7: Non-Linear ADC Output Bins

The limitation of this approach is that it is only incremental in the sense that it only marginally improves the system transient response. To its credit, this approach does not compromise the loop stability since around the target output voltage the ADC behaves linearly.

2.2.3 Time Optimal Control

Time optimal control perhaps achieves the best possible transient response that can be found in the literature [3][4]. This approach combines a form of hysteretic control together with a standard PID controller. The PID controller regulates the output in steady state. During a positive transient event if the output deviates from the target value by a set amount, the converter enters into a hysteretic mode where the top switch is turned on disregarding the duty cycle command set by the PID, see Figure 2.8 and Figure 2.9. The top switch stays on after the inductor current has reached the load current for a time "T_2" which is

determined by the steady state duty cycle "*D*" and the Time it takes for the inductor current to slew up to the load current "T_1". The relationship between "T_2" and "T_1" is shown in equation (2.5). After the inductor reaches its peak value, the top switch is turned off and the bottom switch is turned on to slew the inductor current back to the value of the load current at which point the PID controller takes over again. The time for which the bottom switch is turned on "T_3" also depends on "T_1" and "*D*" as shown in equation (2.6).

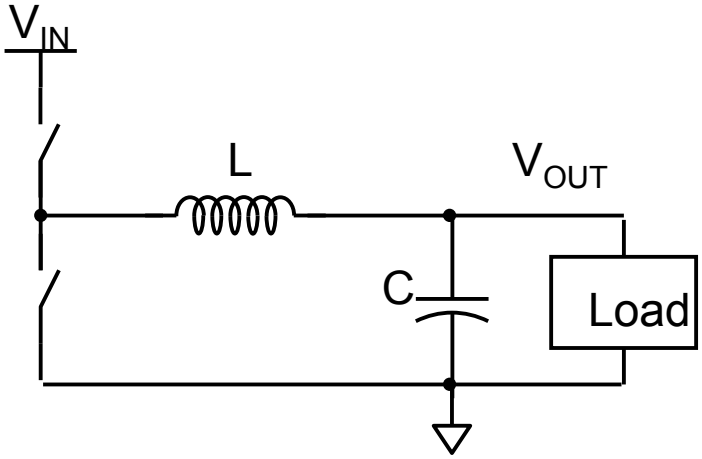


Figure 2.8: Buck Regulator Power Stage

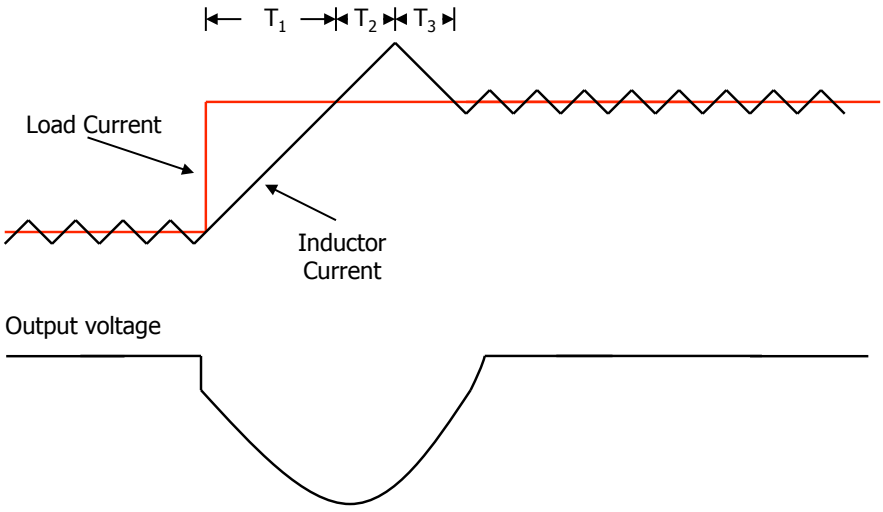


Figure 2.9: Time Optimal Control Waveforms

$$T_2 = \sqrt{D} \cdot T_1 \tag{2.5}$$

$$T_3 = \frac{1-D}{\sqrt{D}} \cdot T_1 \tag{2.6}$$

This approach pushes the transient response to the fundamental limits set by the input voltage, output voltage and the inductor value "*L*". The maximum inductor slew rate is given by equation (2.7), while the time it takes for the inductor current to reach the load current "t_{isp}" is given in equation (2.8) as a function of the Load current step "ΔI_L", the inductor value, the input voltage and the output voltage.

$$Ind.Current + ve\ Slew\ Rate = \frac{V_{IN}-V_{OUT}}{L} \tag{2.7}$$

$$t_{isp} = \frac{\Delta I_{LOAD} \cdot L}{V_{IN}-V_{OUT}} \tag{2.8}$$

The explanation above was for a positive load transient but all the concepts would remain the same for a negative load transient except that the bottom switch would be activated first to slew the inductor current down and then the top switch would be turned on to bring the inductor current back to the load current value. Also the Max slew rate of the inductor current would now only depend on V_{OUT} and the inductor value "*L*" as in the relationship in equation (2.9). Also, the time it takes for the inductor current to reach the load current "t_{isn}" is given in equation (2.10) as a function of the Load current step "ΔI_L", the inductor value, the input voltage and the output voltage.

$$Ind.Current - ve\ Slew\ Rate = \frac{-V_{OUT}}{L} \tag{2.9}$$

$$t_{isn} = \frac{\Delta I_{LOAD} \cdot L}{V_{OUT}} \tag{2.10}$$

There are no real limitations to speak of about this approach except that it is limited by the fundamentals of the input and output conditions as well as the inductor value.

A comparison of the three methods presented in this chapter and their advantages as well as their limitations is given in Table 2.2 below. It is obvious from Table 2.2 that the best solution out of the three is the "Time Optimal Control".

Table 2.2: State of the Art Transient Response Improvement Methods Comparison

Characteristic	*Hysteretic Control*	*Non-Linear ADC*	*Optimal Control*
Suitable for High Currents	No	Yes	Yes
Low Ripple	No	Yes	Yes
Simple	Yes	Yes	No
Requires Large Capacitor	Yes	No	No
Fastest Response	No	No	Yes

CHAPTER 3 PROPOSED SLEW RATE BOOSTING METHOD

As discussed in CHAPTER 2, the best approach to achieve the fastest transient response is the "Time Optimal Control" approach. It is limited only by the power stage fundamentals. The inductor current rises at the fastest rate that it can which is dictated by the input voltage, output voltage and the inductor value as in equation (2.7).

Figure 3.1 illustrates the proposed system power stage that aims to defeat the fundamental limitations of "Time Optimal Control" discussed in CHAPTER 2. The idea is quite simple; the input voltage is stored in capacitors C1 and C2. During a load transient event the capacitors C1 and C2 are switched in a configuration that effectively increases the voltage across the inductor by V_{IN}. The extra voltage across the inductor increases the slope of the inductor current as it slews to the new load current value hence decreasing the time it takes to do so.

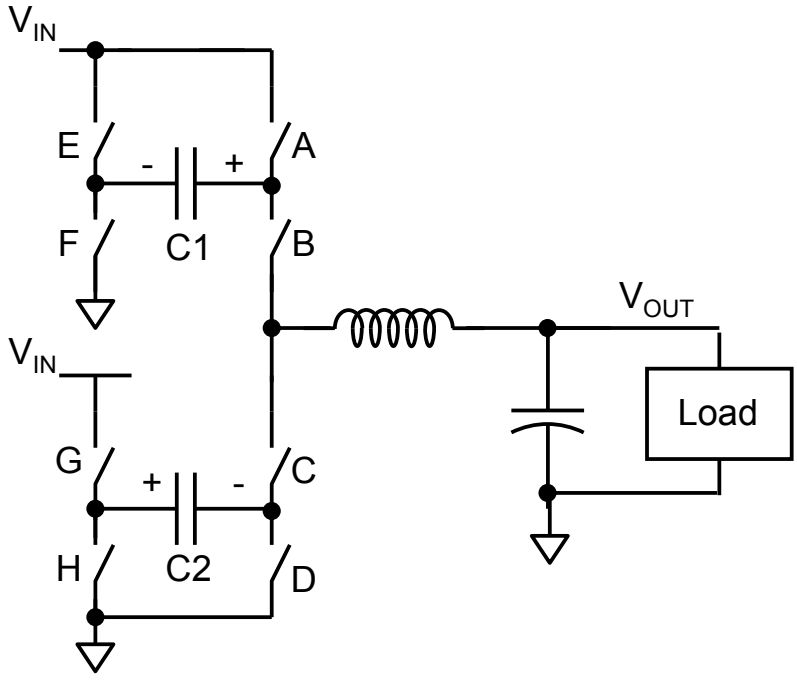


Figure 3.1: Proposed Architecture Power Stage

Switches A, B, E and F are configured in an H-Bridge configuration with capacitor C1 connected at the center of the bridge. Conversely, switches C, D, G and H are also connected in an H-Bridge configuration with C2 at the center of the bridge. In normal operation switches A, F, G and D are in the turned on while switches E and H are turned off. Switches B and C operate as typical step-down converter switches would normally operate. The top switch being switch B and the bottom switch being switch C.

Proposed System

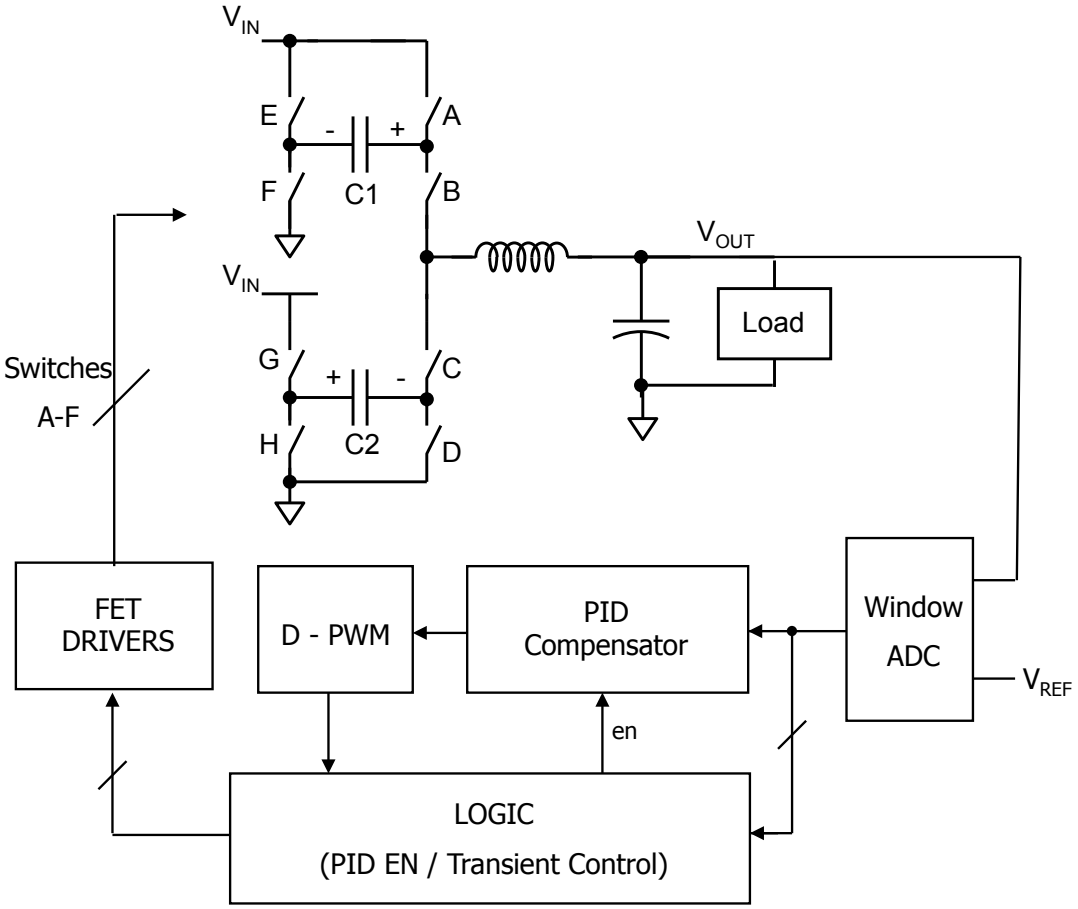


Figure 3.2: Proposed System Architecture

Figure 3.2 shows the complete proposed architecture, both the power stage configuration as well as the control block diagram. In steady state the PID is in control of

the duty cycle and switches B and C are turning on and off based on the duty cycle set by the DPWM. A transient recovery condition is signaled based on the error code from the ADC. Once the logic block flags the transient recovery condition, the PID is ignored and the switches states is not set by the logic block to implement a time optimal algorithm as discussed in CHAPTER 2.

During a positive load transient condition when the load increases to a higher value, switches A and F are turned off while switches B and E are turned on. Switches D and G remain on while switches C and H remain off. With this switch configuration, the effective voltage at the switching node is $2V_{IN}$ rather than V_{IN}. This effectively increases the voltage across the inductor and hence the inductor current slew rate is increased allowing the inductor current to reach the load current value in less time. The Inductor current slew rate is given in equation (3.1) while the time it takes the inductor current to reach the load current is given in equation (3.2).

$$Ind.Current + ve\ Slew\ Rate = \frac{2V_{IN} - V_{OUT}}{L} \tag{3.1}$$

$$t_{isp} = \frac{\Delta I_{LOAD} \cdot L}{2V_{IN} - V_{OUT}} \tag{3.2}$$

In the equations above, "t_{isp}" is the time it takes for the inductor current to reach the load current while "ΔI_L" is the Load current step. Once the inductor current reaches the load current, the switches are kept in the same configuration for a calculated time to recharge the output capacitor after which switch B is turned off and switch C is turned on to bring back the inductor current to the value of the load current. The PID controller now takes over control and the system is back in steady state. Figure 3.3 below shows the

inductor current waveforms compared to time optimal control as introduced in the previous chapter. Equation (3.3) shows the improvement factor "IF_P" in the time it takes to reach steady state in this proposed approach when compared to the time optimal control presented in the previous chapter.

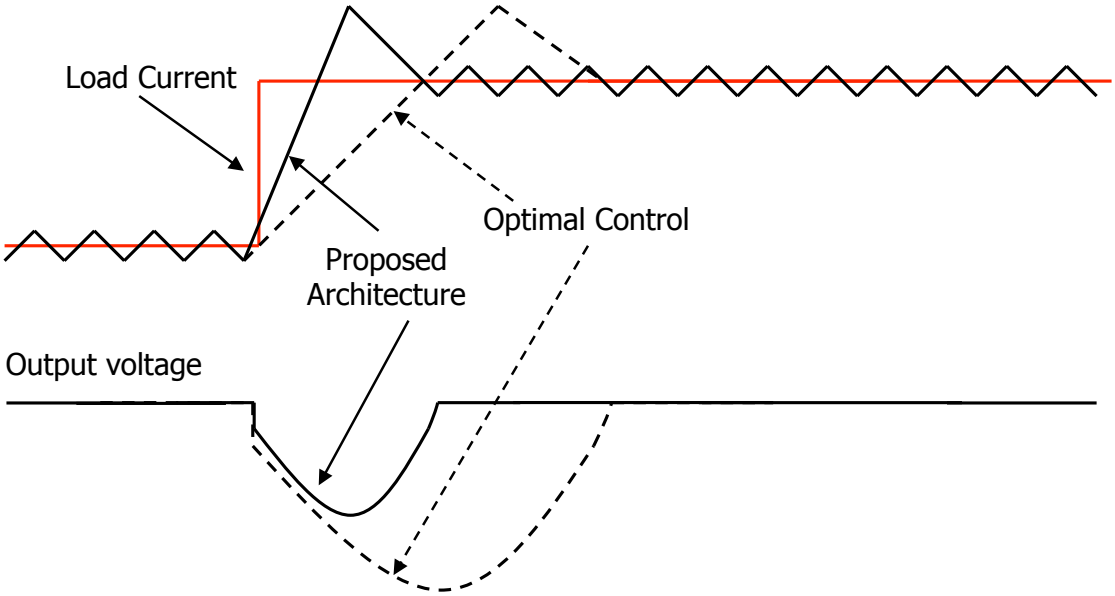


Figure 3.3: Transient Waveforms

$$IF_P = \frac{2-D}{1-D} \tag{3.3}$$

In equation (3.3) above, "D" is the steady state duty cycle determined by V_{IN} and V_{OUT}. It is evident that not only is the time it takes for the inductor to reach the load current decreased by a factor greater than 2 but also the output voltage does not drop during the transient as it would normally drop in time optimal control.

During a negative load transient condition when the load decreases to a lower value, switches D and G are turned off while switches C and H are turned on. Switches A and F remain on while switches B and E remain off. With this switch configuration, the effective voltage at the switching node is negative V_{IN} rather than zero volts. This

effectively increases the voltage across the inductor and hence the inductor current slew rate absolute value is increased allowing the inductor current to reach the load current value is less time. The Inductor current slew rate is given in equation (3.4) while the time it takes the inductor current to reach the load current is given in equation (3.5).

$$Ind.Current - ve\ Slew\ Rate = \frac{V_{OUT}+V_{IN}}{L} \tag{3.4}$$

$$t_{isn} = \left| \frac{\Delta I_{LOAD} \cdot L}{V_{IN}+V_{OUT}} \right| \tag{3.5}$$

In the equations above, "t_{isn}" is the time it takes for the inductor current to reach the load current while "ΔI_L" is the Load current step absolute value. Once the inductor current reaches the load current, the switches are kept in the same configuration for a calculated time to discharge the output capacitor after which switch B is turned on and switch C is turned off to bring back the inductor current to the value of the load current. The PID controller now takes over control and the system is back in steady state.

$$IF_N = 1 + \frac{1}{D} \tag{3.6}$$

Equation (3.6) shows the improvement factor "IF_N" in the time it takes to reach steady state in this proposed approach when compared to the time optimal control presented in the previous chapter. It can be seen that the improvement factor is greater than or equal to 2 for all values of the steady state duty cycle "D". Table 3.1 compares key performance factors for both the proposed architecture and time optimal control as presented on the previous chapter.

Table 3.1: Comparison of Proposed and State of the Art Methods

Characteristic	*Proposed Architecture*	*Optimal Control*
Architecture Complexity	Yes	Yes
Inductor Slew Rate (+ve)	$\frac{2V_{IN} + V_{OUT}}{L}$	$\frac{V_{IN} + V_{OUT}}{L}$
+ve Transient Improvement	$\frac{2 - \mathrm{D}}{1 - D}$	1
Inductor Slew Rate (-ve)	$-\frac{V_{OUT} + V_{IN}}{L}$	$-\frac{V_{OUT}}{L}$
-ve Transient Improvement	$1 + \frac{1}{D}$	1

CHAPTER 4: DESIGN AND SIMULATIONS

In order to verify the proposed architecture the system was designed in Matlab and modeled using Simulink. Three systems were modeled; a conventional PID, a time optimal controller and the system proposed as in Figure 3.2. All three systems were modeled in order to compare the transient response of all three systems to quantify the improvements attained with the proposed system.

Table 4.1: System Design Specification

Parameter	*Value*
V_{IN}	3.3V – 5V
V_{OUT}	1.8V
V_{OUT} *Tolerance* ($V_{OUT\text{-}T}$)	2%
F_{SW}	1MHz
Max I_{LOAD}	8A

4.1 System Design

Given the specifications is Table 4.1, the power stage is first designed per equations (4.1) through (4.3) below. The inductor is chosen to limit the ripple current to less than 20% of the maximum current.

$$\Delta I_L = 0.2 I_{O\max} = 1.6A \tag{4.1}$$

$$L > \frac{V_O(1-D)}{\Delta I_L} T_S = \frac{1.8V(1-0.55)}{1.6A} 1\mu s = 4.7\mu H \tag{4.2}$$

$$C > \frac{\Delta I_L \cdot T_S}{8 \cdot \Delta V_O} = \frac{1.6A \cdot 1\mu s}{8 \cdot 4mV} = 50\mu F \tag{4.3}$$

4.1.1 ADC & DPWM Design Considerations

The Window ADC must have enough resolution to regulate the output at the given tolerance of 2%. To ensure this condition is met the ADC resolution "LSB_{ADC}" is set based on equation (4.4).

$$LSB_{ADC} < V_{OUT} \cdot V_{OUT-T} \tag{4.4}$$

Based on equation (4.1) above, the ADC LSB must be less than 36mV. An LSB of 12mV is chosen to allow for margin and a tightly regulated output. A 4-bit design is chosen with the MSB as the sign bit and the remaining bits as the magnitude of the error.

The DPWM resolution must be carefully chosen with respect to the ADC resolution to avoid limit cycle oscillations as described in CHAPTER 2. The DPWM LSB must satisfy the relationship in equation (4.5) while the number of DPWM bits "n_{DPWM}" can be determined using equation (4.6).

$$LSB_{DPWM} < \frac{LSB_{ADC}}{V_{IN-MAX}} T_{SW} \tag{4.5}$$

$$n_{DPWM} = round_{\ UP}\left[log_2\left(\frac{T_{SW}}{LSB_{ADC}}\right)\right] \tag{4.6}$$

In the equations above, "$V_{IN\text{-}MAX}$" is the maximum input voltage for the design and "T_{SW}" is the switching period. Based on equation (4.2), the LSB for the DPWM should be less than 2.4ns hence a 2ns resolution is chosen. This leaves us with a 9-bit DPWM for this design.

4.1.2 PID Controller

The PID controller is used for steady state operation and does not have special considerations per say. Matlab was used to design a traditional analog voltage mode controller compensation. The analog design is then converted to a digital design using a bilinear transformation yielding the compensator coefficients for the PID controller.

The number of bits used to implement the PID coefficients dictates the accuracy of the compensation and how well it represents an analog equivalent. For this design, 16-bits were used for the PID. The coefficients that were generated by Matlab were then rounded off to their 16-bit equivalents and the final z-domain transfer function was generated and is shown in equation (4.7) below. Please refer to the APPENDIX for the Matlab code.

$$\frac{\tilde{d}}{\tilde{e}} = \frac{59.47z^2 - 117z + 57.52}{z^2 - 1.5z + 0.4997} \tag{4.7}$$

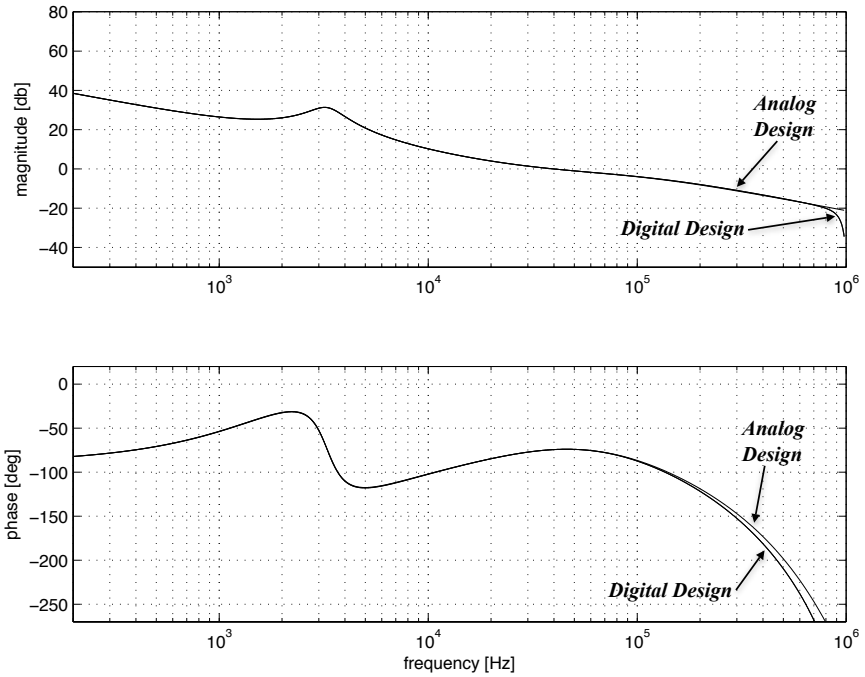


Figure 4.1: Analog and Digital System Bode-plot

The bode-plot of the system AC response is shown in Figure 4.1. The figure shows the ideal analog controller response as well as the design digital controller response and they both match closely as can be seen. The system bandwidth is 35kHz with about 85° of phase margin.

4.1.3 Storage Capacitors & Transient Detection

The storage capacitors "*C1*" and "*C2*" in Figure 3.2 are very key to the operation of this proposed architecture. During the transient event, the capacitors hold up the switch node at the transient recovery voltage and supply the current that flows in the inductor until the output has settled.

It is acceptable for the capacitor voltage to discharge as long as it does not get to the point where it is charged to a negative voltage. To ensure this condition is met, the capacitor must be carefully designed. For a positive load transient, equation (4.8) describes the capacitor value "*C1*" needed to prevent the capacitor voltage from discharging more than a given voltage "ΔV_{cap}" for a given load transient "ΔI_{LOAD}". Equation (4.9) describes the criteria to design capacitor "*C2*" for a negative transient.

$$C_1 > \frac{L(\Delta I_{LOAD})^2}{2(\Delta V_{CAP})(2V_{IN}-V_{OUT})} \tag{4.8}$$

$$C_2 > \frac{L(\Delta I_{LOAD})^2}{2(\Delta V_{CAP})(V_{IN}+V_{OUT})} \tag{4.9}$$

A transient event is triggered if the ADC error code reaches ±5. This tells us that the output voltage went above or below 60mV from the reference. At this point the transient recovery circuit is activated as explained in the previous chapter. Once the error

goes to a code of 1 in the opposite polarity of the transient then the transient recovery circuit is deactivated.

A summary of the design parameters for the system to be modeled can be found in Table 4.2 below.

Table 4.2: System Design Summary

Parameter	*Value*
Ripple Current	1.6A
Inductor	4.7μH
Capacitor	50μF
N_{ADC}	4 bits
LSB_{ADC}	12mV
$N_{PID\text{-}Coefficients}$	16 bits
$N_{PID\text{-}Arithmetic}$	18 bits
N_{DPWM}	9 bits
LSB_{DPWM}	2ns
Transient Detection Threshold	60mV
System Bandwidth	35kHz
System Phase Margin	85°

4.2 Simulation Models & Results

4.2.1 System Models

The Matlab code shown in the APPENDIX generates all the design parameters and PID coefficients needed for the system model. Simulink Models are then used to verify the complete system together including the PID for steady state as well as the control logic for the transient recovery. The Simulink models used to verify the proposed design are shown below.

Figure 4.2 shows the complete system model, while Figures 4.3, 4.4 and 4.5 show the power stage model, the PID model and the transient recovery logic model respectively.

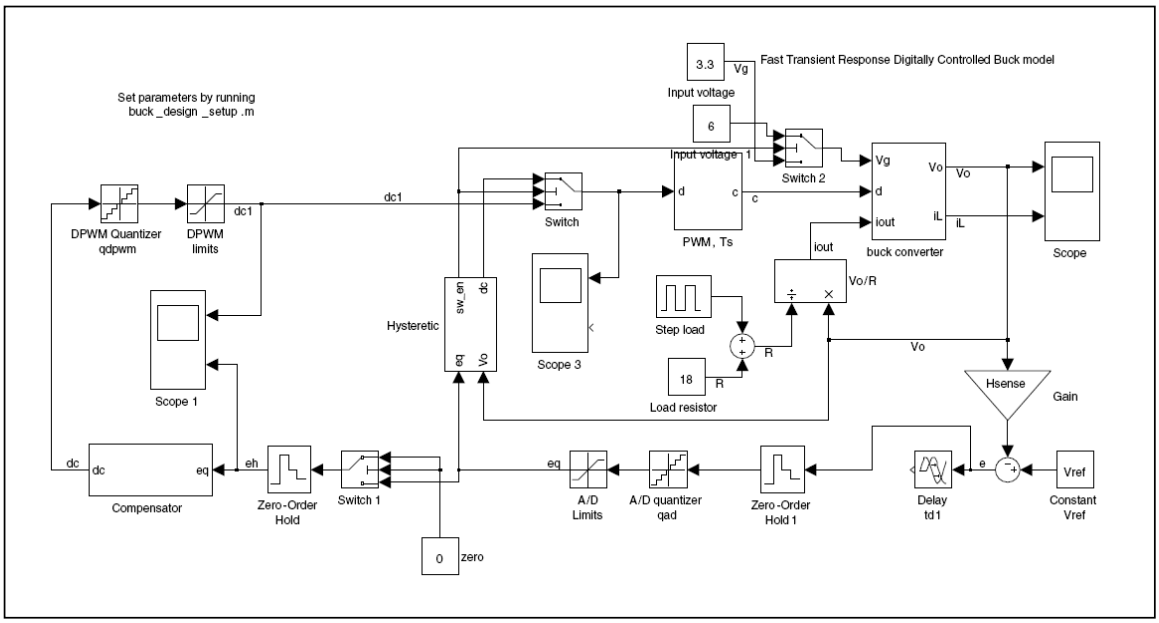


Figure 4.2: Proposed System Simulink Model

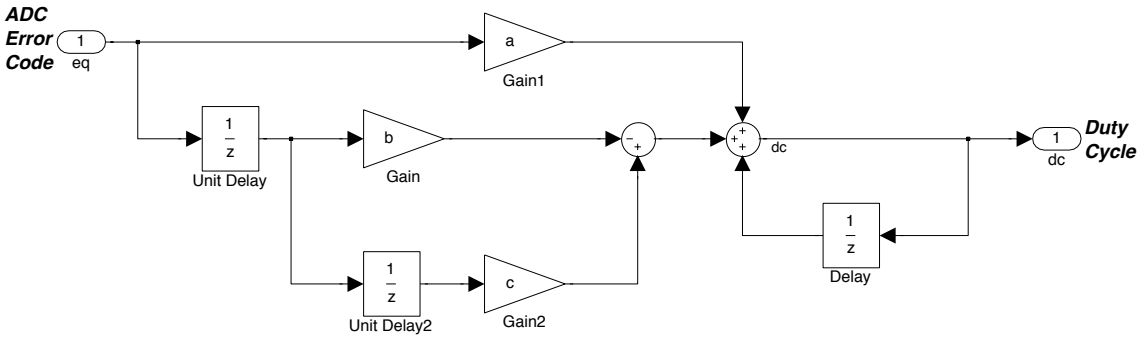


Figure 4.3: PID Simulink Model

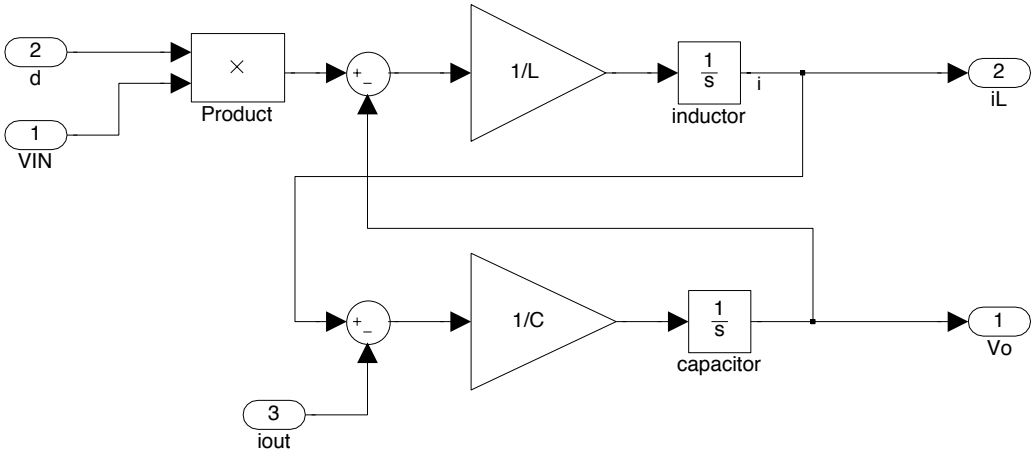


Figure 4.4: Power Stage Simulink Model

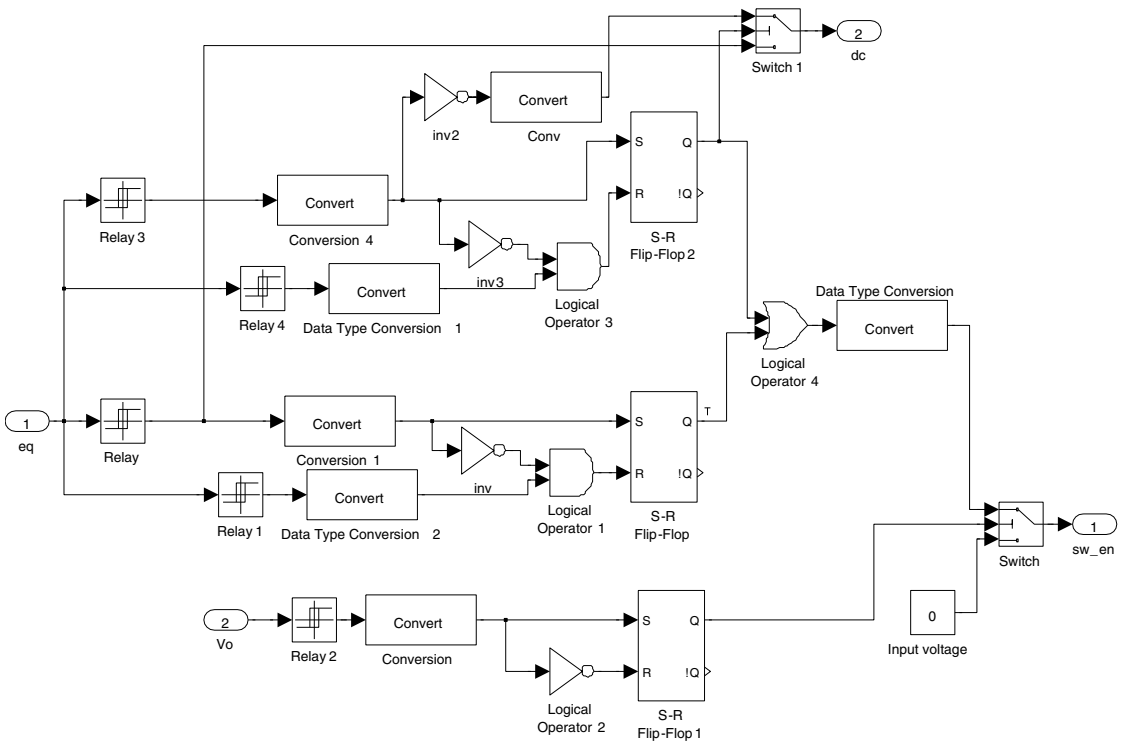
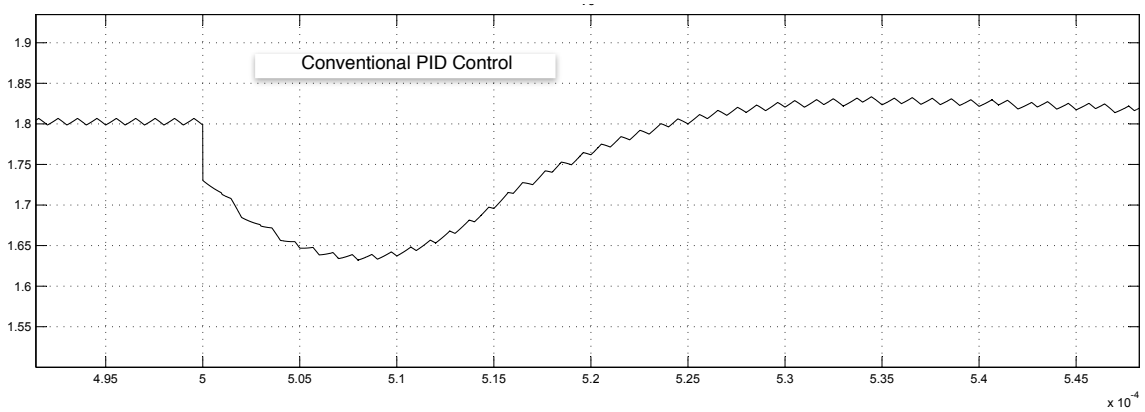


Figure 4.5: Transient Detection and Recovery

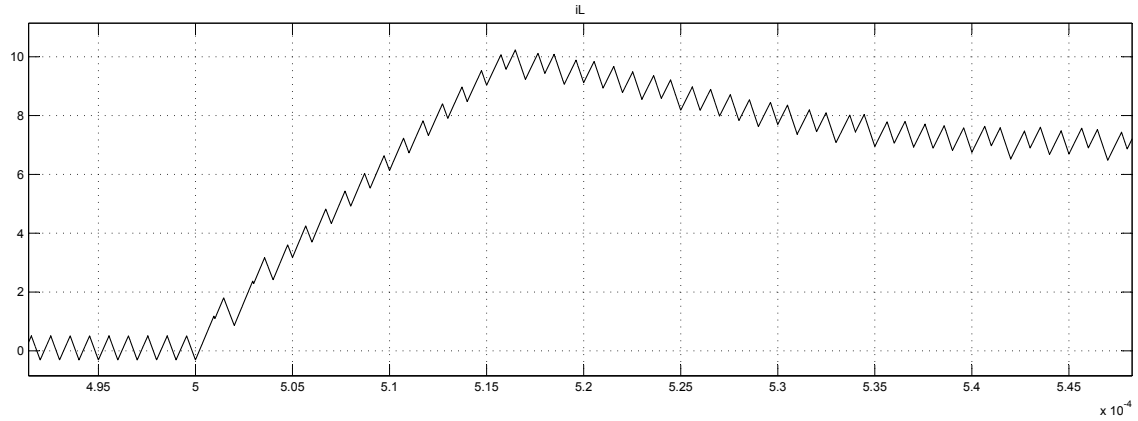
4.2.2 Simulation Results

The Figures below show the Simulink model simulation results of a 7A load transient step. Figure 4.6 shows the conventional PID transient response while Figure 4.7 shows the optimal control transient response and Figure 4.8 shows the proposed control architecture transient response. Finally Figures 4.9 and Figure 4.10 show comparisons of the transient response of the proposed idea with conventional and optimal control respectively.

It can be seen from plots that the proposed control architecture settles to within 2% of the final value over three times faster than optimal control and around one hundred times faster than the conventional PID. Also the output voltage for the proposed controller drops less than half as much as the conventional PID controller drops.



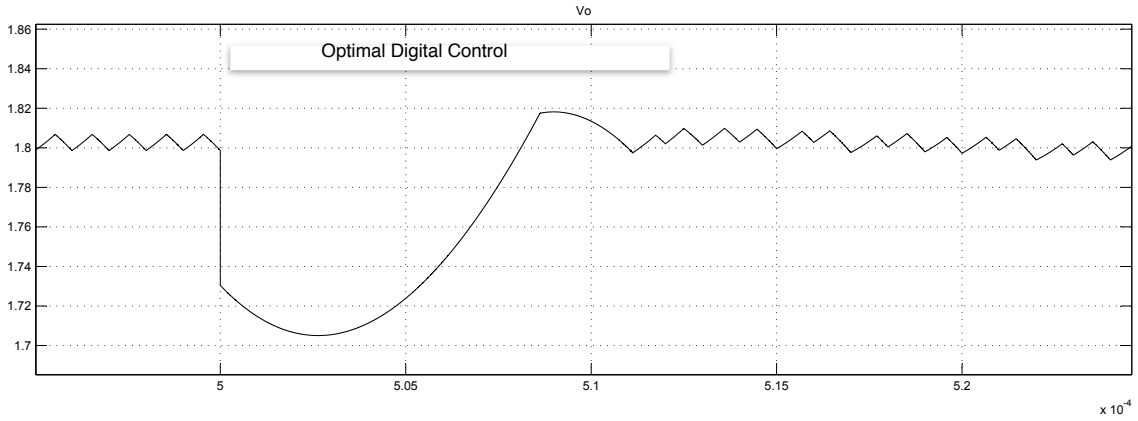
(a)



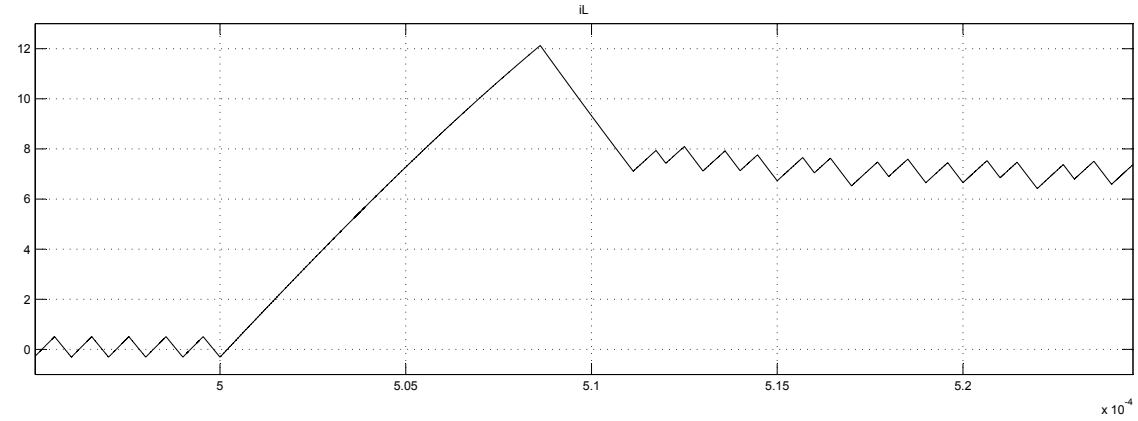
(b)

Figure 4.6: Conventional PID Simulation Results

(a) output voltage response (b) inductor current response



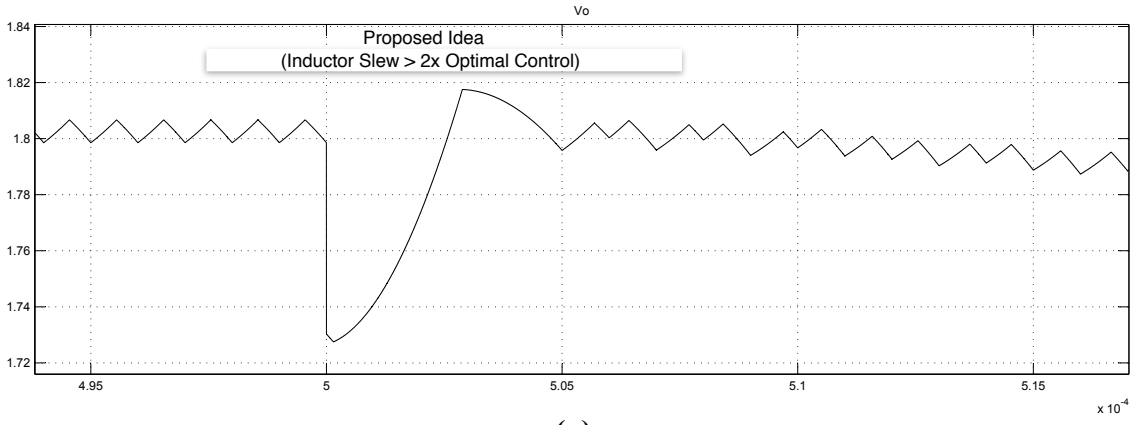
(a)



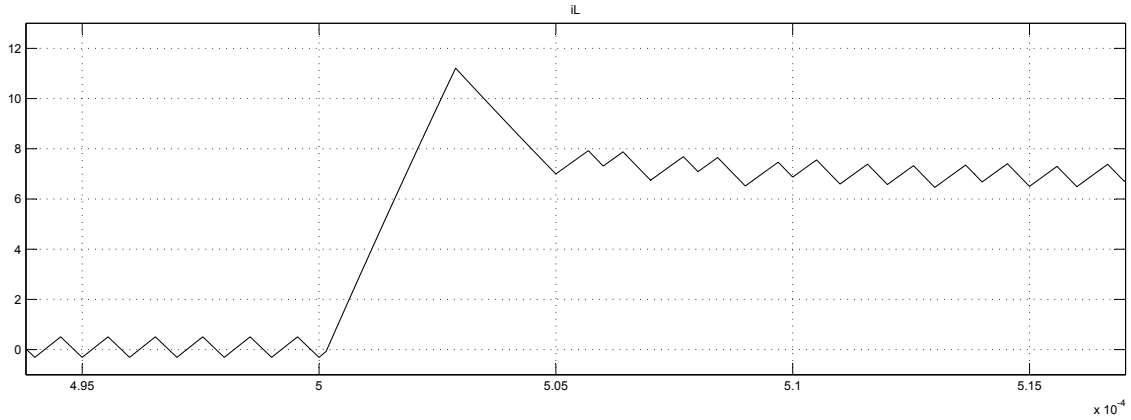
(b)

Figure 4.7: Optimal Control Simulation Results

(a) output voltage response (b) inductor current response



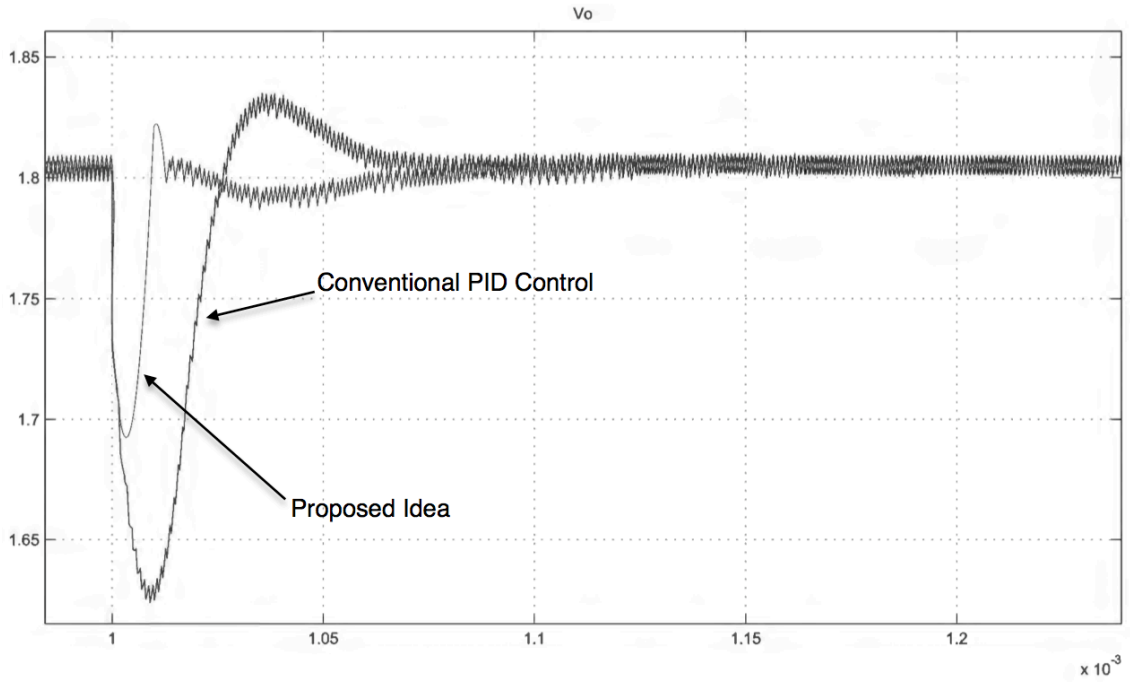
(a)



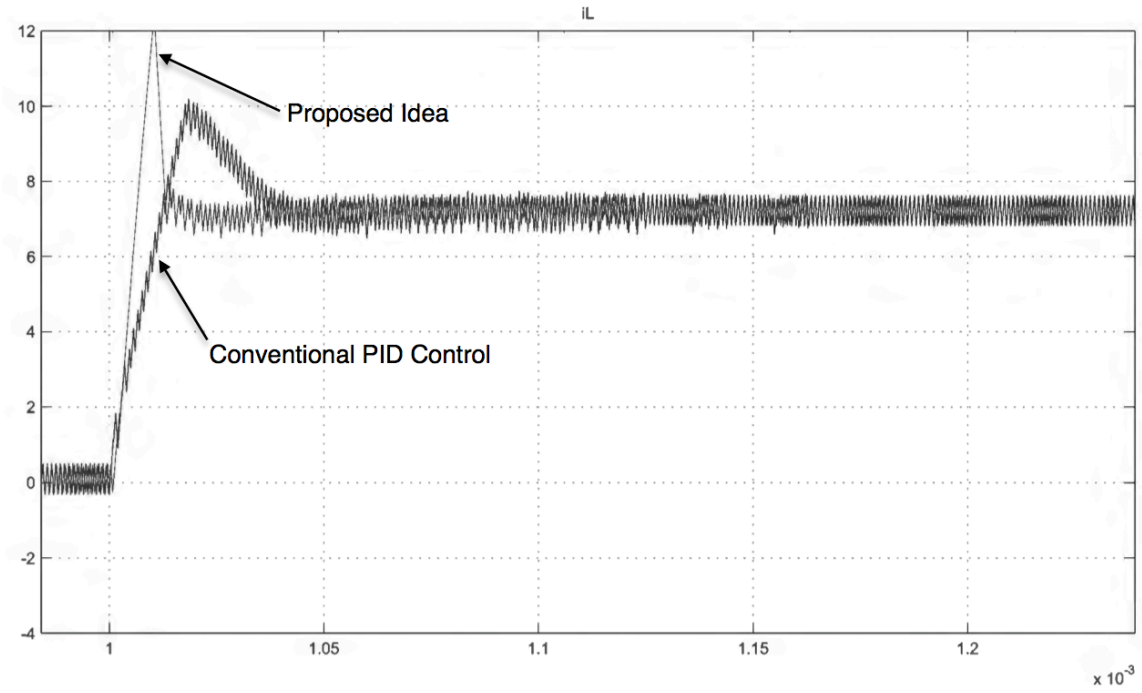
(b)

Figure 4.8: Conventional PID Simulation Results

(a) output voltage response (b) inductor current response



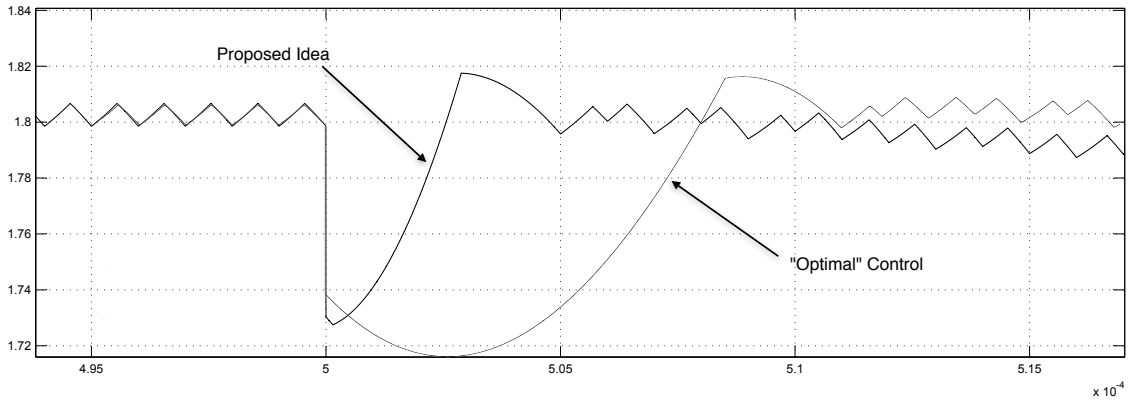
(a)



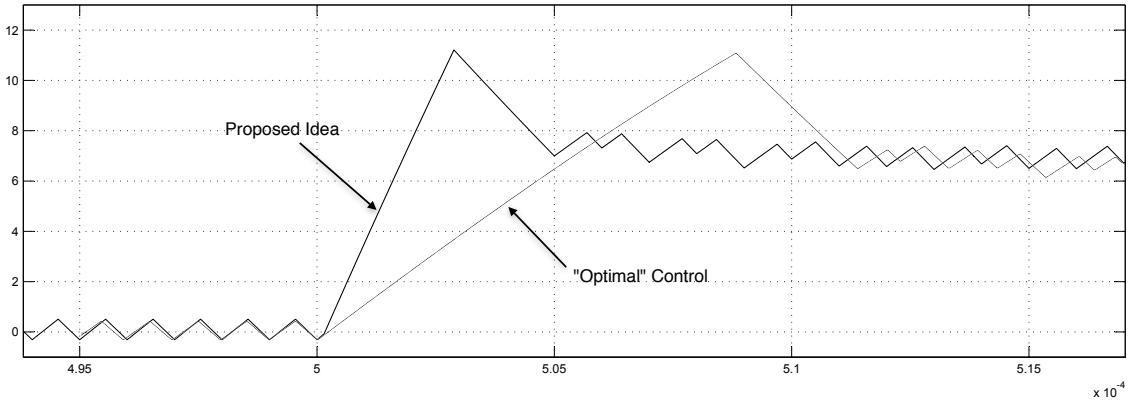
(b)

Figure 4.9: Comparison of Proposed and Conventional PID Control

(a) output voltage response (b) inductor current response



(a)



(b)

Figure 4.10: Comparison of Proposed and Optimal Control

(a) output voltage response (b) inductor current response

Table 4.3: Result Comparison of Proposed, Optimal and Conventional PID

Control Method	*Output Undershoot (mV)*	*2% Settling Time (μs)*
Conventional PID	180	*215*
Optimal Control	80	7.8
Proposed Control	*70*	2.3

4.3 Transistor Level Analog Model & Simulations

To prove the viability of the proposed approach to analog designs as well, an analog system implantation was simulated at the transistor level. Due to process parameter limitations the design was simulated for a 5V input to 3.3V output condition rather than the 3.3V input to 1.8V output conditions used for the digital system. To simplify the design, a non-synchronous implementation was used where a diode is used in place of the bottom switch.

Figure 4.11 shows the toplevel schematic of the analog non-synchronous buck regulator system. Figure 4.12 shows the transient response of both the proposed system as well as the traditional system with the transient recovery circuit disabled. Similar to the digital system simulations, the proposed circuit output voltage dropped half as much as the traditional system and it settled in half the time it took the traditional system to settle.

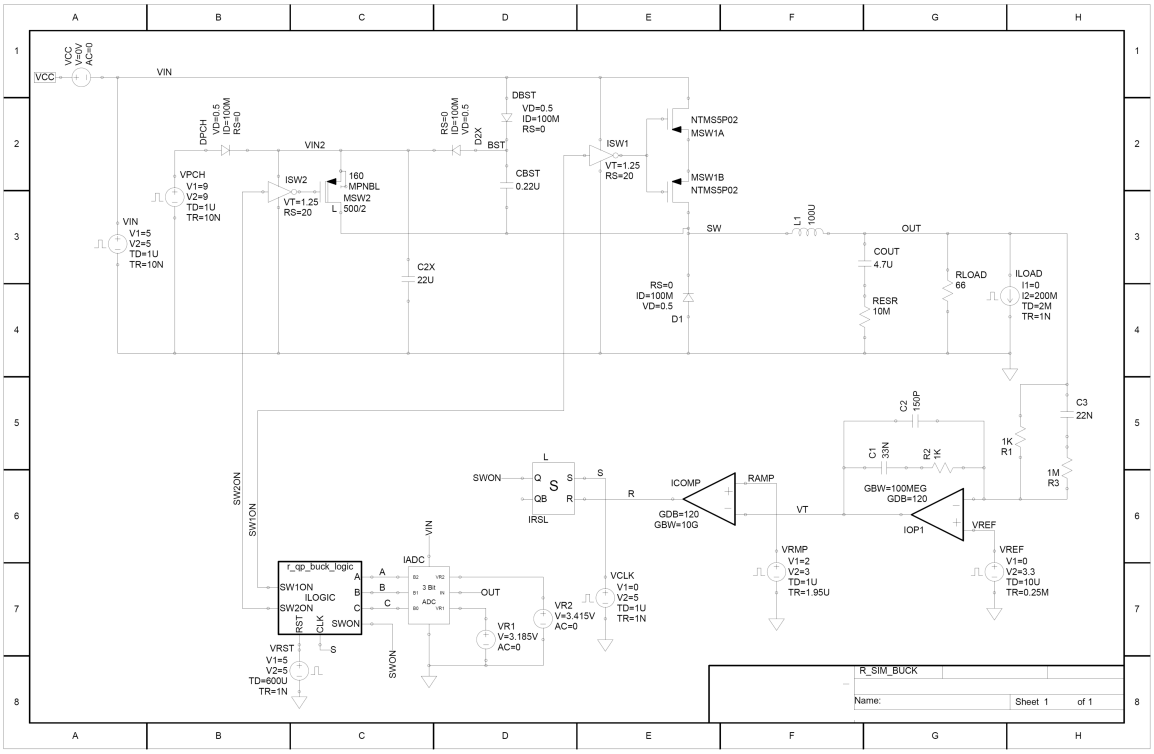


Figure 4.11: Analog System Model

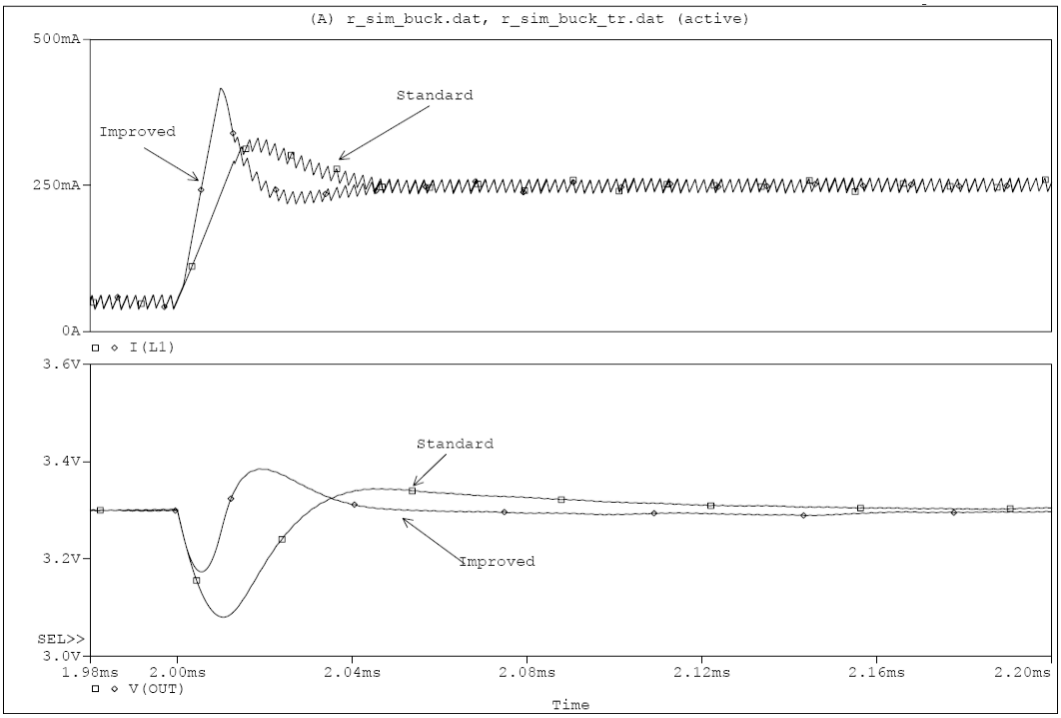


Figure 4.12: Analog System Transient Response

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

5.1 Summary

In this thesis, digital control was discussed at length along with its advantages and limitations. Various approaches to improving the dynamic response were presented. A novel approach that improves the dynamic response beyond what is found in the literature has been presented. Design and simulation results of a 1.8V, 15W, 1MHz digitally controlled step down converter with a 12mV Analog to Digital Converter (ADC) resolution and a 2ns DPWM (Digital Pulse Width Modulator) resolution are presented. The introduced approach was proven to reduce output undershoot as well as the settling time of the converter when compared to conventional PID control or time optimal control.

The proposed approach is not more complex than time optimal control from a control standpoint but there is extra complexity associated with the power stage. Even though it adds extra components, it has ben shown that the transient response improves by a factor greater than three. It is important to point out that this approach is not exclusive only to digital control but rather can also be implemented in analog control approaches.

5.2 Future Work

Implementing the proposed idea on an FPGA would further verify the results found in this thesis. To optimize the performance, an I.C. implantation would be even better than an FPGA implementation.

In a steady state condition, the duty cycle does not change from cycle to cycle. To improve the efficiency of the proposed solution, once the output reaches steady state, the duty cycle command can be stored in a register and the PID can be turned off as well as the ADC. Two comparators to sense when the output moves out of the “0” bin would have to remain active in order to re-enable the PID and the ADC when. This idea can easily be implemented on an FPGA together with the control approach presented in this thesis.

REFERENCES

[1] R. Erickson and D. Maksimovic, "Fundamentals of Power Electronics," Kluwer Academic Publishers, 2001.

[2] E. Meyer, Z. Zhang, Y. Liu "An Optimal Control Method for Buck Converters Using a Practical Capacitor Charge Balance Technique," *IEEE Transactions on Power Electronics, Vol. 23, No. 4, July 2008, pp. 1802-1812.*

[3] Z. Zhao, A. Prodic "Continuous-Time Digital Controller for High- Frequency DC-DC Converters," *IEEE Transactions on Power Electronics, Vol. 23, No. 2, March 2008, pp. 564-573*

[4] A. Corradini, R. Mattavelli, S. Saggini, "Time Optimal Parameters- insensitive Digital Controller for DC-DC Buck Converters," *IEEE PESC , June 2008*

[5] N. Andrews, "The Global Market for Power Supply and Power Management Integrated Circuits," *IEEE Applied Power Electronics Conference and Exposition, pp.126-131, Mar. 2002.*

[6] R. B. Ridley, "A new continuous-time model for current-mode control," *IEEE Trans. Power Electron., vol. 6, pp. 271–280, Apr. 1991.*

[7] Wei Tang, Fred C. Lee, and Raymond B. Ridley, Small Signal Modeling of Average Current Mode Control‖, *IEEE Trans. On Power Electronics, Vol. 8, No.2, April 1993, pp.112-119.*

[8] A. Prodic, D. Maksimovic, and R. Erickson, "Design and implementation of a digital PWM controller for a high-frequency switching DC–DC power converter," *Proc. IEEE IECON Conf., 2001, pp. 893–898.*

[9] T. W. Martin and S. S. Ang, "Digital control of switching converters," *Proc. IEEE Int. Symp. Ind Electron., vol. 2, 1995, pp. 480–484.*

[10] A. Peterchev, S. Sanders, "Quantization Resolution and Limit Cycling in Digitally Controlled PWM Converters," *IEEE Transactions on Power Electronics, Vol. 18, No. 1, January 2003, pp. 301-308.*

[11] B. Miao, R. Zane, D. Maksimovic, "Detection of Instability and Adaptive Compensation of Digitally Controlled Switched Mode Power Supplies," *in Proc. IEEE Appl. Power Electron. Conf. Expo., Austin, TX, Mar. 2005, vol. 2, pp. 63 – 691*.

[12] J. Morroni, R. Zane, D. Maksimovic, "Design and implementation of an adaptive tuning system based on desired phase margin for digitally controlled DC–DC

converters," *IEEE Trans. Power Electron. , vol. 24, no. 2, pp. 559 – 564, Feb 2009.*

[13] D. Maksimović, R. Zane, R. Erickson, "Impact of digital control in power electronics," *in Proc. IEEE Int. Symp. Power Semicond. Dev. ICs, Kitakyushu, Japan, May 2004, pp. 13 – 22.*

[14] S. Saggini, E. Orietti, P. Mattavelli, A. Pizzutelli, A. Bianco, "Fully-digital hysteretic voltage-mode control for dc-dc converters based on asynchronous sampling," *in Proceedings, IEEE APEC, pp. 503–509, Feb 2008.*

[15] H. Hu, D. Maksimovic, "Nonlinear Control for Improved Dynamic Response of Digitally-Controlled DC-DC Converters," *IEEE PESC,* June 2006

[16] H. Hu, V. Yousefzadeh and D. Maksimovic, "Non-uniform A/D quantization for improved dynamic responses of digitally controlled DC-DC converters," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 1998-2005, July 2008.

[17] Intel Corp. "Voltage Regulator-Down (VRD) 11.0, Processor Power Delivery Design Guidelines*," www.intel.com*, November 2011

APPENDIX A

MATLAB DESIGN CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Discrete-time PID compensator design for a buck comverter
%
%
% Last Modified Nov 2012
% Modified by: Ahmed Hashim
%
%
% PID compensator design: two zeros, a pole at zero and a
hf pole
%
% Code based on code from COPEC short course 2005
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define power stage parameters
Vg=3.3;   % input voltage
fs=1e6; % switching frequency
Vref = 1.8; % reference voltage
Hsense = 1; % output voltage sensing gain
L=4.7e-6; % filter inductance
RL=0.020; % series resistance
C=50e-6; % filter capacitance
Resr=0.010; % capacitor esr
Iload = 1;
Rload=Vref/Iload; % Compensator is designed for near-zero-
load case


% A/D, DPWM and coefficient quantization parameters
qad = 12e-3; % LSB of the A/D converter
ndpwm = 9; % number of DPWM bits
qdpwm = 1/2^ndpwm; %
ncoef = 16; % number of bits in a compensator coefficient
word

% td: total sampling, computing, and modulator delay, td =
td1 + DTs
td1 = 0.1e-6;  % td1: delay from the sample instant to the
rising edge of PWM

% Compensator parameters
fcplace = 1/25; % desired cross-over frequency relative to
sampling: fc / fs
```

```
beta = 1/20; % critical frequency relative to sampling:
fcrit / fs
a = 0.5; % desired z-domain hf pole, 1/(z-a), a <= 0;

% Continuous-time compensator templates; two options are
pre-configured:
% (1) two real zeros, relative to the converter filter
cuttof frequency, fo
% for this option, fill in the two placement options below:
z1place = 0.7; % placement parameter, zero 1, relative to
fo
z2place = 0.9; % placement parameter, zero 2, relative to
fo

% 2) pair of resonant zeros, with resonant frequency
relative to fo
% for this option fill in the zero frequency and Q-factor
placement below:
zplace = 1; % placement parameter for the compensator
double zero, relative to fo
qplace = 1; % placement parameter for the compensator Qcmp
relative to the converter Q

% Select the compensator template option with the following
flag
% compoption = 0 : two real zeros
% compoption = 1 : resonant zeros
compoption = 0;

% Call the compensator design file
buck_comp_design




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Discrete-time PID compensator design for a buck converter
%
%
% Last Modified Nov 2012
% Modified by: Ahmed Hashim
%
%
% PID compensator design: two zeros, a pole at zero and a
hf pole
```

```
%
% Code based on code from COPEC short course 2005
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Solve for ADC and DPWM gain
Ga2d = 1/qad;
Gdpwm = qdpwm;

% Solve key power stage characteristics
Qload=Rload*(C/L)^0.5;
Qloss=((L/C)^0.5)/(Resr+RL);
Q=Qload*Qloss/(Qload+Qloss);
wo=1/((L*C)^0.5);
fo = wo/(2*pi);
wesr=1/(C*Resr);
Ts = 1/fs;

% define frequency range of interest (note aliasing above
fs/2)
f=logspace(2,5.99,5000);
w=2*pi*f;

% td: total sampling, computing, and modulation delay
td2 = Ts*Vo/Vg; % td2: computed as DTs, D = steady-state
duty cycle
td = td1 + td2; % total delay

s = zpk('s'); % define s variable for cont time transfer
functions
z = zpk('z',Ts); % define z variable for disc time transfer
functions

% Control to output transfer function of the converter
(averaged model)
Gvd= Vg*(1+s/wesr)/(1+(s/(Q*wo)+(s/wo)^2));

% Control to output transfer function of the converter
(discrete-time model)
% Gvdz is the discrete-time control-to-output tf
% See the MATLAB function "dsmps_control-
to_output_discrete.m" for details
dsmps_control_to_output_discrete;

% Define uncompensated loop gain with total delay, td
Tu = Gvd*Hsense;
```

```
set(Tu,'ioDelay',td); % defines total delay, td

% Plot Tu
%figure(1)
[Tumag,Tuph] = bode(Tu,w);
Tuphu = unwrap(Tuph);
Tumagdb = 20*log10(Tumag);
%subplot(2,1,1)
%semilogx(f,Tumagdb(:,:))
%grid
%ylabel('magnitude [db]')
%axis([200 1e6 -100 50]);
%title('Uncompensated loop gain, Tu = Gvd*Hsense*exp(-
s*td)')
%subplot(2,1,2)
%semilogx(f,Tuphu(:,:))
%axis([200 1e6 -270 20]);
%xlabel('frequency [Hz]')
%ylabel('phase [deg]')
%grid

% Anti-aliasing filter
Gaa = 1/(s/(2*pi*faa) + 1);

% Compensator parameters
fcgoal = fcplace*fs;
wcgoal = 2*pi*fcgoal;

% below, derive alpha to achieve desired z-domain pole "a"
and match
% continuous time mag & phase to discrete time at the
critical freq set by beta
alpha = (beta/tan(pi*beta))*(1-a)/(1+a);
fhf = alpha*fs;
whf = 2*pi*fhf;
fcrit = beta*fs;
wcrit = 2*pi*fcrit;

% compensator cont time templates (real or resonant zero
options)
% Pair of real zeros parameters
wz1 = 2*pi*z1place*fo;
wz2 = 2*pi*z2place*fo;
% Resonant zero parameters
wz = 2*pi*zplace*fo; % define double zero frequency, fz
Qcmp = Q*qplace; % define compensator Q-factor
% Compensator selection, 0: real zeros, 1: resonant pair
```

```
if (compoption == 0)
     Gczeros = (1+s/wz1)*(1+s/wz2); % real zeros for
compensator
else Gczeros = (1+s/(Qcmp*wz)+(s/wz)^2); % resonant zeros
for compensator
end

% solve comp gain to set the desired cross-over frequency
T1 = (1/s)*Gczeros*(1/(1+s/whf))*Tu*Ga2d*Gdpwm; % loop gain
with integral gain of 1
[T1mag, T1ph] = bode(T1,wcgoal); % compute magnitude
response at fc
wk = 1/T1mag;

% define complete cont-time compensator template and
resulting loop gain
Gct = (wk/s)*Gczeros*(1/(1+s/whf));
Tt = Gct*Tu*Ga2d*Gdpwm*Gaa; % loop gain with the template
compensator

% solve & print template loop gain crossover freq and
margins
disp(' ')
disp('Loop gain parameters in template design:');
[Gmt, Pmt, wcgt, wcpt] = margin(Tt);
disp(sprintf('Cross-over Frequency [Hz]: %g',
wcpt/(2*pi)));
disp(sprintf('Phase Margin [deg]: %g', Pmt));
disp(sprintf('Gain Margin [dB]: %g', 20*log10(Gmt)));
disp(sprintf('-180 degrees frequency [Hz]: %g',
wcgt/(2*pi)));

% Design compensator using BLT with prewarp to critical
frequency
Gcd = c2d(Gct,1/fs,'prewarp',wcrit);

% define loop gain in discrete compensator without rounding
Gcdfreq = freqresp(Gcd,w);
Gcdfrd = frd(Gcdfreq,w);
Td = Tu*Gcdfrd*Ga2d*Gdpwm*Gaa;

% solve & print loop gain crossover freq and margins in
discrete design
% without rounding
disp(' ')
disp('Loop gain parameters in discrete design without
rounding:');
```

```
[Gmd, Pmd, wcgd, wcpd] = margin(Td);
disp(sprintf('Cross-over Frequency [Hz]: %g',
wcpd/(2*pi)));
disp(sprintf('Phase Margin [deg]: %g', Pmd));
disp(sprintf('Gain Margin [dB]: %g', 20*log10(Gmd)));
disp(sprintf('-180 degrees frequency [Hz]: %g',
wcgd/(2*pi)));

%Display discrete comp design without rounding
%disp('Discrete compensator w/out rounding')
%Gcd

% Coeeficient selection, 0: Script, 1: User input
if (customcoef == 0)
    %Find comp coefficients
    [Gcdnum, Gcdden] = tfdata(Gcd,'v');
    b1 = Gcdnum(1);
    b2 = Gcdnum(2);
    b3 = Gcdnum(3); % numerator coefficients
    a1 = Gcdden(1); a2 = Gcdden(2); a3 = Gcdden(3); %
denominator coefficients
else
    Gcdnum = [Acoef Bcoef Ccoef]; Gcdden = [1 Dcoef Ecoef];
    b1 = Acoef; b2 = Bcoef; b3 = Ccoef; % numerator
coefficients
    a1 = 1; a2 = Dcoef; a3 = Ecoef; % denominator
coefficients
end

% Convert to parallel form
[R,P,K] = residue(Gcdnum, Gcdden);

%Perform quantization on coefficients
%Round the coefficients into ncoef-bit binary words
R1bin = dec2bin(round(abs(R(1))*qad*2^ncoef));
R2bin = dec2bin(round(abs(R(2))*qad*2^ncoef));
Pbin = dec2bin(round(abs(P(2))*qad*2^ncoef));
Kbin = dec2bin(round(abs(K)*qad*2^ncoef));

%Quantized (rounded) parameters of the parallel form of the
compensator
%These parameters are used in the buck_discrete_design
Simulink model
R1rnd = sign(R(1))*bin2dec(R1bin)/((2^ncoef)*qad);
R2rnd = sign(R(2))*bin2dec(R2bin)/((2^ncoef)*qad);
Prnd = sign(P(2))*bin2dec(Pbin)/((2^ncoef)*qad);
Krnd = sign(K)*bin2dec(Kbin)/((2^ncoef)*qad);
```

```
%Perform quantization on expanded coefficients
%Round the coefficients into ncoef-bit binary words
%Gcdnumtemp = Gcdnum/Gcdden(1);
%Gcdnum1bin =
dec2bin(round(abs(Gcdnumtemp(1))*qad*2^ncoef));
%Gcdnum2bin =
dec2bin(round(abs(Gcdnumtemp(2))*qad*2^ncoef));
%Gcdnum3bin =
dec2bin(round(abs(Gcdnumtemp(3))*qad*2^ncoef));

%Gcddentemp = Gcdden/Gcdden(1);
%Gcdden1bin =
dec2bin(round(abs(Gcddentemp(1))*qad*2^ncoef));
%Gcdden2bin =
dec2bin(round(abs(Gcddentemp(2))*qad*2^ncoef));
%Gcdden3bin =
dec2bin(round(abs(Gcddentemp(3))*qad*2^ncoef));

%Quantized (rounded) parameters of the expanded form of the
compensator
%Gcdnum1rnd = round(Gcdnum(1));
%sign(Gcdnumtemp(1))*bin2dec(Gcdnum1bin)/((2^ncoef)*qad)
%Gcdnum2rnd = round(Gcdnum(2));
%sign(Gcdnumtemp(2))*bin2dec(Gcdnum2bin)/((2^ncoef)*qad)
%Gcdnum3rnd = round(Gcdnum(3));
%sign(Gcdnumtemp(3))*bin2dec(Gcdnum3bin)/((2^ncoef)*qad)

%Gcdden1rnd = round(Gcdden(1));
%sign(Gcddentemp(1))*bin2dec(Gcdden1bin)/((2^ncoef)*qad)
%Gcdden2rnd = round(Gcdden(2));
%sign(Gcddentemp(2))*bin2dec(Gcdden2bin)/((2^ncoef)*qad)
%Gcdden3rnd = round(Gcdden(3));
%sign(Gcddentemp(3))*bin2dec(Gcdden3bin)/((2^ncoef)*qad)

scale = 1; %2^ncoef/abs(Gcdnum(2));

Gcdnum = round(scale*Gcdnum);
Gcdden = round(Gcdden);


Gcdnum1rnd = Gcdnum(1)/scale;
Gcdnum2rnd = Gcdnum(2)/scale;
Gcdnum3rnd = Gcdnum(3)/scale;

Gcdden1rnd = Gcdden(1);
Gcdden2rnd = Gcdden(2);
```

```
Gcdden3rnd = Gcdden(3);


if (customcoef == 0)
    %Find compensator transfer function with rounding
    %Gcdr = tf(Krnd + R1rnd/(z-1) + R2rnd/(z-Prnd));
    Gcdr = (Gcdnum1rnd*z^2 + Gcdnum2rnd*z +
Gcdnum3rnd)/(Gcdden1rnd*z^2 + Gcdden2rnd*z + Gcdden3rnd);
else
    Gcdr = (b1*z^2 + b2*z + b3)/(a1*z^2 + a2*z + a3);
end


%Define loop gain in discrete compensator WITH rounding
Gcdfreqr = freqresp(Gcdr,w);
Gcdfrdr = frd(Gcdfreqr,w);
Tdr = Tu*Gcdfrdr*Ga2d*Gdpwm*Gaa;

% solve & print loop gain crossover freq and margins in
discrete design
% WITH rounding
disp(' ')
disp('Loop gain parameters in discrete design with
rounding:');
[Gmdr, Pmdr, wcgdr, wcpdr] = margin(Tdr);
disp(sprintf('Cross-over Frequency [Hz]: %g',
wcpdr/(2*pi)));
disp(sprintf('Phase Margin [deg]: %g', Pmdr));
disp(sprintf('Gain Margin [dB]: %g', 20*log10(Gmdr)));
disp(sprintf('-180 degrees frequency [Hz]: %g',
wcgdr/(2*pi)));

% check no-limit-cycle conditions
disp(' ')
disp('Check no-limit cycle conditions (A1, A2, B2 should be
< 1)')
% Check A1
disp(sprintf('Check A1: 2*Vg*qdpwm*Hsense/qad equals: %g',
2*Vg*qdpwm*Hsense/qad))
if (2*Vg*qdpwm*Hsense/qad < 1)
    disp('Rule A1 OK')
else disp('Failed A1 (>1)')
end
% Check A2
disp(sprintf('Check A2: 2*Vg*Hsense*Integral_gain equals:
%g', Vg*Hsense*R1rnd*2))
if (2*Vg*R1rnd*Hsense < 1)
```

```
    disp('Rule A2 OK')
else disp('Failed A2 (>1)')
end
% Check B1
[Gvdmx, x] = bode(Gvd,wcgdr);
disp(sprintf('Check B1: 8*|Gvd(wcg)|*qdpwm*Hsense/(qad*pi)
equals: %g', 8*Gvdmx*qdpwm*Hsense/(qad*pi)))
if ((8*Gvdmx*qdpwm*Hsense)/(qad*pi) < 1)
    disp('Rule B1 OK')
else disp('Failed B1 (>1)')
end
% Check B2
disp('Check B2 (GM to be > 10.2)')
if (20*log10(Gmdr) > 10.2)
    disp('Rule B2 OK')
else disp('Failed B2: GM < 10.2 dB')
end

% Print discrete compensator design
disp(' ')
disp('Discrete compensator with rounding')
Gcdr
disp(' ')
%disp('Parallel Form Coefficients:')
%fprintf('\nK = %d\nR1 = %d\nR2 = %d\nP =
%d\n\n\n',Krnd,R1rnd,R2rnd,Prnd);
disp('Expanded Form Coefficients:')
fprintf('\nA = %d\nB = %d\nC = %d\nD = %d\nE =
%d\n\n',scale*Gcdnum1rnd,scale*Gcdnum2rnd,scale*Gcdnum3rnd,
Gcdden2rnd,Gcdden3rnd);
fprintf('\nScale = %0.2f\n\n\n\n',scale);
% Plot loop gain for template & discrete design w/ & w/out
rounding
figure(2)
[Ttmag,Ttph] = bode(Tt,w);
[Tdmag,Tdph] = bode(Td,w);
[Tdrmag,Tdrph] = bode(Tdr,w);
Ttphu = unwrap(Ttph);
Ttmagdb = 20*log10(Ttmag);
Tdphu = unwrap(Tdph);
Tdmagdb = 20*log10(Tdmag);
Tdrphu = unwrap(Tdrph);
Tdrmagdb = 20*log10(Tdrmag);
subplot(2,1,1)
semilogx(f,Ttmagdb(:,:),'k',f,Tdmagdb(:,:),'b',f,Tdrmagdb(:
,:),'r')
text(300,-10,strcat('Rounded System Bandwidth:
```

```
',num2str(round(wcpdr/(2*pi))/1000), ' kHz'), 'FontSize',
12, 'background','w')
grid
title('Template (black), Discrete (blue) and Discrete
Rounded (red) Loop Gain')
ylabel('magnitude [db]')
axis([200 1e6 -50 80]);
subplot(2,1,2)
semilogx(f,Ttphu(:,:),'k',f,Tdphu(:,:),'b',f,Tdrphu(:,:),'r
')
axis([200 1e6 -270 20]);
xlabel('frequency [Hz]')
ylabel('phase [deg]')
text(300,-175,strcat('Rounded System PM:
',num2str(round(Pmdr)), '\circ'), 'FontSize', 12,
'background','w')
grid


% This script  calculates the load transient response
% for a Digital Control Buck regulator that is designed
% in dsmps_buck_setup.m


dsmps_buck_setup

Istep = 2;   % Load current step
% Loop gain
Tol = Gvd*Hsense*Gct*Ga2d*Gdpwm*Gaa;

%Closed loop responce
Tcl = Tol/(1+Tol)

% Calculating open loop output Z
Zlnum = [L RL];
Zlden = [1];
Zl = tf(Zlnum,Zlden); % inductor z
Zcnum = [Resr*C 1];
Zcden = [C 0];
Zc = tf(Zcnum,Zcden); % capacitor z
Zlc = Zl*Zc/(Zl+Zc);

Zol = Zlc*Rload/(Zlc+Rload); % open loop output impedance
Zocl = Zol/(1+Tol); % closed loop output impedance

%figure(2)
%bodemag(Zol,Ts,Zocl)
```

```
%grid on;
figure(3)
step(-Istep*Zocl,1e-3)
grid on
```