FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

The Impact of Timer Resolution in the Efficiency Optimization of Synchronous Buck Converters

Pedro Amaral

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Supervisor: Cândido Duarte (FEUP)

Co-Supervisor: Pedro Costa (Infineon Technologies AG)

June 29, 2015

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES | **2014/2015**

A Dissertação intitulada

"The Impact of Timer Resolution in the Efficiency Optimization of Synchronous Buck Converters"

foi aprovada em provas realizadas em 24-07-2015

o júri

Presidente Professor Doutor Armando Luís Sousa Araújo
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Júlio Manuel Sousa Barreiros Martins
Professor Associado do Departamento de Eletrónica Industrial da Escola de Engenharia da Universidade do Minho

Professor Doutor Manuel Cândido Duarte dos Santos
Professor Auxiliar Convidado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - José Pedro Dias Amaral

Faculdade de Engenharia da Universidade do Porto

Abstract

Power efficiency is now, more than ever, a major topic for systems, households, providers and governments. The amount of electronic devices has increased exponentially in the last decade, and the growth rhythm is not estimated to slow down. Due to an imminent energy crisis and the inability of power suppliers to easily step up their production, the stress of reducing consumption, increasing efficiency and minimizing the power stage footprints is passed through to the system providers.

The switching frequency of power converters is driving the footprint reduction by enabling smaller components (e.g. coils, capacitors, etc), but at the same time imposing another level of optimization in some variables of a power converter. One of the variables that most impacts the efficiency of the converter is linked to the non-optimization of the dead time in synchronous switching. Simultaneously, the market of non-digital power supplies is steadily losing ground to the digitalized counterparts. This creates new possibilities to tackle old issues with unprecedented approaches and create business and innovation opportunities.

This dissertation aims at analyzing the influence of one of the most important characteristics of a digital controller, the timer resolution, in the issue of dead time optimization for synchronous buck converters. An algorithm has been designed and implemented using state-of-the-art hardware from Infineon Technologies to prove the concept and demonstrate the results.

Resumo

A eficiência energética é hoje, mais do que nunca, um tema preponderante em sistemas, lares, produtores e administração. A quantidade de aparelhos electrónicos tem crescido exponencialmente na última década e o ritmo não dá sinais de abrandamento. Devido à iminente crise energética e à incapacidade dos fornecedores de energia em aumentar rapidamente a sua produção, a pressão para reduzir o consumo, aumentar a eficiência e minimizar o tamanho e custo de andares de potência é transmitida para o lado dos fornecedores de sistemas.

A frequência de comutação em conversores de potência tem liderado a redução do seu custo pela diminuição do tamanho dos componentes (como bobinas e condensadores), mas simultaneamente tem imposto um outro nível de optimização das variáveis. Uma dessas variáveis que mais influencia a eficiência do conversor está ligada à não optimização do tempo morto em conversores síncronos. Simultaneamente, o mercado das fontes de potência não-digitais tem vindo a perder terreno para os equivalentes digitais. Isto cria novas possibilidades para abordar problemas antigos com recursos nunca utilizados e assim criar oportunidades de inovação e negócio.

Esta dissertação tem como objectivo analizar a influência de uma das características mais importantes em controladores digitais, a resolução do *timer*, na questão da optimização do tempo morto em conversores abaixadores síncronos. Será desenhado e implementado um algoritmo em tecnologia moderna da Infineon Technologies de forma a provar o conceito e demonstrar os resultados.

Acknowledgments

I would like to start by thanking my supervisors, Cândido Duarte, for his constant support and dedication, scientific experience and helpful advices, and Pedro Costa, for his vast industrial background, the idea of this dissertation and his valuable guidance. Without them, this work would have never taken off.

Furthermore, I would like to acknowledge the engineers of Infineon Technologies AG, who have directly or indirectly contributed to this dissertation, and the technical staff of the Department of Electrical Engineering of this Faculty, for promptly fulfilling my requests.

Moreover, I would like to express my gratitude to my family for supporting me, not only through the duration of this dissertation, but also during the 5 years I had in this university.

I would also like to thank my friends, not only for the valuable technical insights, but for the support, friendship and companionship. If anything, we all had a good laugh over this whole dissertation deal.

Finally, I would like to thank my girlfriend Jenny, for being the wideband low-pass filter to my high switching frequency mood oscillator. Her support and encouragement was invaluable. Thank you.

Pedro Amaral

"There is only one way to avoid criticism:
do nothing, say nothing, and be nothing."

Aristotle

Contents

Abstract iii

Resumo v

1 Introduction 1
- 1.1 Objectives of this Work . . . 5
- 1.2 Organization of the Dissertation . . . 6

2 Literature Review 7
- 2.1 Fixed dead times . . . 7
- 2.2 Switching current sensing . . . 8
- 2.3 Switching voltage sensing . . . 8
  - 2.3.1 Adaptive control . . . 9
  - 2.3.2 Predictive control . . . 10
- 2.4 Sensorless methods . . . 11
- 2.5 Summary and discussion . . . 14

3 Analysis of Dead Time Optimization 17
- 3.1 Modeling of the Algorithm . . . 17
- 3.2 Operation Limits . . . 20
- 3.3 Resource Usage . . . 22
- 3.4 Power Efficiency Improvement . . . 23
- 3.5 Summary . . . 26

4 Experimental Results 29
- 4.1 Powertrain and Microcontroller Unit . . . 29
- 4.2 Software and Algorithm . . . 32
- 4.3 Measurements and Results . . . 36
- 4.4 Simulations . . . 41

5 Conclusions and Future Work 43
- 5.1 Fulfillment of the Objectives . . . 43
- 5.2 Future Work . . . 44

A Calculation of Real Dead Time 47

B Results of the Simulation 49

List of Figures

1.1 Synchronous buck converter. 2
1.2 Typical waveforms of a buck converter. 3
1.3 Rising edge and falling edge dead times. 4
1.4 The current flow through the transistors and the body diode. 4
1.5 The steady growth on the market of digitalized power supplies. 5

2.1 Fixed dead time approach 8
2.2 Switching current sensing method 9
2.3 Adaptive optimization methods 9
2.4 Non-optimal and optimal switching waveforms 11
2.5 Maximum Efficiency Point Tracking algorithm 12
2.6 Accumulator and input current method for sensorless optimization 13
2.7 Sensorless dead time optimization using duty cycle minimization 13

3.1 Switching node voltage v_{sw}. 18
3.2 The four sequential stages of the dead time optimization algorithm. 19
3.3 Representation in $\mathbb{R}^3$ of the set of possible operation points $\mathbf{r}$ 20
3.4 Example of a set of possible operation points defined by the resources of the controller. 21
3.5 Dependency of φ on N_{timer} and N_{ADC}. 23
3.6 Depictions of $\Psi(\gamma)$. 25
3.7 Dependency of the efficiency improvement factor Ψ with ADC and timer resolution. 26

4.1 The proof-of-concept prototype: buck converter kit and XMC4200 control card. . 30
4.2 A view of the microcontroller peripherals required by the project. 30
4.3 High-side current sensing for calculation of the power efficiency. 31
4.4 The buck converter and its control law, realized on a XMC4200 microcontroller. . 32
4.5 Generation of an High Resolution PWM waveform. 33
4.6 The illustrative relation between dead time and duty cycle. 34
4.7 Flowchart of the sensorless dead optimization algorithm. 35
4.8 Screenshot of the xSPY interface designed to control and debug the algorithm. . . 37
4.9 Duty cycle variation after manually triggering the algorithm. Running time is 80 ms. 37
4.10 Duty cycle response to changes in rising edge dead time followed by falling edge dead time. 38
4.11 Buck converter control signals measured with an oscilloscope. 40
4.12 Relation between temperatures of control and synchronous MOSFETs and dead time. 41
4.13 Simulation model used in Simulink. 42
4.14 Simulation results for different N_{ADC} curves. 42

A.1 Representation of the delay between the signal out of the microcontroller pins and the switching node voltage. 47
A.2 Oscilloscope print of the difference between defined and real dead time for both edges. The two signals are measured at the output pins of the microcontroller while the bottom one is measured at the swithing node voltage. 48

B.1 Results of the simulation. 50

List of Tables

2.1 Summary of proposed dead time optimization methods and their claimed results. 15

4.1 Operation parameters for the prototype buck converter. 29
4.2 Experimental results using both low and high resolution PWM. 39

A.1 Time parameters that affect the signal. 48

Abbreviations

AC	Alternate Current
ADC	Analog to Digital Converter
CCM	Continuous Conduction Mode
CCU	Capture and Compare Unit
CS	Control Switch
DC	Direct Current
DCM	Descontinuous Conduction Mode
DPWM	Digital Pulse Width Modulation
DSP	Digital Signal Processor
DTLL	Dead Time Locked-Loop
FPGA	Field Programmable Gate Array
FS	Full Scale
GPIO	General Purpose Input Output
HRPWM	High Resolution Pulse Width Modulation
IC	Integrated Circuit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MEPT	Maximum Efficiency Point Tracking
MPPT	Maximum Power Point Tracking
NVIC	Nested Vector Interrupt Controller
PI	Proportial Integral
PWM	Pulse-Width Modulation
SMPS	Switched Mode Power Supply
SS	Synchronous Switch
VADC	Versatile Analog to Digital Converter
XMC	Cross Market Microcontroller

Chapter 1

Introduction

Since the dawn of electrical engineering, power conversion has been one of the most important disciplines in the area. From the 19th century until the present day, engineers have come up with an incredible variety of electric energy sources and applications. However, these two are not always related, neither in geographic nor in operational terms. The conversion from and into different voltage, current and frequency levels was and still is pivotal to deliver electricity to billions of people, propel the development of new technologies and fuel the modern machinery that supports our society.

But we live in a different time. We are able to transport energy for very long distances and convert it in many ways, but the global energetic and environmental situation pushes us one step forward: we must do it efficiently. This means that the scientific community must pose questions and rethink even the most established design principles in electrical engineering.

Power converters can be divided into four different classes:

1. AC/DC (rectifiers);
2. DC/AC (inverters);
3. DC/DC;
4. AC/AC (transformers).

DC/DC converters are the type of circuit used to convert a direct current supply from one voltage level to another. One family of DC/DC converters are the linear voltage regulators. These circuits act like variable resistors that continuously adjust a voltage divider in order to supply a steady average output voltage. However, they dissipate the extra energy as heat. Switched-mode converters (commonly called DC/DC converters due to their popularity) on the other hand, use a storage element (capacitor or inductor) that is charged and discharged continuously by driving switching elements. Switched-mode converters are much more efficient than linear regulators (theorically, up to 100 %) and have become extremely popular with the declining prices of power switches.

Switched-mode DC/DC converters are available in many topologies, namely buck, boost, buck-boost, Cuk, full-bridge, split-pi, SEPIC, zeta, etc. In this dissertation, the focus will be on

buck converters, as they find their way in a wide variety of applications: residential, commercial, industrial, aerospace, transportation, utilities and telecommunications [1].

The original buck converters had only a control MOSFET (CS in Figure 1.1) and a freewheel diode. This diode is called freewheel because the operation of the converter is similar to the spinning of a wheel to which small impulses are applied, in order to keep the rotation at a steady average speed. With modern low voltage applications (for example, 3.3 V_{CC} electronics), a modification to the original design imposed: the freewheel diode was replaced by the so called synchronous MOSFET (SS in Figure 1.1) due to the fact that the losses in a conducting diode are well above those in a MOSFET [2].

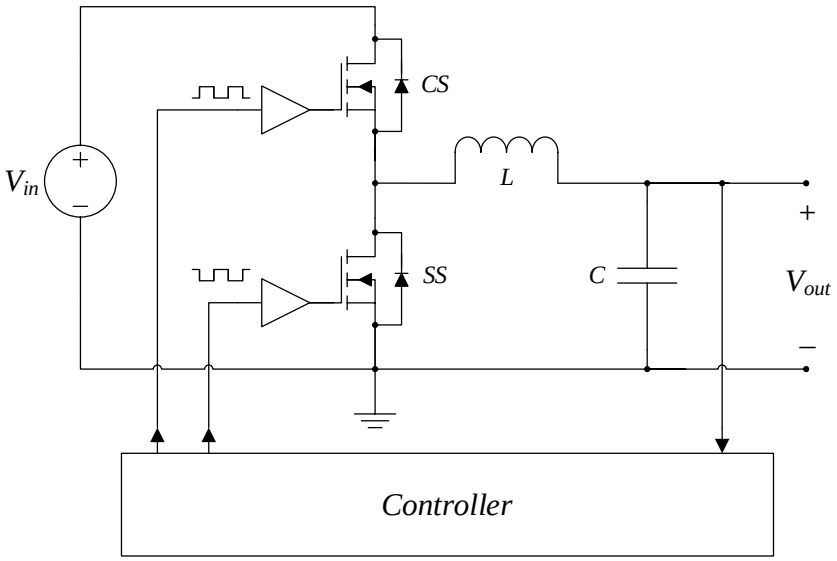


Figure 1.1: Synchronous buck converter.

The MOSFET gates of a synchronous buck converter are driven by PWM signals in a complimentary fashion: when one switch is on, the other is off. A controller measures the output voltage or current, in order to apply corrections to the switch driving and maintain the desired output voltage at different loads. The percentage of a switching cycle in which the control switch is at the on-state is called duty cycle D. The output voltage can be calculated in terms of the duty cycle as in Equation 1.1.

$$V_{out} = \frac{t_{on}}{T_s} V_{in} = D\, V_{in} \tag{1.1}$$

When the control switch is on, a voltage V_{in}–V_{out} appears across the inductor L, which charges with a linear current i_L. The synchronous MOSFET does not conduct any current because it is off and the body diode is reverse biased. When the synchronous switch is turned on (and the control switch is off), the voltage across the inductor is reversed to 0–V_{out} and the current decreases

linearly as the inductor discharges. This current also flows through the synchronous MOSFET. The capacitor C acts as a filter in order to keep V_{out} within an acceptable ripple (Figure 1.2).

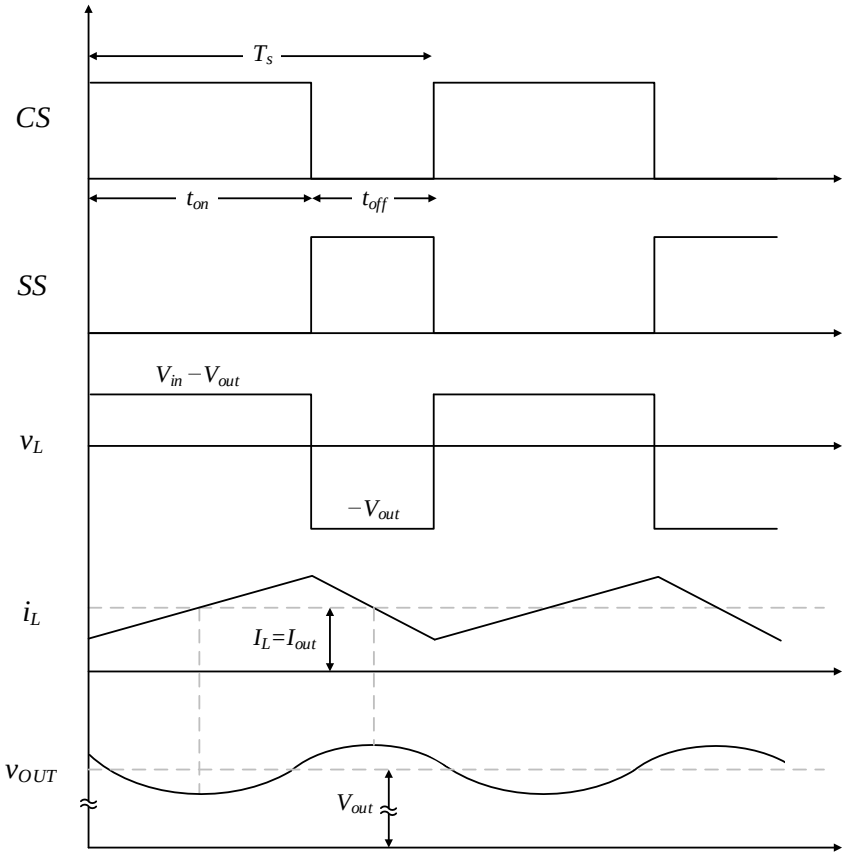


Figure 1.2: Typical waveforms of a buck converter.

For medium to heavy loads (I_{out}), the inductor current never reaches zero, which means the converter is operating on Continuous Conduction Mode (CCM) as seen in Figure 1.2. If the load is light, the inductor current reaches zero and the converter enters Discontinuous Conduction Mode (DCM). In DCM, the switches can be operated normally, in fixed switching frequency (as in CCM), or in a variable frequency mode, known as pulse-skipping. Pulse-skipping increases the efficiency at light loads by reducing the switching losses [3].

In practice, the turn-on of a MOSFET is not instantaneous, which means that a dead time must be added to the PWM waveform in order to prevent a short-circuit between supply and ground (see Figure 1.3). This would cause a high shoot-through current to flow through the transistors, increasing losses and possibly destroying both MOSFETs. Dead time is also called "blanking time" or simply "delay".

In unoptimized systems, this dead time is safely set to the worst-case scenario: a time interval large enough to account for the highest possible load and temperature conditions. However, when both MOSFETs are off, even though the synchronous switch is not conducting, the body diode is

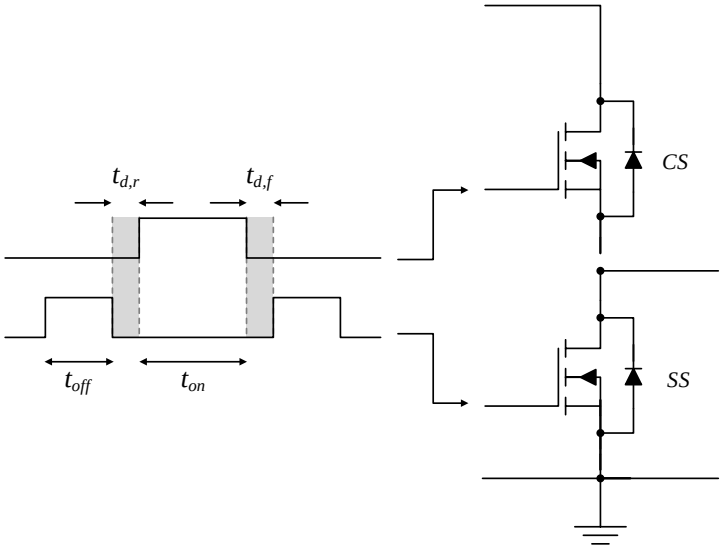


Figure 1.3: Rising edge and falling edge dead times.

forward biased and current is flowing through it, as shown in Figure 1.4. As previously mentioned, a conducting diode has much higher losses than a conducting MOSFET, which means that the existence of dead time decreases the general efficiency of a converter.

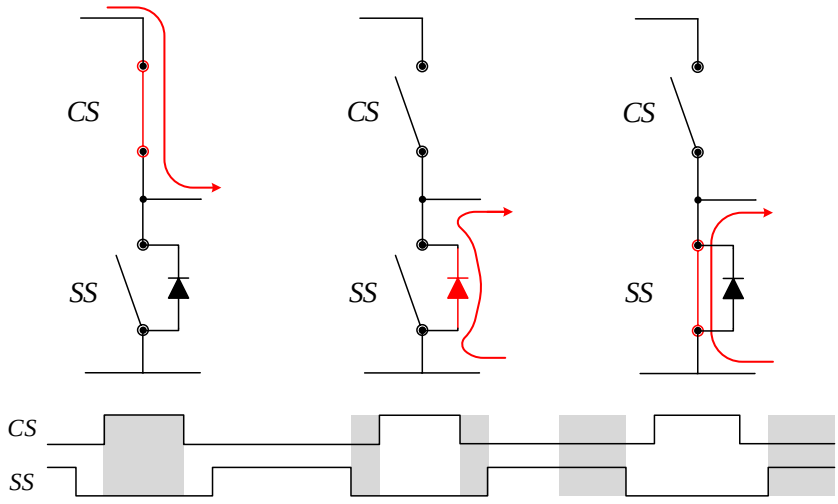


Figure 1.4: The current flow through the transistors and the body diode.

Fortunately, it is not necessary to fix the dead times for the entire operation of the converter. Several dead time optimization methods have been proposed to dynamically adjust both rising and falling edge dead times and, consequently, improve the converter power efficiency. This dissertation proposes an approach to this issue.

1.1 Objectives of this Work

In the last decade, the world market for digitalized power converters has been growing steadily. Digital control introduces features that are not possible in non-digital power supplies, such as communication and diagnostic capabilities. Efficiency can be also addressed with digital control, for example, in the ability to enter different operation modes for particular loads. The decline of microprocessor and microcontroller costs is making digitalized power converters more attractive over the years and the tendency is not stopping, as seen in Figure 1.5. Top technology companies such as Infineon, Microchip, TI and Freescale are investing in these products and widening their portfolios to meet the most demanding customer requirements.

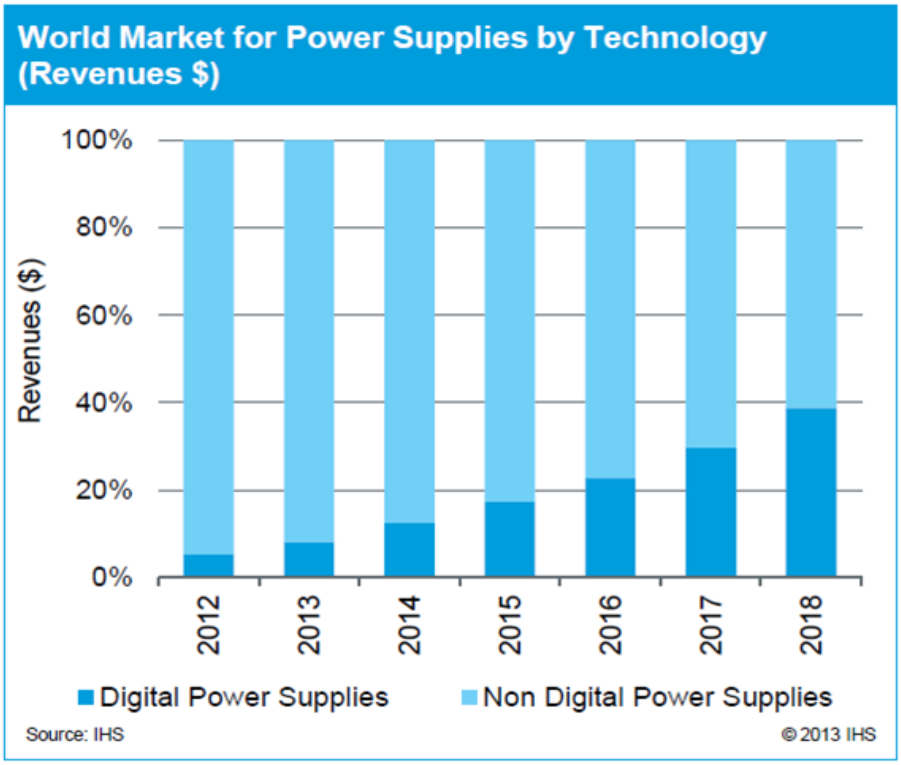


Figure 1.5: The steady growth on the market of digitalized power supplies.

That said, it is scientifically and industrially relevant to address the topic of dead time optimization in light of the use of digital control, particularly microcontrollers.

This dissertation has the following objectives:

- Investigate the relation of an important feature of any digital controller, the timer resolution, with the power efficiency improvement that dead time optimization can offer;
- Design and test an optimization algorithm in a proof-of-concept experimental setup;
- Achieve an algorithm that is simple, computationally inexpensive, effective and cheap to implement, seamlessly taking advantage of the digital controller to resolve the issue of dead time induced power losses.

The new Infineon Technologies Buck Kit will be used as powertrain for the prototype. The control will be performed on a XMC4200 microcontroller and will take advantage of the innovative High-Resolution PWM module (HRPWM). The XMC family is the world's first ARM Cortex-based microcontroller to offer a DPWM resolution of only 150 ps [4].

1.2 Organization of the Dissertation

After this introduction, a literature review on the topic of dead time optimization will be made in Chapter 2. Different approaches to the theme applied in buck converters will be explored and compared to each other.

In Chapter 3, a theoretical analysis of the problem will be done: the buck converter will be analyzed and the operation of an hypothetical algorithm will be related to the characteristics of a generic digital controller. The impact of said characteristics on power efficiency will also be deducted.

Chapter 4 presents a concrete design and implementation of an optimization algorithm. The hardware and software are exposed and discussed. The results are to be presented and related to ones expected from the previous chapter.

The document will end with the conclusions, a balance of the fulfillment of the objectives and suggestions for future work.

Chapter 2

Literature Review

The reduction of power losses by optimization of dead times has been the goal of many academic and industrial publications. There are several proposed methods using different technologies which can be classified as follows:

1. Fixed dead times;
2. Based on the sensing of switching currents;
3. Based on the sensing of switching voltages:
   (a) Adaptive;
   (b) Predictive;
4. Sensorless.

In this chapter, a literature review of the mentioned optimization methods will be done, along with explanations to the peculiarities of certain approaches, illustrated examples and an analysis of advantages and disadvantages to each method.

2.1 Fixed dead times

The fixed dead time control (sometimes referred to as the first generation) is the simplest method but yields the worst performance. It simply consists of choosing a rising edge and falling edge dead time period that stays constant throughout the entire operation. The chosen $t_{d,r}$ and $t_{d,f}$ must be long enough to account for the worst case scenario in terms of process, temperature and load variations, which leads inevitably to a reduced power efficiency, mostly due to unnecessary body diode conduction. This method can be implemented with a cross-coupled latch with delay as explained by Ramachandran in [5] and shown in Figure 2.1. Furthermore, most microcontroller timers and PWM generation ICs also include complementary operation modes with fixed dead time insertion [6].

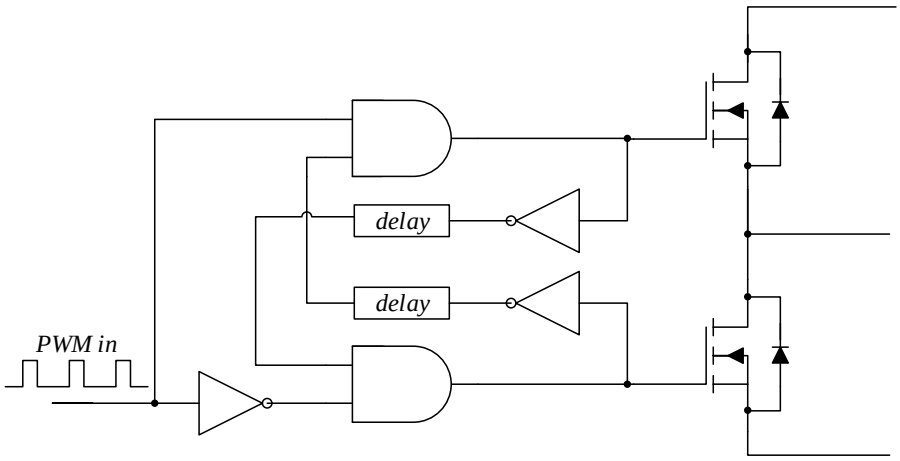


Figure 2.1: Fixed dead time approach [5].

2.2 Switching current sensing

The methods based on switching current sensing are based on the idea of preventing overlapping gate driving commands by measuring the current flowing through the synchronous MOSFET and detecting shoot-through currents (a basic set-up can be seen in Figure 2.2). When this high current is detected, the dead times should be increased to prevent overlapping. Otherwise, they should be decreased [7]. In Huang *et al.* [8] a more advanced method is proposed. By monitoring the switching current, the authors are able to estimate the load and quickly adjust the dead times accordingly, using a digitally implemented tri-mode controller. Despite the power efficiency improvement (up to 5 % in [8]), the fact that these methods depend on high-bandwidth current sensors (such as the "SENSEFET" technology, proposed in [9] and applied in [8]) makes their implementation difficult and sometimes impracticable.

2.3 Switching voltage sensing

The most common dead time control methods in the scientific literature are those based on switching voltage measurement. They are normally sub-divided into the adaptive and predictive type. The adaptive control (also known as second generation methods) consists of measuring the switching voltage across the synchronous rectifier and adjusting the dead times accordingly and on-the-fly. The predictive control (also called third generation methods) differs from the adaptive counterpart by using the information from the previous switching cycle to set the best possible dead time in the current cycle. In the following subsections, some examples of both technologies will be presented and discussed.

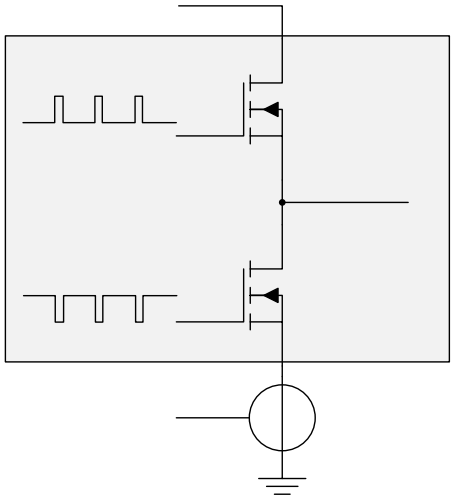
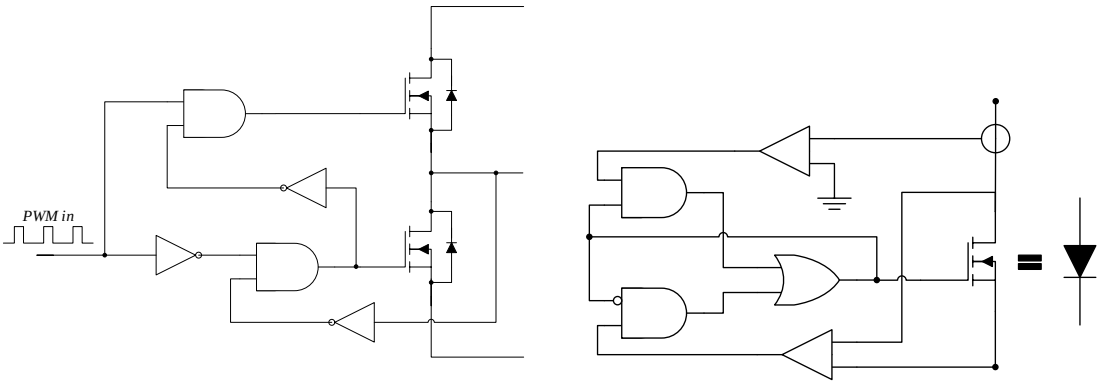


Figure 2.2: Switching current sensing method - basic setup for shoot-through current detection [7].

2.3.1 Adaptive control

A concrete example of an adaptive optimization method is presented in Mappus [10]: as shown in Figure 2.3a, a simple logic circuit is able to detect the zero-voltage crossing at the synchronous rectifier and avoid prolonged body diode conduction by closing it. This emulates the behavior of an ideal diode. The work presented in Krein *et al.* [11] is in fact one of the first articles to emulate multiple switching devices with a MOSFET, including the ideal diode (see Figure 2.3b).



(a) (b)

Figure 2.3: Adaptive optimization methods: (a) Adaptive gate driving circuit [10] and (b) Ideal diode emulation [11].

The same concept is used by Lau *et al.* [12] and Acker *et al.* [13]: when the control MOSFET is turned off, the switch node voltage drops from V_{in} to zero. A comparator detects the zero crossing and immediately turns on the synchronous MOSFET to avoid body diode conduction. Similarly, when the synchronous rectifier is turned off, the body diode is forward biased and the

drain to source voltage goes from zero to negative. By detecting this instant, the control MOSFET is immediately turned on to prevent further diode conduction losses.

The advantages of adaptive control methods are very relevant, particularly when compared to the fixed dead time solution: they allow on-the-fly adjustment for different MOSFETs, temperature and operation conditions. However, despite their popularity, adaptive methods still have some drawbacks: the sensing of the noisy switching voltage, the need for high-speed comparators and the non-compensated MOSFET gate driving delays, which make the adaptive control methods especially ineffective at high switching frequencies.

2.3.2 Predictive control

To overcome the problems mentioned in the previous subsection, the third generation of dead time control methods appeared: the predictive type.

S. Mappus [10] proposes a method that is based on the premise that the dead time required for the next switching cycle will be close to the one required in the current cycle. For the rising edge dead time optimization (PWM CS signal transitions from low to high), a NOR gate is used to sense the drain-to-source voltage of the SS, v_{DS}, and gate-to-source voltage of the CS, v_{GS} and detect when both are low (body diode conduction and CS still off) – see Figure 2.4a and 2.4b. The output of the gate is connected to a 16 bit delay line, in which each bit represents a shift of approximately 4 ns. If the gate output is HIGH, the delay will be reduced by 1 bit while if it is LOW, the delay is increased by the same amount. This continuous forward and backward delay shift is known as dithering and, when optimal, it occurs within an 8 ns window. Conversely, for the optimization of the rising edge dead time (PWM signal transitions from low to high), a comparator is used instead of a NOR gate, in order to avoid false triggering and consequent shoot-through currents (see Figure 2.4c and 2.4d). The authors claim an efficiency improvement over the adaptive method shown in Figure 2.3a of up to 4 %.

Zhao *et al.* [14] proposes a similar scheme with some modifications, called "one-step dead-time correction algorithm". A complex digital circuit, that also includes the NOR gate and delay line from [10], allows the implemented algorithm to converge to the optimal dead time in a single switching cycle, much faster than [10] that needs several steps. It claims an efficiency improvement up to 4 %.

Another example of a predictive dead time optimization method is given by Mei *et al.* [15]. The authors use analog circuits to detect dead-time and optimize it. The optimization circuit is based on a simple RC network and the detection circuit is built on the same die as the synchronous MOSFET in order to provide an accurate indication of body diode conduction. Even though two optimizer circuits are needed for both rising and falling edge dead times optimization, which introduces some component overhead, the method is very fast, unlike the step-based approach of [10]. The authors claim a reduction of total conduction losses up to 1.6 % against a fixed dead time TPS2832 driver IC. However, the results are simulation based and the power consumption of the optimization circuit is not taken into account.

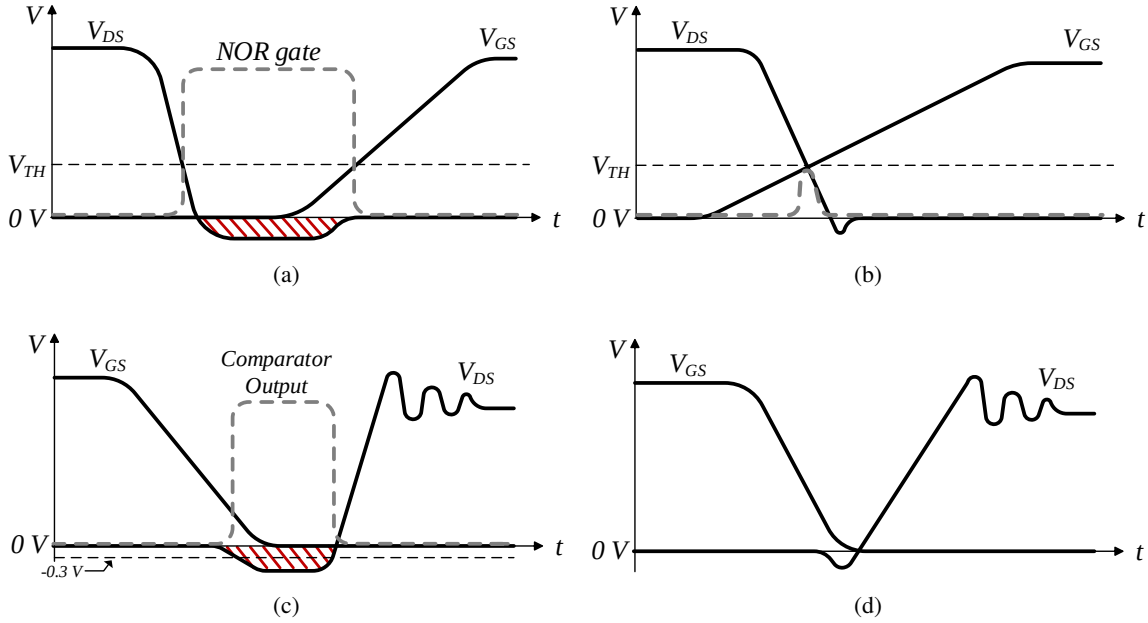


Figure 2.4: Non-optimal and optimal switching waveforms [10]: (a) Non-optimal falling edge switching, (b) optimal falling edge switching, (c) non-optimal rising edge switching, (d) optimal rising edge switching.

Trescases *et al.* [16] is also commonly referenced in the literature. This paper presents a predictive method based on an analog Dead-Time Locked-Loop (DTLL). Two comparators sense the drain-to-source voltage v_{DS} (to detect the zero-voltage crossing) and the gate-to-source voltage v_{GS} (to detect the threshold voltage crossing). The comparators are followed by a phase detector that provides up and down pulses for a charge pump. The charge pump then outputs a linear delay control voltage that determines the dead time for the MOSFET under control, completing the feedback loop. Both the control and synchronous MOSFET have a DTLL block to drive them, which introduces some overhead. However, the experimental results show a significative reduction in body diode losses.

2.4 Sensorless methods

In spite of their undeniable ingenuity and results, the previously mentioned techniques always rely on external circuitry. This can bring cost and effort overhead as well as added power consumption, which counters the benefits of adding said circuitry. Sensorless methods are the answer for applications where extra components cannot be added. They depend exclusively on parameters that are already measured for control or protection of the converter.

An example of sensorless dead time optimization is presented by Abu-Qahouq *et al.* [17]. This method is based on maximum power point tracking (MPPT), commonly used on solar arrays. The approach proposed is called maximum efficiency point tracking (MEPT) because it indirectly measures the efficiency of the converter and uses it to optimize the dead time. Since an absolute

value of the efficiency is not required, only the input current, I_{in}, is monitored. The algorithm evaluates the signal of the gradient of the input current and the dead time shift. If they are the same, the time step should be added to the current dead time. Conversely, if they are opposite, the current dead time should be subtracted a time step. The search stops when the operation point of minimal input current (maximum efficiency) is found, as seen in Figure 2.5. Despite being comparatively slow and thus unsuitable for applications with a fast varying load, this method yields good results with a claimed efficiency improvement of 2 %.

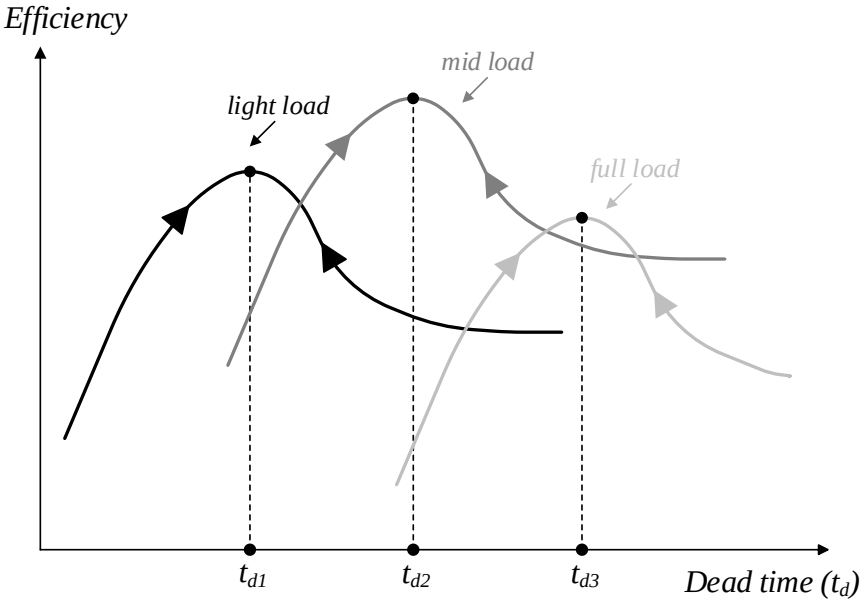


Figure 2.5: Maximum Efficiency Point Tracking algorithm [17].

Pizzutelli *et al.* [18] proposes another sensorless dead time control method based on input current. It claims to improve over [17] because it can never get stuck at local minima and it is suitable for high efficiency (>90 %) applications, where the small current variations cannot be easily measured. In this method, a compare algorithm is used to test two different dead times and choose the one that yields the highest efficiency. This is achieved by summing the input current to an accumulator while one dead time is applied and subtracting while the other dead time is applied. After multiple cycles, the signal of the result from the accumulator indicates which of the dead time values produced a lower input current and consequently a higher efficiency (see Figure 2.6). The authors use the compare algorithm iteratively in a global search algorithm that is capable of ignoring local minima. The experimental results point to a maximum of 1 % efficiency improvement.

Yousefzadeh *et al.* [19] proposes another method of optimization of dead times without sensing. The idea behind it is to optimize the efficiency by minimizing the duty cycle. For a given load, there is an optimal duty cycle: when the dead times are too long (body diode conduction) or too short (shoot-through current) the average value of the switching voltage $\langle v_S \rangle$ is reduced.

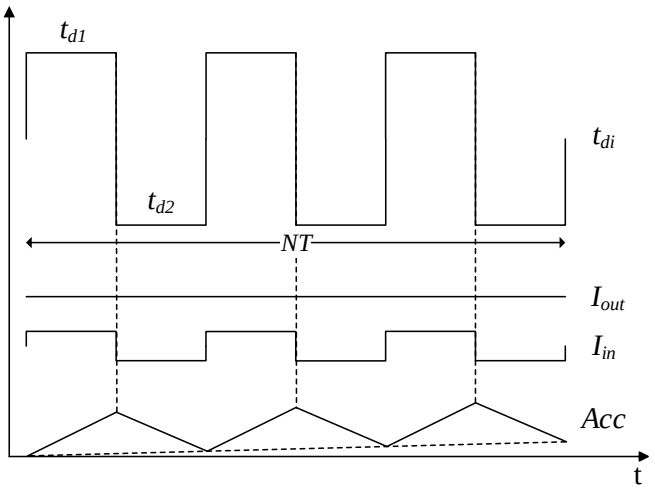


Figure 2.6: The output of the accumulator is positive, which means t_{d2} yields the lowest input current, therefore being more efficient [18].

This forces the duty cycle control loop to apply a correction by adding duty cycle to increase this average. Since the duty cycle control loop is much faster than the dead time control loop, it is possible to reduce the dead time in small steps and monitor the current duty cycle value. In the beginning, the duty cycle should decrease with the decreasing dead time. This means that the efficiency is improving. Once the duty cycle increases, the dead time should stop decreasing (points *a* and *b* of Figure 2.7). The shoot-through condition has already been met and the efficiency will only worsen. Both sides of dead time can be optimized with this method, as seen in Figure 2.7. The authors claim an improvement of efficiency of nearly 4 % against a fixed dead time solution.

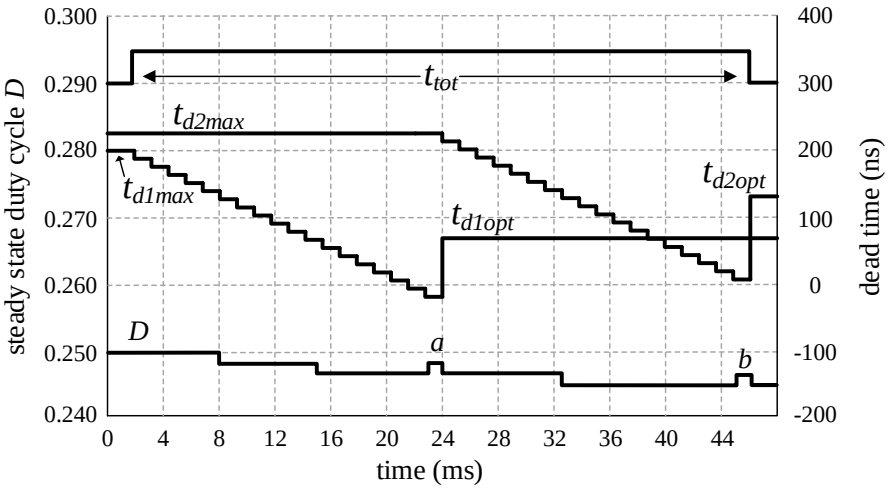


Figure 2.7: Sensorless dead time optimization using duty cycle minimization [19].

Reiter *et al.* [20] proposes another method based on duty cycle monitoring. It is only suitable for multiphase DC/DC converter, a typical automotive architecture. The algorithm introduces a perturbation to the dead times in one phase of the converter and monitors the phase-currents deviation. The most efficient phase is the one with the highest phase-current. A current-based controller compensates this deviation by adjusting the duty cycles. The duty cycle difference between phases allows the detection of the optimal dead times (similarly to [19]). This method achieved an efficiency improvement of 1.40 %.

An interesting controller with incorporated dead time optimization is presented by Peterchev *et al.* [21]. It is based on an extremum-seeking algorithm. Extremum-seeking control is a non-model based real-time optimization approach used when only limited knowledge of the system is available. In this paper, the controller introduces perturbations to the dead times and measures the gradient of a cost function related to power efficiency (input current, temperature, duty cycle, etc.).

2.5 Summary and discussion

A review of the literature related to the optimization of dead times on DC/DC converters has been done. Each method presented has advantages and disadvantages, and it is not always easy to compare them: each approach is tested in unequal conditions and compared with different techniques (see Table 2.1). That said, the efficiency improvement values presented by each author cannot be taken in an absolute manner – the conditions in which they are obtained must be analysed if one wants to infer truthful comparisons.

The 5 % efficiency raise in Huang *et al.* [8] is mainly at light load conditions, when a width controller is able to down size the integrated power MOSFETs. This is not available for larger DC/DC converters made of discrete components. Despite the promising results, this method uses complex circuitry which introduces some overhead.

Zhao *et al.* [14] is appropriate for applications with frequent load variations. However, it is considerably more complex than other predictive methods, such as Mappus [10]. The latter case displays a larger efficiency improvement than the first, because even though they both rank at 4 %, [14] is compared against a fixed dead time solution, while Mappus [10] is tested against an adaptive method which, by itself, already improves over fixed dead time.

The results presented by Mei *et al.* [15] should be taken with caution, since they were not produced under real testing conditions, but rather by simulation in PSIM. The benchmark was a TPS2832, a synchronous buck converter MOSFET driver with adaptive dead time control. The IC was modelled in PSIM with a fixed dead time of 15 ns, its minimal achievable delay.

The proposed sensorless methods have an overall worse performance than the switching voltage and current methods. They are also slower and thus not suitable for fast varying loads. But the fact that no additional components are necessary to implement them, makes them a very attractive option for digitally controlled DC/DC converters with minimal component overhead.

The 2 % efficiency improvement claimed by Abu-Qahouq *et al.* [17] is obtained in comparison to a fixed dead time control of 80 ns, the worst case scenario for the converter used. However, only the rising edge dead time optimization is demonstrated. Furthermore, this method does not cover the existence of local minima that may stop the algorithm before the optimum value is reached. Pizzutelli *et al.* [18] uses a similar algorithm that eliminates that issue and can yield a 1 % improvement against an unspecified fixed dead time.

The method proposed by Yousefzadeh *et al.* [19] yields the largest efficiency improvement amongst sensorless methods. The benchmark is a fixed dead-time of 200 and 220 ns, safe for all operation conditions. However, this comes at the cost of having a high resolution duty-cycle command, which is not always achievable. The efficiency increase obtained by Reiter *et al.*, a similar method, is also measured against a fixed dead time solution of 150 ns.

None of the reviewed methods has been implemented or tested in a microcontroller unit.

Table 2.1: Summary of proposed dead time optimization methods and their claimed results.

Reference	Type	Year	Implementation	Previous Efficiency/ Improvement
Huang *et al.* [8]	Switching currents	2007	IC	87% / 1∼5%
Zhao *et al.* [14]	Predictive control	2010	FPGA	86∼88% / 1∼4%
Mappus [10]	Predictive control	2003	IC	86∼93% / 1∼4%
Mei *et al.* [15]	Predictive control	2013	Analog circuit (PSIM simulated)	82.2∼91.1% / 0.9∼1.6%
Abu-Qahouq *et al.* [17]	Sensorless	2006	FPGA	78∼83% / 1.25∼2%
Pizzutelli *et al.* [18]	Sensorless	2007	FPGA	92% / 1%
Yousefzadeh *et al.* [19]	Sensorless	2006	FPGA	88% / 4%
Reiter *et al.* [20]	Sensorless	2010	DSP	>90% / 0.6∼1.4%

Chapter 3

Analysis of Dead Time Optimization

Multiple variables influence the behavior of a synchronous buck converter in a complex way. In this chapter a theoretical analysis of the processes behind dead time optimization is done. The first section addresses the modeling of the algorithm, in mathematical terms. In the second section the existence and consequences of certain system constraints is explored. The third section is about the resources necessary in the digital controller responsible for running the algorithm. Lastly, in the fourth section, an analysis of the power efficiency improvement is made and related to the characteristics of the controller.

Since one of the objectives of this thesis is to implement a simple, computationally inexpensive algorithm that takes advantage of the preexisting hardware for the elimination of dead time induced power losses, the duty cycle minimizing approach reported by Yousefzadeh *et al.* [19] will be taken as the starting point of this work. However, this does not impose any limitation to the analysis that is made in this chapter.

3.1 Modeling of the Algorithm

Switched mode converters are characterized for having a storage element that is sequentially charged and discharged to provide controlled energy flow. In a synchronous buck converter, this element is the inductor L. The average voltage across an inductor at steady state is null, which implies that integrating the voltage across L over a single switching period T_s must equal 0. This allows us to derive the output voltage as

$$\int_0^{T_s} V_L \, dt = 0 \tag{3.1}$$

$$\int_{t_{d,r}} V_L \, dt + \int_{t_{on}} V_L \, dt + \int_{t_{d,f}} V_L \, dt + \int_{t_{off}} V_L \, dt = 0$$

$$\int_{t_{d,r}} (-V_D - V_{out}) \, dt + \int_{t_{on}} (V_{in} - V_{out}) \, dt + \int_{t_{d,f}} (-V_D - V_{out}) \, dt + \int_{t_{off}} -V_{out} \, dt = 0$$

$$(-V_D - V_{out})\, t_{d,r} + (V_{in} - V_{out})\, t_{on} + (-V_D - V_{out})\, t_{d,f} - V_{out}\, (T_s - t_{d,r} - t_{d,f}) = 0$$

$$V_{out} = V_{in} \frac{t_{on}}{T_s} - V_D \frac{t_{d,r} + t_{d,f}}{T_s} \tag{3.2}$$

where t_{on} is the conduction time of the control switch, t_{off} is the conduction time of the synchronous switch, and t_{on}/T_s is the duty cycle. These quantities are depicted in Figure 1.2.

This mathematical relation[1] can easily be given a physical significance: increasing the duty cycle results in a higher output voltage, while increasing the dead time results in a lower output voltage. In fact, the output voltage equals the average voltage at the switching node v_{sw} (between CS and SS) as seen in Figure 3.1.

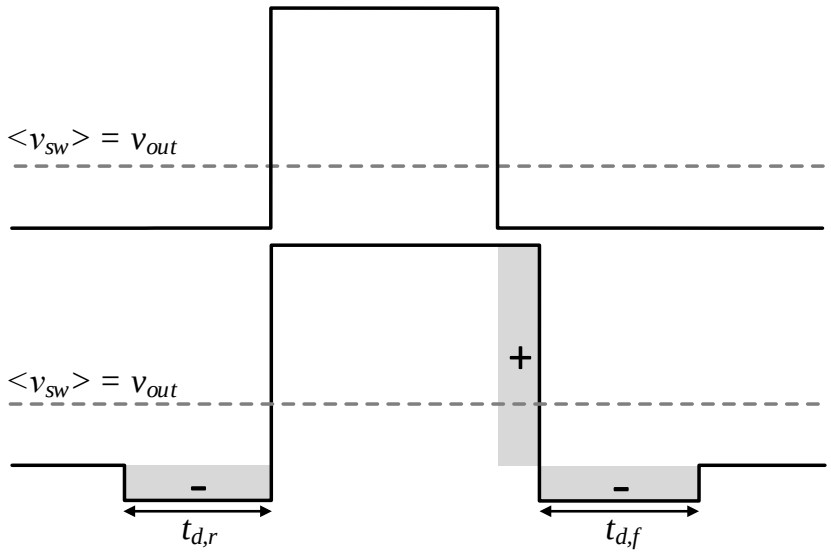


Figure 3.1: The voltage at the switching node is either $-V_D$ or V_{in}. Its average equals the output voltage V_{out}.

Since the output voltage is regulated by the converter control loop, whatever this might be composed of, any perturbation in the dead time has a direct impact on the duty cycle. This is important for a sensorless, duty cycle minimizing algorithm for dead time optimization, such as the one that is going to be used in this work. However this behavior is common to all converters with a regulated output voltage. Considering the diagram of Figure 3.2, it is possible to describe each stage in time as follows:

1. At t_{k-1}, the dead time is changed (Δt_d), leading to a variation on the output voltage (ΔV_{out});
2. At t_k, the controller detects the deviation ΔV_{out} and compensates by regulating the duty cycle (Δt_{on});

[1] Equation 3.2 without the negative term is used for the case of an ideal converter, where the effects of the dead time are not considered.

3. At t_{k+1}, the output voltage returns to its regulated value.

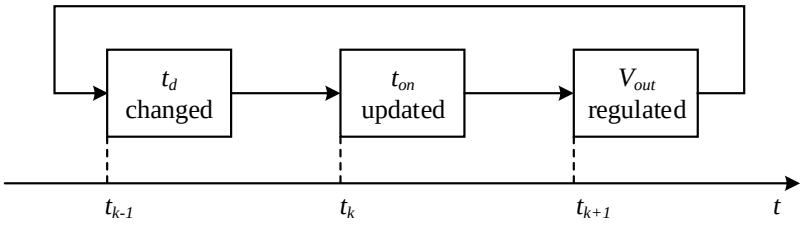


Figure 3.2: The four sequential stages of the dead time optimization algorithm.

The relations between Δt_d, Δt_{on} and ΔV_{out} at all the stages of the algorithm can be quantified using Equation 3.2. Let us assume that $t_d = t_{d,r} + t_{d,f}$ and that positive values of Δt_d and Δt_{on} mean a decrease in t_d and t_{on} respectively. This helps to simplify the calculations and avoid having to always deal with negative numbers.

$$t = t_{k-1}, \quad \Delta t_{on} = 0: \quad \Delta V_{out} = V_D \frac{\Delta t_d}{T_s} \tag{3.3}$$

$$t = t_k, \quad \Delta t_d = 0: \quad \Delta t_{on} = T_s \frac{\Delta V_{out}}{V_{in}} \tag{3.4}$$

Moreover, the output voltage stays constant over a single period of the optimization algorithm (Equation 3.5), which allows for a relation between a change in dead time and a change in duty cycle to be found (Equation 3.6).

$$V_{out}\Big|_{t_{k-1}} = V_{out}\Big|_{t=t_{k+1}} \tag{3.5}$$

$$V_{in}\frac{t_{on}(t_{k-1})}{T_s} - V_D\frac{t_{d,r}(t_{k-1}) + t_{d,f}(t_{k-1})}{T_s} = V_{in}\frac{t_{on}(t_{k+1})}{T_s} - V_D\frac{t_{d,r}(t_{k+1}) + t_{d,f}(t_{k+1})}{T_s}$$

$$V_{in}\Big(t_{on}(t_{k+1}) - t_{on}(t_{k-1})\Big) = -V_D\Big(\big(t_{d,r}(t_{k-1}) + t_{d,f}(t_{k-1})\big) - \big(t_{d,r}(t_{k+1}) + t_{d,f}(t_{k+1})\big)\Big)$$

$$V_{in}\Delta t_{on} = V_D(\Delta t_{d,r} + \Delta t_{d,f})$$

$$\Delta V_{out} = 0: \quad \Delta t_{on} = \frac{V_D}{V_{in}}\Delta t_d \tag{3.6}$$

The changes operated during the optimization algorithm are fully described by Equations 3.3, 3.4 and 3.6. In a single cycle of the algorithm there is only one operation point $(\Delta t_{on}, \Delta t_d, \Delta V_{out})$.

This vector belongs to the set of possible solutions **r** described by the aforementioned equations and represented as a line in the real coordinate 3-dimensional space $\mathbb{R}^3$ (Equation 3.7).

$$\mathbf{r} = (\Delta t_{on}, \Delta t_d, \Delta V_{out}) = \Delta t_{on} \cdot \left(1, \frac{V_{in}}{V_D}, \frac{V_{in}}{T_s}\right) \in \mathbb{R}^3 \tag{3.7}$$

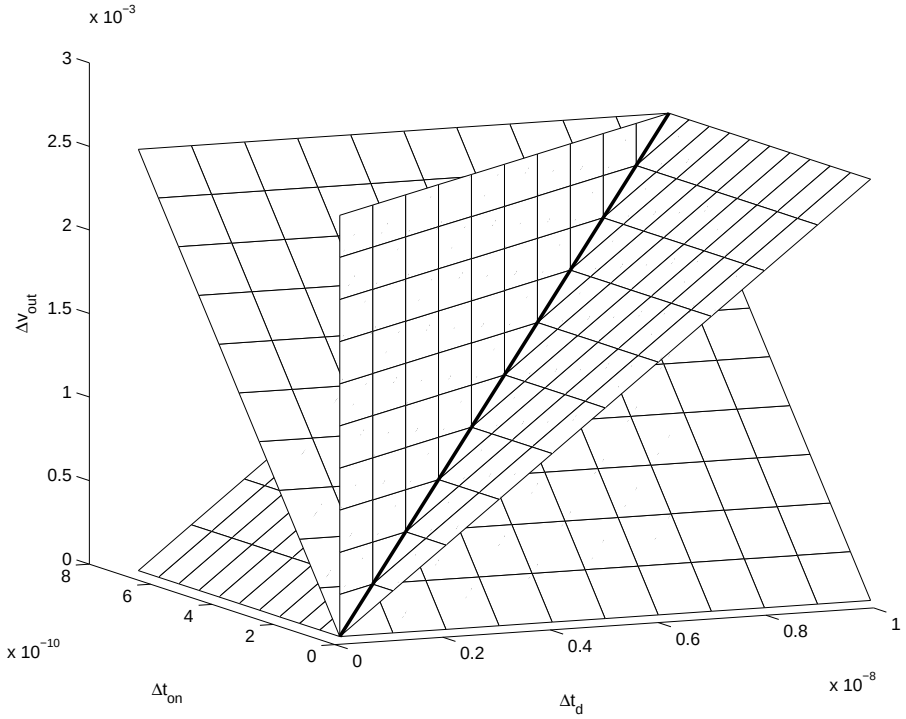


Figure 3.3: Representation in $\mathbb{R}^3$ of Equations 3.3, 3.4 and 3.6, defined by two ramps and a vertical plan. The thicker line where the planes intercept defines the set of possible operation points **r** for the algorithm.

3.2 Operation Limits

In the previous section, the set of possible operation points of the algorithm **r** was determined. However, not all points of this set of solutions are feasible when the algorithm is running in a controller with limited resources. The minimum duty cycle and dead time variations are restrained by the timer resolution, while the minimum output voltage variation is limited to one least-significant-bit of the ADC (3.8).

$$\Delta t_d \geq t_{min}; \quad \Delta t_{on} \geq t_{min}; \quad \Delta V_{out} \geq V_{min} \tag{3.8}$$

The minimum timer and ADC variation can be represented in time/voltage or in bits, which is a more general notation and therefore can be applied to a wider range of digital controllers. The three inequalities in 3.8 add up to Equation 3.7 to define the set of possible operation points for a

given controller with a N_{timer}-bits timer unit and N_{ADC}-bits ADC (Equation 3.9). Said region can be represented in $\mathbb{R}^3$ by a line limited by three planes, as seen in Figure 3.4.

$$\mathbf{r} = (\Delta t_{on}, \Delta t_d, \Delta V_{out}) = \Delta t_{on} \cdot \left(1, \frac{V_{in}}{V_D}, \frac{V_{in}}{T_s}\right) \in \mathbb{R}^3 \text{ subject to:}$$

$$\Delta t_{on},\ \Delta t_d \geq \frac{T_s}{2^{N_{timer}}} \text{ and } \Delta V_{out} \geq \frac{V_{FS}}{2^{N_{ADC}}} \tag{3.9}$$

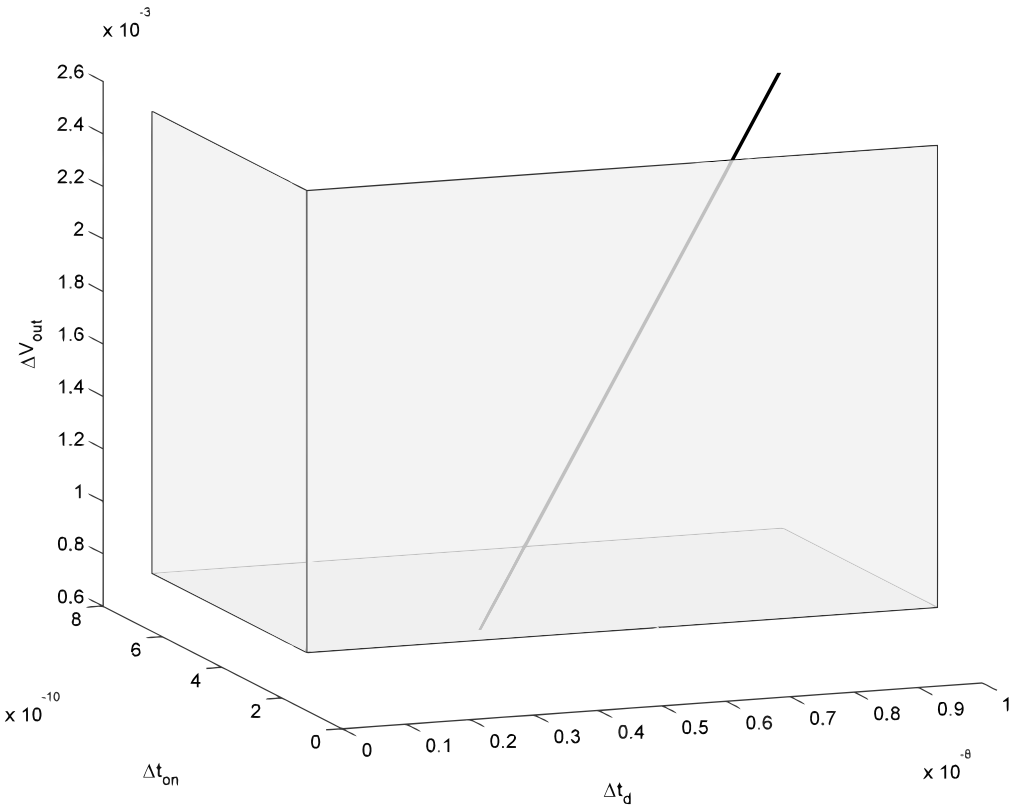


Figure 3.4: Example of a set of possible operation points for $V_{in} = 12\,\text{V}$, $V_D = 0.8\,\text{V}$, $f_s = 320\,\text{kHz}$, $t_{min} = 150\,\text{ps}$ and $V_{min} = 800\,\mu\text{V}$.

The minimum Δt_d possible, taking the given restraints into account, can be derived from the previous relations and expressed as Equation 3.10. Geometrically, this minimized operation point corresponds to the intersection of line $\mathbf{r}$ with one of the three limit planes, as represented in Figure 3.4. The interest of using the lowest possible Δt_d has to do with the fact that the algorithm will have a higher resolution and achieve better results. This idea will become clearer on Section 3.4.

$$\Delta t_d = \frac{V_{in}}{V_D}\Delta t_{on}, \quad \Delta t_d = \frac{T_s}{V_D}\Delta V_{out}$$

$$\Delta t_{d,min} = \frac{V_{in}}{V_D} t_{min} \quad \vee \quad \Delta t_{d,min} = \frac{T_s}{V_D} V_{min}$$

$$\Delta t_{d,min} = \frac{V_{in}}{V_D}\frac{T_s}{2^{N_{timer}}} \quad \vee \quad \Delta t_{d,min} = \frac{T_s}{V_D}\frac{V_{FS}}{2^{N_{ADC}}}$$

$$\Delta t_{d,min} = \max\left\{\frac{V_{in}}{V_D}\frac{T_s}{2^{N_{timer}}}, \frac{T_s}{V_D}\frac{V_{FS}}{2^{N_{ADC}}}\right\}$$

$$\Delta t_{d,min} = T_s\frac{V_{in}}{V_D}\max\left\{2^{-N_{timer}}, \frac{V_{FS}}{V_{in}}2^{-N_{ADC}}\right\} \tag{3.10}$$

At Equation 3.10 the maximum value is used for calculating $\Delta t_{d,min}$ because choosing the minimum would imply that one of the other two variables Δt_{on} or ΔV_{out} would be smaller than t_{min} or V_{min} and violate the restraints imposed by the system.

3.3 Resource Usage

By the end of the previous section, it was shown that the minimum possible dead time variation $\Delta t_{d,min}$ is function of the number of bits available for timer and ADC modules (Equation 3.10). Let us now consider a variable φ representing the distance between the two limits of Δt_d (Equation 3.11).

$$\varphi = 2^{-N_{timer}} - \frac{V_{FS}}{V_{in}}2^{-N_{ADC}} \tag{3.11}$$

If the minimum $\Delta t_{d,min}$ in Equation 3.10 is defined by $2^{-N_{timer}}$, φ is positive and the optimization is restrained by the resolution of the timer. Conversely, if $\Delta t_{d,min}$ is defined by $V_{FS}/V_{in}2^{-N_{ADC}}$, φ is negative and the optimization is restrained by the resolution of the ADC. Being restrained by a resource means that, no matter how much the other resource is improved, $\Delta t_{d,min}$ will not decrease and the optimization will not improve.

$$\begin{cases} \varphi > 0 \implies \text{timer restrained optimization} \\ \varphi < 0 \implies \text{ADC restrained optimization} \end{cases} \tag{3.12}$$

This may also be seen in the geometrical perspective offered in Figure 3.4: if the set of solutions $\mathbf{r}$ intercepts the ΔV_{out} plane (in the bottom), the optimization is ADC restrained. Conversely, if $\mathbf{r}$ intercepts the Δt_{on} plane (on the side), the optimization is timer restrained. Interesting too, is to note that, even though Δt_d is also defined by the timer and therefore limited to its resolution, it will never be a restrain to the algorithm, since $\Delta t_d > \Delta t_{on}$ (as long as $V_{in} > V_D$, which is true for most converters).

Figure 3.5 depicts the value of φ for different combinations of N_{timer} and N_{ADC}. If the operation point is not at the line where $\varphi = 0$, there is a "waste" of resources on the algorithm. This means

that either the timer or the ADC use excessive bits that are not contributing to the optimization. That said, the resource usage is optimal for

$$N_{timer} = N_{ADC} + \log_2\left(\frac{V_{in}}{V_{FS}}\right) \tag{3.13}$$

If a certain operation point where $\varphi = |\varphi_i|$ is moved to a point where $\varphi = |\varphi_f| < |\varphi_i|$, the resource usage is improved. However, this does not guarantee that the power efficiency will improve. Over the next section this will be explained further.

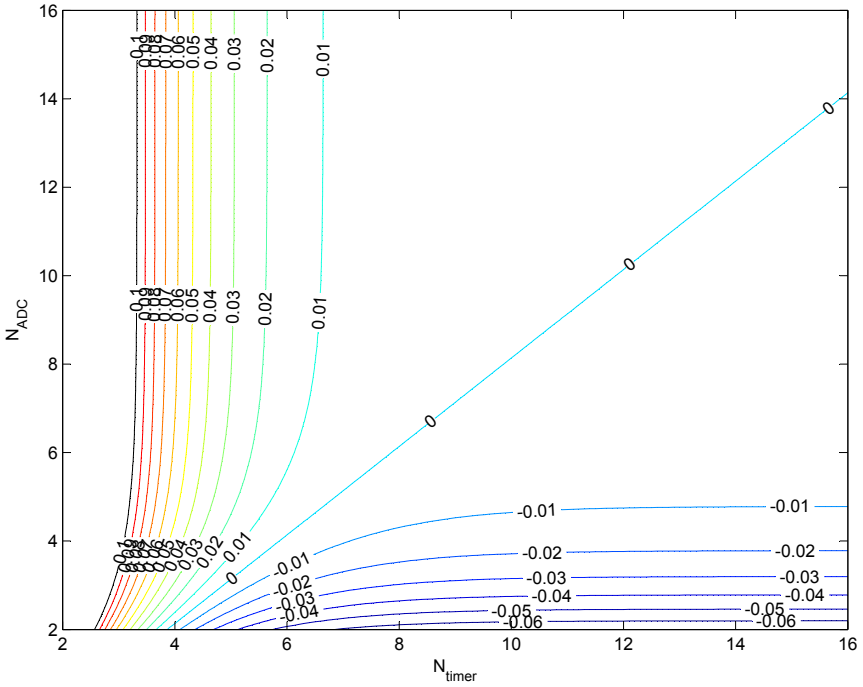


Figure 3.5: Dependency of φ on timer (N_{timer}) and ADC resolution (N_{ADC}) for $V_{in} = 12\,\text{V}$, $V_D = 0.8\,\text{V}$, $f_s = 320\,\text{kHz}$ and $V_{FS} = 3.3\,\text{V}$.

3.4 Power Efficiency Improvement

In Chapter 1 the problem of dead time, diodes and power losses has already been mentioned: asynchronous buck converters lose energy through the freewheel diode during the entire t_{off} time, while synchronous buck converters partially resolve this by replacing the diode with a MOSFET. However, the body diode of the synchronous switch still conducts for t_d. Despite being significantly lower than in the asynchronous counterpart, the diode losses of the synchronous converter still weight in the total converter efficiency.

The dominant part of the losses on a diode comes from conduction. Conduction losses are directly proportional to the diode conduction time, or as it is most commonly called, the dead time (Equation 3.14). Additionally, the body diode suffers from reverse recovery losses. For a diode to

transition from conducting to blocking state, the charge distribution must change. This movement of the free carriers does not occur instantaneously, so during this short period of time, known as reverse recovery time, the diode will not block and, therefore, temporarily conduct a large current in the reverse direction [22]. The losses produced by this non-ideality are represented by Equation 3.15.

$$P_{D,cond} = V_D I_{out} f_s (t_{d,r} + t_{d,f}) \tag{3.14}$$

$$P_{D,RR} = \frac{1}{2} Q_{RR} V_{in} f_s \tag{3.15}$$

Another source of losses due to the existence of dead time is the increase of conduction time of the control switch, in order to maintain the same output voltage, as represented in Equation 3.6. However, these losses are in the order of nW compared to the hundreds of mW of the conduction losses and will not be taken into account. The reverse recovery losses are also considerably smaller than the conduction losses, about 100 times less. Furthermore, they are constantly present, unless the diode is not at all forward-biased to begin with. Therefore, from now on the total diode losses will be approximated by the conduction losses only.

$$P_{loss} \approx V_D I_{out} f_s t_d$$

We are now in conditions to quantify how much the power losses in the body diode change when there is a variation in the dead time. In the previous section of this chapter, the minimum dead time variation $\Delta t_{d,min}$ for a system with certain characteristics was determined (Equation 3.10). This is the dead time variation that also produces the minimum power loss variation.

$$\Delta P_{loss,min} = V_D I_{out} f_s \Delta t_{d,min} \tag{3.16}$$

Let us consider an initial (fixed) dead time $t_{d,i}$ that causes a certain amount of initial power losses $P_{loss,i}$. Since the minimum dead time decrement is given by $\Delta t_{d,min}$, and such variation reduces the power losses in $\Delta P_{loss,min}$, it is possible to determine how much from the initial losses can be eliminated with a given algorithm operation point. Let Ψ be the efficiency improvement factor, a relation between eliminated losses $P_{loss,elim}$ and total initial losses $P_{loss,i}$.

$$\Psi = \frac{P_{loss,elim}}{P_{loss,i}} \tag{3.17}$$

The algorithm will reduce the dead time in steps of $\Delta t_{d,min}$, subtracting with each step $\Delta P_{loss,min}$ from the total losses $P_{loss,i}$. This means that $P_{loss,elim} = \lfloor \gamma \rfloor \cdot \Delta P_{loss,min}$ where $\lfloor \gamma \rfloor$ is the integer number of steps that is possible to take towards null losses. Hence, Ψ can be expressed as

$$\Psi = \frac{\lfloor \frac{P_{loss,i}}{\Delta P_{loss,min}} \rfloor \Delta P_{loss,min}}{P_{loss,i}} \tag{3.18}$$

As explained before, γ is the ideal number of steps that are necessary to achieve total loss elimination. In practice however, steps can only be integer, which allows Equation 3.20 to be formulated, by replacing in Equation 3.18.

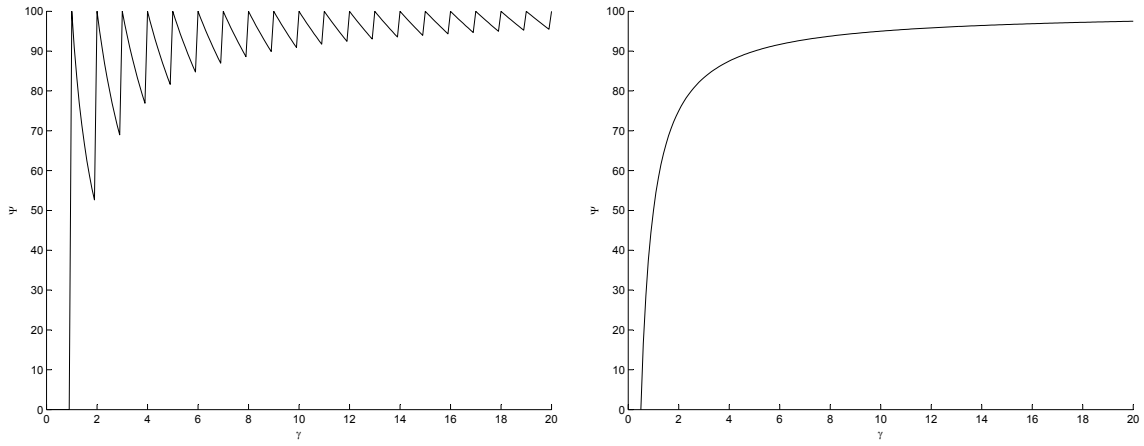
$$\begin{cases} P_{loss,elim} = \lfloor \gamma \rfloor \cdot \Delta P_{loss,min} \\ P_{loss,i} = \gamma \cdot \Delta P_{loss,min} \end{cases}$$

$$\gamma = \frac{P_{loss,i}}{\Delta P_{loss,min}} = \frac{V_D I_{out} f_s t_{d,i}}{V_D I_{out} f_s \Delta t_{d,min}} = \frac{t_{d,i}}{\Delta t_{d,min}} \tag{3.19}$$

$$\Psi = \frac{\lfloor \gamma \rfloor}{\gamma} \tag{3.20}$$

Ψ can be depicted as a function of γ, as seen in Figure 3.6a. Notice that when γ is an integer, even at small values, the losses are fully eliminated. However, this only happens at very precise points: if γ suffers the slighest variation, the improvement will drop significantly. To overcome this, the floor function $\lfloor \cdot \rfloor$ may be approximated by $\lfloor x \rfloor = x - \frac{1}{2} + \frac{1}{\pi} \sum_{k=1}^{\infty} \frac{\sin(2\pi k x)}{k} \approx x - \frac{1}{2}$ and Equation 3.20 ends up being represented as Equation 3.21. Figure 3.6b depicts the approximated, averaged relation of γ with Ψ which better describes what happens in reality than Figure 3.6a.

$$\Psi = 1 - \frac{\Delta P_{loss,min}}{2 P_{loss,i}} = 1 - \frac{\Delta t_{d,min}}{2 t_{d,i}} \tag{3.21}$$



(a) (b)

Figure 3.6: Depictions of $\Psi(\gamma)$: (a) Using the theoretical $\lfloor \gamma \rfloor$, (b) using the floor function approximation that better models the real variations of γ.

A small value of γ means a large initial dead time $t_{d,i}$ or a large step $\Delta t_{d,min}$, which is caused by having a low resolution at either the timer or the ADC. Conversely, a large γ means a small $\Delta t_{d,min}$ or a small $t_{d,i}$, which is the direct consequence of having a high resolution at both resources.

Since Ψ is function of γ, γ is a function of $\Delta t_{d,min}$ and $\Delta t_{d,min}$ is function of N_{ADC} and N_{timer}, a direct relation between the efficiency improvement factor and the characteristics of the digital controller resources can be derived (Equation 3.22).

$$\Psi = 1 - \frac{T_S \frac{V_{in}}{V_D} \max\left\{2^{-N_{timer}},\ \frac{V_{FS}}{V_{in}} 2^{-N_{ADC}}\right\}}{2t_{d,i}} \tag{3.22}$$

Equation 3.22 can be represented in the 3-dimensional space, as seen in Figure 3.7. The "crease" formed in the middle of the surface corresponds to the line $\varphi = 0$, as mentioned in Section 3.3. It is also possible to observe how the optimization algorithm is almost always restrained by the resolution of one of the resources: for example, if the algorithm is running in a controller with a 8 bit ADC, the maximum efficiency improvement is about 90%, whether the timer has a 10 or 20 bits resolution.

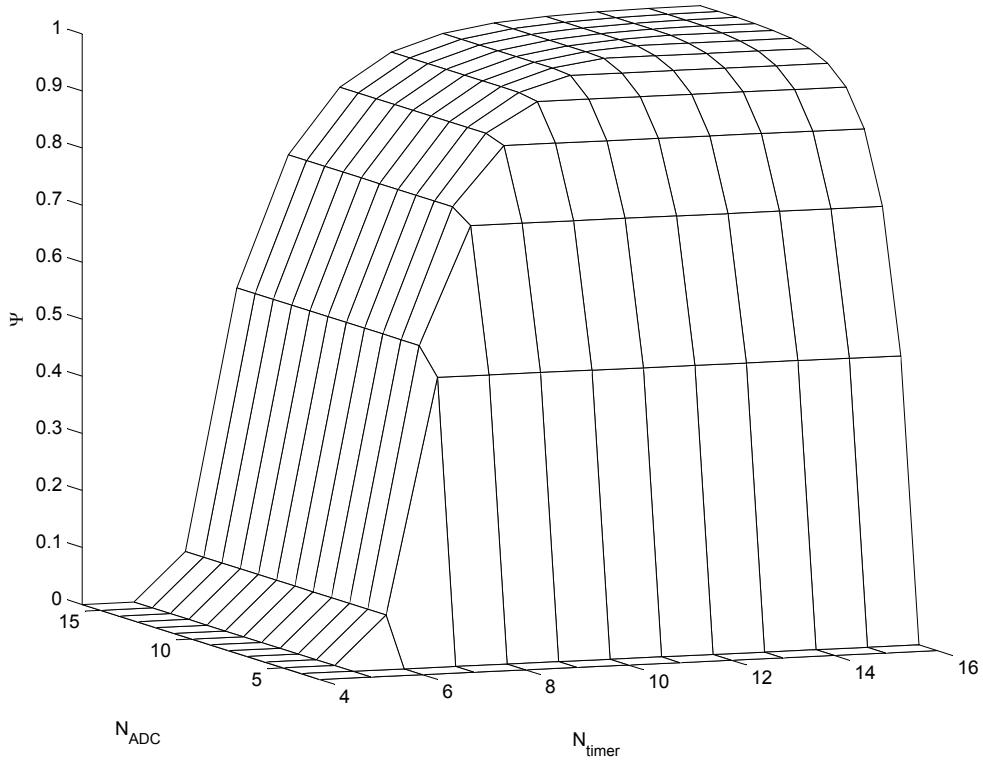


Figure 3.7: Dependency of the efficiency improvement factor Ψ with ADC and timer resolution for $t_{d,i} = 400\,\text{ns}$, $V_{in} = 12\,\text{V}$, $V_D = 0.8\,\text{V}$, $V_{ref} = 3.3\,\text{V}$ and $f_s = 320\,\text{kHz}$.

3.5 Summary

In this chapter the theory behind a generic sensorless, duty cycle minimizing, dead time optimization algorithm was explored and related to the characteristics of a digital controller. It is possible to conclude that the quality of the dead time optimization is dependent on the resolution

of timer and ADC and that these two are also related between each other: the resolution of one of the resources may restrain the optimization, no matter how good the resolution of the non-restraining resource is. Since the algorithm is theoretically proved, the next chapter will present a practical implementation and relate its results with the theory.

Chapter 4

Experimental Results

In the previous chapter, the theory behind the optimization of dead time related losses using a duty cycle minimizing algorithm was presented. Now a concrete application of an algorithm of this kind is presented. In the first section, the hardware and its relevant capabilities are explored as well as the experimental setup used in this work. Next the software is presented, consisting of the voltage control law, the dead time algorithm and its coding. The third section consists of the performed measurements and their relation with the previously expected results. Lastly, the fourth section presents a simulation in Simulink and the extracted data.

4.1 Powertrain and Microcontroller Unit

The most important part of the hardware is the buck converter and the microcontroller. In this work Infineon synchronous buck converter kit with a control card was used as the proof-of-concept prototype. This kit has two channels and is controlled by a detachable card containing an ARM-Cortex M4 based XMC4200 microcontroller [4] and on-board Segger J-Link debugger. The most important parameters and components of the converter and control card are listed on Table 4.1. Figure 4.1 shows the main features of the buck converter kit and the control card, as well as how the assembled setup looks like.

Table 4.1: Operation parameters for the prototype buck converter.

Parameter	Value
V_{in}	12 V
$V_{out,nom}$	1.8 V
$I_{out,max}$	5 A
$P_{out,max}$	9 W
L	33 μH
C	330 μF
f_s	320 kHz

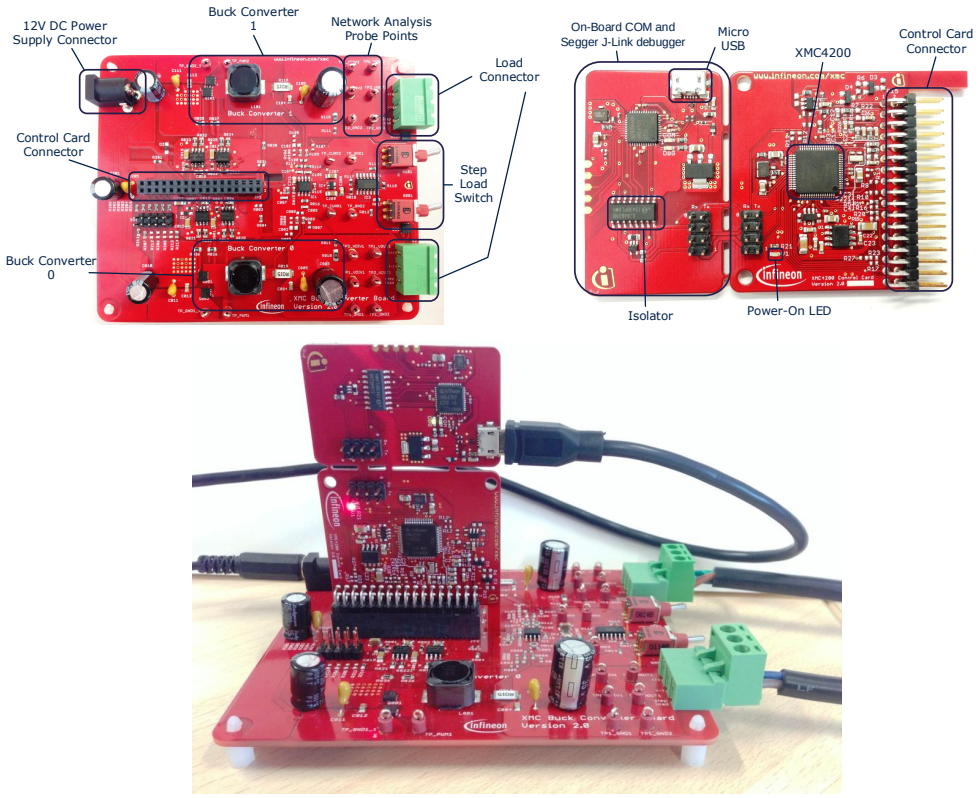


Figure 4.1: The proof-of-concept prototype: buck converter kit and XMC4200 control card.

The load selected is a 0.5 Ω, 15 W power resistor. However, the step load switch also gives the possibility to transition from and to no-load conditions instantaneously. This will be useful for testing load variations, as explained further in this chapter.

The XMC4200 features several peripherals that are relevant for this kind of application, such as the Interrupt Controller (NVIC), Input/Outputs (GPIO) and the Analog-to-Digital Controller (VADC). An overview of all the used peripherals in this application is depicted in Figure 4.2.

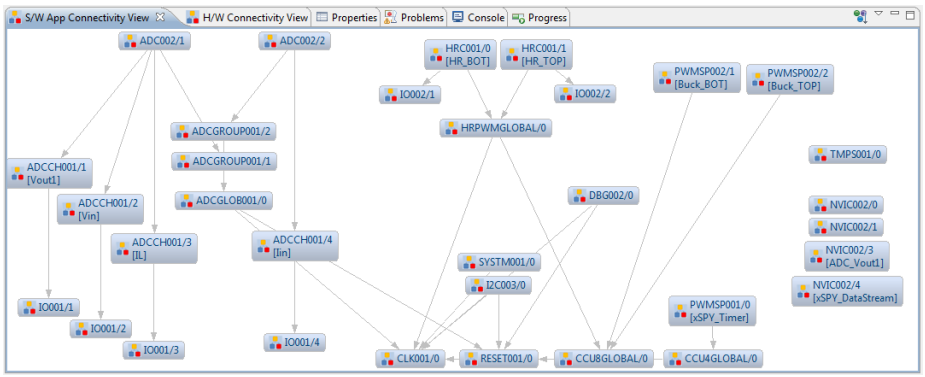


Figure 4.2: A view of the microcontroller peripherals required by the project.

The most important for this work is, however, the High Resolution PWM and the Timer, which is called Capture and Compare Unit (CCU8) in this microcontroller family. The HRPWM

is a peripheral that consists of a Compare and Slope Generator (CSG) for peak current control, slope compensation, clamping, blanking, etc., and a High Resolution Channel (HRC) which can adjust the edges of its input signal by very small steps in the order of picoseconds. The CCU8 (12.5 ns resolution) generates a low resolution PWM waveform which enters the HRPWM (150 ps resolution), is fine adjusted and then can be output to a pin, as a high-resolution PWM waveform (as depicted in Figure 4.5 of the next section). Later, there will be an explanation on how the HRPWM is programmed and used.

Using HRPWM for a dead time optimization algorithm is interesting for two reasons: not only the minimum dead time step $\Delta t_{d,min}$ can be smaller and hence the efficiency improvement factor Ψ increases, but also because the control loop can vary the duty cycle in a much finer way, leading to smaller Δt_{on} values. For an algorithm that uses the duty cycle as a minimization variable, increasing the resolution of the duty cycle is a clear advantage.

The power efficiency of the converter can be expressed as

$$\mu = \frac{P_{out}}{P_{in}} = \frac{V_{out} \cdot I_{out}}{V_{in} \cdot I_{in}} \tag{4.1}$$

Since the load is constant and known in this experimental setup and the output and input voltages are fixed and constant, only the input current is missing in order to calculate the absolute efficiency of the converter. For that purpose, a small high-side current sensing circuit was designed (Figure 4.3). A shunt resistor creates a small drop voltage that is amplified by an AD8219. This current sensor has a 60 V/V gain, 500 kHz dynamic response, may be supplied directly from the inputs and has low input current. A RC low-pass filter was also used at the output and the amplifier package was soldered to an MSOP-to-DIL breakout board for easier prototyping.

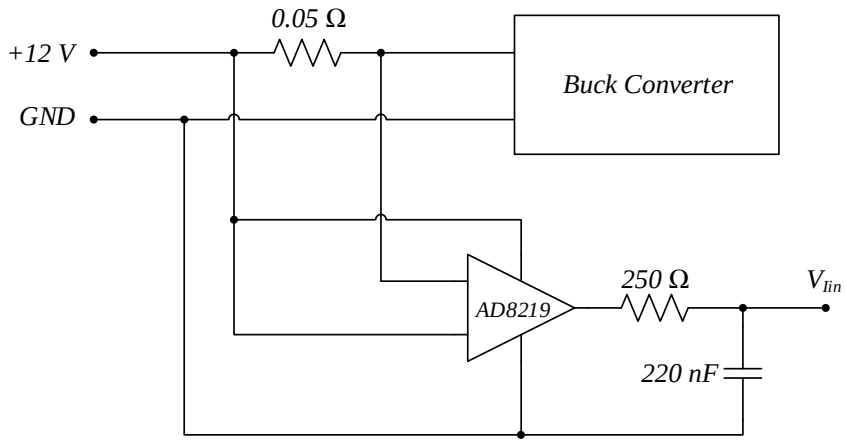


Figure 4.3: High-side current sensing for calculation of the power efficiency.

4.2 Software and Algorithm

The control law of the buck converter is essential to regulate the output voltage. This loop, depicted in Figure 4.4, consists of the following sequence that is executed every 20 μs by the microcontroller: after sensing V_{out}, the voltage error in comparison to the setpoint V_{set} is calculated and fed to a PI controller which outputs a new duty cycle value (Equation 4.2). This new t_{on} is converted to the values that are written in the compare registers. These are used by a timer to set or clear the PWM signal. Additionally, t_{on} is used to optimize the dead times $t_{d,r}$ and $t_{d,f}$. The changes in the PWM waveforms are effected when a so-called shadow transfer occurs. This guarantees that all edges and signals are output simultaneously.

$$\begin{cases} I(k) = K_i \cdot e + I(k-1) \\ U(k) = K_p \cdot e + I(k) \end{cases} \tag{4.2}$$

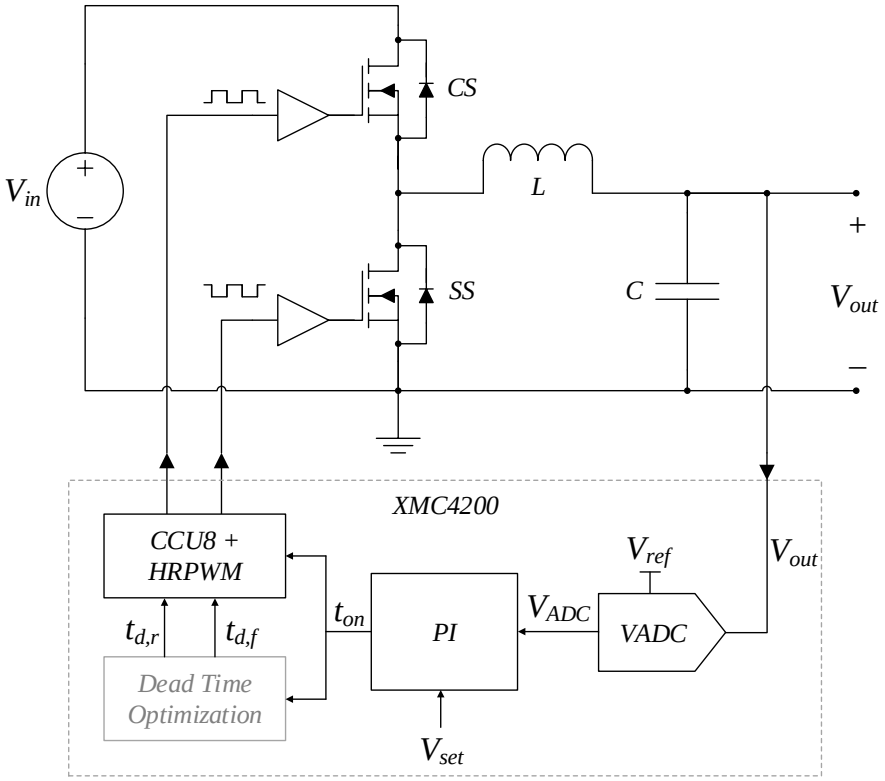


Figure 4.4: The buck converter and its control law, realized on a XMC4200 microcontroller.

The dead time, that is originally fixed at 200 ns, is nothing but an increase of the blocking time t_{off} of the synchronous switch PWM waveform. Therefore, setting a dead time for both rising and falling edge implies changing the values of the same registers that are used for setting the duty cycle. In Figure 4.5 it is possible to see how the PWM waveforms are generated: the timer is compared to registers CR1S and CR2S in order to trigger the set and clear signals. This generates

the low resolution PWM which is sent to the HRC unit. Here, registers SCR1 and SCR2 are used to shift both edges. Each unitary increment in registers CR1S and CR2S adds 12.5 ns while each increment in registers SCR1 and SCR2 adds 150 ps. Two CCU8 and two HRC units (marked as x and y) were used (with only direct output) in order to be able to set independent high resolution dead times, which would not be possible if only one channel of each module was used (with both direct and complementary output).

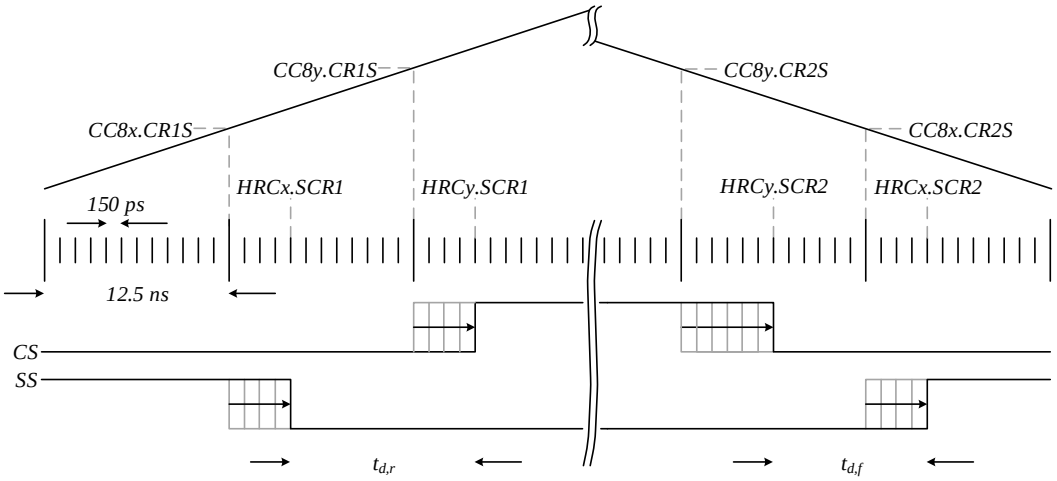


Figure 4.5: Generation of an High Resolution PWM waveform.

Now that it is clear how the dead time can be changed for both rising and falling edge, let us understand the dead time optimization algorithm that was created.

The algorithm aims at the minimization of duty cycle (or conduction time) t_{on}. As seen in Chapter 3, the optimal dead time causes the minimum duty cycle and the smallest loss of power on the body diode. If the duty cycle is higher than the minimum while the dead time is longer than the optimum, there is body diode conduction, whereas if the duty cycle is higher than the minimum while the dead time is shorter than the optimum, there is short circuit. This relation can be visualized in Figure 4.6.

In order to notice small variations of the average duty cycle, it is necessary to filter them out. This eliminates the natural variations that are constantly occurring in order to keep the voltage regulated. For that purpose a simple exponential moving-average low-pass filter was implemented as

$$DC[n] = DC[n-1] + \frac{1}{N} \cdot (x - DC[n-1]), \tag{4.3}$$

where x is the duty cycle calculated on the current iteration of the voltage control loop and $1/N$ is the weighting factor given to the most recent samples of the filter: a large N implies a smoother but slower filtering, while a small N implies a noisier but faster response [23]. In this case, $N = 128$, which is large enough to provide a steady duty cycle value.

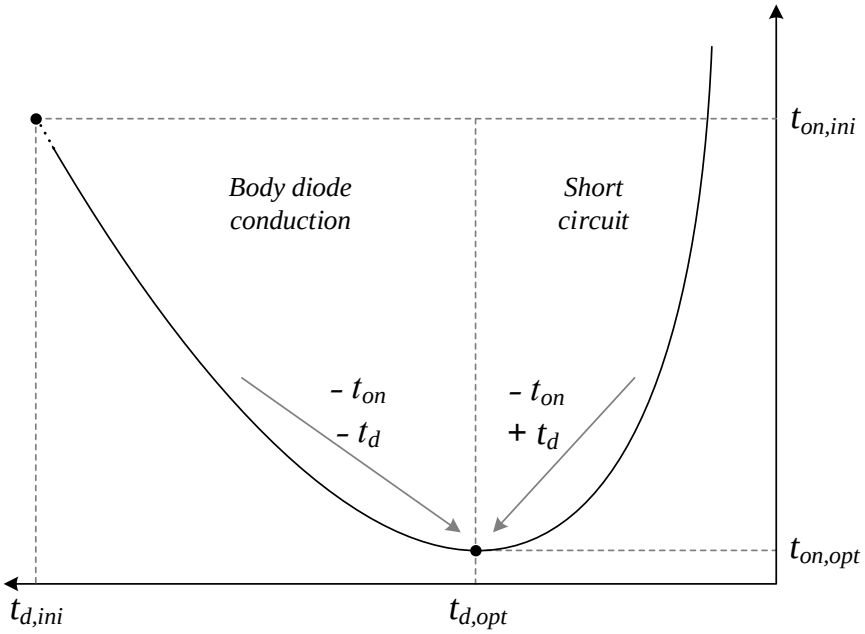


Figure 4.6: The illustrative relation between dead time and duty cycle.

The filter is updated in every iteration of the voltage control law. Whenever a load change is detected (sensed by a change in the filtered duty cycle), the algorithm waits for the transient to be over and starts optimizing the dead time from the predefined fixed value (200 ns). This duty cycle threshold value can be set by the user but it is important that the algorithm is not retriggered when the load is kept fixed: this would not bring any further optimization and would only disturb the voltage regulation. For the practical implementation, the selected duty cycle variation value was 0.5 %.

After being triggered, the algorithm starts by optimizing the rising edge dead time. When it is finished, it repeats the same procedure for the falling edge dead time (Figure 4.7 depicts the algorithm for the optimization of one edge). In each iteration, the variations of duty cycle ΔDC and dead time Δt_d are determined. If ΔDC is smaller than a threshold ε, defined by the user, the algorithm will stop optimizing. Otherwise, the signs are evaluated and compared:

$$\begin{cases} \text{sign}(\Delta DC) = \text{sign}(\Delta t_d) \implies t_d < t_{d,opt} \implies \text{diode conduction region} \\ \text{sign}(\Delta DC) \neq \text{sign}(\Delta t_d) \implies t_d > t_{d,opt} \implies \text{short circuit region} \end{cases} \tag{4.4}$$

If the dead time value of the current iteration is at the body diode conduction region, the dead time should be reduced. Conversely, if it is at the short circuit region, the dead time should be increased. Refer to Figure 4.6 to better understand these two regions. Variable dir keeps track of the direction of optimization. If it changes, the step that is subtracted or added to the current dead time value is divided by two, in order to thin out the optimization. The initial dead time step was

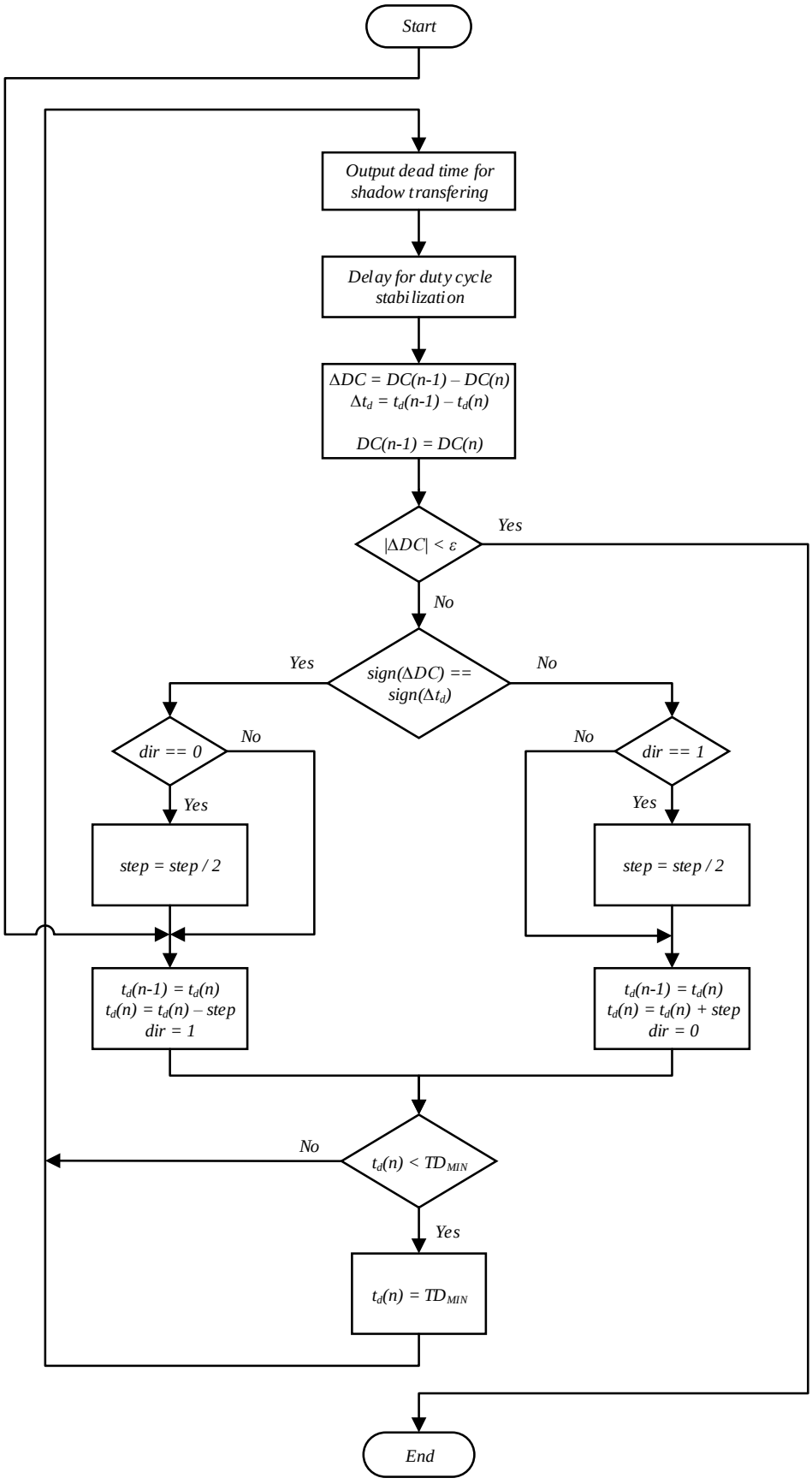


Figure 4.7: Flowchart of the sensorless dead optimization algorithm.

set to 25 ns but it can be changed for other applications. The duty cycle value varies similarly to a ball falling in a round container, bouncing left and right reaching smaller and smaller heights, until it stops at the bottom.

An interesting side note is to understand that ε, the minimum duty cycle variation in order to keep the algorithm running, is equal to the minimum possible duty cycle variation $\Delta t_{on,min}$, derived in Section 3.2.

Another feature is the existence of a user defined minimum dead time. This prevents that when the step is still large (has not been divided yet), the dead time is set to a value well inside the short circuit region, guaranteeing the integrity of the hardware. For this specific application, the minimum was set at 25 ns, slightly into the short circuit operation region.

After an iteration of the algorithm is finished and the current dead time is selected, the delay is armed, the program returns to the control routine and shadow transfers the values of duty cycle and dead time in the registers. The optimization algorithm is only resumed after a certain number of control loop executions, when the delay is over, in order to allow the duty cycle to settle and the filter to obtain the new average value.

4.3 Measurements and Results

The platform that was used to write the code, configure the peripherals, program the microcontroller and debug the project was DAVE3, the integrated development environment for XMC microcontrollers by Infineon Technologies. One of the most useful and interesting features is called xSPY, a plug-in that enables users to visualize data in real time and create graphic interfaces to control and interact with the running application. It is also possible to export data for .csv files in order to plot variables in MATLAB. xSPY was the way that most measurements were produced. Figure 4.8 shows the used interface while the converter is running. It is possible to use a virtual oscilloscope to watch the duty cycle and log the data, as well as manually set dead time values and trigger the optimization algorithm.

With the help of this plug-in it was possible to measure how quick the algorithm is. In Figure 4.9, every unity in the time domain corresponds to 500 μs, the streaming frequency of xSPY. This means that the whole optimization takes roughly 80 ms. It is also possible to observe in the middle and end of the optimization how the algorithm slows down. This corresponds to the time when the minimum dead times are being searched more finely. In Figure 4.10 it is possible to observe a very slow run of the optimization algorithm. The blue line corresponds to the duty cycle while the red represents first the rising edge dead time and second the falling edge dead time. It is interesting to notice that the peaks of duty cycle, corresponding to when the dead time is inside the short circuit region, are small. This demonstrates how the safety of the powertrain is kept at all times by only allowing the dead time to "probe" the short circuit region. At full speed, these peaks are even shorter and the hardware is not compromised.

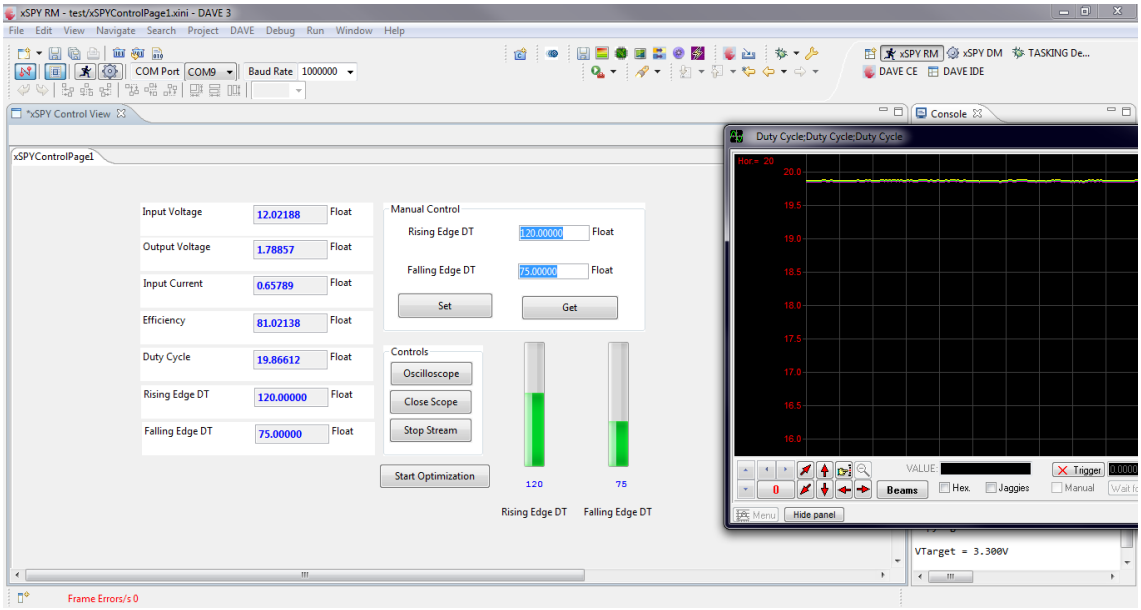


Figure 4.8: Screenshot of the xSPY interface designed to control and debug the algorithm.

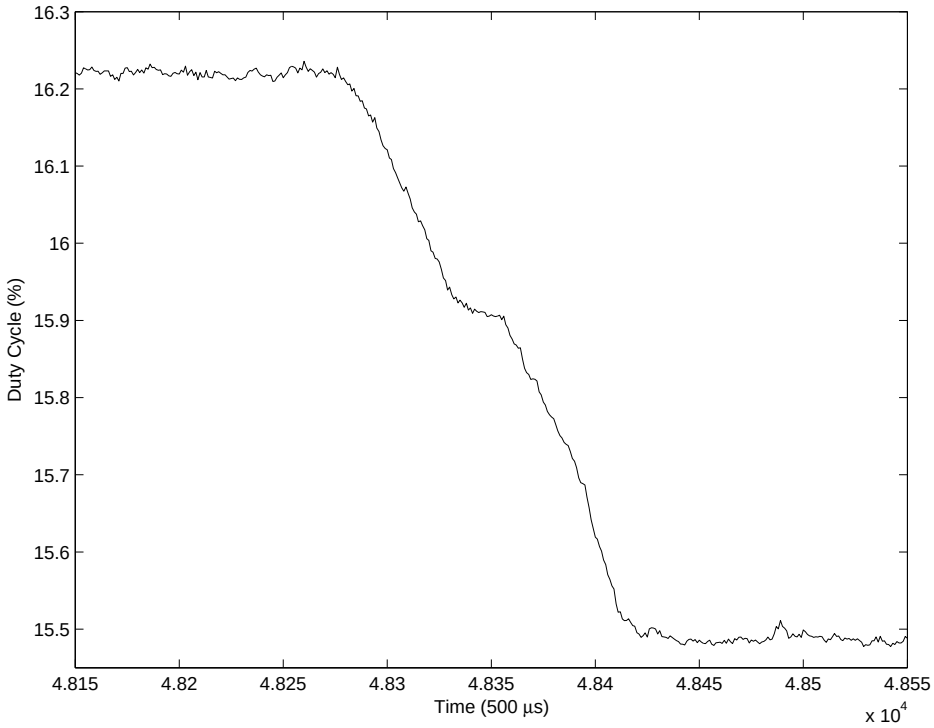


Figure 4.9: Duty cycle variation after manually triggering the algorithm. Running time is 80 ms.

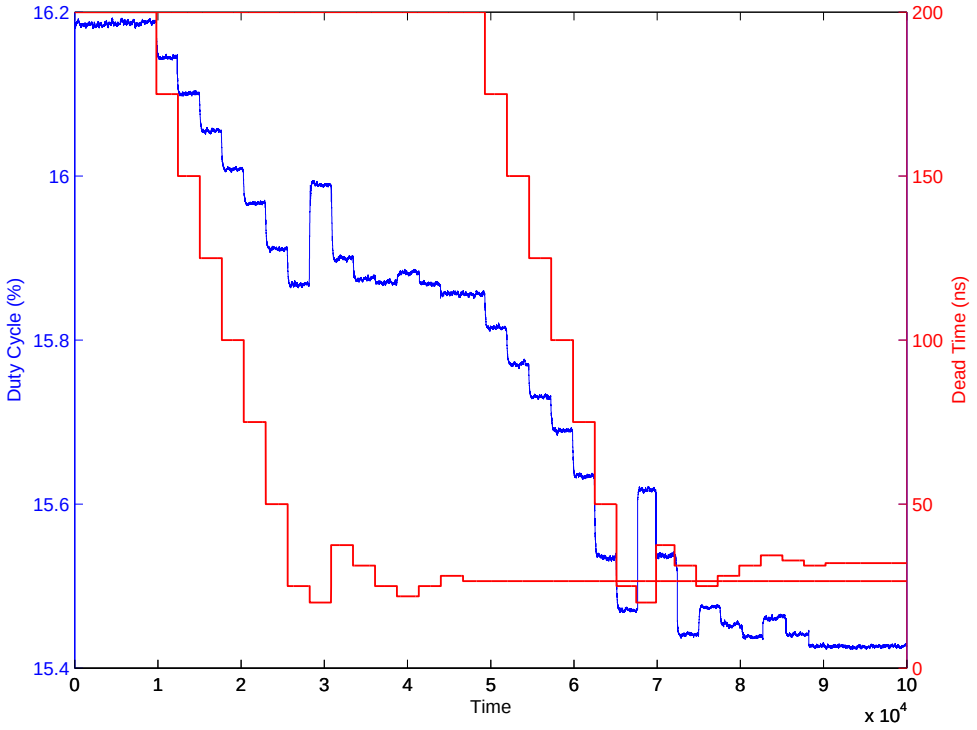


Figure 4.10: Duty cycle response to changes in rising edge dead time followed by falling edge dead time.

The algorithm was implemented in two versions: with and without HRPWM. This way it was possible to test and prove what was theoretically predicted in Section 3.4. For all calculations from now on, it is assumed that $V_{in} = 12\,\text{V}$, $V_D = 0.8\,\text{V}$ [24], $f_s = 320\,\text{kHz}$, $V_{ref} = 3.3\,\text{V}$, $V_{set} = 1.8\,\text{V}$ and $t_{d,i} = 200\,\text{ns} + 200\,\text{ns}$, which leads to $P_{loss,i} = 368.6\,\text{mW}$. The microcontroller defined dead time is slightly different than the real dead time. Appendix A addresses this topic and contains the calculation of the real dead time, based on the time parameters of the microcontroller, gate drivers and MOSFETs.

At low resolution, the minimum timer variation is 12.5 ns which is equivalent to 8 bits. The ADC has a resolution of 12 bits. Using Equations 3.7 and 3.22, it is possible to determine the theoretical operation point and efficiency improvement factor.

$$\Delta t_{d,min} = 187.5\,\text{ns}, \quad \Delta t_{on,min} = 12.5\,\text{ns}, \quad \Delta V_{out,min} = 48\,\text{mV}, \quad \gamma = 2.13, \quad \Psi = 76.5\,\% \tag{4.5}$$

The conduction time (duty cycle) was reduced from 505.9 ns to 486.9 ns, a variation Δt_{on} of 19 ns. The optimization algorithm stopped at

$$t_{d,r} = 25\,\text{ns}, \quad t_{d,f} = 25\,\text{ns} \tag{4.6}$$

The theory predicts that the optimization would result in a Δt_d of $2 \cdot 187.5\,\text{ns}$ but the algorithm

only achieves $2 \cdot 175\,\text{ns}$. This limitation happens because the algorithm does not allow dead times under 25 ns. Additionally, it is expected a Δt_{on} of $2 \cdot 12.5\,\text{ns}$ but the algorithm only achieves 19 ns. The fact that this is possible is due to the implemented duty cycle filter. If the duty cycle is quickly varying between x and $x + \Delta t_{on}$ the algorithm can actually use a value in between and not only on the extremes.

The measured input current decreased from 690 mA to 668 mA. This means that the eliminated power losses were 264 mW, 72 % of the initial body diode losses. Moreover, measurements showed that the microcontroller has an input current of approximately 100 mA. In these conditions, the system efficiency raises from 78.3 % to 80.8 % while the converter efficiency is brought up from 91.5 % to 95.1 %.

At high resolution, the minimum timer variation is 150 ps which is equivalent to 14.3 bits. The ADC still has a resolution of 12 bits. Again, it is possible to determine the theoretical operation point and efficiency improvement factor.

$$\Delta t_{d,min} = 3.125\,\text{ns}, \quad \Delta t_{on,min} = 208\,\text{ps}, \quad \Delta V_{out,min} = 800\,\mu\text{V}, \quad \gamma = 128, \quad \Psi = 99.6\,\% \tag{4.7}$$

The conduction time (duty cycle) reduced from 510.7 ns to 482.1 ns, a variation Δt_{on} of 28.6 ns. The optimization algorithm stopped at

$$t_{d,r} = 27.5\,\text{ns}, \quad t_{d,f} = 31.25\,\text{ns} \tag{4.8}$$

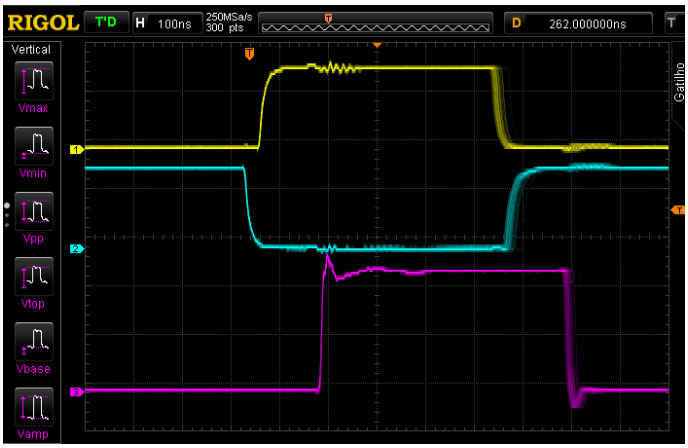
The measured input current decreased from 692 mA to 662 mA. This means that the eliminated power losses were 360 mW, 98.6 % of the initial body diode losses. The input current measurements indicate that the system efficiency raises from 78.0 % to 81.6 % while the converter efficiency is brought up from 91.2 % to 96.1 %.

Table 4.2: Experimental results using both low and high resolution PWM.

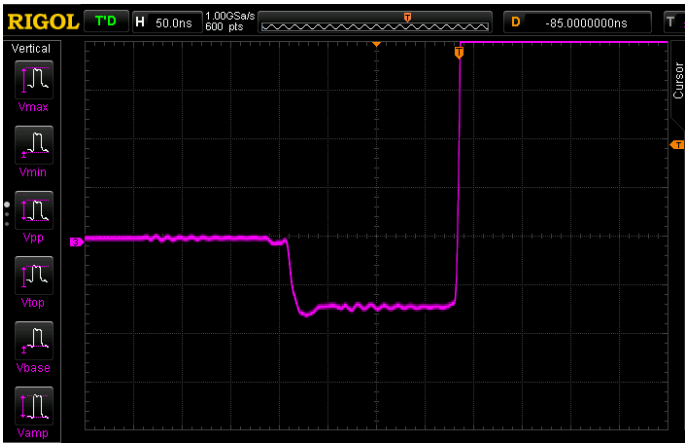
N_{ADC}	N_{timer}	$t_{d,r,opt}$	$t_{d,f,opt}$	$P_{D,loss}$ eliminated	System μ improvement	Converter μ improvement	Converter final μ
12	8	25 ns	25 ns	72 %	2.5 %	3.6 %	95.1 %
12	14.3	27.5 ns	31.25 ns	98.6 %	3.6 %	4.9 %	96.1 %

A 4-channel oscilloscope was also used to observe the DPWM signals for both synchronous and control switch and also the switching node voltage, where the diode conduction can be observed. Figure 4.11a depicts the waveforms before the optimization, with a fixed dead time of 200 ns and Figure 4.11b shows the same signals after the optimization. It is interesting to notice that the falling edge of the switching node voltage is blurred like the signal is "shaking". This is consequence of the control loop constantly changing the duty cycle in order to keep the output voltage regulated. Figure 4.11c is a close-up on the time interval when the body diode conducts.

(a)



(b)



(c)

Figure 4.11: Control switch, synchronous switch and switching node voltage: (a) Fixed dead time of 200 ns, (b) after the optimization algorithm, (c) body diode conduction.

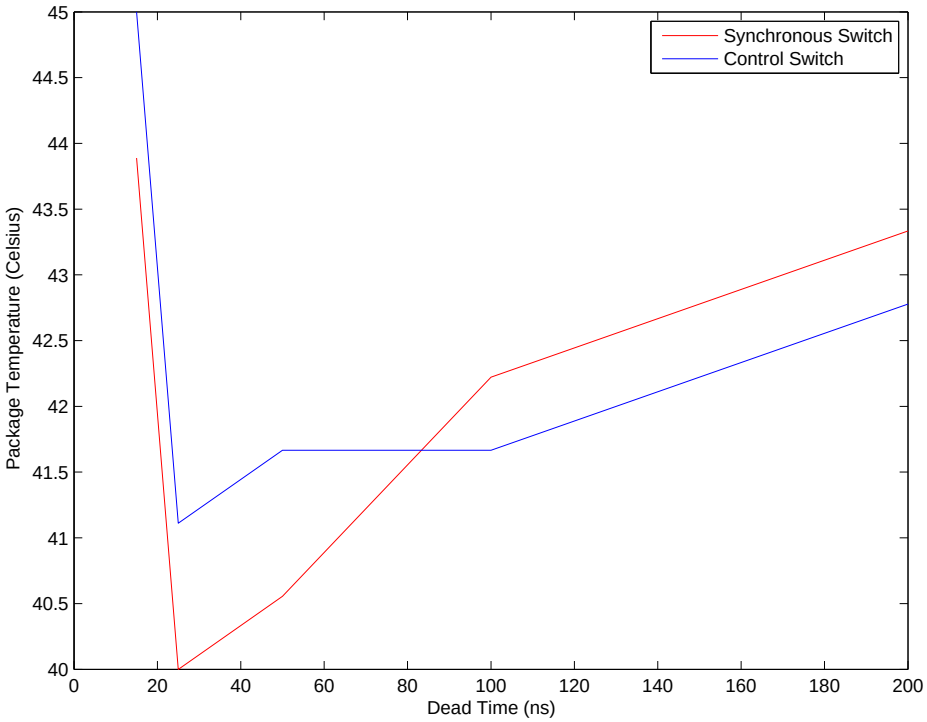


Figure 4.12: Relation between temperatures of control and synchronous MOSFETs and dead time.

Lastly, the temperatures of the MOSFETs were measured to provide further evidence of the optimization capability of the algorithm, as seen in Figure 4.12. The measurements were taken in intervals of 50 ns, starting in 200 ns, due to the low resolution of the available thermometer. An extra measurement was performed with 25 ns, near the optimum dead time, and 15 ns, in short circuit operation. It is noticeable that the synchronous switch, where the body diode conducts and causes power losses, is the hotter when the dead time is large and also cooler when the dead time is near the optimum. The temperature of the control switch increases because, even though its body diode is not conducting, an increase in dead time also causes an increase in conduction time, which heats up the transistor. Unsurprisingly, the temperature during short circuit operation peaks. This measurement was only possible because the MOSFETs in this prototype board are over-dimensioned and short circuits are not a potential threat.

4.4 Simulations

A model in Simulink was developed using SimScape components. This toolbox is used in other works in the area of power electronics ([17], [21], [19]) because it models the electrical parameters of components fairly well. These parameters, such as the ESR of the inductor, the $R_{DS,on}$, C_{oss} and C_{iss} of the MOSFETs or the Q_{rr} of the diodes, among others, were taken from the components datasheets and the bill of materials provided by Infineon for this kit ([24], [25], [26], [27]). The simulation was run for different combinations of N_{ADC} and N_{timer}. A summary

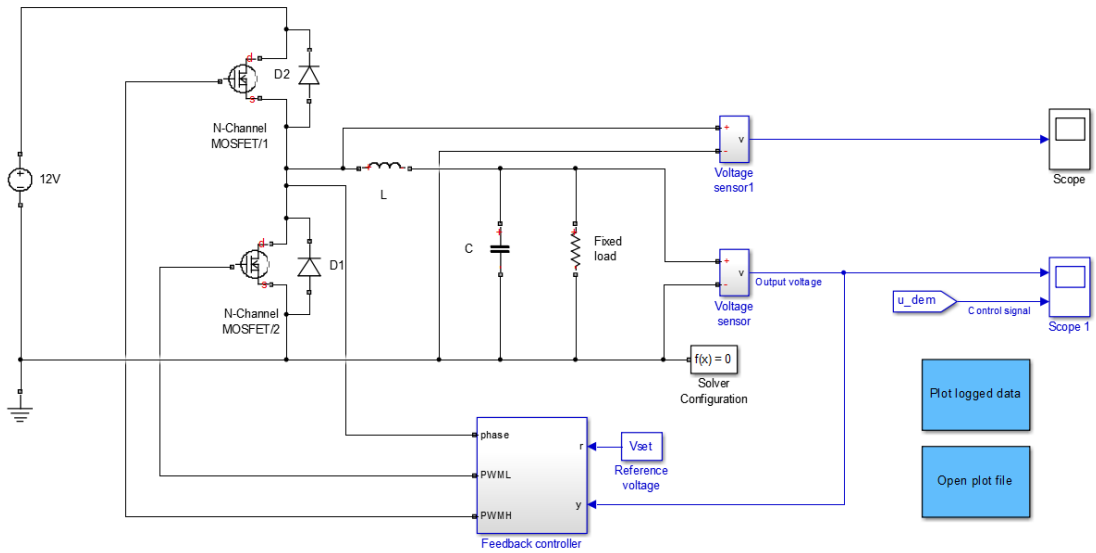


Figure 4.13: Simulation model used in Simulink.

of the results is depicted in Figure 4.14 where several curves for different N_{ADC} values are traced. The full results can be seen in Appendix B. In general, the results are close to the expected, with the efficiency improvement factor Ψ being slightly smaller. This seems consistent with the experimental results which also yielded lower Ψ values (Section 4.3).

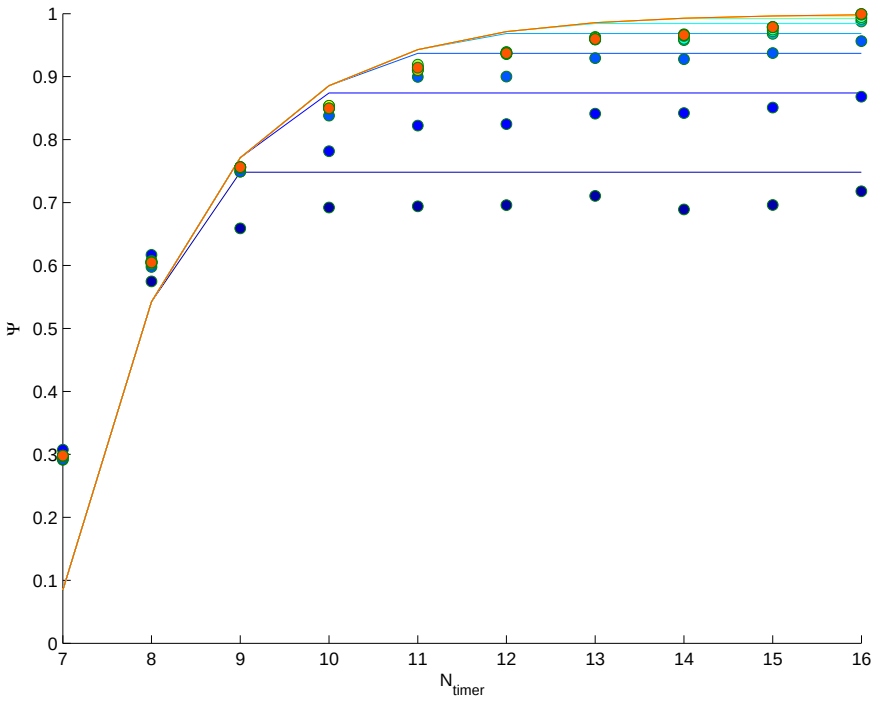


Figure 4.14: Simulation results for different N_{ADC} curves. For each color, the dots represent the simulation while the lines describe the theoretical values.

Chapter 5

Conclusions and Future Work

In this last chapter, a balance to the work presented in this dissertation is made, as well as some suggestions for eventual future works.

5.1 Fulfillment of the Objectives

The objective of this work was the analysis of the influence of timer resolution in the dead time optimization of synchronous buck converters, as well as the design of an optimization algorithm that would prove this impact.

First of all, a literature review of the topic of dead time optimization was done. Different methods and approaches were collected and compared to each other. There are many publications about this topic, however this is the first one (to my knowledge) that uses a microcontroller, in contrast with the traditional use of FPGAs, DSPs or integrated circuitry. This is a novelty in the subject of dead time optimization.

Then a theoretical analysis of the operation of a synchronous buck converter was made, as well as a study of the relation between the characteristics of a digital controller and the possible optimization of power losses caused by the addition of dead time. The author found no other publications as focused on the role of the timer resolution in dead time optimization. This analysis is a novelty and might bring some contributions to the state-of-the-art. It allows the prediction of the efficiency improvement due to the increase of the number of bits in the timer.

Next, based on the previously performed literature review, a general approach for the algorithm design was selected: the sensorless method was chosen for its advantages. We set out to develop an algorithm that was simple, computationally light, needed no additional hardware and yielded good results. Next the concept was implemented and tested in a microcontroller and converter board. The algorithm presented very good results: up to $\sim 5\%$ converter efficiency improvement from a standard fixed dead time solution, guaranteeing hardware integrity, automatic triggering at load changes and enough optimization speed for low frequency varying loads. All this with practically no additional costs.

Even though the premise of duty cycle minimization is common to this work and the approach reported by Yousefzadeh *et al.* [19], the algorithm itself has some differences:

- Step size is fixed and smaller: the approach by Yousefzadeh *et al.* uses a dead time step of 10 ns, while this implementation initially uses a large step of 25 ns that is decremented. This assures that the algorithm selects a dead time much closer to the real optimum;
- Only one optimization direction: the algorithm in [19] only optimizes in the direction of decreasing dead time. In the presented algorithm, subsearches in the opposite direction, using a smaller dead time step are also conducted;
- Stop condition: the condition for breaking the algorithm in [19] is an increase in duty cycle. This might be a problem with such a small step size, because the algorithm can be stuck in local minima right in the beginning. The chances of skipping eventual minima are higher for this algorithm because the initial step size is larger;
- Timer and ADC resolution are smaller: since the ADC resolution is smaller, changes in the output voltage are less likely to trigger a change in the duty cycle, which also varies in larger steps due to the low timer resolution. This causes the duty cycle signal to change less frequently due to the output voltage control, which allows the duty cycle variations due to a change in dead time to be more easily picked up. However, the possible duty cycle variations are also larger in [19], compromising the effectiveness of the algorithm;
- Speed of the optimization: this aspect is also related to the previous one. If changes in duty cycle are easier to detect, a low pass filter is not necessary and the stabilization of new duty cycle values is almost immediate, making the algorithm faster.

The experimental results also confirmed that the impact of timer resolution on a synchronous buck converter may go beyond the usual application on a voltage control loop: when used in the optimization of dead time, an increase in timer resolution will bring up the efficiency. In the proof-of-concept, an increase of 1% was achieved. This may be a decisive factor on the design of a power conversion solution: the cost of added bits on the timer, can be compensated by the energy efficiency gain.

Overall, it can be said that the objectives of this dissertation were successfully completed.

5.2 Future Work

This work can be improved in many ways. In this section, some suggestions are left for future developments over the work presented in this document.

- **Use input current instead of duty cycle**: even though this might imply extra hardware, using the input current as the minimization variable, instead of the duty cycle, might be an interesting approach. In this approach, the resolution of the ADC would have a different impact than in this implementation;
- **Improve the speed of the algorithm**: 80 ms is quite slow for fast varying loads such as processors. A suggestion of a future development would be to work on the filtering of the duty cycle and settling delays, in order to improve the speed of the algorithm;

- **Include second order effects on the analysis**: the theoretical study of the buck converter and the dead time optimization does not take second order effects into consideration, such as ringing on the switching of the MOSFETs. However, these effects are present and can affect the optimization;
- **Avoid local minima**: The algorithm designed in this work did not take local minima into account. The optimization might end up "stuck" far away from the true optimum;
- **Extend this work to other converter topologies**: there are multiple topologies of SMPS only in DC/DC conversion. The work presented in this paper might be adaptable to some of these other topologies or even to other power converter types.

Appendix A

Calculation of Real Dead Time

This appendix contains the calculation of the real dead time for a certain microcontroller-defined dead time. The signal has to pass through the microcontroller output pins, gate driver and MOSFET, with certain delays that are different for both rising and falling edge. The consequence is a deviation from the defined dead time.

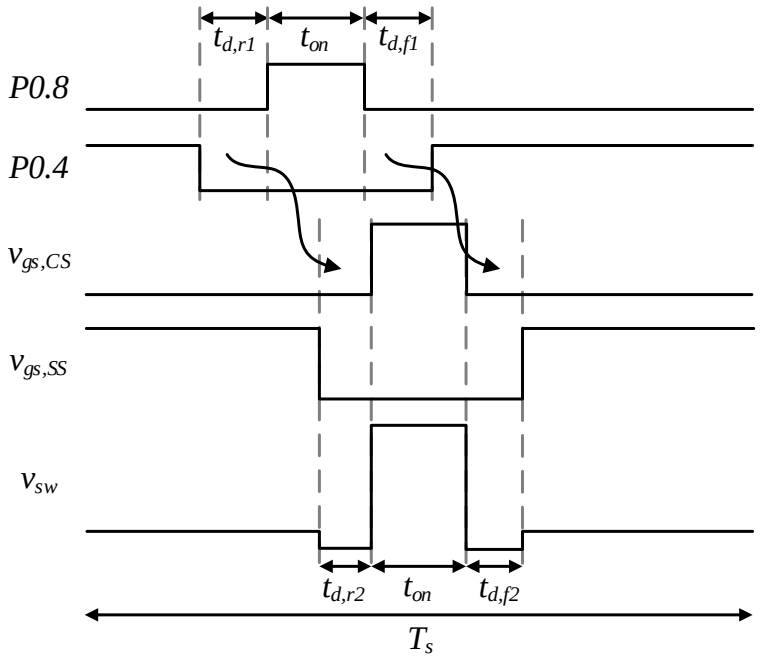


Figure A.1: Representation of the delay between the signal out of the microcontroller pins and the switching node voltage.

Firstly, the parameters that influence the timing of the signals must be fetched from the datasheets of the microcontroller [4], gate driver [25] and MOSFET [24]. The parameters are organized in table A.1.

Table A.1: Time parameters that affect the signal.

Parameter	Component	Value
Fall time, strong pad, soft edge	MCU	16 ns
Rise time, strong pad, soft edge	MCU	16 ns
Input to output propagation delay OFF	Gate driver	105 ns
Input to output propagation delay ON	Gate driver	100 ns
Fall time	Gate driver	20 ns
Rise time	Gate driver	18 ns
Turn-off delay time	MOSFET	11 ns
Turn-on delay time	MOSFET	2.5 ns
Fall time	MOSFET	1.8 ns
Rise time	MOSFET	2.2 ns

Using these parameters the delay for each edge can be calculated, for a defined dead time of 200 ns.

$$delay_{falling} = t_{strongpad,softedge} + t_{G,PDOFF} + t_{G,fall} + t_{Q,d(OFF)} + t_{Q,fall}$$

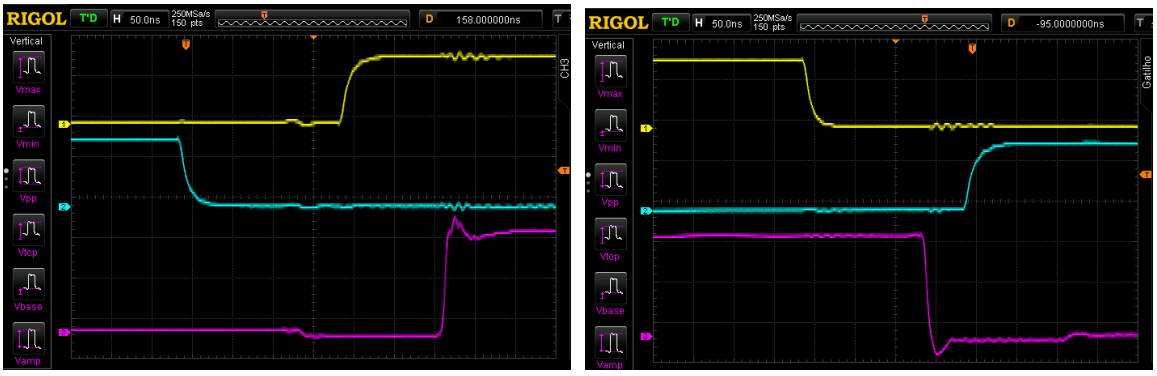
$$delay_{falling} = 16 + 105 + 20 + 11 + 1.8 = 153.8\,\text{ns}$$

$$delay_{rising} = t_{strongpad,softedge} + t_{G,PDON} + t_{G,rise} + t_{Q,d(ON)} + t_{Q,rise}$$

$$delay_{rising} = 16 + 100 + 18 + 2.5 + 2.2 = 138.7\,\text{ns}$$

$$t_{d,r,real} = t_{d,f,real} = t_{d,defined} + delay_{rising} - delay_{falling} = 184.9\,\text{ns}$$

The following oscilloscope captures confirms the calculations for both falling and rising edge.



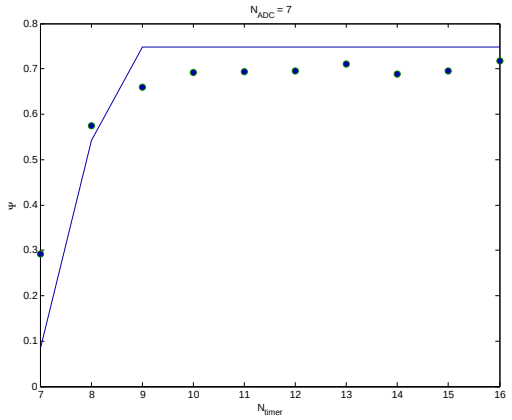
(a) (b)

Figure A.2: Oscilloscope print of the difference between defined and real dead time for both edges. The two signals are measured at the output pins of the microcontroller while the bottom one is measured at the swithing node voltage.

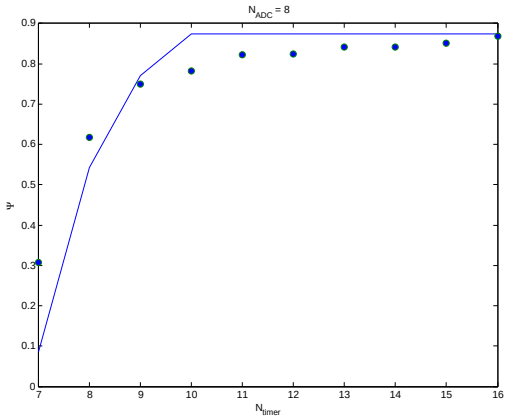
Appendix B

Results of the Simulation

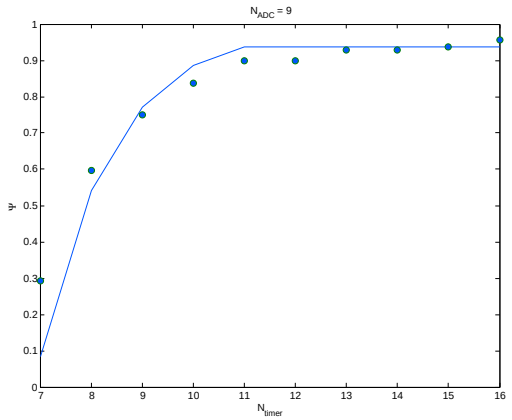
This appendix contains the plots with the results of the simulations performed in Simulink. Each plot represents a different number of bits on the ADC, with the x-axis describing the bits of the timer and the y-axis describing the percentage of body diode losses elimination. The lines are the theoretical values while the crosses show the simulated results.



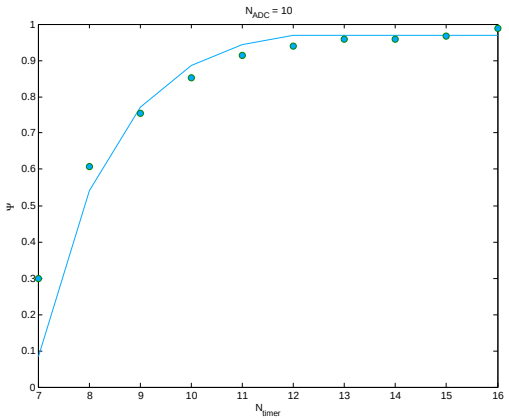
(a)



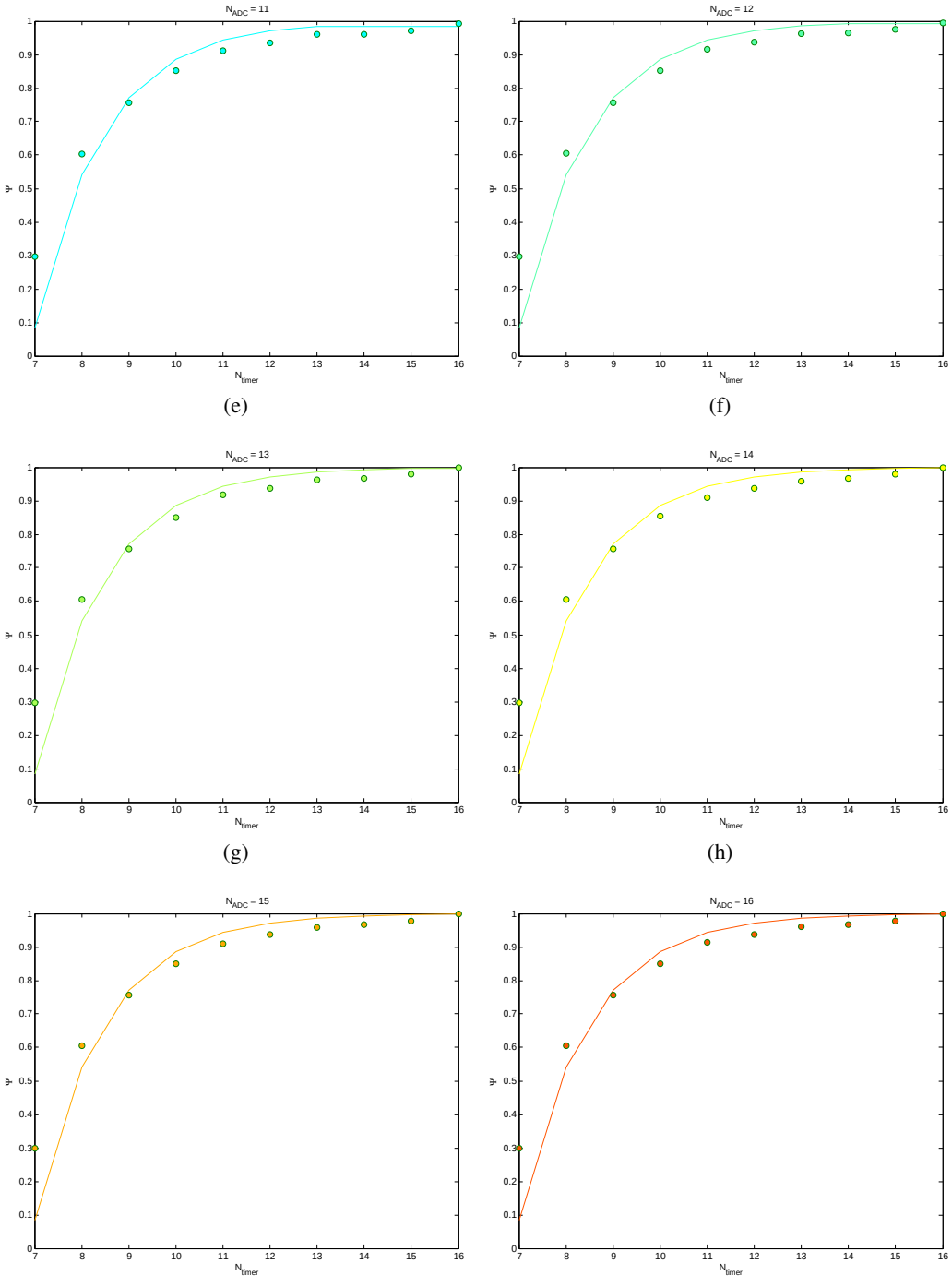
(b)



(c)



(d)



(e) (f) (g) (h) (i) (j)

Figure B.1: Results of the simulation for N_{ADC} from 7 to 16.

Bibliography

[1] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications and Design*, 3rd ed. John Wiley & Sons, Inc., 2003.

[2] P. Markowski, "Estimating MOSFET switching losses means higher performance buck converters," *EETimes*, 2002.

[3] R. Srinivasan and V. P. Ranjan, "Pulse skipping modulated DC-DC converter modeling and simulation," *Journal of Electrical and Electronics Engineering*, vol. 3, no. October, pp. 195–198, 2010.

[4] Infineon Technologies AG, "XMC4200 Reference Manual," 2014.

[5] B. Ramachandran, "Dynamic Operation of Sensorless Dead-time Optimization in Digitally Controlled Synchronous Buck Converters," Ph.D. dissertation, University of Colorado, 2007.

[6] Infineon Technologies AG, "ICE2QS03G Datasheet," 2014.

[7] J. Melanson, "Adaptive dead time control for switching circuits," 2001.

[8] H. W. Huang, C. Y. Hsieh, K. H. Chen, and S. Y. Kuo, "Load dependent dead-times controller based on minimized duty cycle technique for DC-DC buck converters," *PESC Record - IEEE Annual Power Electronics Specialists Conference*, pp. 2037–2041, 2007.

[9] H. Forghani-zadeh and G. Rincon-Mora, "Current-sensing techniques for DC-DC converters," *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, vol. 2, 2002.

[10] S. Mappus, "Predictive Gate Drive Boosts Synchronous DC / DC Power Converter Efficiency," *Texas Instruments Application Report*, no. April, pp. 1–26, 2003.

[11] P. T. Krein and R. M. Bass, "Autonomous control technique for high-performance switches," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 3, pp. 215–222, 1992.

[12] W. L. W. Lau and S. Sanders, "An integrated controller for a high frequency buck converter," *PESC97. Record 28th Annual IEEE Power Electronics Specialists Conference. Formerly Power Conditioning Specialists Conference 1970-71. Power Processing and Electronic Specialists Conference 1972*, vol. 1, 1997.

[13] B. Acker, C. Sullivan, and S. Sanders, "Synchronous rectification with adaptive timing control," *Proceedings of PESC '95 - Power Electronics Specialist Conference*, vol. 1, 1995.

[14] A. Zhao, A. A. Fomani, and W. T. Ng, "One-step digital dead-time correction for DC-DC converters," *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, pp. 132–137, 2010.

[15] L. Mei, D. Williams, and W. Eberle, "A Synchronous Buck Converter Using a New Predictive Analog Dead-Time Control Circuit to Improve Efficiency," *Canadian Journal of Electrical and Computer Engineering*, vol. 36, no. 4, pp. 181–187, 2013.

[16] O. Trescases, W. T. N. W. T. Ng, and S. C. S. Chen, "Precision gate drive timing in a zero-voltage-switching DC-DC converter," *Power Semiconductor Devices and ICs, 2004. Proceedings. ISPSD '04. The 16th International Symposium on*, 2004.

[17] J. Abu-Qahouq, H. M. H. Mao, H. Al-Atrash, and I. Batarseh, "Maximum Efficiency Point Tracking (MEPT) Method and Digital Dead Time Control Implementation," *IEEE Transactions on Power Electronics*, vol. 21, no. 5, pp. 1273–1281, 2006.

[18] A. Pizzutelli, A. Carrera, M. Ghioni, and S. Saggini, "Digital dead time auto-tuning for maximum efficiency operation of isolated DC-DC converters," *PESC Record - IEEE Annual Power Electronics Specialists Conference*, pp. 839–845, 2007.

[19] V. Yousefzadeh and D. Maksimović, "Sensorless optimization of dead times in dc-dc converters with synchronous rectifiers," *IEEE Transactions on Power Electronics*, vol. 21, no. 4, pp. 994–1002, 2006.

[20] T. Reiter, D. Polenov, H. Pröbstle, and H. G. Herzog, "PWM dead time optimization method for automotive multiphase DC/DC-converters," *IEEE Transactions on Power Electronics*, vol. 25, no. 6, pp. 1604–1614, 2010.

[21] A. V. Peterchev and S. R. Sanders, "Digital multimode buck converter control with loss-minimizing synchronous rectifier adaptation," *IEEE Transactions on Power Electronics*, vol. 21, no. 6, pp. 1588–1599, 2006.

[22] P. Haaf and J. Harper, "Understanding Diode Reverse Recovery and its Effect on Switching Losses," *Fairchild Power Seminar*, pp. 23–33, 2007.

[23] H. Tajiri and T. Kumano, "Input Filtering of MPPT Control by Exponential Moving Average in Photovoltaic System," *IEEE International Conference on Power and Energy (PECon)*, no. December, pp. 372–377, 2012.

[24] Infineon Technologies AG, "BSZ060NE2LS Datasheet," 2013.

[25] ——, "1EDI60N12AF Datasheet," 2014.

[26] Bourns, “SRF1280 Datasheet,” 2011.

[27] Infineon Technologies AG, “XMC Buck Converter Kit - Bill of Materials,” 2014.