DESIGN AND MODELING
OF A DIGITAL CONTROLLER
FOR MULTI-MODE DC-DC CONVERTERS

Coordinatore del corso:  
Chiar.mo Prof. Roberto Vescovo  

Dottorando:  
Ing. Marco Meola  

Relatore:  
Chiar.mo Prof. Sergio Carrato  

Correlatore:  
Ing. Emanuele Bodano  

ANNO ACCADEMICO 2008-2009
Abstract

In many different applications, where battery operated devices are involved, dc-dc converters with high efficiency over the whole range of their load values are required. In particular, optimization of the efficiency of these converters when the load current is low, i.e. at light load, has become one of the most challenging issues in designing dc-dc converters. In multi-mode dc-dc converters the controller implements different control strategies according to the output current demand, in order to keep the efficiency as high as possible over the whole range of load values. In this context, control of Switched Mode Power Supplies (SMPS) has been traditionally achieved through analog means with dedicated integrated circuits (ICs). However, as power systems are becoming increasingly complex, the classical concept of control has gradually evolved into the more general problem of power management, demanding functionalities that are hardly achievable in analog controllers. The high flexibility offered by digital controllers and their capability to implement sophisticated control strategies, together with the programmability of controller parameters, make digital control very attractive as an option for improving the features of multi-mode dc-dc converters. On the other side, digital controllers find their major weak point in the achievable dynamic performances of the closed-loop system. Indeed, analog-to-digital conversion times, computational delays and sampling-related delays strongly limit the small-signal closed-loop bandwidth of a digitally controlled SMPS. Quantization effects set other severe constraints not known to analog solutions. For these reasons, intensive scientific research activity is addressing the problem of making digital compensators stronger competitors against their analog counterparts in terms of achievable performances. Moreover, although digital control in power electronics is potentially able to meet the aforementioned requirements of modern power supply systems and electronic equipments, analog control ICs for power converters are still dominating
the market. Indeed, digital control for SMPS lacks of a strong and well established know-how as analog control does and, therefore, is less accessible to designers.

The work of this dissertation finds its origin in this context. The design and the simulation of digital multi-mode controllers are investigated with particular focus on the analysis and the optimization of the converter efficiency at light load. Smart power management is accomplished by the use of dedicated control strategies accordingly to the operating conditions of the converter. A digital multi-mode controller structure is presented for high efficiency low cost and low power dc-dc converter. Moreover, decision criteria for selecting among the control strategies are proposed and tested in an FPGA-based experimental prototype. Finally, the development of a discrete time large-signals model of the converter power stage, particularly suited to predict converter behavior when transitioning between the different control strategies, provides a useful tool for designing SMPS according to the above mentioned decision criteria and highlights the main design issues in the power management.
Abstract


In quest’ambito, il controllo di convertitori a commutazione è tradizionalmente ottenuto per via analogica tramite l’impiego di circuiti integrati dedicati. Tuttavia, mano a mano che i sistemi di potenza diventano sempre più complessi e spesso costituiti a loro volta da sotto-sistemi fra loro interagenti, il classico concetto di controllo si è gradualmente evoluto nella più generale tematica del power management, richiedendo funzionalità difficilmente implementabili nei controllori analogici. L’elevata flessibilità offerta dai controllori digitali e la loro predisposizione ad implementare sofisticate strategie di controllo, insieme alla programmabilità dei parametri del controllore, fanno del controllo digitale una attrante alternativa per il miglioramento delle prestazioni dei convertitori dc-dc multi-modo. Tuttavia, il punto debole più evidente di un controllore digitale risiede nelle prestazioni dinamiche a catena chiusa da esso ottenibili. I tempi impiegati per la conversione analogico-digitale della grandezza da controllare, i ritardi di calcolo così come i ritardi associati al campionamento pongono limiti severi alla massima banda di controllo ottenibile in un convertitore controllato digitalmente. Ulteriori limitazioni sono inoltre imposte dagli effetti di quantizzazione nella catena di controllo.

Per le ragioni sopra esposte, la realizzazione di controllori digitali in grado di essere competitivi (in termini di prestazioni dinamiche) rispetto alle classiche soluzioni analogiche è materia di
intensa attività scientifica nonché interesse industriale. Inoltre, sebbene il controllo digitale appa- 
pare capace di soddisfare le esigenze sopra menzionate, i convertitori dc-dc a controllo analogico 
dominano ancora il mercato. Infatti, il controllo digitale di convertitori dc-dc soffre della man-
canza del solido know-how posseduto dai controllori analogici, risultando così meno accessibile. 
Questo lavoro di tesi si inquadra nel contesto così delineato. L’attività principale svolta riguarda 
la progettazione e simulazione di controllori dc-dc a controllo digitale con l’obiettivo di studiare 
l’ottimizzazione dell’efficienza per piccole correnti di carico.

In questa tesi la struttura di un controllore digitale multi-modo per convertitori a basso costo 
e bassa consumzione potenza è presentata. Criteri decisionali sulla scelta della strategia di con-
trollo a seconda delle condizioni di carico sono proposti e testati su prototipo sperimentale di 
convertitore dc-dc dove il controllo digitale è implementato in una FPGA. Inoltre, lo sviluppo 
di un modello a larghi segnali a tempo discreto dello stadio di potenza del convertitore, studiato 
appositamente per modellizzare il comportamento del convertitore nel passaggio da una strate-
gia di controllo all’altra, costituisce un utile strumento per la progettazione di convertitori a 
commutazione sulla base dei criteri decisionali proposti, mettendo in luce le problematiche della 
progettazione di sistemi per il power management.
Acknowledgments

I would like to warmly thank my advisor Prof. Sergio Carrato and my technical advisor Ing. Emanuele Bodano for their guide and support during my PhD activity, as well as Ing. Alberto Gola for the opportunities given to me to grow up professionally joining Infineon Technologies in Padova and the University of Colorado in Boulder. A special ”Grazie” goes to the Instrumentation and Detector Laboratory at Elettra Sincrotrone Trieste that took care of me for almost five years. In particular, I want to thank Luigi Stebel for the nice time we spent together in the lab., Rudi Sergo and Christian Amasoli for all the technical teachings and Giuseppe Cautero for his support and the warm hospitality in his group.

This work and PhD fellowship have been entirely sponsored by Infineon Technologies AG., I would like to thank all the colleagues in the Padova Design Center for the nice experience I had in Padova before starting my PhD activity.

Part of my PhD activity has been carried out at the Colorado Power Electronics Center (CoPEC), University of Colorado at Boulder. It has been a real honor for me to have the chance to work for Prof. Dragan Maksimovic and Prof. Regan Zane during my nine months research period in Colorado, and a huge opportunity to grow both professionally and personally. I thank them both for their warm hospitality and their support. A big thank you goes to all the CoPEC guys and especially to Xu Zhang for his support while working together.

Finally my thanks go to my mother, who always supported me in my academic career and always encouraged me to do what I like and I am interested in, as well as to my friend Luca, for being always close to me, and my girlfriend Ingrid, who gave me all the energy I needed to complete this work.

Marco Meola
# Contents

1 Introduction ................................................. 9

2 The buck converter and its operating modes .......................... 17
   2.1 Overview of buck converter operation .......................... 17
   2.2 The Continuous Conduction Mode (PWM-CCM) .................. 19
   2.3 Discontinuous Conduction Mode (PWM-DCM) ................. 21
   2.4 The Pulse Frequency Modulation (PFM) mode of operation .... 23
   2.5 Power losses analysis in CCM and DCM ...................... 24
   2.6 Power loss in PWM-CCM ...................................... 25
      2.6.1 Conduction loss .......................................... 25
      2.6.2 Switching loss ........................................... 26
      2.6.3 Stray inductance loss .................................... 29
      2.6.4 Controller quiescent current ............................ 29
   2.7 Power loss in PWM-DCM ...................................... 29
      2.7.1 Conduction loss .......................................... 29
      2.7.2 Switching loss ........................................... 30
   2.8 Power loss in PFM .......................................... 31
   2.9 Efficiency characterization in PFM mode ....................... 31
   2.10 Dc-dc converter system design ................................ 36
      2.10.1 Digital controller system specification ................ 36
      2.10.2 Output filter design ..................................... 36
3 Multi-mode buck converter

3.1 AC equivalent modeling in PWM-CCM ........................................... 39
3.2 AC equivalent modeling in PWM-DCM ........................................... 41
3.3 AC equivalent modeling in Pulse Frequency Mode (PFM) of operation .... 44

4 Design of digitally controlled dc-dc converters .......................... 46

4.1 Delays in the digital loop ............................................................... 48
  4.1.1 Modulator delay ................................................................... 50
4.2 Design of the digital compensator ............................................... 52
  4.2.1 Discrete time equivalent system of the dc-dc converter ............... 53
4.3 Design of the digital compensator in PWM-CCM ......................... 55
4.4 Design of the digital compensator in PWM-DCM ......................... 58
4.5 Compensator structures ............................................................... 60
4.6 Serial PID architecture ................................................................. 62
4.7 Sources of quantization ............................................................... 65
  4.7.1 ADC resolution .................................................................. 65
  4.7.2 Effect of finite word length of compensator parameters .......... 66
  4.7.3 DPWM resolution ............................................................... 67
4.8 Limit cycle oscillations ............................................................... 68

5 Implementation of the digital multi-mode controller ................... 72

5.1 ADC converter ............................................................................ 74
5.2 Compensator structure ............................................................... 74
5.3 DPWM structure ........................................................................ 77
5.4 Sigma delta modulator ............................................................... 78
5.5 Digital multi mode controller structure ....................................... 80
5.6 Automatic selection of the mode of operation of the converter ....... 83
  5.6.1 Transition from PWM-CCM to PWM-DCM ......................... 83
  5.6.2 Transition from PWM-DCM to PWM-CCM ......................... 85
  5.6.3 Transition from PWM to PFM ............................................ 85
5.6.4 Transition from PFM to PWM ........................................ 85

6 Simulation of the digital multi-mode buck converter .......... 88
   6.1 Discrete time large signal model of converter power stage .......... 89
   6.2 Implementation of the discrete buck converter model ............. 93
   6.3 FPGA-based digital multi-mode buck converter prototype .......... 93
   6.4 Experimenta results .............................................. 94

7 Conclusions and future work ........................................ 104
   7.1 The continuous to digital block .................................... 119
       7.1.1 Aliasing ....................................................... 120
   7.2 The digital to continuous block .................................... 122
       7.2.1 The effect of sampling ....................................... 124
   7.3 Selection of the sampling frequency ................................ 125
   7.4 Digital compensator design ....................................... 126
       7.4.1 Design of discrete equivalents via numerical integration ....... 126
       7.4.2 Zero-pole matching equivalents ................................ 129
Chapter 1

Introduction

Control of switched mode power supplies (SMPS) intended for consumer market has been traditionally achieved by analog means. Nowadays, analog control ICs are available at low price and for a variety of power applications and converter topologies. These controllers typically integrate one or more error amplifiers, modulation circuitry, a temperature-compensated voltage reference, overvoltage/overcurrent protections as well as soft-start, standby and automatic shutdown features. Depending on the power rating, gate drivers and power switches may be integrated or left as off-chip components. Off-chip passive circuitry is used to program the controller behavior, define the shape of the compensator transfer function and provide feedback and sensing interfaces between the chip and the power converter.

With the rapid development and diffusion of battery operated portable devices, requirements such as high efficiency, tight output voltage regulation, area optimization and minimization of external components are demanded. Many analog solutions have been developed for applications such as MP3 players, PDAs, mobile phones, portable instruments, portable hard disk drivers etc. where the efficiency is kept as high as possible over a wide range of output currents by the use of dedicated control strategies that involve complex analog circuitry to properly manage the operations of the converter. Examples of available products are [1–3] where, in order to satisfy the above mentioned requirements, features such as efficiency optimization at light load, small output voltage ripple, high efficient solution for load transient optimization, high switching frequency and automatic transition between the different control strategies have been
implemented. Nevertheless, the implementation of sophisticated controllers by the use of analog solutions have come to limitations due to the hardware involved. To further improve converter performances, therefore, digital solutions have become attractive in this field.

Digital control for SMPS has started to gain popularity (at least at academic level) since the years 2000 – 2001 ([4–7]). The main advantages deriving from the implementation of a digital loop in dc-dc converter systems are represented by the high degree of programmability and computational power, the reduced need for external passive components and the consequent decreased sensitivity to tolerances and other sources of parametric variations, the possibility to implement complex control strategies as well as to easily switch through different modes of operation, targeting for highest efficiency or optimized dynamic performances. System monitoring functions are of extreme importance for high-reliability applications and their implementation strongly points to digital solutions able to collect and process environmental data. Self-tuning, also known as autotuning functions, allow a digital compensator to adapt its parameters to the specific power plant under control, eliminating the need for manual design or calibration and enhancing controller modularity and versatility.

In general, compared to the existent analog solutions, digital controlled SMPS would not able to obtain the same features, in terms of output voltage regulation, if additional functionalities and more complex control strategies were not involved, functionalities that justify the cost and the effort to be developed. In fact, digital control wins where the algorithms of the operation is too complex for analog implementations. No analog controller exhibits the same degree of programmability and versatility as a digital controller does. Compensator parameters can be stored in a nonvolatile memory and loaded in a programmable controller at system power-on. This way, different sets of pre-calculated parameters can be run for many environmental conditions on the same control hardware. More evolved tuning algorithms literally perform an automatic design of the compensator parameters through a number of online measurements and post-processing operations.

Different control laws can be adopted over time by the same control structure, depending on the particular situation. A typical example is efficiency maximization in a converter switching from heavy-load to light-load operation. Detection of this transition allows to choose the
most suitable control strategy in order to track the point of highest efficiency, this being of utmost importance for portable and battery-operated equipments. From these considerations, real breakthrough of digital control in consumer power electronics will come from cost-effective solutions featuring capabilities not available in analog ICs. The research for hardware, power and area optimization in digital ICs for SMPS control has stimulated the scientific community over the recent years, leading to the achievement of increasingly improved performances.

Although digital control of SMPS provides many interesting features it suffers from the lack of a strong and well-established know-how compared to its analog counterpart. Indeed, the research activity on this topic has drastically increased in the past few years and many research groups are currently focusing their research on digital control of SMPS but, at marketing level, only the 20-30% of the converter are digital. The reason of this is basically the lack of education on digital control, especially whereas dc-dc converters are involved, because the standard theory of sampled data system does not always apply to the control of dc-dc converters and many different topics on the design and modeling of digitally controlled dc-dc converters are being separately investigated.

As reported in the *Applied Power Electronic Specialists conference (APEC)* in 2007 (the most important conference in the power electronic field), the number of papers sessions, seminars etc. has greatly increased since the year 2000 (1.1 (a)) but the spread of digital control in the semiconductor manufacture companies has not been so popular due to the lack of educational programs (Fig. 1.1 (b)).

From the research point of view all the main contributions to digital solutions for SMPS presented in literature aim to demonstrate control approaches with minimum hardware resources and reduced complexity [4–23]. Feasibility of completely integrated digital controllers was demonstrated for the first time in [8, 9], in which innovative solutions for the main constituents of a digital controller, namely the compensator, the A/D converter and the digital pulse-width modulator, were presented. Based on a look-up table structure, the PID compensator employed in [8] presented reduced complexity. Delay-line and windowed ADCs were used in these works for fast conversion times and small area requirements. A ring oscillator-MUX DPWM was implemented in [9], while in [8] a hybrid counter/delay line architecture was con-
Figure 1.1: a) Papers, sessions, seminars, and rap sessions presented at APEC 2000-2007 in the areas of high-frequency digital SMPS control b) Research status year 2007.

Considered as a suitable trade off between resolution, area and power consumption. Further works in the area exploited ring-oscillator ADCs [11, 13] and other hybrid DPWM structures. Further examples of DPWM architectures can be found in [24–27].

Along with solutions aimed to an increasingly deeper integration, the research activity also focused on exporting control approaches widespread in the analog world into the digital domain, an example being the investigation and development of digital current-mode control techniques [16, 20].

Research background

Besides the research topics mentioned so far, one of the most intense and interesting research activities concerns the development and implementation of hardware-effective autotuning techniques for digital compensators [28–39]. Results presented in literature indeed point to a number of viable approaches for robust and repeatable autotuning. Moreover, identification approaches have been presented which allow for system health monitoring functions to be implemented [40–42], which enhance the controller with fault detection features and can as well be employed to undertake preventing actions against possible future faults. Beside purely digital control solutions, mixed-signal controllers are worth to be mentioned [43–46]. These approaches combine simple integrated analog blocks and digital provisions with the aim to achieve analog-
like dynamic performances, but still retaining the advantages of digital systems like program-
ability, low passive component count and control robustness.

The State of the Art (commercial solutions)

As a state of the art, in the marketing, two products deserve to be mentioned. The first one is a digital synchronous Point of Load PWM controller from Texas Instrument (UCD9240) [47] meant for networking equipment, telecommunication equipment, servers, etc. The controller can control up to four voltage rails and up to eight phases. This controller implements all the main features provided so far by the application of digital control to dc-dc converters. By the use of a PC-Based design tool the controller can be fully configured, simulated and the power supply performances can be monitored as well. The switching frequency of the controller is programmable in a range from 15kHz up to 2MHz and a window ADC converter is used to provide up to 1mV of close loop resolution of the output voltage. A digital programmable soft start is provided to move the output voltage inside the regulation band at the system turn on. Moreover, four DPWM with 250ps resolution are implemented in the chip and a third order compensator [22] is used to perform the adjustment of the output voltage. Transient performance are enhanced by the use of a nonlinear gain block at the output of the ADC converter, as in [48]. Autotuning is available as well, using one of the procedure presented in [33–39].

The second product worth to be mentioned is the Intersil ZL2008 synchronous digital controller with drivers integrated in the chip [49]. This digital controller is meant for telecom/datacom equipment and server/storage equipment as well as the UDC9240 with switching frequency programmable from 200kHz up to 1.4MHz and ±1% as output voltage accuracy.

One of the main features of the ZL2008 is the introduction of a Non-Linear Response block which provides a second error signal path that bypasses the primary path when the output begins to transit outside the regulation band. By the use of this non linear response loop the response time and the output voltage deviation in the event of a sudden output load current step results drastically decreased. In order to keep the efficiency as high as possible optimization of dead times is performed by a dedicated closed loop algorithm based on duty cycle minimization, as in [50]. Moreover, at light load the efficiency is optimized allowing the converter to work
in discontinuous conduction mode keeping the low side switch off for the whole length of the
switching period.

The work of this dissertation

This research activity focuses on the design and simulation of digital multi-mode controllers
targeted to automotive applications. In general, in such applications requirements on the output
voltage regulation are not as tight as the requirements demanded by portable devices such as
MP3 players, PDAs etc. Instead, the converter has to guarantee high efficiency over either a
wide range of input voltage values and output current values. To meet this requirement the
converter has to adopt different control strategies (i.e. operating modes) according to the load
value.

The main design issues of multi-mode converters regard the optimization of the converter effi-
ciency at light load, the decision criteria for determining when a control strategy has to be used
and the management of the output voltage regulation while transitioning between the operating
modes. In such converters digital control is very attractive because of its capability to implement
complex control algorithms, required to properly manage the operations of the converter in any
of its operating modes, as well as its programmability, a key point when different control strate-
gies are implemented by the controller. Moreover, one of the main features of digital control is
the testability of all the functionalities of the controller by the use of a low cost FPGA-based
experimental prototype of the converter without the need of developing a test chip, at least at
the first stages of the design.

Although digital solutions for multi-mode converters can be found in literature [11,13,17,51–55],
just a small part of the research activity on digital control of SMPS has been focused on the
design issues of these controllers, topics that are still being investigated. On one side the non-
linear behavior of the converter while transitioning from one operating mode to another limits
the investigation of the strategies for the mode management via mathematical analysis, on the
other side the lack of simple converter models that accurately predict the converter behavior in
any of its operating modes, and especially when the converter is transitioning between them,
limits the development of efficient strategies to perform automatic detection of the operating conditions of the converter.

The main objective of this research activity is to investigate the efficiency optimization of multi-mode converters at light load. Converter power loss in such load condition is first researched, deriving a control strategy for maximizing the efficiency as a function of the output current by means of an efficient output current estimation technique [56]. Afterwards, the structure of a digital multi-mode controller is developed and implemented on a FPGA-based prototype for a buck converter to investigate transitions between the operating modes. A new discrete time large signal model of the converter is presented [57], especially meant for accurately predicting converter behavior in any operating condition. By the use of the model converter design issues are carried out and decision criteria for the automatic transition between the different control strategies are proposed. Eventually a serial compensator structure for small area, high efficiency, low power multi-mode converters is proposed [58].

The thesis is structured as follow: in Chapter 2 the basic operations of a buck converter are reviewed in all the most common operating modes. Converter power losses are also addressed together with the development of an efficiency characterization system for the investigation of control techniques aimed to efficiency optimization at light load [56]. Indeed, an automated efficiency characterization system is presented based on a Labview controlled laboratory setup and communication with the digital controller via serial link. Experimental efficiency characterization results, in conjunction with a power loss model, are used to address optimum selection of control parameters in Pulse Frequency Modulation (PFM) mode of operation.

In Chapter 3 the small signal models of the converter are discussed for the operating modes addressed in Chapter 2, while the design of the digital compensator and the compensator structures available are discussed in Chapter 4. In this chapter the design procedure is carried out highlighting the design issues introduced by the digital feedback loop on the compensator itself. Moreover a novel serial architecture of the digital compensator specifically tailored for multi-mode converters is proposed [58]. The adoption of a specific floating point representation of controller parameters minimizes the area requirements while allowing the representation of a
wide range of parameter values. Large multipliers and wide look-up tables are thus avoided, making this architecture well suited to applications where small size and low power consumption are required.

The implementation choices of the digital multi-mode controller are addressed in Chapter 5 where the structure of the multi-mode controller is explained. Here strategies for the automatic detection of the converter operating conditions are proposed, together with an efficient output current estimation technique specifically designed for light load conditions [56]. Finally a discrete time large signal model of the buck converter ([57]) is presented in Chapter 6 allowing the fully testability of the functionalities of the multi-mode controller. Indeed the proposed model accurately predicts the behavior of the converter in any of its operating conditions allowing the designer to easily investigate the converter issues when transitioning from one mode of operation to another. Simulation results are then validated comparing the simulated waveforms of the converter to the experimental results obtained from the FPGA-based prototype of the converter. Conclusions and future works are discussed in Chapter 7.
Chapter 2

The buck converter and its operating modes

In this dissertation a buck converter is used as an example of dc-dc converter because of its wide use and ease of understanding. The basic operation of a buck converter is overviewed in this chapter addressing the most common modes of operation. Power losses in the converter are addressed as well, in order to motivate, later on, the choice of the operating mode accordingly to the load condition of the converter. Moreover, an automated efficiency characterization system is presented to investigate converter power losses for low values of the output current. By the use of a power loss model, experimental results can be used to address the selection of controller parameters in such operating conditions, as presented at the IEEE Power Electronics Specialists Conference (PESC) in 2008 [56].

2.1 Overview of buck converter operation

A buck converter, also called a step-down converter, produces a lower average output voltage $V_{\text{out}}$ than the DC input voltage $V_{\text{in}}$. Fig. 2.1(a) shows a buck converter system. The converter consists of a switch network (transistors $M_1$ and $M_2$) and a second order $LC$ low pass filter. The input voltage from the battery gets chopped by the power switches and the average input voltage is thus reduced. Assuming ideal switches, the chopped voltage $v_{\text{sw}}$ at the switching node
is a square wave with duty ratio $D = t_{on}/T_s$, where $t_{on}$ is the on time of the transistor $M1$ and $T_s$ the switching period as shown in Fig. 2.1(b). The low pass filter, instead, is used to pass the DC component of $v_{sw}$ while attenuating the AC component to an acceptable ripple voltage. Neglecting the loss in the converter, the DC output voltage $V_{out}$ is given by:

$$V_{out} = V_{in}D. \quad (2.1)$$

The converter is said to be synchronous or asynchronous whether the low side power switch ($M2$) is used or is kept off during the whole length of the switching period. Converter state variables are the current flowing through the inductor $L$ ($i_L$) and the voltage across the capacitor $C$ ($v_c$) that coincides with the output voltage $V_{out}$ except for the voltage drop across the Equivalent Series Resistance (ESR) of the output capacitor $C$. The controller, therefore, computes the control law based on one or both of the state variables.

According to the output current $I_{out}$ the converter can enter two different modes of operation, the Continuous Conduction Mode (CCM) and the Discontinuous Conduction Mode (DCM) [59]. From this point forward we will use the notation $PWM-CCM$ and $PWM-DCM$ to refer to the continuous conduction mode and discontinuous conduction mode of the converter respectively, where PWM indicates that the switching frequency of the converter is kept constant.
Figure 2.2: Buck converter power stage. The two power transistor $M_1$ and $M_2$ are driven by the signals $HS$ and $LS$. The inductor series resistance ($R_L$) and the output capacitor ESR ($R_{esr}$) are also represented in the schematic.

### 2.2 The Continuous Conduction Mode (PWM-CCM)

Let us consider the buck converter power stage in Fig. 2.2. The waveforms characterizing the converter in this operating mode are shown in Fig. 2.3. Moreover, in Fig. 2.3 the waveforms of the output voltage and inductor current are approximated to be piecewise linear functions, approximation that will be adopted from here on.

In steady state the output voltage is given by (2.1) while the output current $I_{out}$ equals the average value of the inductor current. The inductor current is always positive for the whole length of the switching period $T_s$ as long as $I_{out} > \Delta i_L/2$, while when $I_{out} < \Delta i_L/2$ the inductor current becomes negative momentarily in the switching cycle and this corresponds to the converter discharging the output capacitor $C$ through the inductor $L$. As long as $i_L$ flows continuously, the converter is considered to be in continuous conduction mode.

In PWM-CCM the inductor current ripple $\Delta i_L$ is given by

$$\Delta i_L = \frac{V_{out}t_{off}}{L} = \frac{V_{out}(1 - D)T_s}{L}$$

(2.2)

while there are two ripple components in the output voltage: one is due to the output capacitor $C$ and the other is due to its equivalent series resistance (ESR) $R_{esr}$. Let $R_{esr}$ be zero for now, the output voltage ripple $\Delta v_{out}$ can be calculated by estimating the total charge accumulated on $C$ when $i_L$ is higher than $I_{out}$.

$$\Delta v_{out} = \frac{\Delta Q}{C} = \frac{1}{2} \frac{\Delta i_L T_s}{2 C} = \frac{1}{8} \frac{\Delta i_L T_s}{C}$$

(2.3)
Substituting $\Delta i_L$ from equation (2.2) yields

$$\Delta v_{\text{out}} = \frac{1}{8} V_{\text{out}} (1 - D) \frac{T_s^2}{LC} \quad (2.4)$$

Recalling that the switching frequency is $f_s = \frac{1}{T_s}$ and the cut-off frequency of the LC filter is $f_c = \frac{1}{\sqrt{2\pi}LC}$, then $\Delta v_{\text{out}}/V_{\text{out}}$ can be expressed as

$$\frac{\Delta v_{\text{out}}}{V_{\text{out}}} = \frac{\pi^2}{2} (1 - D) \frac{f_c^2}{f_s^2} \quad (2.5)$$

Equation (2.5) shows that the output voltage ripple can be reduced by selecting the LC cut-off frequency $f_c$ to be much lower than the switching frequency $f_s$ of the converter. With non-zero $R_{\text{esr}}$, the output voltage ripple is higher than with zero $R_{\text{esr}}$. It can be shown that, assuming the ripple current is piecewise linear, for $D < 50\%$, the voltage ripple $\Delta V_{\text{out}}$ is

$$\Delta v_{\text{out}} = \begin{cases} 
\frac{\Delta i_L T_s}{8C} + \Delta i_L R_o \frac{\tau_o}{2T_s} \left( \frac{1}{1-D} + \frac{1}{D} \right) & \text{for } \tau_o \leq \frac{1}{2} DT_s \\
\frac{\Delta i_L T_s}{8C} (1-D) T_s + \Delta i_L R_o \frac{\tau_o}{2(1-D)T_s} + \frac{\Delta i_L}{2} R_o & \text{for } \frac{1}{2} DT_s < \tau_o \leq \frac{1}{2} (1-D) T_s \\
\Delta i_L R & \text{for } \tau_o > \frac{1}{2} (1-D) T_s 
\end{cases} \quad (2.6)$$
Figure 2.4: Asynchronous buck converter. The low side power transistor $M_2$ is either replaced by a power diode or kept off during the whole length of the switching period allowing its body diode to conduct when $M_1$ is off.

and for $D \geq 50$

$$\Delta v_{out} = \begin{cases} 
\frac{\Delta i_L T_s}{8C} + \Delta i_L R_o \frac{\tau_o}{2T_s} \left( \frac{1}{1-D} + \frac{1}{D} \right) & \text{for } \tau_o \leq \frac{1}{2} (1-D) T_s \\
\frac{\Delta i_L T_s}{8C} DT_s + \Delta i_L R_o \frac{\tau_o}{2DT_s} + \frac{\Delta i_L}{2} R_o & \text{for } \frac{1}{2} DT_s < \tau_o \leq \frac{1}{2} (1-D) T_s \\
\Delta i_L R_o & \text{for } \tau_o > \frac{1}{2} DT_s
\end{cases} \tag{2.7}$$

where $\tau_o$ is the time constant of the output capacitor defined as $\tau_o = RC$. It turns out that smaller output voltage ripple can be achieved by choosing output capacitors with lower time constant values. The output voltage ripple is one of the specifications of the converter and its maximum value depends on the application.

### 2.3 Discontinuous Conduction Mode (PWM-DCM)

Replacing the low side switch $M_2$ in the synchronous converter with a power diode $D$ results in a conventional (or asynchronous) buck converter, as shown in Fig. 2.4.

When output current $I_{out}$ is higher than $i_L/2$, the inductor current flows continuously and the converter still works in continuous conduction mode, with $V_{out}$ satisfying (2.1). When $I_{out}$ is lower than $\Delta i_L/2$, the diode conducts while the inductor current decreases. The inductor discharges stored energy to the output capacitor until the current drops to zero. The diode then blocks the reverse current and the inductor current remains zero until the next switching cycle. This operation mode is called discontinuous conduction mode because the inductor current is
Figure 2.5: Converter waveforms in PWM-DCM mode of operation. \( t_{on} \) is the on time of the power transistor \( M_1 \) while \( \Delta i_L \) and \( \Delta v_{out} \) are the inductor current ripple and the output voltage current ripple respectively.

Identically zero for finite intervals. Fig. 2.5 shows converter waveforms in this mode of operation.

In a synchronous buck converter the body diode of the transistor \( M_2 \) is usually used to allow the converter to work in discontinuous conduction mode (\( M_2 \) is kept off for the whole length of the switching cycle).

In discontinuous conduction mode, the DC value of the output voltage of the converter \( V_{out} \) does not satisfy (2.1). Instead, the new expression of \( D \) is given by (2.8)

\[
D = \frac{V_{out}}{V_{in}} \sqrt{\frac{I_{out}/I_{out,cr}}{1-V_{out}/V_{in}}}
\]  

(2.8)

where \( I_{out,cr} \) is the maximum value of average inductor current at the edge of continuous conduction mode if \( V_{out} \) is constant:

\[
I_{out,cr} = \frac{T_s V_{out}}{2L}
\]

The inductor current ripple \( \Delta i_L \) equals the inductor current peak and is given by:

\[
\Delta i_L = \frac{V_{in} - V_{out}}{L} DT_s
\]

(2.9)

In general, as the average duty cycle value \( D \) in (2.8) is lower than the value in (2.1) when
the converter is in continuous conduction mode, the output voltage ripple in PWM-DCM is not usually considered as a design variable.

2.4 The Pulse Frequency Modulation (PFM) mode of operation

Unlike the PWM mode of operation, where the switching frequency of the converter is kept constant, the PFM mode of operation allows the converter to work with variable switching frequency as a function of the output current. This approach is widely used when the converter works with very low values of output current.

The basic behavior of the PFM controller is shown in Fig. 2.6. The output voltage is now adjusted comparing its value to the reference value $V_{th_{PFM}}$. The output of the comparator is sampled by the signal sample so that whenever $v_{out}$ is lower than $V_{th_{PFM}}$ a pulse of constant length $t_{on}$ is generated to bring $v_{out}$ above the reference. The switching period $T_{s_{PFM}}$ is thus variable and depends on the value of the output current. After a time $t_{on} + t_{off}$ the inductor current reaches zero and the output capacitor $C$ has to supply the current requested at the output of the converter. Because of the low value of $I_{out}$ the capacitor $C$ slowly discharges leading to a long switching period $T_{s_{PFM}}$. Its value can be approximated to:

$$f_{s_{PFM}} = \frac{1}{T_{s_{PFM}}} \approx \frac{I_{out}}{Q} = \frac{2I_{out}}{\Delta i_{L}} \frac{v_{in}}{t_{on}v_{out}}$$  \hspace{1cm} (2.10)

In PFM the converter is allowed to work with an output voltage ripple $\Delta v_{out}$ higher than those allowed in PWM-CCM. The value of $\Delta v_{out}$ is usually set to derive the length $t_{on}$ of the signal HS used to adjust the output voltage. Once $t_{on}$ is chosen the peak value of the inductor current is computed ($\Delta i_{L}$) in order to be sure that the power mosfets are not overstressed.

When $t_{on}$ is kept constant the controlled does not provide any feed forward and, therefore, an increase of the input voltage can lead to high peak values of the inductor current and then damage the power transistors. To avoid this a constant current peak PFM control can be implemented. The inductor current is sensed and its value is compared to a pre-set value $I_{p}$.

The condition $i_{L}(t) = I_{p}$ sets the length of the signal HS and, since the inductor current slope during $t_{on}$ is a function of the input voltage $V_{in}$, a feed forward compensation is performed.
Figure 2.6: Converter waveforms in PFM mode of operation. $\Delta i_L$ and $\Delta v_{out}$ are the inductor current ripple and the output voltage ripple respectively. $T_{s,PFM}$ is the switching period achieved for the corresponding value of output current. The sample signal samples the output of the comparator on the output voltage.

2.5 Power losses analysis in CCM and DCM

In any dc-dc converter the power consumption of the converter can be divided into three different categories [60]:

- **Conduction loss ($P_{cond}$):** Loss associated to the resistive elements of the converter. This loss included the power loss of the on resistance of the power transistors as well as the forward voltage drop of the power diode, the inductor series resistance $R_L$ and the capacitor ESR $R_{esr}$. In general this loss is a function of the output current $I_{out}$.

- **Switching loss ($P_{sw}$):** Loss associated to the switching action of the converter. During the turn-on and turn-off transitions of the power transistors and/or the power diode their output capacitance has to be charged (during the turn-on time) or discharged (during the turn-off time) loosing part of the power that has to be delivered to the load. Gate driver losses, voltage/current overlap at switching transitions and inductor core losses have also to be included in this category. As a general rule this loss is proportional to the switching frequency of the converter.
- Fixed loss ($P_{\text{fixed}}$): Power loss due to the controller standby current and leakage currents of transistors, diodes, etc. This loss is independent of the output current $I_{\text{out}}$

Naming $W_{\text{sw}}$ the energy lost during the transistor turn-on and turn-off transitions, dependent on the switching frequency $f_s$, the total power loss of the converter ($P_{\text{loss}}$) can be expressed as:

$$P_{\text{loss}} = P_{\text{cond}}(I_{\text{out}}) + W_{\text{sw}}f_s + P_{\text{fixed}}$$

(2.11)

In multi-mode converters minimization of the total power loss, in order to guarantee high efficiency over a wide range of output current values, is one of the most challenging issues. At high values of $I_{\text{out}}$ the converter conduction loss $P_{\text{cond}}$ is dominant and high efficiency is usually achieved replacing the power diode in Fig. 2.2 with a power transistor [61–63]. Indeed, the voltage drop across the on resistance of the transistor results to be smaller than the forward voltage of the diode thus reducing the conduction loss. As the output current decreases the inductor current can assume negative values for a portion of the switching period leading the capacitor $C$ to discharge through the inductor $L$ and then consuming additional power. The converter is then allowed to work in PWM-DCM in order to avoid this effect. From 2.11 it can be seen that, as the output current value further decreases, the switching loss becomes predominant and additional power can be saved if the switching frequency is reduced. A variable frequency regulation method, such as the PFM mode, is then adopted so that the switching losses decreases with the value of the output current.

In the following sections the power loss for the PWM and PFM modes are investigated.

### 2.6 Power loss in PWM-CCM

#### 2.6.1 Conduction loss

The conduction loss is mainly due to the finite on-resistance of the high side ($M_1$) and the low side ($M_2$) switches, denoted by $R_{\text{dsonHS}}$ and $R_{\text{dsonLS}}$ respectively, and the series resistance of the output inductor $R_L$. Let $R_{\text{dson}}$ be the equivalent on-resistance of the power switches seen by the inductor current. $R_{\text{dson}}$, the sum of the on-resistance of the power transistors, weighed
by conduction time:

\[ R_{dson} = R_{dsonHS}D + R_{dsonLS}(1 - D) \]  

(2.12)

In steady state, the average inductor current equals the output current \( I_{out} \). If we assume that the inductor current ripple is small compared to the average inductor current, the conduction loss are given by

\[ P_{cond} = (R_{dson} + R_L)I_{out}^2 \]  

(2.13)

To avoid the shoot through current between the high side and the low side switches, dead time must be inserted in the PWM signals that controls the two switches, to make sure the two switches are not on simultaneously. During the dead time, both power transistors are off, and the continuous flow of inductor current is relying on the body diodes of the transistors. If \( V_d \) is the voltage drop across the diode junction when the diode is on, the conduction loss on the diode, denoted by \( P_D \), is

\[ P_D = \frac{V_d \cdot I_{out} \cdot t_{deadtime}}{T_s} \]  

(2.14)

where \( t_{deadtime} \) is the total dead time in one switching cycle. If the dead time is designed properly, the conduction loss from the diode should also be small compared to \( P_{cond} \).

### 2.6.2 Switching loss

The power switches conduct momentarily in saturation mode during the turn-on and turn-off transient. The high voltage across the power device and the inductor current that flows through it can cause significant loss. This is often referred as hard switching. The switching loss in a dc-dc converter is mainly the loss due to hard switching and the loss in gate drives.

#### Hard switching loss

The profile of the power loss for the high side transistor and the low side transistor are quite different and therefore needs to be considered separately [64, 65]. Staring the analysis from the high side switch, the switching time is divided into 5 periods \( (t_1 - t_5) \) as illustrated in Fig.
Figure 2.7: a) High side transistor turn on waveforms b) Low side transistor turn on waveforms. The top diagrams show the voltage across the transistors and the current through them while the bottom diagrams represent $V_{gs}$ as a function of time. In the figure $V_{sp}$ is the switching point voltage or plateau voltage, $V_{spec}$ is the target voltage to reach and $Q_{g(sw)}$ is the total charge to deliver to the gate to complete the switch transition. The turn off losses are the same and the turn off waveforms result symmetrical to the turn on ones.

2.7(a). The top diagram of Fig. 2.7(a) shows the voltage across the transistor and the current through it while the bottom diagram represents $V_{gs}$ as a function of time.

Let us suppose that the gate of the high side transistor is being driven by a constant current. The switching interval begins when the driver of the high side switch turns on and begins to supply current to the gate of the switch $M1$ to charge its input capacitance. There is no switching loss until $V_{gs}$ reaches the threshold voltage $V_{th}$ of the transistor. When $V_{gs}$ reaches $V_{th}$, the input capacitance ($C_{iss} = C_{gs} + C_{gd}$) is being charged and the drain current $I_d$ of the mosfet is rising linearly until it reaches the inductor current $i_L$ which is presumed to be $I_{out}$. During this period ($t_2$) the mosfet is sustaining the entire input voltage across it. During $t_3$ instead, $I_{out}$ is flowing through $M1$, and its $V_{ds}$ begins to fall. In this case, all of the gate current will be going to recharge the capacitance $C_{gd}$. $C_{gd}$ is similar to the “Miller” capacitance of a bipolar transistor, so $t_3$ could be thought of as ”Miller time”. During this time the current is constant (at $I_{out}$) and the voltage is falling fairly linearly from $V_{in}$ to 0.
During \( t_4 \) and \( t_5 \), the mosfet is just fully enhancing the channel to obtain its rated \( R_{\text{dsonHS}} \) at a rated \( V_{gs} \). The losses during this time are very small compared to \( t_2 \) and \( t_3 \), when the mosfet is simultaneously sustaining voltage and conducting current, so they are usually ignored in the analysis. The switching loss for any given edge is just the power that occurs in each switching interval, multiplied by the duty cycle of the switching interval:

\[
P_{\text{swHS}} = \frac{V_{in}I_{\text{out}}}{2}(t_2 + t_3)f_s
\]

(2.15)

Low side switching loss for each turn on and turn off transitions can be calculated in a similar fashion to the high side switching loss:

\[
P_{\text{swLS}} \approx \left(t_2V_d + t_3V_d + I_{\text{out}} \cdot R_{\text{dsonL}}\right) I_{\text{out}}f_s
\]

(2.16)

where \( V_{in} \) in (2.15) has been replaced by \( V_d \), the schottky diode drop (approximated as 0.6V) in (2.16). Moreover, there is almost no Miller effect for the low-side mosfet, since \( V_{ds} \) is increasing (becoming less negative) as the device is turned on and then the gate driver is not having to supply charge to \( C_{gd} \). Since the body diode of the low side switch conducts before the switch to turn on or off the low side transistor is operating at zero voltage transition and therefore this loss is much lower than the loss associated to the high side transistor.

**Gate driver loss**

The power dissipation in the gate drives is mostly dynamic power used to charge and discharge parasitic capacitors of the power transistors. Two processes are important in understanding this type of loss: the charging of gate-source capacitor \( C_{gs} \), and charging of the miller capacitor \( C_{gd} \). The charge accumulated on \( C_{gs} \) when the transistors turn on is:

\[
Q_{gs} = (\Delta V_{gs})C_{gs}
\]

(2.17)

and the charge delivered to the miller capacitor is:

\[
Q_{gd} = (\Delta V_{gd})C_{gd}
\]

(2.18)
where $\Delta V_{gs}$ and $\Delta V_{gd}$ are the gate-source and gate-drain voltage change respectively during turning-on transient. Thus, $\Delta V_{gs} = V_{in}$, and $\Delta V_{gd} = 2V_{in}$. Therefore, the gate drive power loss can be given by:

$$P_{gate} = f_s(Q_{gs} + Q_{gd})V_{in} = f_s(C_{gs} + 2C_{gd})V_{in}^2$$

(2.19)

The loss in gate drives is usually smaller than the hard switch loss when the load current is high. However, the hard switch loss has dependency on the output current. The gate drive loss, on the other hand, is independent of output current. Thus, the gate drive loss can become the dominant component in the switching loss when the load is light (i.e. for low output current values).

### 2.6.3 Stray inductance loss

The stray inductance $L_s$ in the loop formed by input decoupling capacitor and power transistors has a power dissipation that equals to

$$P_{Ls} = \frac{1}{2T_s} L_s I_{max}^2$$

(2.20)

where $I_{max}$ is the maximum inductor current.

The value of $L_s$ depends on the PCB layout, packaging, etc. and can be reduced by minimizing the loop that contains the high current.

### 2.6.4 Controller quiescent current

The equivalent bias power that the controller of the dc-dc converter dissipates is called controller quiescent power. Typically, the quiescent power of the PWM controller is much lower than the sum of the switching and the conduction losses of the converter in continuous conduction mode.

### 2.7 Power loss in PWM-DCM

#### 2.7.1 Conduction loss

To compute the conduction loss for the high side and low side switches the rms values of the inductor waveform during the on time of the high side switch and the low side switch have to
be computed. The conduction loss due to the on resistance of the high side and the low side mosfets are:

\[ P_{\text{condHS}} = R_{\text{dsonHS}} \left( \frac{V_{\text{in}} - V_{\text{out}}}{L} \right)^2 \frac{f_s}{3} \]  
(2.21)

\[ P_{\text{condLS}} = R_{\text{dsonLS}} \left( \frac{V_{\text{out}}}{L} \right)^2 \frac{f_s}{3} \]  
(2.22)

At the same manner, the contribution to the conduction loss of the inductor series resistance is given by:

\[ P_{rl} = \frac{f_s \cdot i_{\text{peak}} \cdot L}{3} \frac{V_{\text{in}}}{V_{\text{out}}}(V_{\text{in}} - V_{\text{out}})R_L \]  
(2.23)

where \( i_{\text{peak}} \) is the peak value of the inductor current \( i_L \) \( (i_{\text{peak}} = ((V_{\text{in}} - V_{\text{out}})/L)t_{\text{on}} \) in a buck converter). Eventually, the capacitor equivalent series resistance (ESR) loss is given by:

\[ P_{\text{esr}} = \frac{V_{\text{in}}}{V_{\text{out}}}(V_{\text{in}} - V_{\text{out}})L \frac{f_s}{3} \left( \frac{i_{\text{out}}^3}{3} + \frac{i_{\text{peak}}}{2} \right)I_{\text{out}} R_{\text{esr}} \]  
(2.24)

If \( R_{\text{esr}} \) is small enough this conduction loss can be safely neglected.

When the low side switch is kept off during the whole length of the \( t_{\text{off}} \) time the conduction loss of its body diode has to be considered:

\[ P_{D} = \frac{i_{\text{peak}} \cdot V_{d} \cdot t_{\text{off}}}{2} f_s \]  
(2.25)

### 2.7.2 Switching loss

In the discontinuous conduction mode, the high side transistor will have a voltage/current overlap loss when it is turned off. When \( M1 \) turns on, instead, the inductor current is zero so no power loss is associate to this transition. When transistor \( M2 \) is turned on, the voltage across it is approximately zero, and when it is turned off, the inductor current is zero. If \( t_{\text{fallHS}} \) is the fall time of the high side transistor, then the overlap power loss is approximately given by:

\[ P_{\text{swHS}} = \frac{V_{\text{in}} \cdot i_{\text{peak}} \cdot t_{\text{fallHS}}}{2} f_s \]  
(2.26)

Another important contribution to the switching loss is given by the energy required to charge the equivalent capacitance \( C_{\text{sw}} \) seen at the switching node of the converter. It can be safely
assumed that the total energy lost in the charging process is given by

\[ E_{\text{sw, tot}} = \frac{1}{2} C_{\text{sw}} V_{\text{in}}^2 \]  

(2.27)

and therefore

\[ P_{\text{Csw}} = E_{\text{sw, tot}} \cdot f_s \]  

(2.28)

The capacitance \( C_{\text{sw}} \) can be directly estimated from converter waveforms during the time both the high side and the low side switches are off (\( t_{\text{idle}} \)). Indeed, during \( t_{\text{idle}} \) the capacitance \( C_{\text{sw}} \) exchanges energy with the inductor \( L \) originating an oscillation on the voltage at the switching node. Denoting \( f_{\text{osc}} \) the frequency of the oscillation, \( C_{\text{sw}} \) is given by

\[ C_{\text{sw}} = \left( \frac{1}{2\pi f_{\text{osc}}} \right)^2 \cdot \frac{1}{L} \]  

(2.29)

2.8 Power loss in PFM

When the converter is working in PFM mode, in steady state, the switching frequency is constant to the value \( f_s, \text{PFM} \). Therefore the converter is operating in discontinuous conduction mode, since the inductor current is not flowing continuously to the load, and the power loss can be compute in the same manner as for the PWM-DCM case.

2.9 Efficiency characterization in PFM mode

One of the key advantages of digital control is that a digital system interface enables easy programming of control parameters as well as collection of data from the controller. This feature has been used to develop a Labview based testing system to perform efficiency measurements of the converter in PFM mode of operation, with the objective to achieve an automated efficiency characterization and use the results to study efficiency optimization at light load.

The prototype system used to perform the efficiency measurements is shown in Fig. 2.8. This system consists of a PC-based Labview tool communicating with the digital controller through a serial interface, digital multimeters to acquire input and output voltages and currents, and a programmable load. In this case, the purpose of the efficiency characterization system is to
Figure 2.8: Efficiency characterization system. $V_{ref0}$ is the threshold for adjusting $v_{out}$ in PFM mode and the signal ADC0 enables the generation of the power transistors gate drive signals through the digital pulse width modulator (DPWM) module. The Labview tool communicates with FPGA-based digital controller via serial interface and set the value of the controller parameters.

perform efficiency measurements in PFM for different $t_{on}$ time (on time of the high side switch), and $t_{off}$ time (on time of the low side switch, including $t_{off} = 0$ to test asynchronous operation) under various load and input voltage conditions. The role of the Labview tool is to manage the entire efficiency measurement cycle by exchanging information with the digital controller using a set of pre-defined commands.

Operation of the tool for efficiency characterization is illustrated in Fig. 2.9. At the beginning of each measurement cycle, initial $t_{on}$ and $t_{off}$ time values for the DPWM are set. Afterwards, $t_{on}$ is incremented and the new value of $t_{off}$ is computed based on the acquired values of input ($V_{in}$) and output ($V_{out}$) voltages through the relationship:

$$t_{off} = \left(\frac{V_{in}}{V_{out}} - 1\right)t_{on}$$  \hspace{1cm} (2.30)

For each $t_{on}$ time the controllers analog-to-digital converter (ADC) monitors whether the output voltage is in regulation or not (through the signal ADC0), and measures the output voltage ripple (the signal ADC5 is used to determine whether the output voltage ripple exceeds a pre-set value). If the output is in regulation, input and output voltages, input and output currents, as well as voltage and current for the power mosfet driver are acquired from digital multimeters and sent
Figure 2.9: Block diagram of the entire efficiency measurement cycle managed by Labview tool.

to the Labview tool through GPIB interface. The digital controller also sends an estimate of the switching frequency to the Labview tool.

The Labview tool increases the $t_{on}$ time and acquires data until the output voltage ripple reaches a predetermined value (50mV in the experiments). This value is detected by the controller itself looking at the output of the output voltage ADC comparators (signal ADC5). Efficiency characterization results are collected into a spreadsheet and correlated with a power loss model based on [64–68].

In the power loss model of the experimental prototype the following losses have been considered:

- Power consumed to drive one of the two power mosfets ($P_{gate}$), since high side and low side mosfets are the same in the experimental prototype.

- Conduction losses of the high side and low side mosfets (respectively $P_{condHS}$ and $P_{condLS}$)

- Switching losses occurring during high side mosfet turn off and low side mosfet turn on (respectively $P_{swHS}$ and $P_{swLS}$)

- Energy lost to charge the equivalent capacitance at the switching node ($P_{C_{sw}}$). The equiv-
The equivalent capacitance $C_{sw}$ has been determined experimentally by measuring the oscillation frequency $f_{osc}$ at the switching node at the time when both mosfets are off in discontinuous conduction mode. Its value has then been computed from

$$C_{sw} = \frac{1}{4\pi^2 f_{osc}^2 L}$$

- Conduction loss due to output inductor series resistance ($P_{rl}$)

Power losses due to high side mosfet turn on and low side mosfet turn off have been neglected since the inductor current is almost zero when switching occurs. In the experimental example NXP PH3830L power mosfets (LFPAK package) have been used as both high side and low side mosfets with LM5101AM (NSC) gate driver. The power mosfets in the converter are sized for large output current in PWM mode of operation, resulting in relatively high switching losses at light loads and the need for efficiency optimization in PFM mode of operation.

The total equivalent capacitance measured at the switching node is found to be $C_{sw} = 3\text{nF}$, while the output filter series resistance ($R_L$) is 0.63mΩ. Conduction losses of the transistors have been computed as in [64].

Results obtained by the automated efficiency characterization system compared to the power loss model predictions are illustrated in Fig. 2.10. It can be observed that efficiency increases as a function of $t_{on}$ time, which means that switching losses dominate in the prototype converter. Using the characterization results and the power model, which correlates well with the experiments as shown in Fig. 2.10(a), it is possible to investigate details of loss contributions as shown in Fig. 2.10(b), using an analysis similar to [65]. Fig. 2.10(b) shows that the switching loss gives the highest contribution to the total power consumption of the experimental prototype.

Based on the efficiency characterization and power loss model, in the experimental prototype example of Fig. 2.8, the light load efficiency in PFM mode of operation can be improved by selecting the longest on time $t_{on}$ possible. However, longer $t_{on}$ time contributes to increased output voltage ripple. Therefore the on time values are chosen to maximize efficiency subject to maximum allowed output voltage ripple specification.

As a proof of the concept the range of output current values has been divided into windows where $(t_{on}, t_{off})$ times are set according to the characterization results. Indeed, $(t_{on}, t_{off})$ times
Figure 2.10: (a) Comparison between efficiency measured by the efficiency characterization system and the efficiency predicted by the power loss model; (b) Power losses as a function of $t_{on}$ time. In the legend $P_{condHS}$ and $P_{condLS}$ are conduction losses of high side and low side mosfets, $P_{swHS}$ and $P_{swLS}$ are switching losses associated to the power transistors, $P_{gate}$ is the power consumed to drive one of the two mosfets, $P_{rl}$ is the conduction loss associated with inductor series resistance and $P_{Csw}$ is the energy lost to charge the equivalent capacitance at the switching node.

in each window are set to obtain an output voltage ripple of 30mV at the upper boundary of the window. A point of load (POL) buck converter has been used for this purpose with the following parameters: input voltage $V_{in} = 5 – 12V$, output voltage $V_{out} = 1, 3V$ and $I_{out} = 0 – 10A$. PFM operation is intended for current values up to about 1A, or 12% of the output power of the converter.

The range of load current values has been divided into 10 windows where each window has a width of 125mA. Efficiency measurements have been taken for different load values and for three values of the input voltage. Results using the adaptive ($t_{on}, t_{off}$) approach are compared to constant, non-adaptive values set at minimum load current (106mA in this case).

Fig. 2.11 highlights the efficiency improvements obtained by the adaptive on time digital PFM control technique at light loads when the converter operates in PFM. An efficiency gain of up to 4% is reached at the high end of the light load range and the improvement is almost independent of the input voltage.
2.10 **Dc-dc converter system design**

2.10.1 **Digital controller system specification**

Some of the main specifications for the buck converter are listed in table 2.1.

2.10.2 **Output filter design**

The output filter in a buck converter is typically a second order LC filter. As shown by equations (2.5), (2.6) and (2.7), the output voltage ripple could be capacitance dominated, equivalent series resistance (ESR) dominated, or have significant components due to each. In the capacitance dominated case, the voltage ripple is quadratically dependent on the ratio of $LC$ filter cutoff frequency and the converter switching frequency. For a specified output voltage ripple and a given switching frequency, the product of the two filter components values can be determined. And the specific $L$ and $C$ values can be decided by constraints of current ripple, cost and profile requirements of output capacitors. A ceramic capacitor is usually preferred due to its smaller time constant $\tau_o$, smaller capacitor physical profile, and higher reliability. Moreover, from (2.6) and (2.7) turns out that a smaller capacitor time constant leads to reduced output voltage ripple.

Figure 2.11: Efficiency improvement obtained with the DPFM buck converter with adaptive $t_{on}$ time PFM control. Efficiency gains are almost independent of the input voltage.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{out_{\text{max}}}$</td>
<td>Maximum load current</td>
<td>2</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>$V_{in}$</td>
<td>Input voltage</td>
<td>5</td>
<td>20</td>
<td>45</td>
<td>V</td>
</tr>
<tr>
<td>$V_{out}$</td>
<td>Output voltage</td>
<td>4</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$I_{\text{peak_lim}}$</td>
<td>Switch peak current limit</td>
<td>2</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>$\Delta V_{out}/V_{out}$</td>
<td>PWM mode DC output voltage precision</td>
<td>3%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta v_{out_PWM}$</td>
<td>PWM mode output voltage ripple</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
</tr>
<tr>
<td>$\Delta v_{out_PFM}$</td>
<td>PFM mode output voltage ripple</td>
<td>80</td>
<td></td>
<td></td>
<td>mV</td>
</tr>
</tbody>
</table>

Table 2.1: Dc-dc converter specifications
Chapter 3

Multi-mode buck converter

Multi-mode dc dc converters have the feature to perform output voltage regulation with high efficiency for a wide range of output current values. This feature is achieved by the use of different control strategies according to the current requested at the output of the converter.

When the load current $I_{\text{out}}$ has a high value such that the inductor current $i_L$ never gets negative (heavy load condition) the converter operates in PWM-CCM where a PID regulator is required to maintain the output voltage to its adjusted value. As the current $I_{\text{out}}$ decreases the inductor current $i_L$ gets momentarily negative forcing the output capacitor $C$ to discharge through the inductor $L$ thus raising the conduction loss of the converter. Power can be saved if the converter is forced to work in PWM-DCM by keeping the low side switch $M2$ off for the whole length of the switching period and thus allowing the conduction of its body diode. In discontinuous conduction mode (PWM-DCM) a PI regulator is sufficient to perform the output voltage regulation.

For even lower values of $I_{\text{out}}$ the switching loss becomes dominant in the converter and additional power can be saved if a variable switching frequency control technique is adopted. At light load, therefore, the converter is forced to enter the Pulse Frequency Mode (PFM) so that the switching frequency becomes a function of the output current and the switching loss is thus reduced.

The design of a feedback loop for a dc-dc converter requires the use of dynamics models of the converter for any of its modes of operation. While analog controllers usually adopt a current and voltage feedback together (current mode feedback) to perform the output voltage regulation and be sufficiently fast for matching the desired specifications, the flexibility of digital controllers and
their ability to implement sophisticated control strategies, such as non linear control techniques, allow the designer to rely on a simple voltage feedback (voltage mode) to close the feedback loop.

In this chapter the converter dynamic models for all the operating modes will be derived. In general, converter signals such as the output voltage $v_{out}$ and the inductor current $i_L$ can be represented as the sum of a small signal component and a periodic ripple component with period $T_s$ due to the switching nature of the converter [59]. Since the bandwidth of the system is lower than the switching frequency, the ripple component does not contribute to converter dynamics so that it can be removed by averaging the converter waveforms over the switching period. In this way, the small signal model of the converter for any of the operating modes can be easily derived. Once the averaging action is performed a common procedure is to linearize converter waveforms with respect to their quiescent operating point, so that converter signals can be expressed by their DC components plus small-signal ac variations:

$$
\begin{align*}
v_{out}(t) &= V_{out} + \hat{v}_{out}(t) \\
i_L(t) &= I_L + \hat{i}_L(t)
\end{align*}
$$

(3.1)

In (3.1) $I_L$ and $V_{out}$ represent the DC values of the inductor current and the output voltage respectively while $\hat{i}_L$ and $\hat{v}_{out}$ are their small-signal ac variations. This notation will be adopted from here on to study the dynamics of the converter.

### 3.1 AC equivalent modeling in PWM-CCM

Let us consider the buck converter power stage in Fig. 3.1. In PWM-CCM the converter is synchronous and therefore both power switches $M1$ and $M2$ are driven. The main target of the dynamic model of the converter is to model the important dominant behavior of the system, while neglecting other insignificant phenomena. Among the converter parasitic components, such as on resistance of the power mosfets and stray capacitances, inductor series resistance, etc., only the inductor series resistance $R_L$ and the equivalent series resistance ESR of the output capacitor $C$ will be considered in the model, the last one because of the high frequency zero associated to it.

In any dc-dc converter the control variable is the duty cycle $d(t)$ of the signal $HS$ driving the
power switch $M_1$ while the variable to control is, of course, the output voltage $v_{out}$. Therefore, the small signal model of the converter has to describe how the converter reacts (in terms of output voltage variation $\hat{v}_{out}$) to a change of the duty cycle ($\hat{d}$). Assuming the input voltage $V_{in}$ to be constant, the dynamic of the converter is described by the control to output transfer function $G_{vd}(s)$ defined as:

$$G_{vd}(s) = \frac{\partial \hat{v}_{out}}{\partial \hat{d}} \bigg|_{\hat{v}_{in}(s)=0}$$

Derivation of $G_{vd}$ is well known in literature and can be found for example in [59]. For a buck converter in continuous conduction mode $G_{vd}(s)$ is found to be

$$G_{vd}(s) = \frac{V_{in} s CR_{esr} + 1}{s^2 (LC(R+R_{esr})) + s \left( \frac{LCR_{esr}}{R} \right) + 1}$$

As it can be seen from (3.3) the DC gain is independent from the output current $I_{out}$ and depends on the input voltage $V_{in}$. This sets the need for a voltage feed forward to maintain the desired features of the compensator for a wide range of input voltages.

The bode plot of (3.3) is shown in Fig. 3.2. Following the procedure in [59] the model can be improved considering also the effects of other parasitic components. In general the contribution given from the parasitic resistance in the forward path of the current from input to output (i.e. on resistance of power mosfets and inductor series resistance $R_L$) will introduce an attenuation
in the \( G_{vd} \) decreasing the cut-off frequency of the real system.

In continuous conduction mode there is an energy exchange between the output capacitor \( C \) and the output inductor \( L \) which is typical of a second order system with complex conjugate poles. This is because the inductor current \( i_L \) flows continuously for the whole length of the switching period. Of course this energy exchange between \( L \) and \( C \) originates power loss and therefore the efficiency of the converter gets higher as the current ripple \( \Delta i_L \) decreases.

3.2 AC equivalent modeling in PWM-DCM

The derivation of the control to output transfer function of the converter in PWM-DCM is carried out as in [69]. In this case the state space averaging technique has been adopted, where the converter dynamic is described by the state equations of the converter. Considering the inductor current \( i_L \) and the voltage across the capacitor \( C (v_c) \) as state variables, the state space equations of the converter are:

\[
\begin{align*}
\frac{di_L}{dt} &= - \left( \frac{R_L}{L} + \frac{(R//R_{esr})}{L} \right) i_L - \frac{2i_L}{\tau} V_{in}(R+R_{esr}) - (R R_{esr} + R_L (R+R_{esr})) i_L - R_{esr} \hat{v}_c + \frac{d}{\tau} V_{in} \\
\frac{dv_c}{dt} &= \left( 1 - \frac{R//R_{esr}}{R} \right) \frac{V_{in}}{C} i_L - \frac{1}{C} \frac{\hat{v}_c}{(R+R_{esr})}
\end{align*}
\] (3.4)

Figure 3.2: Bode plot of the control to output transfer function \( G_{vd} \) of the converter in continuous conduction mode.
The equations in (3.4) can be expressed using a matrix representation

$$\frac{d}{dt} \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix} = A \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix} + B \begin{bmatrix} \hat{v}_{in} \\ \hat{d} \end{bmatrix}$$

(3.5)

where

$$v_{out} = C \begin{bmatrix} \hat{i}_L \\ \hat{v}_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} \hat{v}_{in} \\ \hat{d} \end{bmatrix}$$

(3.6)

Defining two equivalent resistances $r_a$ and $r_b$ as:

$$\begin{align*}
  r_a &= R_L + R_{esr} \frac{1}{R} \\
  r_b &= \frac{R}{R + R_{esr}}
\end{align*}$$

(3.7)

terms of the matrix (3.5) are found to be:

$$\begin{align*}
  a_{11} &= -\frac{r_a}{L} + \frac{r_b(r_a M - 1)(2r_b) M}{D^2 r_a^2 (r_a^2 - 1)} \\
  a_{12} &= \frac{2 D r_a (r_a - r_b) M}{D^2 (1 - (r_a^2 + r_b) M) r_a^2} \\
  a_{21} &= \frac{r_b}{C} \\
  a_{22} &= -\frac{r_b}{C R}
\end{align*}$$

(3.8)

$$\begin{align*}
  b_{11} &= \frac{D}{L} + \frac{2 r_b^* M^2}{D^2 r_b^* (1 - (r_a^* + r_b^*) M) r_b^*} \\
  b_{12} &= \frac{V_{in} L}{E} + \frac{2 r_b^* M^2 V_{in}}{D^2 r_b^* (1 - (r_a^* + r_b^*) M)} \\
  b_{21} &= 0 \\
  b_{22} &= 0
\end{align*}$$

(3.9)

where

$$c_1 = \frac{R}{R_{esr}} \quad c_2 = r_b$$

(3.10)

where

$$r_a^* = \frac{r_a}{R} \quad r_b^* = \frac{r_b}{R}$$
The control to output transfer function can be computed solving the equations in (3.5) using tools such as Matlab. $G_{vd}(s)$ will be given by:

$$G_{vd}(s) = \frac{\partial \hat{v}_{out}}{\partial d} \bigg|_{\hat{v}_{in}=0} = C[(sI - A)^{-1}B] \quad (3.11)$$

and the bode plot of (3.11) is shown in Fig. 3.3.

In the discontinuous conduction mode the inductor current does not flow continuously in the converter but it gets null for a portion of the switching period $T_s$. Intuitively, this means that there is no energy exchange between $C$ and $L$ as in the continuous conduction mode and, therefore, we expect to find two separate poles in the control to output transfer function of the converter. Moreover, the time the inductor current stays to zero depends on the output current value (which, in steady state, equals the average value of the inductor current over the switching period). This sets a relation of the pole due to $L$ and the DC gain of the converter with the output current value, as shown in Fig. 3.4.
Figure 3.4: Control to output transfer function $G_{vd}$ of a buck converter for different values of the output current: $I_{\text{out}} = 0.13A$ (violet), $I_{\text{out}} = 0.11A$ (blue), $I_{\text{out}} = 0.1A$ (green), $I_{\text{out}} = 0.05A$ (red), $I_{\text{out}} = 0.01A$ (cyan). The converter is in PWM-CCM for $I_{\text{out}}$ higher than 0.11A.

3.3 AC equivalent modeling in Pulse Frequency Mode (PFM) of operation

The small signal model of the converter operating in PFM for a constant current peak control can be derived as in [60, 66]. The control to output transfer function of the converter is reported in (3.12) which results in a first order low pass filter due to the low frequency pole of the capacitor $C$. It can be noticed that the effect of the inductance $L$ is here omitted since it takes place to frequencies much higher than the steady state switching frequency ($f_{s,PFM} = 1/T_{s,PFM}$).

$$G_{vf}(s) = \frac{\partial \hat{v}_{\text{out}}}{\partial \hat{f}} = \frac{I_2^2 L V_{\text{in}} R}{2V_{\text{out}} (2V_{\text{in}} - 3V_{\text{out}})} \left(1 + \frac{s}{\omega_p}\right)$$  \hspace{1cm} (3.12)

where

$$\omega_p = \frac{3V_{\text{out}} - 2V_{\text{in}}}{RC(V_{\text{in}} - V_{\text{out}})}$$  \hspace{1cm} (3.13)

The bode plot of the control to output transfer function is shown in Fig. 3.5.
Figure 3.5: Control to output transfer function $G_{vf}$ of the converter in PFM mode of operation.
Chapter 4

Design of digitally controlled dc-dc converters

This chapter focuses on the design of the digital compensator in the two PWM modes of operation. In PFM mode, instead, the controller generates a pulse of a fixed pulse width whenever the output voltage is lower than a pre-set threshold, therefore no compensator is needed to adjust the output voltage. The structure of the chapter follows the procedure that has to be adopted for the design highlighting design issues of these converters. The compensation strategies used are explained and different implementation choices are presented as well. Moreover, a novel serial pid structure is presented that, together with the adopted representation of controller parameters (which will be described in the next chapter), minimizes the area requirements while allowing the representation of a wide range of parameter values [58]. The architecture results well suited for applications where small devices and low power consumption are demanded.

The basic blocks implementing the digital loop for a dc-dc converter are shown in Fig. 4.1 (b). Unlike the analog loop in Fig.4.1 (a) the output voltage signal $v_{out}(t)$ is compared to a reference voltage $V_{ref}$ and the corresponding error signal $e(t) = V_{ref} - v_{out}(t)$ is converted into a digital signal by an ADC converter. The digital error $e[n]$ is then processed by the digital compensator which outputs the duty cycle value $d[n]$. Eventually a Digital Pulse Width Modulator (DPWM) generates the gate signals for the high side and low side power mosfets.

Unlike the analog case, the introduction of the ADC and the DPWM blocks in the feedback
loop originates some design issues that are not present when designing an analog feedback loop for dc-dc converter. The first thing that has to be taken into account is that both the ADC converter and the DPWM blocks are quantizer blocks, because they introduce a quantization of their input signals. The combination of these two nonlinear blocks can originate limit cycle oscillation (LCO), i.e. oscillations due to the round-off errors introduced by the quantization of the converter variables. This oscillation affects converter stability and can not be predicted by the stability theory developed for linear time invariant systems. Therefore, additional conditions will be carried out to avoid LCO in the design of the digital controller.

In addition, the ADC and the DPWM blocks introduce a delay in the feedback path that degrades the phase of the closed loop system, so that additional effort has to be made to achieve the same features of analog dc-dc converters. Indeed, in analog dc-dc converters the duty cycle command $d(t)$ is generated from the continuous time error signal $e(t)$ which monitors the behavior of the output voltage of the converter at any time instant $t$. As a consequence, the delay introduced by the analog loop, i.e. the time required by the converter to react to a change in the output voltage, is due only to the propagation time of the analog compensator and the gate driver of the two power mosfets. Since this delay is usually short compared to the switching period $T_s$ the phase degradation associated to it is usually neglected. In digitally controlled dc-dc converters, instead, all the converter variables are sampled at a sampling period $T$ and, therefore, the output voltage is not monitored at any instant $t$ but only at instants $t = nT$ where $n$ is an integer number in the range $[0, +\infty]$. Intuitively it can be noticed that the sampling of the converter variables, together with the delay introduced by the other blocks involved in the

Figure 4.1: a) Analog dc-dc converter system. b) Digital dc-dc converter system.
digital loop, will introduce a delay in the feedback loop that will be longer than the one in the analog case and, therefore, it has to be taken into account while designing the digital controller. As addressed in Appendix A, the sampling period of the ADC converter has to be chosen properly to avoid aliasing in the spectrum of the sampled error voltage. Because of the periodic behavior of the output voltage of a dc-dc converter, the most intuitive choice is to set the sampling frequency equal to the switching frequency of the converter so that the relation $T = T_s$ is satisfied. In this case, all the signals of the digital controller will be updated once every switching cycle and the power consumption of the digital controller can be limited if all the blocks work at the sampling frequency.

The approach that will be adopted to design the digital compensator is the emulation method. In this way all the know-how of the design of analog dc-dc converters can be used to design the digital loop making digital control for dc-dc converter more accessible to analog designer.

### 4.1 Delays in the digital loop

Unlike the design of analog dc-dc converters, when designing the digital feedback loop the delay introduced by each block in the feedback path has to be considered for a proper design of the digital compensator. Indeed, the total delay in the control loop ($t_d$) is responsible for a phase degradation of the system open loop transfer function and therefore a more sophisticated compensator than the lag network in the analog case has to be used to achieve the target specifications on the output voltage adjustment. In general, $t_d$ is the sum of different contributions, as shown in Fig. 4.2. The total delay in the loop is then given by:

$$t_d = t_c + t_{d1} + t_{dpwm} + t_g$$  \hspace{1cm} (4.1)

where

- $t_c$ is the ADC conversion time
- $t_{d1}$ is the processing time of the digital compensator
Figure 4.2: Timing diagram of the delays in the control loop. \( v_{\text{out}}(t) \) is the output voltage of the converter, \( v_{\text{sw}}(t) \) is the switching node voltage and \( H_S(t) \) is the high side mosfet gate signal of the converter.

- \( t_{\text{dpwm}} \) is the modulator delay
- \( t_g \) is the gate-driver propagation delay

The most important contributions to \( t_d \) are the modulator delay \( t_{\text{dpwm}} \) and the ADC conversion time \( t_c \). Since \( t_{\text{dpwm}} \) is strictly dependent on the modulator topology, a great effort is usually done to reduce the conversion time of the ADC so that the total delay \( t_d \) is kept as short as possible. If \( G_{\text{vd}}(s) \) is the control to output transfer function of the converter, the delay term \( t_d \) corresponds to the term \( e^{-st_d} \) in the s-domain, so that the new control to output transfer function affected by the loop delay becomes:

\[
G_{\text{vd}}(s) \bigg|_{t_d} = G_{\text{vd}}(s)e^{-st_d} \tag{4.2}
\]

The effect of \( t_d \) on the control to output transfer function is shown in the bode plot in Fig. 4.3 for a buck converter working in PWM-CCM.
Effect of the total delay of the control loop on the control to output transfer function of a buck converter in PWM-CCM with switching period $T_s = 1.28\mu s$. In the bode plot $t_{dpwm} = 490\text{ns}$, $t_c = 260\text{ns}$ and $t_d = 880\text{ns}$. In general the main contributions to $t_d$ are the computation time of the ADC ($t_c$) and the modulator delay ($t_{dpwm}$).

### 4.1.1 Modulator delay

In the design of the digital loop for dc-dc converters, it is of fundamental importance to understand the effect of the DPWM block in the control loop. Comparing the digital loop in Fig. 4.1 (b) to the standard digital loop used in the theory of the sampled data systems (addressed in appendix A), the DPWM might be seen as a digital to analog converter block (DAC). Moreover, always from this point of view, the combination of the DAC and ADC blocks might originate a delay in the feedback loop that is approximately equal to half of the sampling period $T_s$, where $T = T_s$ in the case of a dc-dc converter. In digitally controlled dc-dc converters this ZOH effect does not exist. This is because the ZOH effect only occurs in real systems where the samples are converted back to the analog world through a filter operation. The DPWM, instead, is not performing any filtering action, the only equivalent "hold" in the system is a characteristic of the switching converter from the discrete duty cycle to the continuous output. Even though the modulator block do introduces a delay in the loop, delay that depends on the topology of the
modulator used.

In Fig. 4.4 comparison between the analog leading edge modulator and the most common digital modulator topologies is presented. By definition, the delay originated by the DPWM is the time gap between the sampling instant of its input signal, i.e. the duty cycle $d[n]$, and the switching instants of the signal output by the modulator. In the analog case (Fig. 4.4 (a)) the duty cycle $d(t)$ is never sampled and the signal output by the PWM gets active at the beginning of each switching cycle. The control action, i.e. the amount of charge that have to be transferred from the input of the converter to its output, is determined by the intersection of the duty cycle with the ramp signal. This action can be seen as a sampling of the duty cycle and because of that analog PWMs are called naturally sampled PWMs [70]. The trailing edge of the output signal of the analog PWM occurs exactly at the instant when the duty cycle is sampled and, therefore, there is no delay between the sampling action and the switching instant of the output signal. The delay introduced by the analog PWM is therefore equal to zero ($t_{dpwm} = 0$).

In Fig. 4.4 (b) the corresponding digital version of the analog leading edge PWM is shown. The duty cycle $d[n]$ is sampled at the beginning of each switching cycle and its value is kept constant until the next cycle. Therefore there is no delay from the sampling instant of the duty cycle and the turn on instant of the signal output by the DPWM, but there is a delay between the
Table 4.1: Summary of the delay introduced by the modulator in the most common modulation schemes. LE stands for leading edge modulation while TE stands for trailing edge modulation.

<table>
<thead>
<tr>
<th>Modulation Type</th>
<th>Delay Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog LE PWM</td>
<td>$t_{dpwm} = 0$</td>
</tr>
<tr>
<td>Digital LE PWM</td>
<td>$t_{dpwm} = DT_s$</td>
</tr>
<tr>
<td>Digital TE PWM</td>
<td>$t_{dpwm} = \frac{T_s}{2}$</td>
</tr>
</tbody>
</table>

sampling instant of the duty cycle and the turn off instant of the output signal. The modulator delay, in this case, can be easily determined to be $t_{dpwm} = DT_s$, where $D$ is the average duty cycle value. In the case of a triangular DPWM (Fig. 4.4) there is both a turn on and turn off delay and the average delay of the modulator is computed to be $t_{dpwm} = T_s/2$ [71, 72]. Table 4.1 summarizes the results obtained.

4.2 Design of the digital compensator

Once the control to output transfer function of the converter power stage is known and the total delay in the digital control loop is computed, the digital compensator can be designed. The design procedure presented in this section is based on the emulation approach addressed in appendix A so that all the strong know-how held by analog designer can be used to design the compensator. The target of the emulation approach is to encompass the transfer functions of all the elements in the feedback loop that wraps the digital compensator (Fig. 4.5). In general, the ADC converter can be assumed to have transfer function as in (4.3), while the DPWM has transfer function shown in (4.4).

\[
G_{ADC}(s) = \frac{K_{ADC}}{1 + \frac{s}{\omega_{adc}}} e^{-st_c} \tag{4.3}
\]

\[
G_{DPWM}(s) = K_{DPWM} e^{-st_{dpwm}} \quad K_{DPWM} = \frac{1}{V_m} \tag{4.4}
\]

where $V_m$ is the peak value of the ramp signal used to generate the output signal of the DPWM (Fig. 4.4).

The analog plant, together with the ADC and DPWM block, will have the following transfer
Figure 4.5: a) Digital dc-dc converter block diagram. b) Equivalent block diagram of the converter when the emulation approach is used to design the digital loop.

function:

\[ G_p(s) = G_{vd}(s)G_{DPWM}(s)G_{ADC}(s)e^{-s(t_c+t_d)} \]  

(4.5)

that has to be used to design the analog compensator \( G_{c_{eq}}(s) \) and that will be mapped in the Z-domain using one of the mapping techniques addressed in appendix A.

To properly design the digital compensator an accurate model of the analog plant is required. Nevertheless, the control to output transfer function of the converter \( (G_{vd}(s)) \) is derived by averaging the converter signals over the switching period and, therefore, it is an approximate model of the converter. The model, thus, does not take into account of high frequency effects usually neglected while analog dc-dc converters and that becomes important when designing digital dc-dc converters with bandwidth higher than \( \frac{f_s}{20} \).

### 4.2.1 Discrete time equivalent system of the dc-dc converter

As mentioned in section 4.1.1, the switching nature of the converter originates an equivalent "hold" from the discrete duty cycle to the continuous output \( v_{out} \). This effect can be easily seen when the converter operates in PWM-DCM, as shown in Fig.4.6. In Fig. 4.6 an analog PWM is considered for shake of simplicity. With the purpose of deriving a small-signal model of the
Figure 4.6: Principle of operation of the DPWM block. The small ac perturbation \( \hat{d}(t) \) of the duty cycle can be approximated to a string of pulses that occur at the instants where the ramp signal crosses the duty cycle \( d(t) \). This action can be seen as a sampling of \( d(t) \).

Converter the duty cycle signal \( d(t) \) can be linearized around its steady state value \( D \) as in (4.6).

\[
d(t) = D + \hat{d}(t)
\]  

(4.6)

The small-signal variation \( \hat{d}(t) \) of the duty cycle command originates a variation on the amount of charge transferred to the output filter LC, proportional to \( \hat{d} T_s \). Therefore, the signal \( \hat{d}(t) T_s \) can be seen as a sequence of pulses that occur periodically with period \( T_s \). This originates an equivalent hold effect on the inductor current, as depicted in Fig. 4.7.

The small-signal variation \( \hat{i}_L \) of the inductor current can then be described in the s-domain by (4.7) and the "hold" effect originates a second pole at frequency \( \frac{2f_s}{D_2} \) in the control to output transfer function \( G_{vd}(s) \).

\[
\hat{i}_L(s) \approx \frac{1 - e^{-sD_2T_s}}{s} \hat{d}(s) T_s
\]

(4.7)

Intuitively, from Fig. 4.6 it can be seen that system described by \( G_{vd}(s) \) is driven by a stream of Dirac pulsed and it is followed by the ADC converter which, hereby, is considered to be an
Figure 4.7: Hold effect on the inductor current $i_L$

Ideal sampler. If $g_{vd}(t)$ is the impulsive response of the converter, the impulsive response of its equivalent discrete time system is

$$g_{vd}[k] = T_s g_{vd}(t = nT_s) \quad (4.8)$$

and, therefore, the transfer function of the discrete time equivalent of the continuous time system $G_{vd}(s)$ has transfer function reported in (4.9):

$$G_{vd}(z) = T_s Z_{T_s \{ G_{vd}(s) \}} = \sum_{n=0}^{+\infty} g_{vd}(t = nT_s) z^{-n} \quad (4.9)$$

The system described by (4.9) takes into account of the sampling effect in the converter and, once mapped back to the s-domain using, for example, the bilinear transformation, is used to design the compensator $G_{c,eq}(s)$.

### 4.3 Design of the digital compensator in PWM-CCM

The procedure followed to design the digital compensator is summarized below:

- Derivation of the analog plant transfer function $G_p(s)$

- Compensator poles and zeros placement based on the open loop transfer function of the system affected by the total delay in the control loop
- Conversion of the compensator transfer function from s-domain to Z-domain using bilinear transformation (BLT)

Because of the phase degradation introduced by the delay term $t_d$ the design of the compensator is usually carried out with the target to have the highest bandwidth possible. In a standard design, the bandwidth of the system can be set to $f_c = \frac{f_s}{20}$, while additional effort will have to be done to achieve higher bandwidth, such as $f_c = \frac{f_s}{10}$ or $f_c = \frac{f_s}{6}$.

In continuous conduction mode a PID compensator has to be used to achieve the desired bandwidth. The compensator transfer function is shown in (4.10).

$$G_{\text{pid}}(s) = \frac{K}{s} \left( 1 + \frac{s}{2\pi f_{z_1}} \right) \left( 1 + \frac{s}{2\pi f_{z_2}} \right) \left( 1 + \frac{s}{2\pi f_{h}} \right)$$

(4.10)

Compensator zeros $f_{z_1}$ and $f_{z_2}$ can be placed in the following manner:

I. $f_{z_1}$ and $f_{z_2}$ can compensate exactly the conjugate pole pair of the control to output transfer function of the converter.

II. $f_{z_1}$ and $f_{z_2}$ can be placed close the converter resonant frequency $f_0$. Position of the two poles can be expressed as a function of $f_0$:

$$f_{z_1} = m_1 f_0$$

(4.11)

$$f_{z_2} = m_2 f_0$$

(4.12)

A common choice is to set $f_{z_1} = 0.7f_0$ and $f_{z_2} = 0.9f_0$ in order to have a smooth phase transition. An additional high frequency pole $f_{hf}$ is placed at frequency higher than the crossover frequency and is used to perform a prewarping of the compensator transfer function. The bilinear transformation will be used to convert the continuous time transfer function of the compensator into its equivalent one in the Z-domain. The main advantages of using the BLT transformation are:

- The BLT correctly maps the s-domain stability axis $j\omega$ to the Z-domain unit circle

- The entire s-domain left half plane (LHP) is mapped inside the Z-domain unit circle with no aliasing of pole and zero location. This means that there is no aliasing in added poles
and, therefore, the additional high frequency pole can be added to the compensator transfer function

Since there is no method that performs an exact matching of magnitude and phase of the continuous time system and the discrete time system at all frequencies, the BLT transformation allows designer to set a frequency value (called critical frequency $f_{crit}$) such that magnitude and phase of the continuous and discrete systems exactly matches at frequency $f_{crit}$. This turns out to be very useful in the design of the digital controller using the emulation approach because $f_{crit}$ can be set to the crossover frequency value $f_c$ so that the equivalent discrete time compensator will have the same phase margin of the its corresponding analog one.

The insight of the high frequency pole $f_{hf}$ is strictly related to the BLT transformation. The BLT maps the continuous time transfer function in the Z-domain such that the number of poles and zeros are equal. As a consequence, if the number of poles or zeros are different in the analog case, the missing poles or zeros are mapped to $z = -1$ (the highest frequency in the discrete time domain). It is advisable, then, to add poles and zeros so that the number of poles and zeros are the same, so that predictable results are obtained.

The high frequency pole should be placed directly in the Z-domain, i.e. $z_{hf} = a$, and it corresponds to the continuous time domain high frequency pole $f_{hf}$ by definition of (4.13).

$$f_{hf} = \frac{f_{crit}}{\tan \pi \cdot f_{crit}/f_s} \frac{1 + a}{1 - a}$$

(4.13)

The parameter $a$ has to be chosen in the range $a \in [0; -0.2]$. The choice of $a = 0$ corresponds to a simple delay in the controller implementation and leads to the minimum hardware implementation. If $a < 0$, the negative pole boosts the phase margin but degradates the gain margin, while the opposite is obtained for $a > 0$.

Fig. 4.8 highlights the location of zeros and poles of the compensator on the open loop transfer function of the system. Once poles and zeros of the compensator are placed the gain $K$ of the compensator in (4.10) is adjusted to achieve the desired cross over frequency:

$$K = \frac{1}{||G(j(2\pi f_c))||}$$

(4.14)
The transfer function $G_{pid}(s)$ is now ready to be mapped to the Z-domain using the BLT transformation. The generic compensator transfer function obtained by the BLT transformation will have the structure in (4.15):

$$G_{pid}(z) = \frac{b_2 z^2 + b_1 z + b_0}{z^2 - (1 - a)z - a}$$

that corresponds to the control law (4.16):

$$d[n] = (1 - a)d[n - 1] + ad[n - 2] + b_2 e[n] + b_1 e[n - 1] + b_0 e[n - 2]$$

### 4.4 Design of the digital compensator in PWM-DCM

When the converter operates in discontinuous conduction mode the design of the digital compensator is much easier than in the continuous conduction mode case and can be carried out directly in the Z-domain. In PWM-DCM, indeed, no high bandwidth is required since the output current $I_{out}$ is low and a simple PI compensator is enough to provide the output voltage regulation. In this case the control to output transfer function of the converter consists of two
real separate poles, one at very low frequency due to the output capacitor, and the other to high frequency due to the inductor current dynamics. The location of the low frequency pole of the converter is a function of the output current value, i.e. the load resistor $R$, and so is the DC gain of the converter. For a buck converter Eq. (4.17) yields.

$$G_{vo} = \frac{2V_{out}}{D} \frac{1 - M}{1 + M} \quad \omega_p = \frac{2 - M}{(1 - M)RC} \quad (4.17)$$

where $M = V_{out}/V_{in}$ is the conversion ratio and $D$ is the steady state duty cycle value as in chapter 2.

The design of the compensator is accomplished by the use of the control to output transfer function with $I_{out}$ close the boundary value $I_{out,cr}$. In general, the location of $\omega_p$ does not vary significantly with the output current, therefore the converter can be easily compensated placing a zero directly at the frequency of the output capacitor pole, as shown in Fig. 4.9.

The compensator transfer function is:

$$G_{pu}(s) = K \frac{z - zf}{z - 1} \quad (4.18)$$
Once the poles and zeros of the compensator have been placed, the gain $K$ is adjusted to achieve the desired bandwidth.

### 4.5 Compensator structures

Once the transfer function of the digital compensator is derived, several implementations of the same compensator exist. In general, designers try to reduce the number of resources used to implement the compensator so that both power and area are saved. From this consideration, the most common solution are hereby derived for the generic minimum hardware compensator, i.e. when the parameter $a$ in (4.15) is chosen to be zero.

The most intuitive implementation derives from the PID controller in the continuous time domain

$$d(t) = K \left( e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \dot{e}(t) \right) \quad (4.19)$$

where $K$, $K/T_I$ and $KT_D$ are the proportional, integral and derivative parameters respectively. Making the derivative of the generic signal $x(t)$ and approximating it through the backward rectangular rule in (4.20)

\[
\begin{cases}
\dot{x}(t) \approx \frac{x[n]-x[n-1]}{T_s} \\
\ddot{x}(t) \approx \frac{x[n]-2x[n-1]+x[n-2]}{T_s^2}
\end{cases}
\quad (4.20)
\]

the digital PID control law becomes:

$$d[n] = d[n-1] + \alpha e[n] + \beta e[n-1] + \gamma e[n-2] \quad (4.21)$$

that corresponds to the controller transfer function:

$$G_{std,pid}(z) = \frac{\alpha z^2 + \beta z + \gamma}{z(z-1)} \quad (4.22)$$

Equating the transfer function of the compensator just designed to (4.22) parameters $\alpha$, $\beta$ and $\gamma$ can be derived. The implementation of the control law (4.21) is shown in Fig. 4.10.

In Fig. 4.10 it is not straightforward to determine the proportional, integral and derivative
coefficients ($K_p$, $K_i$ and $K_d$ respectively) of the compensator. Therefore another possible implementation can be derived separating these three parts starting from the control law in (4.23)

$$d[n] = d[n-1] + K_p \cdot e[n] + K_i \cdot e[n] + K_d(e[n] - e[n-1])$$  \hspace{1cm} (4.23)

The corresponding transfer function of (4.23) is reported in (4.24)

$$G_{par_{pid}}(s) = K_p + K_i \frac{z}{z-1} + K_d \frac{z - 1}{z}$$  \hspace{1cm} (4.24)

Always equating the transfer function obtained by the BLT transformation to (4.24) the values of $K_p$, $K_i$ and $K_p$ are determined. Fig. 4.11 shows the implementation of the control law (4.23).

The structures shown in Fig. 4.11 and Fig. 4.10 are parallel structures. In these two structures the computation time of the compensator is minimized since there are no delays elements in

![Diagram](image)
Figure 4.12: Implementation of the control law (4.23) when the products $\alpha \cdot e[n]$, $\beta \cdot e[n-1]$ and $\gamma \cdot e[n-2]$ are pre-computed and stored into LUTs.

the forward path. The resources used to implement the compensator are not optimized instead, since any of the structures (4.23) and (4.21) require three multipliers and one adder to compute the control law. The computation of the duty cycle $d[n]$ can involve multipliers with a high number of bits, leading to an extensive area required to implement them. If the regulation of the output voltage is performed within a range of values (called regulation band) centered to its nominal value $V_{out}$ the error signal $e[n]$ can be represented using a small number of bits. In this case it could be convenient to pre-computate the products $K_x \cdot e[n]$ for any value of $e[n]$ and store the results into LUTs [14, 73] (where $K_x$ is the generic compensator parameter). As an example, if the structure in Fig. 4.10 is chosen the products $\alpha \cdot e[n]$, $\beta \cdot e[n-1]$ and $\gamma \cdot e[n-2]$ can be stored so that the new implementation will look like the one in Fig. 4.12.

### 4.6 Serial PID architecture

In order to minimize the area as well as the number of resources required to implement the PID compensator a serial PID architecture can be used [58]. The algorithm to compute the control law is derived from the standard linear PID law in (4.23).

\[
\begin{align*}
    d[n] &= K_p e[n] + K_i \sum_{r=0}^{n} e[r] + K_d (e[n] - e[n-1]) \\
    &= (K_p + K_i) e[n] + K_i \sum_{r=0}^{n-1} e[r] + K_d (e[n] - e[n-1])
\end{align*}
\] (4.25)
If $K_p$ is much greater than $K_i$ therefore the term $K_i + K_p$ can be approximated to $K_p$ so that (4.25) becomes

$$d[n] \approx K_pe[n] + K_i \sum_{r=0}^{n-1} e[r] + K_d(e[n] - e[n-1])$$

$$= K_pe[n] + d_i[n-1] + K_de[n] - e[n-1])$$

where $d_i[n] = K_i \sum_{r=0}^{n} e[n]$ denotes the integral part of the duty cycle $d[n]$ at the time $nT_s$.

The sequence of operations to compute the duty cycle $d[n]$ is derived from (4.26) once the system clock is defined. The system clock sets the timing for the executions of the operations. Each clock period $T_{sys}$ is associated to one computation step of the sequence of computations as defined in Table 4.2. In the serial architecture of the PID controller only one adder and one multiplier are involved. Three registers are used to store the values $d_i[n]$, $d[n]$ and $e[n]$ (i.e. the integral register, the output register and the input register respectively) while an additional register (main register) is used to store intermediate results $d_{tmp}[n]$ of the computation of $d[n]$.

<table>
<thead>
<tr>
<th>Number of $T_{sys}$</th>
<th>Computation steps</th>
<th>Mode</th>
<th>$K_x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$d_i[n-1] = K_ie[n-1] + d_i[n-2]$</td>
<td>I</td>
<td>$K_i$</td>
</tr>
<tr>
<td>2</td>
<td>$d_{tmp}[n] = d_i[n-1] + K_de[n-1]$</td>
<td>D</td>
<td>$K_d$</td>
</tr>
<tr>
<td>3</td>
<td>$d_{tmp}[n] = d_i[n-1] - K_de[n-1] + K_de[n]$</td>
<td>D</td>
<td>$K_d$</td>
</tr>
</tbody>
</table>

Table 4.2: Sequence of operations performed by the serial PID controller to compute the control law. The signal mode is used to select the operation to be computed at each step. Integral (I), derivative (D) and proportional (P) parts of the law (4.23) are computed sequentially.
Figure 4.13: Implementation of the serial PID compensator. A control unit generates the signal mode and the enable signals main_en, integ_en and out_en to perform the sequence of operations defined in Table 4.2. The signal mode selects the value of the parameter $K_x$ and the path to be followed by the output of the main adder. The integral path (dashed path) is used to compute the integral term of the duty cycle (step 1) while the main path (dotted path) is used to do all the remaining operations to complete the computation of the control law.
input, output and main registers. Table 4.2 also reports the value assumed by the signal mode and the parameter $K_x$ at any clock cycle. Moreover, the signal sign of the adder is used to select the sign of the algebraic sum. By definition, the computation time of the compensator is the delay introduced from the time when the error signal value $e[n]$ is updated to the time when the new duty cycle value is available to be processed by the DPWM. From Table 4.2 it has to be noticed that the value of the error signal is updated at the second clock period $T_{sys}$ while the new duty cycle value $d[n]$ is available at the fifth clock period. As a consequence the computation time results to be equal to $3*T_{sys}$ and, therefore, the approximation done in (4.26) allowed us to shorten the computation delay by one clock period, that would have been $4*T_{sys}$ otherwise.

Extension to the DCM mode, where a PI controller is used for the regulation, can be easily done by redefining the serial control algorithm of Table 4.2 as in (4.27).

$$d[n] = d_i[n-1] + (K_p + K_i)e[n]$$  \hspace{1cm} (4.27)

### 4.7 Sources of quantization

In a digitally controlled dc-dc converter there are three points in the feedback loop where quantization effects come into play: the A/D conversion, the DPWM quantization and the digital compensator, where finite-precision arithmetic causes quantization errors to affect both the compensator coefficients and the calculations.

Uniform quantizers, i.e. characterized by a constant quantization step, will be considered. Uniform quantizers are common in digital control, where a fixed-point arithmetic is commonly adopted for calculation purposes. Indeed, because of the high overheads in terms of both hardware requirements and speed required by the floating point arithmetic, this solution is usually avoided.

#### 4.7.1 ADC resolution

The resolution of the ADC converter, expressed by the width $n_{ad}$ of the binary word output by the converter itself, introduces a quantization error affecting the error signal $e[n]$ processed
by the digital compensator. Thus, $n_{ad}$ limits the extent to which the actual error is known to the digital compensator. Denoting with $FSR$ the full scale range of the ADC converter, the quantization step $q_{ad}$ on the voltage error signal is:

$$q_{ad} = \frac{FSR}{2^{n_{ad}}}$$

(4.28)

In most cases it is more meaningful to refer the quantization step to the converter state variable being acquired [74]:

$$q_{v,ad} = \frac{FSR}{2^{n_{ad}}} \frac{1}{H_{sense}(0)}$$

(4.29)

where $H_{sense}(0)$ is the DC gain of the sensing circuitry used to sense the output voltage of the converter. The quantization step $q_{v,ad}$ represents an upper limit to the regulation accuracy achievable by a given control loop. Of particular importance is the zero error bin, which represents the interval of output voltage values that result in $e[n] = 0$ and therefore are not distinguished by the digital compensator. Any steady state output voltage lying within the zero-error bin does not produce any further regulating action (open loop). As a consequence, given a setpoint $V_{ref}$, the ADC quantization causes the digital compensator to be able to regulate $v_{out}$ only to within $q_{v,ad}$.

### 4.7.2 Effect of finite word length of compensator parameters

Once the structure of the compensator is chosen the compensator has to be implemented in hardware by the use of a finite word length arithmetic. Indeed, in hardware, multiplier coefficients are rounded based on the number of bits used to perform the computation of the duty cycle value $d[n]$. On one side designers try to reduce the number of bits used in the compensator in order to limit the number of resource used as well as save area and power, on the other side a small number of bits will modify the location of zeros and poles defined in the design step [75]. As a consequence this could lead to system instability. High frequency and low frequency poles and zeros, i.e. poles and zeros with values close to $z = -1$ and $z = 1$, require an high number of bits to be distinguished from each other.

As an example, in Fig. 4.14 the bode plot of the open loop transfer function of the compensated buck converter in PWM-CCM is shown for a compensator involving 14, 11 and 10 bits.
Figure 4.14: Effect of the use a finite word length to represent compensator parameters is shown on the open loop frequency response of a buck converter working in PWM-CCM. In the figure the following cases are compared: blue: compensator parameters have an infinite resolution, black: 14 bits are used, green: 11 bits are used, red: 10 bits are used.

to represent compensator parameters in fixed point notation. In general, the effect of using a finite length arithmetic depends on the position of zeros and poles, the compensator structure an the notation chosen to represent compensator parameters. Therefore each design should be analyzed separately and verified by simulation.

As an assumption, in the design of digitally controlled dc-dc converter the most common procedure is to assume that the number of bits used to compute the duty cycle value $d[n]$ is such that the stability of the system is not affected and the only effect due to the quantization of the system variables is due to the finite resolution of the ADC and DPWM blocks.

4.7.3 DPWM resolution

The DPWM resolution $n_{dpwm}$ is usually limited by either area and power consumption requirements. For this reason the DPWM digital input is usually stored in a binary word which is
smaller than the word length of the duty cycle signal $d[n]$ produced by the control algorithm. The necessary conversion, i.e. truncation, round-off etc, represents a quantization for the duty cycle signal which is given by:

$$q_{dpwm} = \frac{1}{2^{n_{dpwm}}}$$  \hspace{1cm} (4.30)

Thus, $n_{spwm}$ limits the extent to which a desired duty cycle $d[n]$ can be realized to drive the power stage. Even if an infinite resolution ADC converter is considered, (4.30) degrades the output regulation accuracy. Referring again the quantization step to the converter output voltage, $q_{v,dpwm}$ is then given by:

$$q_{v,dpwm} = \frac{1}{2^{n_{dpwm}}} G_{vd}(0) = q_{dpwm}V_{in}$$  \hspace{1cm} (4.31)

where $G_{vd}(0)$ is the DC gain of the control to output transfer function of the converter in the operating mode considered.

4.8 Limit cycle oscillations

Limit cycle oscillations (LCO) represent the most typical and undesired nonlinear effect in a feedback-based digital system. LCOs represent persistent, amplitude-limited oscillations originated by the nonlinearities present in the feedback loop. Beside affecting the regulation of the output voltage, LCOs may be undesirable also from an electromagnetic interference (EMI) point of view, as they represent additional frequencies which are usually much lower than the converter switching frequency. The study of LCOs present many mathematical difficulties due to their intrinsic nonlinear nature. General criteria and guidelines have nevertheless been proposed in [76–81] for establishing necessary conditions to avoiding LCOs in digitally controlled converters.

In general two sets of limit cycle conditions have been derived in literature:

- **Static conditions**: Conditions that assure the existence of a DC solution for the output voltage of the digital dc-dc converter. If these conditions are not met the output voltage will never reach the zero error bin.
- **Dynamic conditions:** Describing functions are used to replace the nonlinear quantizers (i.e. the ADC converter and the DPWM blocks) with an amplitude and/or frequency dependent transfer function. In this way the effect of the quantizers in the feedback loop can be studied with the standard control theory and conditions for avoiding LCOs can be derived.

Some of these conditions will be addressed here with the purpose to understand the origin of the LCOs, while a more detailed discussion is found in [76].

To start the analysis of the limit cycle conditions let us assume that the converter is operating in PWM-CCM mode where a PID compensator is employed to perform the regulation of the output voltage of the converter. A first simple consideration can be formulated: if there is no DPWM quantization level that maps the output voltage in the ADC zero error bin, then the system will exhibit limit cycle oscillations. The reason is that, under this condition, no DC operating point exists that nulls the error \( e[n] \). On the other hand, a constant nonzero error would be indefinitely integrated by the PID integrator, thus violating the DC hypothesis. As a result, \( e[n] \) will oscillate around the zero error bin maintaining a zero average value. Fig. 4.15 clarifies this concept.

From this discussion, the existence of a DC operating point compatible with \( e[n] = 0 \) is ensured if the DPWM resolution, referred to the converter output voltage, is finer than the ADC resolution:

\[
q_{v,dpwm} < q_{v,ad}
\]  

(4.32)

In practice, it is recommended to realize (4.8) with a certain margin. An equivalent DPWM resolution two or three bits higher than the ADC resolution is usually advisable.

Condition (4.8) is not sufficient to guarantee the existence of a DC solution for the dc-dc converter. Indeed, even with an infinite resolution DPWM (i.e. \( q_{v,dpwm} = 0 \)) limit cycle can occur because of the integral gain of the PID compensator. To show this, let us consider a steady state condition in which \( e[n] = 0 \) and let us assume that a transient perturbation occurs (i.e. \( e[n] = q_{ad} \)), after which the controller restores the correct DC operating point reaching the zero

69
Figure 4.15: Origin of limit cycle oscillations: a) when the resolution of the DPWM \( (q_{dpwm}) \) is low the output voltage \( v_{out} \) cannot enter the zero error bin of the ADC converter and therefore \( v_{out} \) oscillates around the zero error bin originating a sequence of error values \( \{-1, +1, -1, ...\} \). b) If there is at least one quantization level of the DPWM that maps the output voltage inside the zero error bin of the ADC converter a DC solution is found and limit cycle oscillations are avoided.

error bin \( e[n] = 0 \). Considering only a simple integrator as a compensator:

\[
d[n] = d[n - 1] + K_i e[n] \tag{4.33}
\]

the response of the compensator to the input perturbation will be a step of amplitude \( K_i q_{ad} \) that occurs at the same instant the perturbation is applied. This means that, even with infinite resolution of the DPWM, the smallest possible increment in the duty cycle command is:

\[
\Delta d[n] = K_i q_{ad} \tag{4.34}
\]

As a consequence if the gain of the integrator \( K_i \) is not low enough to map the output voltage \( v_{out} \) inside the zero error bin limit cycle oscillations will occur. LCOs are then avoided whenever

\[
0 < G_{vd}(0)K_i = V_{in}K_i < \alpha \tag{4.35}
\]

where \( \alpha \) is a safety parameter usually set to \( \alpha = 0.5 \).

The two conditions (4.34) and (4.35) are static, necessary non limit cycle conditions. It must be underlined that the existence of a DC solution compatible with \( e[n] = 0 \) by no means guarantees
that the converter will actually converge to this particular steady-state mode of operation. Dynamic no-limit cycling conditions are derived in [76] and are here omitted since they involves a complex mathematical analysis that is out of the scope of this thesis. Nevertheless all the no-limit cycle conditions that designers might be aware of satisfying when designing a digitally controlled dc-dc converter are listed below:

- **Static conditions**

  **Condition A.1:** DPWM resolution sufficiently high
  
  \[ V_{in}q_{dpwm} < \alpha q_{v,ad} \]

  **Condition A.2:** Integral gain \( K_i \) of the compensator sufficiently low
  
  \[ 0 < G_{vd}(0)K_i = V_{in}K_i < \alpha \]

- **Dynamic conditions**

  **Condition B.1:** Need of a high resolution DPWM
  
  \[ \frac{4}{\pi} ||G_{vd}(j\omega_x)||H_{sense}(0)q_{dpwm} < \alpha q_{v,ad} \]

  where \( \omega_x \) is the frequency such that the phase of the open loop transfer function of the compensated system is \(-180^\circ\).

  **Condition B.2:** Gain margin of the converter \((GM_L)\) must be sufficiently large
  
  \[ GM_L < 4.2dB - 20\log{\alpha} \]

  where \( \alpha = 0.5 \) gives a practical safety margin.
Chapter 5

Implementation of the digital multi-mode controller

The implementation of digital multi-mode controller is presented in this chapter and all the blocks constituting the digital feedback loop are analyzed. The structure of the multi-mode controller, i.e. the controller specifically designed for the operating modes management, is presented and decision criteria for automating detection of converter operating conditions are explained as well. In PWM mode of operation area and power are saved by the use of the the serial pid structure [58] addressed in the previous chapter, and details on its implementation will be carried out in section 5.2. At light load, instead, the efficiency of the converter is optimized allowing the converter to operate in PFM mode. A new load current estimation technique used to determine whether the converter has to enter PWM mode is presented [56]. The load current estimator uses the existing analog-to-digital converter hardware, requires no additional sensing circuitry and operates over short time intervals compared to the switching period, therefore reducing the controller power consumption in PFM.

The structure of the whole digital controller is shown in Fig.5.1. The digital feedback loop is performed in the same fashion as described in chapter 4, with the only difference that an additional block is inserted in between the digital compensator and the DPWM. This block, i.e. the $\Delta-\Sigma$ modulator, has the purpose to relax the hardware requirements of the DPWM modulating the high resolution duty cycle signal output by the digital compensator (denoted as $d_{hr}[n]$ in
Fig. 5.1) around its steady state value. Taking advantage of the low pass filtering action of the dc-dc converter power stage, a less precise duty cycle command \( d_{lr}[n] \) is thus required at the input of the DPWM to obtain the same regulation feature that it would have obtained using the high resolution duty cycle command \( d_{hr}[n] \).

As addressed in chapter 4, each block in the feedback loop is characterized by its own processing time that delays the response of the system to an input perturbation. Because of the delays introduced by these elements, a synchronizer block is needed to sample the output signals of the elements in the feedback loop at the appropriate time. The signals \( \text{sample\_adc} \), \( \text{sample\_pid} \) and \( \text{sample\_dpwm} \) are used for this purpose.

Finally, the main controller finite state machine (FSM) block is used to manage the operations of the dc-dc converter. In general, the purpose of this controller is to set up all the controller parameters at the system turn on and move the output voltage inside the regulation band. Once the steady state is reached the converter will then operate as previously described, computing the duty cycle command \( d_{hr}[n] \) from the error signal \( e[n] \) and acting on the dc-dc converter power stage by means of the DPWM. In the case of the multi-mode controller developed this block is also responsible for the management of the operating modes of the converter, modes that are selected through the signals \( \text{DCM\_ext} \) and \( \text{PFM\_ext} \). The signals \( \text{CCM/DCM} \) and \( \text{PWM/PFM} \) are used to monitor the status of the converter and inform the digital feedback elements on the
control strategy to adopt.

The signals \texttt{DCM\textunderscore ext} and \texttt{PFM\textunderscore ext} can be either generated by an external logic or by an additional circuitry developed to perform automatic detection of the operating conditions of the converter. In the FPGA-based converter prototype this feature is implemented as well: the study of the possible solution to accomplish this task has been done and all the different strategies implemented to detect the operating conditions of the converter will be addressed in section 5.6.

In the following sections a description of all the implementation choices adopted in the digital multi-mode buck converter will be provided.

### 5.1 ADC converter

The ADC converter, together with the DPWM, is the most delicate block to design in a digitally controlled dc-dc converter. Due to the tight requirements on the output voltage adjustment that are usually demanded, fast and high resolution ADC converters would have to be employed in the digital loop. Indeed, as explained in chapter 4 the ADC converter is responsible for one of the highest contributions of the delay in the digital loop. Moreover, due to the high resolution requirements a high number of bits is usually required to codify the output voltage over the whole range of possible values. To overcome these limitations a \textit{window ADC} can be utilized (Fig. 5.2). The output voltage will then be codified only in a range of values centered at the nominal output voltage value $V_{out}$. When tight regulation of the output voltage is demanded, just few bits are thus needed to codify the output voltage inside the regulation band. As an example, if $V_{out} = 4V$ and a resolution of $q_{v,ad} = 20\text{mV}$ is required in a regulation band of $\pm 80\text{mV}$, 8 bits are required to represent the error voltage $e[n]$. Using the configuration in Fig. 5.2, instead, only 3 bits are needed to achieve the same features.

### 5.2 Compensator structure

The structure of the compensator implemented in the FPGA-based prototype is the serial compensator presented in chapter 4 and its implementation is reported in Fig. 5.3 for the shake
Figure 5.2: Window ADC converter. A group of comparators is used to sense the output voltage of the converter inside the regulation band determined by the reference values \( V_{ref,1} \) and \( V_{ref,n} \). The center of the regulation band coincides with the desired nominal value \( V_{out} \) of the output voltage.

of simplicity. In order to reduce the number of bits to represent compensator parameters and, at the same time, allows the representation a wide range of parameter values an alternative representation is used against the standard fixed point representation [58]. In the serial PID regulator the generic parameter \( K_x \) is expressed as:

\[
K_x = 2^{n_mant} (1 + m_x \cdot 2^{-n_mant}) \cdot 2^{E_x} \cdot 2^{K_x const} \cdot 2^{-n}
\]  

(5.1)

where \( n_mant \) is the number of bits used to represent the mantissa of the coefficient \( K_x \), that gives the precision of the number, while \( E_x \) is an integer value in the range \([0, 2^{n_exp} - 1]\) representing the value of the exponential of \( K_x \). \( E_x \) sets the range of parameter values that can be represented using \( n_exp \) bits for the exponent of \( K_x \) while \( m_x \) is an integer number in the range \([0, 2^{n_mant} - 1]\) that defines the value of its mantissa. The term \( 2^{K_x const} \) is used to shift the range of values of \( K_x \) so that \( K_i \), \( K_p \) and \( K_d \) can be represented using the same notation. The resolution of the controller parameters is given by the term \( 2^{n_mant} \cdot 2^{K_x const} \cdot 2^{-n} \) where \( 2^{-n} \) is the value of the LSB when \( n \) bits are used to represent the duty cycle value. It has to be noticed that the resolution is not the same for the various the parameters \( K_p \), \( K_i \) and \( K_d \), as it would be for the fixed point representation, but is weighted by the term \( 2^{K_x const} \).
Figure 5.3: Implementation of the serial PID compensator. The integral path (dashed path) is used to compute the integral term of the duty cycle while the main path (dotted path) is used to do all the remaining operations to complete the computation of the control law.

The floating point representation in (5.1) is only used to reduce the number of bits to be stored in LUTs to represent the values of the parameters $K_p$, $K_i$ and $K_d$, as well as the area of the multiplier. Only a small size multiplier is thus required to perform the multiplication of the mantissa $m_x$ of $K_x$ and the error signal $e[n]$, while a shift register is used to multiply the output of the multiplier by the term $2^{E_x} \cdot 2^{K_x \cdot \text{const}}$. The result of the product $K_x \cdot e[n]$ is expressed in fixed point representation so that also all the operations performed to compute the control law are done using fixed point arithmetic. In this way non linear and non-deterministic effects of the quantization error of the floating point arithmetic are avoided [82]. The use of floating point representation for the filter coefficients, instead, introduces a quantization error that is deterministic and, therefore, straightforward to analyze. In fact, this quantization error affects the frequency response of the controller by the approximation of poles and zeros location and it can be easily taken into account computing the quantized equivalent values of the filter coefficients. By the use of the notation (5.1) the generic PID parameter $K_x$ can assume any
value in the range \([k_{x,\text{min}}, K_{x,\text{max}}]\), defined by
\[
K_{x,\text{min}} = 2^{n_{\text{mant}}(1 + 0 \cdot 2^{-n_{\text{mant}}}2^0)} \cdot 2^{K_{x,\text{const}} \cdot 2^{-n}}
\]
\[
K_{x,\text{min}} = 2^{n_{\text{mant}}(1 + 2^{n_{\text{mant}}} \cdot 2^{-n_{\text{mant}}}2^n_{\text{exp}} \cdot 2^{K_{x,\text{const}} \cdot 2^{-n}}}
\]
(5.2)

In general, the integral coefficient \(K_i\) always has \(2^{K_{i,\text{const}} = 2^0}\) because it is the smallest parameter value to be represented, while the other parameters \(K_p\) and \(K_d\) are scaled accordingly.

### 5.3 DPWM structure

The digital PWM (DPWM) can be seen as the digital version of the standard analog PWM involved in analog dc-dc converters. In the digital case, the voltage ramp is replaced by a counter that generates a digital ramp with resolution \(\frac{1}{2^{n_{\text{dpwm}}}}\), where \(n_{\text{dpwm}}\) is the number of bits of the DPWM. The clock frequency \(f_{ck}\) of this counter-based PWM is then given by:
\[
f_{ck} = 2^{n_{\text{dpwm}}} f_s
\]
(5.3)

As an example, if 10 bits are required by the DPWM, the (5.3) would give a clock frequency \(f_{ck} = 12.78\,\text{GHz}\). To overcome this issue, an hybrid DPWM is commonly employed. The structure of this type of DPWM can be found in [25] and it has the feature to combine the counter based DPWM with a structure counter+delay line (DLL) to achieve a high resolution DPWM without the need of wide counters with extremely high clock frequencies.

The basic idea behind an hybrid DPWM is the following: the duty cycle word \(d[n]\) of \(n_{\text{dpwm}}\) bits can be divided into two words \(d_{\text{MSB}}[n]\) and \(d_{\text{LSB}}[n]\) encompassing the first \(n_{\text{MSB}}\) MSBs of the original word \(d[n]\) and the remaining \(n_{\text{LSB}} = n_{\text{dpwm}} - n_{\text{MSB}}\) LSBs. A low resolution counter-based is then used, together with the word \(d_{\text{MSB}}[n]\), to generate the coarser part of the signal driving the converter power stage. The DLL, together with the word \(d_{\text{LSB}}[n]\), is then used to generate the finer part of the output signal taking advantage of the small time resolution given by the delay of the single element of the DLL.

The DPWM used in the FPGA-based experimental prototype is an hybrid DPWM where a triangular modulation is employed. The basic modulation scheme is shown in Fig. 5.4. The main advantage of this type of modulation is that the duty cycle is updated while the output
signal driving the converter power stage is being generated and, therefore, it reacts faster to output voltage perturbations compared to the standard leading edge DPWM scheme.

5.4 Sigma delta modulator

To further relax the hardware requirements of the DPWM a $\Sigma - \Delta$ modulator is used. The basic function of this block is the same found in the $\Sigma - \Delta$ ADC converters where its main purpose is to shift the quantization noise at higher frequencies modulating the high resolution input signal and providing at the output a lower resolution signal affected by a quasi-random perturbation [83].

The discretization operation performed by the ADC converter can be viewed as an injection of quantization noise at the output of the ADC converter itself. The noise power level depends on the quantization step and can be assumed to be white and uniformly spread over a band going from the DC to $f_s/2$ (Nyquist frequency). When the quantization step is increased the noise also will increase uniformly in the Nyquist band.

The noise shaper principle is to apply a high pass filtering to the quantization noise so that its floor is decreased in a small bandwidth close to DC, the base band, while it increases at
high frequency. Finally a low pass filtering is filtering out the high frequency components thus reducing the noise at the output of the device.

In the case of a dc-dc converter the low pass filtering action is performed by the second order $LC$ output filter of the converter [84–88]. As a consequence, the band of interest in which a low noise floor has to guaranteed is roughly defined by the output filter resonant frequency [89]. The increase of resolution in such systems is traded with the system operating frequency. In particular it is important to define the oversampling ratio, i.e. the ratio between the operating frequency of the $\Sigma - \Delta$ modulator and the corner frequency of the baseband. In the case of a dc-dc converter the $\Sigma - \Delta$ modulator is usually operating at the switching frequency $f_s$ of the converter and the maximum frequency of the signals in the loop is approximately defined by the bandwidth $f_c$ of the converter. As a rule of thumb one bit resolution is added each doubling of the oversampling ratio.

In general, as in the $\Sigma - \Delta$ ADC converter, the effective resolution at the output of the dc-dc converter can be defined by the effective number of bits (ENOB) of the converter, given by:

$$ENOB = \frac{SNR_{dB} - 1.76}{6.02} \tag{5.4}$$

where $SNR_{dB}$ is the signal to noise ratio of the converter expressed in $dB$. For a $\Sigma - \Delta$ converter with oversampling factor $OVR$, $L$-th order filter and $N$ bits of output quantization, the maximum theoretical SNR is given by:

$$SNR_{dB} = \frac{3}{2} OVR^{2L+1}(2^N - 1)^2 \left( \frac{2L + 1}{\pi^{2L}} \right) \tag{5.5}$$

The hardware implementation of the $\Sigma - \Delta$ modulator is shown in Fig. 5.5(a). As already mentioned, the discrete time filter $1 - NTF(z)$ operates at a frequency equal to $f_s$ and involves processing of only $n_{comp} - n_{dpcm}$ bits. The output duty cycle command can then be expressed by

$$d_{\text{lr}}[n] = d_{hr}[n] + NTF(z)q_{\text{adc}} \tag{5.6}$$

where the quantization error $q_{\text{adc}}$ is modeled as a white noise having RMS magnitude equals to $1/(2^{n_{\text{adc}}} \sqrt{12})$. The most common choice of the noise transfer function $NTF(z)$ is

$$NTF(z) = (1 - z^{-1})^2 \tag{5.7}$$
Figure 5.5: a) Error feedback architecture of the $\Sigma - \Delta$ modulator. b) Implementation of the second order $\Sigma - \Delta$ modulator.

and its implementation is shown in Fig. 5.5 as presented in [87].

5.5 Digital multi mode controller structure

Among the standard elements constituting the digital control loop of a dc-dc converter, additional circuitry is required to manage the operations of the blocks implementing the digital feedback loop of the converter. This task is performed by the digital multi mode controller, controller that also has the target to manage the different operating modes of the converter in order to achieve the desired features, in terms of output voltage adjustment, while transitioning between an operating mode to another.

The main structure of the controller is given by the finite state machine in Fig. 5.6. Initially a manual reset man$_{\text{rst}}$.n moves the controller to its initial state init. In this state the output voltage of the converter is null and no control signal is yet generated. An internal reset pmurst.n is then produced to initialize all the controller parameters. This internal reset can be also used to reset the system if some unexpected conditions are detected during the normal functioning of the converter.

Afterwards, the start up state is entered. Here the output voltage is moved into the regulation band by linearly increasing the reference value $V_{ref}$ until the output voltage reaches its nominal value. During this start up phase the converters operates in PWM-CCM so that the steady state operation can be reached independently of the value of the output current. When the

80
Figure 5.6: Main controller finite state machine. The converter is in steady state when the main state is reached. In the main state the converter is allowed to work in three different operating modes: PWM-CCM, PWM-DCM and PFM modes according to the output current value.

output voltage is regulated the controller state machine finally reaches its main state where it will remain until the system is shut down.

The multi-mode controller is designed to allow the converter to work in continuous conduction mode (PWM-CCM), discontinuous conduction mode (PWM-DCM) and pulse frequency modulation mode (PFM) according to the value of the output current. The operating mode of the converter is selected through the external signals DCM_ext and PFM_ext and, once the operating mode is entered, the signals CCM/DCM and PWM/PPM are used to monitor the status of the converter. The signals DCM_ext and PFM_ext can be either activated by the user or by an additional logic for performing automatic detection of the operating conditions of the converter.

In Fig. 5.7 the structure of the main state of the controller state machine is shown, together with the allowed operating mode transitions. The signals go2CCM and go2DCM are used to change the operating mode of the converter from PWM-CCM to PWM-DCM while the signals go2PFM, go2PWM are used to select between fixed switching frequency regulation approach (PWM) and a variable switching frequency regulation approach (PFM). In addiction, an 8bits counter is associated to any of the states in Fig. 5.7 so that, every time a new operating mode is entered, the state of the controller cannot be changed until the counter reaches a pre-set value. In this way the controller waits for the output voltage to reach its steady state value before allowing the converter to change the mode of operation, preventing the converter from bouncing between the modes and then bringing the output voltage out of regulation. The signal CCM/DCM is used by the
digital compensator as well to choose the control algorithm to implement and the compensator parameters. In PFM mode, instead, the ADC converter is disabled and the error signal $e[n]$ is set to zero so that the last duty cycle value $d[n]$ is stored. The controller implements a constant on-time PFM control and the signal $\text{PWM/PFM}$ is used to signal whether the converter is working in PWM or PFM mode.

As an example, after the start up phase where the converter is kept in PWM-CCM mode, the operating conditions may be changed to PWM-DCM (if the signal go2DCM is active) or PFM mode (if go2PFM is now active). During the transient the converter is always working in the operating mode that provides the best features in terms of adjustment of the output voltage (i.e. PWM-CCM). Based on this concept, from Fig. 5.7 it can be noticed that the transition from PFM to PWM-DCM is not allowed. This is because the converter works in PFM mode for very low output current values. When a positive step is applied to $I_{out}$ the controller will have to compensate for the lack of energy requested by the load by quickly charging the output capacitor to have the output voltage adjusted again to its nominal value.
5.6 Automatic selection of the mode of operation of the converter

One of the main topics of this research activity is to investigate the possible strategies to perform the detection of the operating conditions of the converter in order to allow the controller to select the control strategy to employ. To accomplish this, an additional logic has been developed with the purpose of driving the signals DCM$_{\text{ext}}$ and PFM$_{\text{ext}}$ when one of the conditions for the operating mode change is verified. The following strategies have been implemented in the FPGA-based converter prototype.

5.6.1 Transition from PWM-CCM to PWM-DCM

In general, when the output current is high, the converter operates in PWM-CCM mode and its steady state duty cycle, output by the digital compensator, is higher than its ideal value $D = V_{\text{out}}/V_{\text{in}}$. Indeed, the controller has to compensate for the high conduction loss occurring in the converter power stage. As the output current decreases and approaches its boundary value ($I_{\text{out,cr}}$) before entering the PWM-DCM, the conduction loss becomes lower and the steady duty cycle is then close to its ideal value. This condition can be used to determine whether the converter is approaching the PWM-DCM mode.

The following operations are performed by the controller: once the average duty cycle gets lower than a given value the controller has to check if the PWM-DCM mode has to be entered. From Fig. 5.8 (b) it can be seen that when the output current is such that the inductor current becomes negative, during the dead times (i.e. when both the switches are off) the voltage at the switching node will quickly jump to the input voltage value $V_{\text{in}}$ plus the forward voltage of the body diode of the high side switch. This is due to the resonant action of the equivalent capacitance seen at the switching node and the output inductor $L$. The slew rate of this transition is determined by the value of the equivalent resistance at the switching node that slows down the transition.

In this case, if the signal driving the low side switch (i.e. the signal LS(t)) is shortened as a pulse width a comparator can be used to determine whether the switching node voltage becomes positive before the end of the switching period. When such a condition is verified then
Figure 5.8: Detection of the transition to PWM-DCM mode

a) the converter is working in PWM-CCM and $i_L > 0$ for the whole length of the switching period. b) The output current has now decreased and during the dead times, when $i_L < 0$, the voltage $v_{sw}$ at the switching node jumps to the value $V_{in} + V_d$. c) The signal $LS(t)$ is then shorten and the transition of $v_{sw}$ from $-V_d$ to $V_{in} - V_d$ can be used to detect when the PWM-DCM mode has to be entered.
the operating mode is changed to the PWM-DCM one.

5.6.2 Transition from PWM-DCM to PWM-CCM

Once the converter is working in PWM-DCM its steady state duty cycle $D_{dcm}$ is lower than the steady state duty cycle $D_{ccm}$ in PWM-CCM and it is a function of the output current $I_{out}$. Moreover, as $I_{out}$ increases, $D_{dcm}$ gets closer to the value $D_{ccm}$.

In PWM-DCM the on time of the low side switch ($t_{off}$) is given by:

$$t_{off, dcm} = D_{off, dcm} T_s = \left(\frac{V_{in}}{V_{out}} - 1\right) D_{dcm} T_s$$

(5.8)

The same duty cycle value, instead, would be associated to a $t_{off}$ time in PWM-CCM as in (5.9).

$$t_{off, ccm} = D_{off, ccm} T_s = (1 - D_{dcm}) T_s$$

(5.9)

As the output current increases the value $D_{off, dcm}$ approaches the value $D_{off, ccm}$ until it gets higher for output currents above its critical value $I_{out, cr}$. If the condition $D_{off, dcm} > D_{off, ccm}$ is verified then the operating mode of the converter has to be changed to PWM-CCM.

5.6.3 Transition from PWM to PFM

The PFM mode of operation is entered for values of the output current lower than 50mA. When the converter is working in one of the PWM modes, either the CCM or the DCM mode, the easiest and safest way to detect such a condition is to sense the inductor current or the current flowing through one of the switches. In a buck converter IC the sensing of the current is integrated in the chip and therefore fully optimized on this purpose. In the experimental prototype, instead, the sensing of the current had to rely on discrete components and therefore our decision was not to implement it and replace the sensing with a simple, but less safe, condition on the average duty cycle value.

5.6.4 Transition from PFM to PWM

The transition from PFM to PWM-CCM mode would require a sensing of the inductor current to detect whether the operating mode has to be changed. Direct measurement of the load current...
in PFM is difficult (since the switch or the inductor current waveshapes are not directly related to the DC output current), and can be costly in terms of additional losses or sensing circuitry. Estimation of the load is therefore much preferred. In PFM operation, the switching frequency is related to the load current. However, digital measurement of the switching period, which can be very long at light loads, would require long counters (or custom circuits [51]) and continuous operation in PFM, with penalties in additional controller power losses. An alternative load current estimation technique, using a short counter and the existing ADC hardware, has been developed and tested in the experimental prototype of the converter [56].

For a buck converter the output current in discontinuous conduction mode (i.e. when \( i_L \) does not flow continuously to the load) is given by:

\[
I_{\text{out}} = i_L(t) - i_{\text{cap}}(t) \tag{5.10}
\]

where \( i_{\text{cap}} \) is the current flowing through the output capacitor. During the idle time \( t_{\text{idle}} \), (i.e. when both the power switches are off) \( i_L = 0 \) and therefore (5.10) becomes:

\[
I_{\text{out}} = -C \frac{dv_{\text{out}}}{dt} \tag{5.11}
\]

Eq. (5.11) states that the information of the output current is encompassed in the slope of the output voltage \( v_{\text{out}} \) during the idle time. As a consequence this information can be exploited using two of the comparators used as an ADC in the PWM modes. If we denote \( \text{ADC0} \) the output signal of the comparator used to perform the output voltage adjustment in PFM and \( \text{ADC1} \) the output signal of the comparator with threshold \( V_{\text{th,PFM}} + q_{v,\text{ad}} \), then the value of \( I_{\text{out}} \) can be estimated measuring the time \( t_{\text{meas}} \) required by output voltage to decrease of the quantity \( q_{v,\text{ad}} \) (Fig. 5.9(a)). The time \( t_{\text{meas}} \) is easily measured by a counter with system clock activated asynchronously, upon detecting a falling edge of the \( \text{ADC1} \) signal. In the experiments, the system clock is 25MHz. Because of the nonlinearity between the measured time \( t_{\text{meas}} \) and the output current \( I_{\text{out}} \) in (5.11) the resolution of the estimated load current is not constant over the range of load current values but it decreases at light load currents. Furthermore, slow variation of the output voltage during the estimation may lead to a jitter in the comparators output signals \( \text{ADC0} \) and \( \text{ADC1} \) which, in turn, results in additional errors in the estimated value of the load current. Comparisons among twenty load current measurements and their experimentally
Figure 5.9: a) Buck converter waveforms in PFM mode. The output current can be estimated from 5.11 measuring $t_{\text{meas}}$. b) Maximum and minimum load current estimation errors experimentally measured over 20 measurements for each load current value $I_{\text{out}}$. Errors are expressed as percentage of the measured output current.

Estimated values under different load conditions are shown in Fig. 5.9(b). The relative error is less than 7% at all tested loads. Tolerances or variations in the output filter capacitance can introduce a further error in the load current estimation. This error is proportional to the relative tolerance in $C$ and it can be removed by calibration.
Chapter 6

Simulation of the digital multi-mode buck converter

One of the main issues when designing multi-mode converters is the choice of the approach that has to be adopted to model the whole system. In general, while digital controllers are described in the discrete time domain by the use of HDL language, dc-dc converter power stages are modeled in the continuous time domain relying upon state-space averaging [59, 69]. However, from the point of view of the controller, the converter power stage can be seen as a stimulus generator for the digital controller itself and, therefore, it is advantageous to model the behavior of the analog blocks in a digital fashion. Thus, extensions of the HDL languages to the analog world, i.e. VHDL-AMS or Verilog-AMS, have been used for this purpose.

At system level, the simulation time is a key parameter for speeding up the design process, therefore the use of event driven tools, i.e. tools involved in the simulation of pure digital circuits, is in general preferred to the use of simulation tools involving mixed signal simulation even from the earliest stages of the design. In [90], for example, a discrete time model of a flyback converter is implemented in VHDL language and comparison of this model with its corresponding VHDL-AMS model is done in [91] to motivate the choice of a pure HDL language model. The model presented in [90] is a first order behavioral model where the switching elements of the converter (i.e. the switch network) are modeled as ideal switches. This approach is not suited for modeling the behavior of the converter when the operating mode is changed since
parasitic elements of the switches are not considered. In [92], instead, equations describing the behavior of the converter are developed separately for each operating mode. As a consequence, this introduces a discontinuity in the behavior of the model when the converter transits from one operating mode to another. To avoid this, an appropriate modeling of the switch network of the converter is thus required in order to derive a model of the power train where equations are valid in any operating condition. This is the key point for multi-mode converters to address design issues related to mode transitions and to allow the analysis of conditions for performing automatic transition between operating modes.

A large-signal model of dc-dc converters has thus been developed [57], which can be implemented by any HDL language and is particularly meant for the design of multi-mode converters where digital control is applied. The model accurately describes the behavior of synchronous/asynchronous converters in PWM-CCM, PWM-DCM, and PFM modes of operation, with particular benefit in DCM and PFM. By the use of the model, conditions for automatic mode transition can be easily determined from simulation results and all the functionalities of the digital controller can be easily tested as well, before being implemented in the experimental prototype. HDL language implementation of the model allows the system level simulation of digitally controlled SMPS without the need of mixed-signal simulator tools, resulting in a reduction of the verification time when compared to the same simulation using a VHDL-AMS model of the converter.

6.1 Discrete time large signal model of converter power stage

Without loss of generality, the main idea behind this work is applied to the buck converter in Fig. 6.1(a), for its wide use and ease of understanding. In Fig.6.1(a), the switch network and the passive components of the converter are identified with the purpose of modeling them as separated blocks, which eventually will be connected together. The key point of the model is the way the switch network of the converter is modeled. Referring to Fig. 6.1(a), the switch network can be seen as a three port network as in Fig.6.1(b), where the power devices have been
replaced by variable controlled resistors defined by:

\[
\begin{align*}
gm_{HS} &= \begin{cases}
\frac{1}{R_{\text{on},HS}} & \text{PWM}_{HS} = 1' \\
gm_{\text{OFF},HS} & \text{PWM}_{HS} = 0'
\end{cases} \\
gm_{LS} &= \begin{cases}
\frac{1}{R_{\text{on},LS}} & \text{PWM}_{LS} = 1' \\
gm_{\text{OFF},LS} & \text{PWM}_{LS} = 0'
\end{cases}
\end{align*}
\]  

(6.1)

where \(R_{\text{ON},HS}\) and \(R_{\text{ON},LS}\) are the on resistances of the power mosfets while \(g_{\text{mOFF}}\) is the conductance of the power mosfets when they are off. Compared to the solution with ideal switches [90], the approach in Fig. 6.1 (b) allows the current \(i_c\) (the inductor current \(i_L\) in the case of a buck converter) to be negative when both the power mosfets are off. This allows us to derive the equation which models the behavior of the switching node voltage \(v_c\), and which remains valid for any operating mode. Fig. 6.1 (b) also shows the RC network \((R_{\text{sw}}\) and \(C_{\text{sw}}\)) at the switching node, which models the electrical equivalent of all the parasitic components at the node itself. \(R_{\text{sw}}\) and \(C_{\text{sw}}\), together with the output inductor \(L\), are responsible for the ringing at the switching node \(v_c\) when the converter operates in PWM-DCM or in PFM mode.

In fact, if \(R_{\text{sw}}\) and \(C_{\text{sw}}\) are not included in the model, whenever the current \(i_c\) becomes negative the switching node voltage \(v_c\) immediately jumps to the value of the input voltage \(V_{\text{in}}\) plus the forward voltage of the body diode of the high side mosfet. In the real circuit this variation is slowed down by an equivalent parasitic capacitance at the switching node, that here is modeled by \(C_{\text{sw}}\). Moreover, to model the damping of the ringing due to all the parasitic resistances in
Figure 6.2: a) Inductor current $i_L(t)$ and switching node voltage $v_c(t)$ waveforms obtained from a buck converter model where a) the power mosfets are modeled as ideal switches, b) $R_{sw}$ and $C_{sw}$ are included in the model. The converter works in PWM-DCM mode with asynchronous rectification.

the path followed by $i_c$, the equivalent resistance $R_{sw}$ is included in the model. $R_{sw}$ and $C_{sw}$ can be estimated directly from the experimental prototype waveforms by measuring amplitude and frequency of the oscillation.

In Fig.6.2 comparison between the model of the buck converter using ideal switches (Fig. 6.2(a) and the model using the variable resistors (Fig. 6.2(b)) is provided to highlight the main contribution given by the model developed. in Fig. 6.2(a) the inductor current is not allowed to assume negative values. Therefore the voltage at the switching node remains at the value $V_d$ for all the time where $i_L(t) = 0$. In Fig. 6.2(b), instead, the equivalent capacitance $C_{sw}$, together with the equivalent resistance $R_{sw}$, are added to the model of the switch network to allow the switching node voltage $v_c(t)$ to oscillate. When the ringing is over the voltage $v_c(t)$ stabilizes at the value of the output voltage $V_{out}$.

Accurate modeling of the voltage at the switching node is the key parameter to be able to estimate switching frequency and output voltage ripple of the converter in PFM mode.

The discrete time model can be easily derived by considering the Kirchoffs laws and component relations around the structure:

$$i_a = i_b + i_c + i_p \quad (6.2)$$
where $i_a = gm_{HS}(v_a - v_c)$, $i_b = gm_{LS}(v_c - v_b)$, $i_p = v_c(sC_{sw})/(1 + sC_{sw}R_{sw})$.

Solving with respect to the voltage at the switching node $v_c$ and moving to the discrete time domain by mean of the backward Euler transform $s = (1 - z^{-1})/T$ for a suitable sampling time $T$, Eq. (6.3) is obtained:

$$v_c = \frac{(gm_{HS}v_a + gm_{LS}v_b + i_c)(1 - \alpha_{sw}z^{-1}) + v_c z^{-1} \alpha_{sw}(gm_{HS} + gm_{LS} + gm_{sw})}{gm_{HS} + gm_{LS} + gm_{sw} \alpha_{sw}}$$

(6.3)

where $\tau_{sw} = C_{sw}R_{sw}$, $gm_{sw} = 1/R_{sw}$ and $\alpha_{sw} = \tau_{sw}/(T + \tau_{sw})$. The choice of $T$ determines the time resolution of the simulation and the matching between the real circuit and the model.

For digitally controlled SMPS, it might be useful to set $T$ equal to the resolution of the DPWM ($T_{DPWM}$) because this is the minimum time variation of the pulse width modulated signal that controls the converter. Since $T_{DPWM}$ is much smaller than the switching period $T_s$ the Euler transformation does not introduce significant distortion in the converter waveform [93].

Similarly, output inductor and capacitor have to be modeled in the discrete time as well, using the same procedure. Therefore, starting from the continuous time equation of the inductor current $i_L$ and output voltage $v_{out}$, (6.1) is obtained:

$$i_L(z) = \frac{T}{L(1-z^{-1})}(v_c - v_{out} - R_Li_L(z))$$

(6.4)

$$v_{out}(z) = \frac{T}{C(1-z^{-1})}i_{cap} + R_{esr}i_{cap}$$

The whole discrete time model of the buck converter power stage is achieved implementing (6.3) and (6.1) in HDL language and connecting all the blocks together, as shown in Fig. 4. In Fig. 4 the switch model represents the switch network of Fig. 6.1 (b) while inductor and capacitor models implement equations in (6.1). It has to be noticed that the switch network in Fig. 6.1 (b) is identified by three electric terminals, each described by means of the current flowing in it and its voltage referred to ground. Input signals $PWM_{HS}$ and $PWM_{LS}$ can be considered standard digital signal while $v_a$, $v_b$, $v_c$ and $i_a$, $i_b$, $i_c$ can be either input or output signals. Therefore the switch model can be extended to any dc-dc converters, such as boost, inverting buck-boost and non inverting buck-boost, once the input signals are defined according to the dc-dc converter topology to model. Once the power stage is modeled, close loop simulation can be performed when the behavioral models of the mixed analog-digital block DPWM and ADC are available.
6.2 Implementation of the discrete buck converter model

Implementation of the model by HDL language is accomplished rewriting equations (6.3) and (6.1) as difference equations after having defined the following discrete time variables:

\[ F_1[n] = g_m\text{sw} \cdot \alpha_{sw} + g_mH[n] + g_mL[n] \]  
\[ F_2[n] = g_m\text{sw} + g_mH[n] + g_mL[n] \]  
\[ K_{drv}[n] = g_mH[n] \cdot v_a[n] + g_mL[n] \cdot v_b[n] - i_c[n] \]

where \( n \) denotes the discrete sampling time \( nT \). Therefore, the switching node voltage \( v_c \) in (6.3) can be expressed as a function of the sampling step \( n \):

\[ v_c[n] = (\alpha_{sw} \cdot (F_2[n-1] \cdot v_c[n-1] - K_{drv}[n-1]) + K_{drv}[n]) / F_1[n] \]  
\[ i_L[n] = (1/K_{RL})i_L[n-1] + (K_L/K_{RL}) \cdot (v_c[n] - v_{out}[n]) \]  
\[ v_{out}[n] = v_{out}[n-1] + K_{resr}(i_L[n] - I_{out}[n]) - R_{resr}(i_L[n-1] - I_{out}[n-1]) \]

where \( K_L = T/L \), \( K_{RL} = (1 + (T/L)R_L) \) and \( K_{resr} = (R_{esr} + T/C) \). Equations (6.8), (6.9) and (6.10) constitute the main equations of the converter model and their implementation in HDL language now results to be straightforward.

6.3 FPGA-based digital multi-mode buck converter prototype

In order to be able to test the digital multi-mode controller and validate the simulation results a digital multi-mode buck converter prototype has been developed. The experimental prototype consists of a discrete components buck converter power stage, a load board (including the ADC converter and the variable load for the converter) and a Xilinx Virtex 4 FPGA development
board where the digital controller is implemented. Fig. 6.3 shows the experimental prototype developed.

Parameters of the converter are listed in Table 6.1. The converter operates in PWM-CCM using a PID compensator to obtain a bandwidth \( f_c \) of \( f_s/20 \) while a PI regulator is used in PWM-DCM to achieve a bandwidth of \( f_s/35 \). The PFM mode of operation is entered when the output current is below 50mA and the maximum output voltage ripple allowed is 80mV.

6.4 Experimenta results

In order to validate simulation results, inductor current \( i_L \) and switching node voltage \( v_c \) waveforms of the experimental prototype are compared to simulation results obtained for the discrete time model of the converter implemented in verilog HDL language. Fig. 6.4 and 6.5 show switching node voltage \( v_c \) and inductor current \( i_L \) of the buck converter prototype working in PWM-CCM (output current \( I_{out} = 250\text{mA} \)) and PWM-DCM (output current \( I_{out} = 100\text{mA} \)) are
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{\text{in}}$</td>
<td>20V</td>
<td>Input voltage</td>
</tr>
<tr>
<td>$V_{\text{out}}$</td>
<td>4V</td>
<td>Output voltage</td>
</tr>
<tr>
<td>$f_s$</td>
<td>780kHz</td>
<td>Switching frequency</td>
</tr>
<tr>
<td>$L$</td>
<td>10$\mu$H</td>
<td>Output inductor</td>
</tr>
<tr>
<td>$R_L$</td>
<td>130m$\Omega$</td>
<td>Inductor DCL resistance</td>
</tr>
<tr>
<td>$C$</td>
<td>50$\mu$F</td>
<td>Output capacitor</td>
</tr>
<tr>
<td>$R_{\text{esr}}$</td>
<td>5m$\Omega$</td>
<td>Output capacitor ESR</td>
</tr>
<tr>
<td>$C_{\text{sw}}$</td>
<td>1.3nF</td>
<td>Equivalent capacitance at the switching node</td>
</tr>
<tr>
<td>$R_p$</td>
<td>1.97$\Omega$</td>
<td>Equivalent resistance at the switching node</td>
</tr>
<tr>
<td>$R_{\text{on}}$</td>
<td>13m$\Omega$</td>
<td>On resistance of the two switches</td>
</tr>
<tr>
<td>$g_{\text{mOFF}}$</td>
<td>$1\mu\Omega^{-1}$</td>
<td>Off resistance of the two switches</td>
</tr>
<tr>
<td>$q_{\text{v,ad}}$</td>
<td>20mV</td>
<td>Output voltage resolution</td>
</tr>
<tr>
<td>$q_{\text{dpwm}}T_s$</td>
<td>1.25ns</td>
<td>Time resolution of the DPWM</td>
</tr>
<tr>
<td>$n_{\text{dpwm}}$</td>
<td>10</td>
<td>Number of bits required by the DPWM</td>
</tr>
<tr>
<td>$n_{\text{comp}}$</td>
<td>14</td>
<td>Number of bits required for computing the control law</td>
</tr>
</tbody>
</table>

Table 6.1: Digital multi-mode buck converter parameters.
Figure 6.4: Inductor current $i_L$ and switching node $v_c$ waveforms obtained from a) the discrete time model of the buck converter implemented in verilog HDL, b) the experimental prototype. The converter is working in PWM-CCM with output current $I_{out} = 250\text{mA}$.

compared to the waveforms obtained by the discrete time model of the converter implemented in verilog HDL language. Similarly, Fig. 6.6 shows converter waveforms when converter operates in PFM mode. The output voltage $v_{out}$ is shown only when the converter works in PFM (Fig. 6.6) because in PWM (Fig. 6.4 and 6.5) the comparison does not provide any additional insight, while it becomes a key parameter in PFM mode.

As shown in Fig. 6.5 and 6.6 modeling of the switching node parasitic components leads to accurate modeling of converter behavior in DCM and converter parameter prediction in PFM mode of operation. Table 6.2 summarizes the experimental measurements of main converter
Figure 6.5: Inductor current $i_L$ and switching node $v_c$ waveforms obtained from a) the discrete time model of the buck converter implemented in verilog HDL, b) the experimental prototype. The converter is working in PWM-DCM with output current $I_{out} = 100\text{mA}$.
Figure 6.6: Inductor current $i_L$ and switching node $v_c$ and output voltage $v_{\text{out}}$ waveforms obtained from a) the discrete time model of the buck converter implemented in verilog HDL, b) the experimental prototype. The converter is working in PFM with output current $I_{\text{out}} = 15\, \text{mA}$. 

98
parameters in PFM mode.

In Fig. 6.7 transition from PWM-DCM to PWM-CCM is used as an example to highlight the capability of the model to predict converter dynamics. Since the digital controller uses different control algorithms to perform output voltage regulation in PWM-DCM and PWM-CCM (PI and PID compensators respectively), in Fig. 6.7 the integral part of the PID compensator slowly charges up once PWM-CCM mode is entered, leading to unacceptable undershoot of the output voltage $v_{out}$, as predicted by the model. The easiest solution to avoid this issue is to set an initial value for the integral duty cycle so that, any time the PWM-CCM mode is entered, the integral part of the duty cycle is already at its steady state value. Since the duty cycle in PWM-CCM equals its ideal value $D = V_{out}/V_{in}$ for currents close to the boundary PWM-CCM - PWM-DCM (because the conduction loss is minimized), the initial duty cycle value can be set to its ideal value $D$. Fig. 6.8 shows the converter waveforms when this initial condition is applied. An alternative would be to store the last value of the duty cycle before changing the operating mode to PFM. Similarly, the same issue is found when transitioning from PFM to PWM-CCM as well.

In Fig. 6.9, instead, another issue is reported when the output current $I_{out}$ approaches its critical value $I_{out_{cr}}$ in PWM-DCM mode. Since a mere PI controller is used to perform output voltage

<table>
<thead>
<tr>
<th>$I_{out}$ [mA]</th>
<th>$\Delta i_L$ A</th>
<th>$\Delta v_{out}$ mV</th>
<th>$f_{s_{PFM}}$ [kHz]</th>
<th>$\Delta i_L$ A</th>
<th>$\Delta v_{out}$ mV</th>
<th>$f_{s_{PFM}}$ [kHz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6</td>
<td>74</td>
<td>5.26</td>
<td>1.63</td>
<td>68.5</td>
<td>4.2</td>
</tr>
<tr>
<td>34</td>
<td>1.6</td>
<td>72</td>
<td>11.63</td>
<td>1.63</td>
<td>66.8</td>
<td>9.57</td>
</tr>
<tr>
<td>41</td>
<td>1.6</td>
<td>76</td>
<td>14.28</td>
<td>1.63</td>
<td>66</td>
<td>11.83</td>
</tr>
<tr>
<td>49</td>
<td>1.6</td>
<td>74</td>
<td>16.67</td>
<td>1.63</td>
<td>66.5</td>
<td>14.08</td>
</tr>
</tbody>
</table>

Table 6.2: Comparison among experimental prototype and discrete time model waveforms in PFM mode of operation for different output current values ($I_{out}$). $\Delta i_L$ is the inductor current peak, $\Delta v_{out}$ is the output voltage ripple while $f_{s_{PFM}}$ is the obtained switching frequency.
regulation the amount of charge provided to the output capacitor in response to a unity step of the error signal $e[n]$ exceeds the amount required to regulate the output voltage. The output capacitor, therefore, quickly charges up driving the output voltage to hit the upper level of the regulation band with a consequent reaction of the compensator that will set the duty cycle value to its previous value. The output capacitor will then slowly discharge due to the low output current required by the load until the output voltage will hit the lower level of the regulation band again. The error signal $e[n]$ results in a periodic signal with pattern 1 0 -1 0 0 0 0 0 1 corresponding to a sawtooth oscillation of the output voltage with amplitude of exactly $q_{v,ad}$. Since the oscillation has a triangular waveshape it cannot be predicted with the standard theory of the describing functions [94] while it is predicted by the model.
Figure 6.7: Output voltage $v_{out}$ and inductor current $i_L$ for a) the discrete time model of the converter implemented in verilog HDL, b) the experimental prototype, when the converter changes operating mode. Since PWM-DCM and PWM-CCM have different control algorithms (PI and PID controllers respectively) the integral part of PID controller in CCM slowly charges up leading to an unacceptable undershoot of the output voltage.
Figure 6.8: Output voltage $v_{out}$ and inductor current $i_L$ for a) the discrete time model of the converter implemented in verilog HDL, b) the experimental prototype, when the converter changes operating mode. An initial condition on the average duty cycle in PWM-DCM is applied solving the issues in Fig. 6.7.
Figure 6.9: Output voltage $v_{out}$ and inductor current $i_L$ for a) the discrete time model of the converter implemented in verilog HDL, b) the experimental prototype when output current is close to the value $I_{out,r}$. The frequency of the oscillation is 16kHz in the experimental prototype while 18kHz is the frequency predicted by the model.
Chapter 7

Conclusions and future work

The main topic investigated in this work was the efficiency optimization of dc-dc converters, with particular attention to the light load case. First, converter power loss was investigated in PFM. A Labview based testing system was set up to control efficiency measurements and to communicate with an FPGA based digital controller. The objectives was to achieve automated efficiency characterization and to use these results to enable adaptive controller operation aimed at optimizing light-load efficiency. In general, depending on the switching and conduction losses in the converter, efficiency improvement at light load can be achieved by adjusting the energy transferred to the load during a PFM pulse as a function of the load. Following this concept, in this dissertation an automated efficiency characterization system was presented which, in conjunction with a power loss model, enables selection of controller timing parameters (such as the switch on time in PFM). A simple digital load current estimation technique has been proposed to enable adaptive operation in PFM.

Afterwards, the structure of a digital multi-mode controller for dc-dc converters has been described with the purpose to highlight the main design issues related to the operating mode management of the converter. In order to design the multi-mode controller and simulate all its functionalities directly in the digital domain, a discrete time large-signal model of the converter power stage has been developed with the aim to properly model the behavior of the converter, especially when its operating mode is changed. The main target of the model is to allow designers to study converter design issues related to the transition from one operating mode to another.
and to be able to develop and test controllers where an operating mode management algorithm is implemented. Conditions for automatic detection of the operating conditions of the converter have been proposed and tested in a FPGA-based multi-mode controller validating simulation results. Finally a serial digital PID architecture has been presented for applications where small devices and low power consumption are demanded. Area minimization is achieved by the use of the serial architecture that, together with the floating point representation adopted for the regulator parameters, involves only one small multiplier and one main adder compared to the most common implementations; large multipliers or wide size LUTs used to store regulator parameters are thus avoided. In fact, only a small number of bits is used to represent a large range of converter parameters, reducing the size of the LUTs and making the architecture suitable for multi-mode converters.

in the context of multi-mode converters there are still some subjects that deserve to be investigated. Many research activities are focused on the development of PFM controllers featuring low power consumption and adaptive controller parameters for maximizing the efficiency as a function of the output current. In digital controllers the current peak PFM controller is commonly replaced by a constant on time PFM controller and therefore strategies for efficiently performing voltage feed forward (i.e. compensation for input voltage variations) need to be investigated. Moreover, efficiency at light load can be further improved if synchronous rectification is used, at the expense of additional logic to compute the on time of the low side switch. Transition from the PFM mode to PWM-CCM and PWM-DCM modes can be further optimized investigating alternative control strategies to quickly supply the energy demanded by the load to the output capacitor, thus resulting in a faster transient. Indeed, the controller has to force the output voltage to be within the regulation band to achieve the targeted specifications. In case the compensator fails and the output voltage drops out of the window of the ADC converter, the derivative part of the PID compensator gets to zero and no exact information is available of the output voltage. Therefore, the PID algorithm is reduced to a PI control law thus leading to a slower transient. Nonlinear control strategies, therefore, need to be researched to either avoid the output voltage to drop outside the regulation band or perform a fast recovery whenever this
situation occurs.

Finally an interesting topic to be analyzed is the origin of the limit cycle characterizing the behavior of the converter for output currents close to the critical value $I_{\text{out,cr}}$ and conditions for avoiding such oscillation (if existing) need to be determined.
Bibliography


Appendix A

Review of sampled data systems

The main target of this chapter is to introduce the design of digital controllers starting from an equivalent analog system and go through the basic theory of the sampled data systems to explain the basic elements of a digital loop.

Starting from a generic analog system shown in Fig. 7.1, our target is to design a digital system using the know how of analog design we already own. The approach that will be followed in this chapter is to \textit{emulate} the behavior of a continuous controller in order to find the best digital approximation of the the controller itself. The final digital system will then have the structure presented in Fig. 7.2.

The analog system in Fig. 7.1 consists of the analog plant we want to control, described by its transfer function $G_p(s)$, and the analog controller with $G_c(s)$ as a transfer function. The variable to be adjusted is the output signal of the plant $y(t)$ which will be sensed by a sensor

![Figure 7.1: Generic analog system. The analog plant $G_p(s)$ is controlled by the analog compensator $G_c(s)$ that adjusts the output signal $y(t)$ of the plant to the desired value $r(t)$.](image)

Figure 7.2: Generic digital system. The same analog plant of Fig. 7.1 is controlled by a digital controller carefully designed to approximate the action of the analog controller $G_c(s)$.

having transfer function $H(s)$. The output of the sensor $s(t)$ is then compared to a reference signal $r(t)$ and the resulting error signal $e(t) = r(t) - s(t)$ is processed by the analog controller to generate the control signal $c(t)$, according to the control law implemented by the controller. Compared to the analog system the digital system involves two new blocks to convert signals from the continuous to discrete time domain and viceversa: the continuous to digital converter block and the digital to continuous converter block.

In the following sections these two blocks will be analyzed pointing out the effects introduced by the conversion of a signal from a domain to the other.

### 7.1 The continuous to digital block

The continuous to digital block has the role to convert the continuous time signal at its input into a digital one. This block can be modeled as a sampler followed by an Analog to digital converter (A/D) block. The ideal sampler samples the value of its continuous input signal $s(t)$ repetitively at regular instances of time which are $T$ seconds apart. $T$ is one of the important design parameters in digital control and is called the sampling period. Afterwards, samples of the continuous signal are converted into (binary) numbers for being processed by the digital compensator. This part is carried out by the Analog-to-Digital (A/D) converter.

Given a sampling period $T$, we denote the discrete samples of a continuous signal $s(t)$ by $s(nT)$ or, with a little abuse of notation, by $s(n)$ where $n$ is an integer number.

In Fig. 7.3 a sampler is shown schematically together with its waveforms. The continuous time
signal \( x(t) \) is sampled by the sampler with period \( T \) and the resulting signal \( x^*(t) \) is still a sequence of pulses with amplitude equal to the amplitude assumed by the original signal \( x(t) \) at the sampling instant. After being processed by the analog to digital block the digital signal \( x[n] \) is obtained.

### 7.1.1 Aliasing

The ideal sampler in Fig.7.3 is used to achieve a sampled version of its input signal with sampling rate \( f = 1/T \). The sampled signal \( x^*(t) \) can be seen as the product of the original signal \( x(t) \) with a string of impulses. It is defined as:

\[
x^*(t) := \sum_{k=-\infty}^{+\infty} x(t) \delta(t - kT)
\]

and its Laplace transform is given by:

\[
X^*(s) = \frac{1}{T} \sum_{i=-\infty}^{+\infty} R(s - j\omega_i)
\]
Figure 7.4: a) Spectrum of the original signal $x(t)$. b) Spectrum of the sampled signal $x^*(t)$.

where we assume $\omega_s = 2\pi f = 2\pi/T$.

Imposing $s = j\omega$ the frequency response of $x^*(t)$ can be found:

$$X^*(j\omega) = \frac{1}{T} \sum_{i=-\infty}^{+\infty} R(j(\omega - \omega_s i)) \quad (7.3)$$

Therefore (7.3) shows that the spectrum of the sampled signal still preserves the spectrum of the original signal $x(t)$ with the addition of replicas centered at multiplies of the sampling frequency. This result sets a first limit of the digital systems. Let us suppose that the signal $x(t)$ has a limited bandwidth where $\omega_b$ is its maximum frequency in the spectrum, and let us sample it with sampling frequency much higher than $\omega_b$. The sampled signal $x^*(t)$ then will have the spectrum shown in Fig. 7.4. The spectrum of the original signal can then be extracted filtering the sampled signal $x^*(t)$ by the use of a low pass filter with cutoff frequency selected in the interval $[\omega_b, \omega_s - \omega_b]$. This means that the original signal $x(t)$ can be reconstructed from its sampled version $x^*(t)$. If the sampling frequency is as low as that $\omega_b > \omega_s - \omega_b$ the baseband spectrum and its replica centered at $\omega_s$ mixed up so that if the same low pass filter is now used to extract the spectrum of $x(t)$ the resulting spectrum will have some components of its replica leading to a distorted version of $x(t)$ (Fig. 7.5). This effect is called aliasing. Based on these considerations the sampling (or Nyquist-Shannon) theorem states that, if the signal $x(t)$ has a spectrum with maximum frequency $\omega_b$, the signal $x(t)$ can be reconstructed from its sampled version $x^*(t)$ whenever the sampling frequency satisfies

$$\omega_s > 2\omega_b \quad (7.4)$$

The sampling theorem states that the switching frequency has not to be chosen too low so that
the sampled version of the input signal holds the same behavior of its original signal. On the other side, there is a practical restriction when choosing $f$ too high as well. This restriction is due to the fact that the digital compensator uses words of a finite resolution to compute the control law, as will be addressed later on in this chapter. The Analog to Digital converter (A/D) itself also uses a limited number of bits to codify the continuous input signal into a digital one. The resulting round off error makes the device is nonlinear. The effect of the nonlinearities of the A/D converter in the system are application dependent and therefore will not be considered in this chapter.

### 7.2 The digital to continuous block

The digital to continuous block converts the digital signal coming from the digital compensator back to a continuous time signal. To accomplish this, a Digital-to-Analog (D/A) Converter is used to convert binary numbers into analog voltages. Then a second circuitry, called hold circuit, is used to convert the voltages into a continuous time signal. The easiest conversion method is to hold voltage constant across a sampling period, that is, the value of the continuous signal $x(t)$ will be held constant at $x[n]$ between $nT; (n + 1)T)$. Such a method is called zero-order hold (ZOH), where a polynomial of degree zero is used to connect the sample points. If the sample points are connected by a straight line we call the hold circuit a first-order hold. The structure of the digital to continuous block is shown in Fig. 7.6 together with its waveforms. When representing samples by a string of pulses, the function of a ZOH can be succinctly expressed.
using its transfer function, $ZOH(s)$. Derivation of $ZOH(s)$ is straightforward. Let $\delta(t)$ be the input to a $ZOH$. The signal output by the $ZOH$ is a rectangular shaped signal starting at time 0, with height 1 and width $T$. The system transfer function would then be the Laplace transform of this rectangular function, which can be easily found to be:

$$ZOH(s) = \frac{1 - e^{-sT}}{s} \quad (7.5)$$

Since the $ZOH$ is used to reconstruct the original signal from its digital version, it is of our interest to analyze its frequency response to understand this process even from a spectral point of view. From 7.5 imposing $s = j\omega$ the (7.6) is achieved.

$$ZOH(j\omega) = e^{-j\omega T/2} T \text{sinc} \left( \frac{\omega T}{2} \right) \quad (7.6)$$

This means that the $ZOH$ acts as a low pass filter attenuating the frequency components higher than the sampling frequency $f$ but not completely removing them. This is not a problem, as most plants have low pass characteristics and remove the high frequency aliases. The corresponding effect in time domain is replacing the high frequency jumps between steps with a smooth curve connecting the samples.
Figure 7.7: Delay introduced by the ZOH circuit. Compared to the original signal a) the average of the signal output b) by the ZOH has a delay of $T/2$.

### 7.2.1 The effect of sampling

The introduction of the sampler and the ZOH in the loop leads to a degradation of converter features, such as step response and stability, at high frequency as a function of the sampling frequency. Once the signal to control is sampled and the error signal has been processed by the controller, the control signal generated by the controller must be converted to the continuous time domain by a D/A and hold before being accepted by the plant. The output signal of the ZOH is a staircase function that, as illustrated in Fig. 7.7, lags the values of $x(nT)$, on the average, by $T/2$. This effect can be seen also from its frequency response (7.6) because of the presence of the term $e^{-j\omega T/2}$ which degradates the phase of the system to control and therefore the phase marging of the system. In general a delay in the feedback loop can be expressed by a term $e^{-st_d}$ where $t_d$ is the total delay in the loop.

In order to easily analyze the effect of this delay it may be useful to approximate the delay term $e^{-st_d}$ with the structure in (7.2.1). $G_d(s)$ is called first order Padé approximant.

$$G_d(s) = \frac{1 - \frac{1}{2}t_ds}{1 + \frac{1}{2}t_ds}$$  \hspace{1cm} (7.7)

The negative phase shift introduced by the ZOH is approximately equal to:

$$\delta\Phi \approx -\frac{\omega T}{2}$$  \hspace{1cm} (7.8)
7.3 Selection of the sampling frequency

One of the first parameters to set when designing a digital feedback loop is the sampling frequency, which strictly depends on the application the loop is designed for. Even though, few considerations can be made in order to address this choice.

As stated by the sampling theorem, the sampling frequency cannot be chosen too low otherwise the sampled signal will result distorted compared to its continuous time counterpart. Moreover, the condition $\omega_s > 2\omega_b$ is not enough to guarantee a smooth time response. As a rule of thumb, a smooth response is obtained when $\omega_s \geq 20\omega_b$.

On the other hand, there are some practical limitations for choosing the sampling frequency too high. As the behavior of a continuous time system is described by a transfer function in the Laplace ($s$) domain, a discrete system is described by a transfer function in the $Z$ domain. Comparing the definitions of the Laplace transform and the $Z$-transform for a continuous signal $x(t)$ and its equivalent discrete signal $x[n]$ sampled with sampling period $T$, it turns out that the $Z$-transform $X(z)$ is the sampled version of the Laplace transform $X(s)$, whenever $z = e^{sT}$.

\[
\begin{align*}
X(s) &= L\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-st} \quad \text{Laplace transform} \\
X(z) &= Z\{x(t)\} = \sum_{k=-\infty}^{\infty} x[k] z^{-k} \quad \text{$Z$-transform}
\end{align*}
\]

Equation 7.9 establishes a relationship between the location of zeros and poles and the sampling frequency $f = 1/T$. In general, the higher the sampling frequency the closer the zeros and poles to $-1$ (the highest frequency in the digital domain). Therefore, at high sampling frequencies, zeros and poles of the digital compensator will be clustered around the value $-1$ leading to the need of a high number of bits to distinguish one from the others. As a general rule, high sampling frequency leads to an improvement of the performances of the digital controller but with the drawbacks that a faster conversion speed of the A/D is required, less time is available for the computation of the control law that also involves words with a larger number of bits. As a consequence this results in a cost increase of the system.
7.4 Digital compensator design

In this section the design of the digital compensator will be addressed. In general, the digital compensator can be designed following one of the two approaches:

1. **Direct design approach:** In this case zeros and poles of the compensator are placed directly in the Z domain so that the transfer function of the digital controller $G_c(z)$ is directly obtained. This approach requires the designer to have a good knowledge of the digital domain and results not so suitable to designers having a good experience of analog design.

2. **Emulation approach:** This is the approach that will be followed in the next chapters because it allows designers to take advantage of the knowledge they already own on the design of analog controllers for designing a digital compensator which has the same features if its equivalent analog counterpart. The compensator is designed in the analog domain taking into account of the delay $T/2$ introduced by the ZOH. Afterward a discrete equivalent of the designed compensator is derived.

Assuming that the emulation approach is chosen we investigate the techniques for deriving the discrete equivalent of the designed analog controller. There are two methods to accomplish this: the first method is to use numerical integration to approximate a system described by a set of linear differential equations. The second method, instead, uses the idea of pole-zero mapping to find the discrete-time counterpart of a continuous-time transfer function, recalling that a continuous-time pole located at $s = s_0$ corresponds to a discrete-time pole at $z = e^{s_0 T}$.

7.4.1 Design of discrete equivalents via numerical integration

To obtain the discrete equivalent of a transfer function via numerical integration, the first step is to write the system differential equations and afterwards the derivative of the state variables
of the system will be approximated by one of the following numerical integration techniques:

$$\dot{x}(n) \approx \frac{x[n+1] - x[n]}{T} \quad \text{Forward rectangular rule}$$

$$\dot{x}(n+1) \approx \frac{x[n+1] - x[n]}{T} \quad \text{Backward rectangular rule} \quad (7.10)$$

$$\frac{\dot{x}(n)+\dot{x}(n+1)}{2} \approx \frac{x[n+1] - x[n]}{T} \quad \text{Trapezoid rule}$$

where $\dot{x}(n)$ is the derivative of the signal $x(t)$.

This means that, given a continuous signal $x(t)$, the area delimited by the signal can be approximate with a staircase function in the case of the forward and backward rectangular rule or with a function that connect two adjacent sampling point through a trapezoidal function, as for the case of the trapezoid rule. Fig. 7.8 clarifies this concept. The operation can be carried out directly on transfer function if the above equations are translated into frequency domain. Recalling that each discrete time left-shift by $n$ corresponds to a $z^n$ multiplying factor in the Z-domain, and each $\frac{d^n}{dt^n}$ in continuous time domain corresponds to an $s^n$ multiplying factor in the Laplace domain, the following relationships between $s$ and $z$ are achieved:

$$s \rightarrow \frac{z-1}{T} \quad \text{Forward rectangular rule}$$

$$s \rightarrow \frac{z-1}{zT} \quad \text{Backward rectangular rule} \quad (7.11)$$

$$s \rightarrow \frac{2(z-1)}{Tz+1} \quad \text{Trapezoid rule or bilinear transformation}$$
The trapezoid rule is also called Tustin’s method. Each of the above relations can be viewed as a map from z-plane to s-plane, therefore the first concern is to analyze how the stable region of s-plane (that is, the left-half plane described by $\text{Re}(s) < 0$) is mapped to z-plane, in order to see whether the stability is maintained in the Z-domain. As illustrated in Fig. 7.9, under trapezoid rule the stable region of s-plane is mapped exactly to the stable region of z-plane. Moreover we can say that

1. Under trapezoid rule, the discrete time system is stable if and only if the continuous time system is stable.

2. Under backward rectangular rule, the discrete time system is stable if the continuous time system is stable.

Despite the congruence of stability under the trapezoid rule, the scheme suffers from a serious drawback: the entire imaginary axis is mapped to only $2\pi$-length of the unit circle. This results in a great amount of distortion, which is overcome by introducing an extension of Tustin’s rule. The approach is called bilinear with prewarping. The idea is to ensure that at a pre-specified frequency, the discrete time transfer function has exactly the same characteristics (gain and phase) as the corresponding continuous time transfer function. The transformation is bilinear like Tustin’s, however the constant coefficient (which used to be $\frac{2}{\tau}$) is now different. The new
mapping function is:
\[ s \rightarrow \frac{\omega_{cr}}{\tan(\omega_{cr}T/2)} \frac{z - 1}{z + 1} \text{ Bilinear transformation with prewarping} \quad (7.12) \]

Therefore the new discrete equivalent \( G_{BP}(z) \) of the analog controller \( G_c(s) \) can be expressed as:

\[ G_{BP}(z) = G_c(s) \bigg|_{s = \frac{\omega_{cr}}{\tan(\omega_{cr}T/2)} \frac{z - 1}{z + 1}} \quad (7.13) \]

### 7.4.2 Zero-pole matching equivalents

As seen before, a pole \( s_0 \) of the Laplace transform of a continuous time signal is related to a pole \( z_0 \) of the Z-transform of signal samples according to \( z_0 = e^{s_0T} \). The idea of zero-pole matching is to use this mapping to determine the location of zeros as well. The procedure is outlined below.

1. A continuous-time pole at \( s = s_0 \) is mapped to a discrete-time pole at \( z = e^{s_0T} \)

2. A continuous-time, finite, zero at \( s = s_0 \) is mapped to a discrete time zero at \( z = e^{s_0T} \)

3. Let \( m \) and \( n \) be the degree of numerator and denominator of a continuous time transfer function. If \( m < n \), then the system will have \( n - m \) zeros at infinity. Each continuous time zero at \( s = 1 \) is mapped to a discrete time zero at \( z = -1 \). In general we would like the highest continuous time frequency of \( s = j\infty \) to correspond to the highest discrete time frequency of \( z = e^{j\frac{\pi}{2T}} = -1 \)

4. Finally, we adjust gain of the discrete equivalent by making continuous time and discrete time systems gains equal at a pre-specified frequency, in other words, we demand the following equality:

\[ G_{zp} \left( e^{j\omega_0T} \right) = G_c(j\omega_0) \quad (7.14) \]

Often in practice we require the DC gains to be equal, that is, \( \omega_0 = 0 \), in which case we must have:

\[ G_{zp}(1) = G_c(0) \quad (7.15) \]
Appendix B

MATLAB design source code

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
% CONFIGURATION FILE FOR DIGITAL VOLTAGE MODE BUCK CONVERTERS %
%
% Author: Marco Meola %
% University of Trieste %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% display('//////////////////////////////////////////////////////////////////');
display('');
display('DIGITAL CONTROL FOR DC-DC CONVERTERS TRAINING');
display('');
display('Author: Marco Meola');
display(' University of Trieste');
display('');
display('This is the configuration file for a digital voltage mode buck converter.');
Three operating modes are considered:

1) Continuous Conduction Mode (PWM-CCM)
2) Discontinuous Conduction Mode (PWM-DCM)
3) Pulse Frequency Modulation (PFM)

% BUCK CONVERTER PARAMETERS DEFINITION

clear all;
% Output inductance characteristics
L=10e-6;
Rl=130e-3;
% Output capacitor characteristics
C=50e-6; %100e-6;
ESR=5e-3;
% Switching frequency
fs = 780e3; % (Hz)
Ts=1/fs;
%Buck converter definition
B=Buck_vm(L,Rl,C,ESR,Ts);

% OPERATING CONDITIONS
Vin=20; % Nominal input voltage (V)
Vo=4.01; % Nominal output voltage (V)
M=Vo/Vin; % Conversion ratio
Io=1.5; % Load current in CCM (A)
Rload=Vo/Io; %Load (Ohm)  
Hsense=1; %Gain of the sensing for the Vout  
Vref=Vo*Hsense; %Voltage reference to be used (V)  

% % PARAMETERS OF ADC AND DPWM BLOCKS  
q_vout = 20e-3; %Quantization step of the output voltage  
quad=(q_vout)*Hsense; %Quantization step of the ADC  
ndpwm=14; %number of bits used to represent the duty  
%cycle value d(n) input to the DPWM  
qudpwm=1/(2^ndpwm); %equivalent quantization step of the DPWM  
RegBandAmp=4; % Amplitude of the regulation band for a  
% window flash ADC  
esat=RegBandAmp*qad; % Saturation value of the error signal  

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%  
% CONTINUOUS CONDUCTION MODE %  
%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  

display('--------------------------------------');  
display('CONVERTER OPERATING CONDITIONS in CCM')  
display('--------------------------------------');  

%Defines the operating mode of the converter  
op=Buck_op_vm(B,Vin,Vo,Rload)  

%Hardware requirements for the DPWM  
disp('');  
disp('DPWM requiremets')
disp(sprintf('Minimum number of bits of the DPWM required : %g', round(log2(Hsense*Vin/qad))));
display('');

% Expected features of the converter in CCM
[DiL,LIR,DVosh,DVo]=BuckDesignData(Vin,Vo,fs,Io,L,C,ESR);
display(' Critical load resistance:')
Rcrit=(2*B.L)/((1-M)*B.Ts)

% % OPEN LOOP TRANSFER FUNCTION OF THE BUCK CONVERTER
display(' Control-to-output transfer function with ESR')
Gvd_ESR=Buck_tf_vm_ESR(B,op)
figure(1)
set(1,'Name','OPEN-LOOP Control-Output tf with ESR')
bode(Gvd_ESR,2*pi,2*pi*fs*10),grid
title('OPEN-LOOP Control-Output tf with ESR')

% % DELAYS DEFINITION AND OPEN LOOP TRANSFER FUNCTION OF THE CONVERTER
% AFFECTED BY THE LOOP DELAY
t_adc=(31-18)*40e-9; % delay introduced by ADC
t_comp=0; % computation delay
t_dpwm=M*Ts; % delay introduced by the trailing edge
% modulation
tg=0; % mosfets driver delay

td=t_adc+t_comp+t_dpwm+tg; % total loop delay
%Effect of digital loop delay in the frequency response of the converter
Gvd_used=Gvd_ESR;
set(Gvd_used,'inputdelay',td);

% DC gain of the power stage
[Gvdo,pho]=bode(Gvd_used,0);

%
% EQUIVALENT OPEN LOOP TRANSFER FUNCTION IN THE Z-DOMAIN
% Uncompensated loop transfer function of the converter
display(' Uncompensated loop transfer function affected by digital delays')
T=Hsense*Gvd_used;
Gvd_z=Ts*c2d(T,Ts,'imp')

figure(2)
set(2,'Name','OPEN-LOOP Control-Output tf with digital loop delays');
bode(Gvd_z,Gvd_ESR*Hsense,2*pi,2*pi*fs*10),grid
title('OPEN-LOOP Control-Output tf with(blue)/without(green) digital loop delays');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% COMPENSATION NETWORK DESIGN %
% %
% % CONTINUOUS CONDUCTION MODE %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fo=1/(2*pi*(L*C)^0.5); % resonant frequency
fc=fs/20; % cross over frequency
% choose your compensator!!

CompensatorCCM = 1; % CompensatorCCM = 0 : COPEC zero-pole matching PID
% CompensatorCCM = 1 : bilinear PID

if CompensatorCCM == 0

%----------------------------------------------------
% BILINEAR DIGITAL PID COMPENSATOR DESIGN
%----------------------------------------------------

%Standard PID parallel implementation
[Gcz, alpha, Kp, Ki, Kd] = bilinear(fo, fs, fc, Gvd_z);
[Gcrnd_z, alpharnd, Kprnd, Kirnd, Kdrnd, eprnd, eirnd, edrnd]
= standardPID(alpha, Kp, Ki, Kd, fs, RegBandAmp, qad, ndpmw);

%Parallel PID implementation (COPEC Short Term Course)
%[Gcz, K1, K2, K3, P, Ki] = bilparPID(fo, fs, fc, Gvd_z);
%[Gcrnd_z, K1rnd, K2rnd, K3rnd, Prnd, e1rnd, e2rnd, e3rnd]
= parallelPID(K1, K2, K3, P, fs, RegBandAmp, qad, ndpmw);

else

%----------------------------------------------------
% ZERO-POLE MATCHING DIGITAL PID COMPENSATOR DESIGN
%----------------------------------------------------

%Copec PID implementation
[Gcz, alpha, beta, gamma, Ki] = zero_pole_comp(fo, fs, fc, Gvd_z);
[Gcrnd_z, alpharnd, betarnd, gammarnd, e_alpha_rnd, e_beta_rnd, e_gamma_rnd] =
copecPID(alpha,beta,gamma,fs,RegBandAmp,qad,ndpwm);

% Standard PID parallel implementation

% [Gcrnd_z, alphanrd, Kprnd, Kirnd, Kdrnd, eprnd, eirnd, edrnd] =
standardPID(alpha, Kp, Ki, Kd, fs, RegBandAmp, qad, ndpwm);

end

% Open-loop compensated transfer function in the Z-domain
Tcomp = Gcz * Gvd_z;
figure(3)
set(3, 'Name', 'Compensated loop in CCM');
bode(Tcomp, 2*pi, 2*pi*fs*10), grid
title('Compensated loop in CCM');

display('');
display('--------------------------------------');
display('COMPENSATOR FEATURES IN CCM');
display('--------------------------------------');

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

136
% Open-loop compensated transfer function with rounded PID coefficients
Tcomprnd=Gcrnd_z*Gvd_z;
figure(4)
set(4,'Name','Compensated loop in CCM including rounding effects');
bode(Tcomp,Tcomprnd,2*pi,2*pi*fs*10),grid
title('Compensated loop in CCM with(green)/without(blue)
including the rounding effects ');

%
%---------------------------------------------------------------------
% LIMIT CYCLE CONDITIONS IN CCM
%---------------------------------------------------------------------
display(' ');
display('--------------------------------------');
display('LCO CONDITIONS IN CCM');
display('--------------------------------------');
% % check no-limit-cycle conditions
disp(' ')
disp('Check no-limit cycle conditions (A1, A2, B2 should be < 1)')
% Check A1
disp(sprintf('Check A1: 2*Gvdo*qdpwm*Hsense/qad equals: %g', 2*Vin*qdpwm*Hsense/qad))
if (2*Vin*qdpwm*Hsense/qad < 1)
disp('Rule A1 OK')
else disp('Failed A1 (>1)')
end
% Check A2
disp(sprintf('Check A2: 2*Gvdo*Hsense*Integral_gain equals: %g', Vin*Hsense*Ki*2))
if (2*Vin*Ki*Hsense < 1)
disp('Rule A2 OK')

else disp('Failed A2 (>1)')
end
%
% Check B1

[Gvdmx, x] = bode(Gvd_used,wcgt);
disp(sprintf('Check B1: 8*|Gvd(wcg)|*qdpwm*Hsense/(qad*pi) equals: %g', 8*Gvdmx*qdpwm*Hsense/(qad*pi)))
if ((8*Gvdmx*qdpwm*Hsense)/(qad*pi) < 1)
disp('Rule B1 OK')
else disp('Failed B1 (>1)')
end
%
% Check B2

disp('Check B2 (GM to be > 10.2)')
if (20*log10(Gmt) > 10.2)
disp('Rule B2 OK')
else disp('Failed B2: GM < 10.2 dB')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% DISCONTINUOUS CONDUCTION MODE %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
display(' ');
display(' ');
display(' ');
display('--------------------------------------');
display('CONVERTER OPERATING CONDITIONS IN DCM');
display('--------------------------------------');

Io_dcm=0.09; % load current used for the design
R_dcm=Vo/Io_dcm;
%Defines the operating mode of the converter
op_dcm=Buck_op_vm(B,Vin,Vo,R_dcm)

% OPEN LOOP TRANSFER FUNCTION OF THE BUCK CONVERTER

Gvd_ESR_dcm=Buck_tf_vm_ESR(B,op_dcm);
figure(5)
set(5,'Name','OPEN-LOOP Control-Output tf with ESR in DCM');
bode(Gvd_ESR_dcm,2*pi,2*pi*fs*10),grid
title('OPEN-LOOP Control-Output tf with ESR in DCM');

% OPEN LOOP TRANSFER FUNCTION OF THE CONVERTER AFFECTED BY THE LOOP DELAY

% Effect of digital loop delays in the frequency response of the converter
Gvd_used_dcm=zpk(Gvd_ESR_dcm);
set(Gvd_used_dcm,'inputdelay',td);
% DC gain of the power stage
[Gvdo_dcm,pho_dcm]=bode(Gvd_used_dcm,0);

%%
% EQUIVALENT OPEN LOOP TRANSFER FUNCTION IN THE Z-DOMAIN
display(' ');
display(' Uncompensated loop transfer function affected by digital delays')
Tunc_dcm=Hsense*Gvd_used_dcm;
Gvd_z_dcm=Ts*c2d(Tunc_dcm,Ts,'imp')

figure(6)
set(6,'Name','OPEN-LOOP Control-Output tf with digital loop delays in DCM');
bode(Gvd_z_dcm,Gvd_ESR_dcm*Hsense,2*pi,2*pi*fs*10),grid
title('OPEN-LOOP Control-Output tf with with digital loop delays in DCM');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% COMPENSATION NETWORK DESIGN %
% %
% % DISCONTINUOUS CONDUCTION MODE %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc_dcm=fs/45; %cross over frequency in DCM
%
% Compensator design procedure based on direct Z domain design
[Gcz_dcm,KpDCM,KiDCM]=PI_DCM(fs,fc_dcm,Gvd_z_dcm);
%
% Round off effects
[Gcrndz_dcm,KpDCMrnd,KiDCMrnd,epDCMrnd,eiDCMrnd]=standardPI(KpDCM,KiDCM,fs,RegBandAmp,qad,ndpwm);

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

% Compensated loop in DCM
Tcomp_dcm=Gcz_dcm*Gvd_z_dcm;
figure(6)
set(6,'Name','Compensated loop in DCM');
bode(Tcomp_dcm,2*pi,2*pi*fs),grid
title('Compensated loop in DCM');

display('--------------------------------------');

display('COMPENSATOR FEATURES IN DCM')
% solve & print template loop gain crossover freq and margins

disp(' Loop gain parameters in template design (DCM):');

[Gmt_dcm, Pmt_dcm, wcgt_dcm, wcpt_dcm] = margin(Tcomp_dcm);
disp(sprintf('Cross-over Frequency [Hz]: %g', wcpt_dcm/(2*pi)));
disp(sprintf('Phase Margin [deg]: %g', Pmt_dcm));
disp(sprintf('Gain Margin [dB]: %g', 20*log10(Gmt_dcm)));
disp(sprintf('-180 degrees frequency [Hz]: %g', wcgt_dcm/(2*pi)));

%plot closed loop tf with quantization effects
Tcomprnd_dcm=Gvd_z_dcm*Gcrndz_dcm;
figure(7)
set(7,'Name','Compensated loop in DCM with quantization effects');
bode(Tcomp_dcm,Tcomprnd_dcm,2*pi,2*pi*fs),grid
title('Compensated loop in DCM with(green)/without(blue) quantization effects');

%--------------------------------------
% LIMIT CYCLE CONDITIONS IN DCM
%--------------------------------------

IoPFM=50e-3; %Limit to enter PFM mode
Rlightl=Rcrit:0.1:Vo/IoPFM; %Light loads
Iolightl=Vo./Rlightl; %Light load output current range
% Output voltage accuracy in CCM and DCM

% CCM
Dvpccm=(1/(2^ndpwm))*(Vin);

% DCM
Dvpdcm=(1/(2^ndpwm))*sqrt((2*Rlightl*Ts)/L)*((1-M)^(3/2)/(M*(2-M)))*Vo;
figure(8);
set(8,'Name','Output voltage accuracy in CCM and DCM');
plot(Iolightl,Dvpccm,'bo-',Iolightl,Dvpdcm,'g*--');%Iolightl,qad,'r*-.'), grid on;
title('Output voltage accuracy in both CCM (blue) and DCM (green)');
xlabel('Output current Io (A)');
ylabel('Output voltage accuracy DVpp (V)');

% check no-limit-cycle conditions

disp('')
disp('Check output voltage accuracy @ IoPFM')
disp(sprintf('Check Dvp_dcm: %g',
(1/(2^ndpwm))*sqrt((2*(Vo/IoPFM)*Ts)/L)*((1-M)^(3/2)/(M*(2-M)))*Vo))
if ( (1/(2^ndpwm))*sqrt((2*(Vo/IoPFM)*Ts)/L)*((1-M)^(3/2)/(M*(2-M)))*Vo < q_vout )

"DPWM time resolution requirements for digitally controlled dc-dc

disp('Rule "output voltage accuracy at light load" OK')
else disp('Failed output voltage accuracy condition at light load')
end

disp('Check no-limit cycle conditions (A1, A2, B2 should be < 1)')

% Check A1

142
disp(sprintf('Check A1: 2*Gvdo*qdpwm*Hsense/qad equals: %g',
2*Gvdo_dcm*qdpwm*Hsense/qad))
if (2*Gvdo_dcm*qdpwm*Hsense/qad < 1)
disp('Rule A1 OK')
else disp('Failed A1 (>1)')
end

% Check A2
disp(sprintf('Check A2: 2*Gvdo*Hsense*Integral_gain equals: %g',
Gvdo_dcm*Hsense*KiDCM*2))
if (2*Gvdo_dcm*KiDCMrnd*Hsense < 1)
disp('Rule A2 OK')
else disp('Failed A2 (>1)')
end

% Check B1
[Gvdmx_dcm, x] = bode(Gvd_used_dcm,wcg_dcm);
disp(sprintf('Check B1: 8*|Gvd(wcg)|*qdpwm*Hsense/(qad*pi) equals: %g',
8*Gvdmx_dcm*qdpwm*Hsense/(qad*pi)))
if (((8*Gvdmx_dcm*qdpwm*Hsense)/(qad*pi) < 1)
disp('Rule B1 OK')
else disp('Failed B1 (>1)')
end

%Check B2
disp('Check B2 (GM to be > 10.2)')
if (20*log10(Gmt_dcm) > 10.2)
disp('Rule B2 OK')
else disp('Failed B2: GM < 10.2 dB')
end

%
\begin{verbatim}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LIGHT LOAD OPERATION %
% %
% PFM CONTROLLER ANALYSIS %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

display('');
display('--------------------------------------');
display('PFM MODE @ LIGHT LOAD')
display('--------------------------------------');
tauo=ESR*C;
Dcrit=M;

\text{toncritPFM}=Dcrit*Ts;
Ipeaklim=3; % Maximum value of the inductor current peak
\text{tonlimPFM}=(Ipeaklim*L)/(Vin-Vo); %ton time @ Ipeaklim

\text{tonPFM}=0.01*Ts:0.01*Ts:tonlimPFM;
\text{toffPFM}=((1/M)-1)*tonPFM;
sz=size(toffPFM);
Ipeak=((Vin-Vo)/L)*tonPFM;
Dvomax=Vo*0.03; % 3\% of the nominal output voltage

% Computes the output voltage ripple in PFM
for i= 1:sz(2);
if(tauo<=toffPFM(1,i))
  \%display('tauo<=toff');
  \text{DvoPFM(i)}=(Ipeak(1,i)/(2*C))\text{tonPFM(1,i)}*(1/M)+(Ipeak(1,i)/(2*C))\text{tauo}^2/toffPFM(1,i));
else
  \%display('tauo>toff');
\end{verbatim}
DvoPFM(i) = (Ipeak(1,i)/(2*C)) * tonPFM(1,i) + Ipeak(1,i) * ESR;
end

if(DvoPFM(i) > Dvomax) indexPFMmax = i-1;
break;
end

end

tonPFMmax = tonPFM(indexPFMmax);
disp(' ');
disp(sprintf(' Maximum ton in PFM mode for a given output voltage ripple Dvomax:
  %g',tonPFMmax));
figure(9);
set(9,'Name','Output voltage and inductor current peak in PFM');
plot(0.01*Ts:0.01*Ts:tonPFMmax,Ipeak(1:indexPFMmax),'bo-',
0.01*Ts:0.01*Ts:tonPFMmax,DvoPFM(1:indexPFMmax),'g--',
0.01*Ts:0.01*Ts:tonPFMmax,Dvomax,'r*-.'), grid on
title('Output voltage ripple(green)& inductor current peak(blue) vs ton');
xlabel('ton (s)');
ylabel('Dvo (V)& iL peak (A)');

Ipeak.tonmax = ((Vin-Vo)/L) * tonPFMmax;
disp(sprintf(' Inductor current peak value @ ton_max: %g',Ipeak.tonmax));

Function Buck_op_vm.m

function op = Buck_op_vm(B,Vin,Vo,R)

op = BUCK_OP_VM(B,Vin,Vo,R)
\% Author: Marco Meola
\% University of Trieste
\%
\%-------------------------------------------------------------------------
\% Determines the operating mode (CCM or DCM) of a given buck converter B
\% defined by the function Buck_vm and a given resistive load R
\%
\% PARAMETERS
\% Vo: nominal output voltage [V]
\% Vin: nominal input voltage [V]
\% D: steady state duty cycle
\% R: load resistor [Ohm]
\% MODE: - CCM = Continuous Conduction Mode
\% - DCM = Discontinuous Conduction Mode
\%
\% See also Buck_vm

M=Vo/Vin;
k=2*B.L/(R*B.Ts);
I=Vo/R;

\%-------------------------------------------------------------------------
\% CCM Erickson Model
\%
\%from: R.W. Erickson, D. Maksimovic, "Fundamentals of power electronics",
%Springer 2001
\%-------------------------------------------------------------------------
if (Vo/R) > (M*B.Ts*(Vin-Vo-B.Resl*I))/(2*B.L)
D=(Vo+B.Resl*I)/Vin;
Mode=['CCM'];

else
% model with losses:
ra=B.Resl+(B.ESR*R)/(B.ESR+R);
rb=R/(B.ESR+R);
ras=ra/R;
rbs=rb/R;
% steady state duty cycle ratio for a desired conversion ratio M
ad=B.Ts*(M*(ras+rb)-1);
bd=M*B.Ts*ras*(1-M*(ras+rb));
cd=2*rbs*B.L*M^2;
% display('Steady state duty cycle');
D=(-bd-sqrt(bd^2-4*ad*cd))/(2*ad)
Mode=['DCM'];
end

op=struct('Vo',Vo,'Vin',Vin,'R',R,'D',D,'Mode',Mode);
Function BuckDesignData.m

function [DiL,LIR,DVosh,DVo]=BuckDesignData(Vin,Vo,fs,Io,L,C,ESR)

% [DiL,LIR,DVosh,DVo]=BuckDesignData(Vin,Vo,fs,Io,L,C,ESR)
%
% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
% Computes the main features of a Buck converter power stage with
% output filter LC when operating in CCM. Main features are:
%
% - Inductor current ripple
% - $LIR = \frac{\text{Inductor current ripple}}{\text{output current}}$
% - Output voltage overshoot
% - Output voltage ripple
%
% Maximum value of the output parameters can be found setting $Vin=Vin_{\text{max}}$
% and $Io=Io_{\text{max}}$
%
% Input Parameters:
%
% $Vin = \text{converter input voltage [V]}$
% $Vo = \text{converter output voltage [V]}$
% fs = switching frequency [Hz]
% Io = load current [A]
% L = output filter inductor [H]
% C = output filter capacitor [F]
% ESR = output capacitor equivalent series resistance
%
% Output Parameters:
%
% DiL = inductor current ripple [A]
% LIR = (Inductor current ripple)/(output current)
% DVosh = Output voltage overshoot [V]
% DVo = output voltage ripple [V]

% Further information can be found in:
% www.powerelectronics.com - D. Schelle, G. Castorena, "Buck-converter
design demystified"

Ts=1/fs;
DiL=((Vin-Vo)/L)*(Vo/Vin)*Ts;
LIR=DiL/Io;
DVosh=sqrt(Vo^2+(L*(Io+DiL/2)^2)/(C))-Vo;
DVo=((Vin-Vo)/(2*C*L))*(Vo*Ts/Vin)^2+DiL*ESR;

% Function Buck_tf_vm_ESR.m

function Gvd=Buck_tf_vm_ESR(B,op)
% This function returns the Control-to-Output transfer function of the
% Buck converter defined by the structure 'B' with Operating Point defined
% by the structure 'op'.

% The Control-to-Output function in CCM is derived from
% 'R. Erickson, D. Maksimovic - Fundamentals of Power Electronics - Second Edition'
% Springer Ed. 2001,

% while the DCM model, instead, is derived from:

% J. Sun, M. Mitchell, M. F. Greuel, P. T. Krein, "Averaged modeling of PWM
% Converters Operating in Discontinuous Conduction Mode" IEEE Trans. on
% Power Elec. 2001, Volume 16, Page(s): 482-492

% modeling inductor series resistance and capacitor ESR effects.

% See also Buck_vm, Buck_op_vm

if op.Mode=='CCM' % CCM Erickson Model
    display( 'CCM Erickson Model')
%Gvd(s)
Gvdo=(op.Vo)/(op.D);
zd=1/(B.C*B.ESR);
wd=sqrt(op.R/(B.L*B.C*(op.R+B.ESR)));
Qd=sqrt(op.R/(wd*(B.L+B.C*op.R+B.ESR)));
G_pole=tf([1/zd 1],[1/(wd^2) 1/(wd*Qd) 1]);
Gvd=Gvdo*G_pole;

else %DCM Erickson Model
  display('DCM Sun, Mitchell Model')
ra=B.Resl+(B.ESR*op.R)/(B.ESR+op.R);
rb=op.R/(B.ESR+op.R);

%Small signal analysis
Vc=op.Vo;
Il=Vc/op.R;

%from diI/dt:
  a11=-ra/B.L+((rb*Vc-op.Vin)*(2*rb*Vc))/(op.D*B.Ts*(op.Vin-ra*Il-rb*Vc)^2);
  a12=((ra*Il-op.Vin)*2*rb*Il)/(op.D*B.Ts*(op.Vin-ra*Il-rb*Vc)^2);
  b11=op.D/B.L+(2*rb*Il*Vc)/(op.D*B.Ts*(op.Vin-ra*Il-rb*Vc)^2);
  b12=(2*rb*Il*Vc)/(op.D^2*B.Ts*(op.Vin-ra*Il-rb*Vc)+op.Vin/B.L);

%from dvc/dt:
  a21=rb/B.C;
  a22=-rb/(B.C*op.R);
  b21=0;
  b22=0;
% from vo:
c1=(op.R*B.ESR)/(op.R+B.ESR); c2=rb; d1=0; d2=0;

Ass=[a11 a12; a21 a22];
Bss=[b11 b12; b21 b22];
Css=[c1 c2];
Dss=[d1 d2];

% small signal state space system
sysDCM=ss(Ass,Bss,Css,Dss,'statename','iL' 'vc','inputname','vg' 'd','outputname','vo');

[num,den]=ss2tf(Ass,Bss,Css,Dss,2);
temp=tf(num,den);
Gvd=zpk(temp);

end

Function bilinear.m

function [Gcz,alpha,Kp,Ki,Kd]=bilinear(fo,fs,fc,Gvdz)

% [Gcz,alpha,Kp,Ki,Kd]=bilinear(fo,fs,fc,Gvdz)

% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
% COPEC PID design based on bilinear transformation as presented in:

152
% COPEC summer short course: Digital Control in Switched Mode Power
% Supplies - University of Colorado at Boulder
%
% %------------------------------------------------------------------------
% Input Parameters:
%
% % fo = resonant frequency of the converter [Hz]
% % fo = 1/(2*pi*(L*C)^2)
% % fs = switching frequency [Hz]
% % fc = desired cross-over frequency [Hz]
% % Gvdz = control to output transfer function of the converter affected by
% % loop delays and DPWM and ADC gains (Z domain)
% %
% % Output Parameters:
%
% % alpha,Kp,Ki,Kd = digital PID controller parameters
%
% % Gcz(z)=Kp+Ki/(1-z^-1)+Kd*(1-z^-1)/(1+alpha*z^-1)

ziplace=0.7; %location of the first zero with respect to fo
z2place=0.9;%0.9; %location of the second zero with respect to fo

wcgoal=2*pi*fc;
wz1=2*pi*z1place*fo;
wz2=2*pi*z2place*fo;
fkrit=fc; %perfect match @ the cross over frequency
a=0;
fhf=((fkrit)/tan(pi*fkrit/fs))*((1-a)/(1+a)); %high frequency pole
whf=2*pi*fhf;
\[ K = \frac{2\pi \cdot f_{\text{crit}}}{\tan(\pi \cdot f_{\text{crit}}/f_s)}; \]

% Analog equivalent uncompensated loop \( T = G_{\text{vd}} \cdot G_{\text{dpwm}} \) for compensator design
\[ G_{\text{vd}}_{\text{eq}} = \text{zpk}(\text{d2c}(G_{\text{vdz}}, '\text{prewarp}', 2\pi \cdot f_{\text{crit}})); \]

% Computes the gain in order to get the desired cross-over frequency
\( s = \text{tf}('s'); \)
\[ G_{\text{cs}} = \frac{1}{s} \cdot \left( \frac{1}{s} + \frac{1}{w_{z1}} \right) \cdot \left( \frac{1}{s} + \frac{1}{w_{z2}} \right) \cdot \left( \frac{1}{s} + \frac{1}{w_{hf}} \right); \]
\[ T_{\text{unc}} = G_{\text{vd}}_{\text{eq}} \cdot G_{\text{cs}}; \]
% compensated loop used to compensator gain design
\[ [T_{\text{uncmag}}, T_{\text{uncph}}] = \text{bode}(T_{\text{unc}}, w_{\text{goal}}); \]
\[ w_k = \frac{1}{T_{\text{uncmag}}}; \]
\[ G_{\text{cz}} = \text{c2d}(w_k \cdot G_{\text{cs}}, 1/f_s, '\text{prewarp}', 2\pi \cdot f_{\text{crit}}); \]

% extraction of digital PID compensator coefficients
\[ [G_{\text{cz num}}, G_{\text{cz den}}] = \text{tfdata}(G_{\text{cz}}, 'v'); \]

% general digital compensator transfer function
% \( G_{\text{cz}}(z) = \frac{a_1 z^2 + a_2 z + a_3}{b_1 z^2 + b_2 z + b_3} \)
\[ a_1 = G_{\text{cz num}}(1); \quad a_2 = G_{\text{cz num}}(2); \quad a_3 = G_{\text{cz num}}(3); \]
\[ b_1 = G_{\text{cz den}}(1); \quad b_2 = G_{\text{cz den}}(2); \quad b_3 = G_{\text{cz den}}(3); \]

% standard PID parallel structure
% \( G_{\text{cz}}(z) = K_p + \frac{K_i}{1-z^{-1}} + \frac{K_d}{(1-z^{-1})(1+\alpha z^{-1})} \)
\[ \alpha = a; \]
\[ K_p = -\frac{a_1 b_3 + a_2 + a_3 (2-b_3)}{((1-b_3)^2)}; \]
\[ K_i = \frac{a_1 + a_2 + a_3}{(1-b_3)}; \]
\[ K_d = \frac{a_1 b_3^2 + a_2 + a_3}{((1-b_3)^2)}; \]
Function standardPID.m

function [Gczrnd, alpharnd, Kprnd, Kirnd, Kdrnd, ephex, eihex, edhex] = standardPID(alpha, Kp, Ki, Kd, fs, RegBandAmp, qad, ndpwm)

% [Gczrnd, alpharnd, Kprnd, Kirnd, Kdrnd, ephex, eihex, edhex] = standardPID(alpha, Kp, Ki, Kd, fs, RegBandAmp, qad, ndpwm)
%
% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
% Effect of coefficient round-off in digital PID compensators using the
% structure:
% Gcz(z) = Kp + Ki/(1-z^-1) + Kd(1-z^-1)/(1+alphaz^-1)
%
% Digital PID compensator with rounded coefficients has the structure
%
% Gczrnd(z) = Kprnd + Kirnd/(1-z^-1) + Kdrnd(1-z^-1)/(1+alpharndz^-1)
%
%-------------------------------------------------------------------------
% Input Parameters:
%
% alpha, Kp, Ki, Kd = digital PID compensator parameters
% fs = switching frequency of the converter [Hz]
% RegBandAmp = number of bins used to code the output voltage:
% if the error is in the interval [-esat, esat] and qad is the
% quantization step of the ADC at the output of the converter then
% RegBandAmp=esat/qad
% qad = quantization step of the ADC at the output of the converter [V]
% ndpwm = number of bits used to represent the duty cycle d[n] input to
% the DPWM
%
% Output Parameters:
%
% Gczrnd = digital PID controller transfer function (Z domain) with
% rounded coefficients
% alpharnd,Kprnd,Kirnd,Kdrnd = rounded PID coefficients
% ephex,eihex,edhex = decimal words to store into LUTs

%------------------------------------------------------------------------
% QUANTIZATION EFFECTS ON PID COMPENSATOR IN CCM
%------------------------------------------------------------------------

for i = 1:1:RegBandAmp
    ep(i,1)=Kp*i*qad*2^ndpwm;
    ei(i,1)=Ki*i*qad*2^ndpwm;
    ed(i,1)=Kd*i*qad*2^ndpwm;
end

%Coefficient stored into LUT - binary format
epbin=dec2bin(abs(ep));
eibin=dec2bin(abs(ei));
edbin=dec2bin(abs(ed));

%Hexadecimal values
ephex=dec2hex(bin2dec(epbin));
eihex = dec2hex(bin2dec(eibin));
edhex = dec2hex(bin2dec(edbin));

% Coefficient stored into LUT - decimal format
eprnd = sign(ep).*bin2dec(epbin)/(2^ndpwm);
eirnd = sign(ei).*bin2dec(eibin)/(2^ndpwm);
edrnd = sign(ed).*bin2dec(edbin)/(2^ndpwm);

% Rounded coefficients
alpharnd = alpha;
for i = 1:1:RegBandAmp
    Kprnd = eprnd(i,1)/(i*qad);
    Kirnd = eirnd(i,1)/(i*qad);
    Kdrnd = edrnd(i,1)/(i*qad);
end

% Definition of digital PID with rounded coefficients
% Gcz = (a1rndz^2 + a2rndz + a3rnd) / (b1rndz^2 + b2rndz + b3rnd)

z = tf('z',1/fs);
a1rnd = Kprnd + Kirnd + Kdrnd;
a2rnd = Kprnd*(alpharnd-1) + Kirnd*alpharnd - 2*Kdrnd;
a3rnd = Kdrnd - Kprnd*alpharnd;
b1rnd = 1;
b2rnd = alphanrnd - 1;
b3rnd = -alpharnd;
Gczrnd = (a1rnd*z^2 + a2rnd*z + a3rnd) / (b1rnd*z^2 + b2rnd*z + b3rnd);

Function bilparPID.m
function [Gcz,K1,K2,K3,P,Ki]=bilparPID(fo,fs,fc,Gvdz)

% [Gcz,K1,K2,K3,P,Ki]=bilparPID(fo,fs,fc,Gvdz)
%
% Author: Marco Meola
% University of Trieste
%
% COPEC PID design based on bilinear transformation as presented in:
% COPEC summer short course: Digital Control in Switched Mode Power
% Supplies - University of Colorado at Boulder
%
% with parallel structure
%
% Input Parameters:
%
% fo = resonant frequency of the converter [Hz]
% fo = 1/(2*pi*(L*C)^2)
% fs = switching frequency [Hz]
% fc = desired cross-over frequency [Hz]
% Gvdz = control to output transfer function of the converter affected by
% loop delay and DPWM and ADC gains (Z domain)
%
% Output Parameters:
% K1,K2,K3,P = digital PID controller parameters where
% Gcz = K1+K2/(z-1)+K3/(z-P)

z1place=0.7; %location of the first zero with respect to fo
z2place=0.9; %location of the second zero with respect to fo

wcgoal=2*pi*fc;
wz1=2*pi*z1place*fo;
wz2=2*pi*z2place*fo;
fcrit=fc; %perfect match @ the cross over frequency
a=0;
fhf=(((fcrit)/tan(pi*fcrit/fs))*((1-a)/(1+a)); %high frequency pole
whf=2*pi*fhf;
K=(2*pi*fcrit)/tan(pi*fcrit/fs);

%Analog equivalent uncompensated loop T=Gvd*Gdpwm for compensator design
Gvd_s_eq=zpk(d2c(Gvdz,'prewarp',2*pi*fcrit));

%Computes the gain in order to get the desired cross-over frequency
s=tf('s');
Gcs=(1/s)*(1+s/wz1)*(1+s/wz2)*(1/(1+s/whf));
Tunc=Gvd_s_eq*Gcs; %compensated loop used to compensator gain design
[Tuncmag,Tuncph]=bode(Tunc,wcgoal);
wk=1/Tuncmag;

Gcz=c2d(wk*Gcs,1/fs,'prewarp',2*pi*fcrit);

% extraction of Cz coefficients
[Gcz_num,Gcz_den]=tfdata(Gcz,'v');
\[
Gcz(z) = \frac{(a_1z^2 + a_2z + a_3)}{(b_1z^2 + b_2z + b_3)}
\]

\[
a_1 = Gcz.num(1); \quad a_2 = Gcz.num(2); \quad a_3 = Gcz.num(3);
\]

\[
b_1 = Gcz.den(1); \quad b_2 = Gcz.den(2); \quad b_3 = Gcz.den(3);
\]

\% parallel structure - transfer function
\[
[Rc, Pc, Kc] = \text{residue}(Gcz.num, Gcz.den);
\]

\[
K1 = Kc; \quad K2 = Rc(1); \quad K3 = Rc(2);
\]

\[
P = Pc(2);
\]

\[
Ki = K2;
\]

Function parallelPID.m

function [Gczrnd, K1rnd, K2rnd, K3rnd, Prnd, e1rnd, e2rnd, e3rnd] = parallelPID(K1, K2, K3, P, fs, RegBandAmp, qad, npwm)

\% Author: Marco Meola
\%
\% University of Trieste
\%
\%-------------------------------------------------------------------------
\%
\% Effect of coefficient round-off in digital PID compensators using the
\% parallel structure:
\%
\% Gcz(z) = K1 + K2/(z-1) + K3/(z-P)
\%
\% Digital PID compensator with rounded coefficients has the structure
\%
\% Gczrnd(z) = K1rnd + K2rnd/(z-1) + K3rnd/(z-Prnd)

160
% Input Parameters:
%
% K1,K2,K3,P = digital PID compensator parameters
% fs = switching frequency of the converter [Hz]
% RegBandAmp = number of bins used to code the output voltage:
% if the error is in the interval [-esat,esat] and qad is the
% quantization step of the ADC at the output of the converter
% RegBandAmp=esat/qad
% qad = quantization step of the ADC at the output of the converter [V]
% ndpwm = number of bits of the DPWM
%
% Output Parameters:
%
% Gczrnd = digital PID controller transfer function (discrete domain) with
% rounded coefficients
% K1rnd,K2rnd,K3rnd,Prnd = rounded PID coefficients
% e1rnd,e2rnd,e3rnd = decimal words to store into LUTs

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% QUANTIZATION EFFECTS ON PID COMPENSATOR IN CCM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i = 1:1:RegBandAmp
  e1(i,1)=K1*i*qad*2^-ndpwm;
  e2(i,1)=K2*i*qad*2^-ndpwm;
  e3(i,1)=K3*i*qad*2^-ndpwm;
end
```matlab
% Coefficient stored into LUT - binary format
e1bin=dec2bin(abs(e1));
e2bin=dec2bin(abs(e2));
e3bin=dec2bin(abs(e3));

% Coefficient stored into LUT - decimal format
e1rnd=sign(e1).*bin2dec(e1bin)/(2^ndpwm);
e2rnd=sign(e2).*bin2dec(e2bin)/(2^ndpwm);
e3rnd=sign(e3).*bin2dec(e3bin)/(2^ndpwm);

% Rounded coefficients
Prnd=P;
for i = 1:1:RegBandAmp
K1rnd=e1rnd(i,1)/(i*qad);
K2rnd=e2rnd(i,1)/(i*qad);
K3rnd=e3rnd(i,1)/(i*qad);
end

% Definition of digital PID with rounded coefficients
% Gcz=(a1rndz^2+a2rndz+a3rnd)/(b1rndz^2+b2rndz+b3rnd)

z=tf('z',1/fs);
a1rnd=K1rnd;
a2rnd=K2rnd+K3rnd-K1rnd*(Prnd+1);
a3rnd=Prnd*(K1rnd-K2rnd)-K3rnd;
b1rnd=1;
b2rnd=-(Prnd+1);
b3rnd=Prnd;
```
Gczrnd=(a1rnd*z^2+a2rnd*z+a3rnd)/(b1rnd*z^2+b2rnd*z+b3rnd);

Function zero_pole_comp.m

function [Gcz,alpha,beta,\gamma,Ki]=zero_pole_comp(fo,fs,fc,Gvdz);

% [Gcz,alpha,beta,\gamma,Ki]=zero_pole_comp(fo,fs,fc,Gvdz);
%
% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
% COPEC LUT PID design from:
% A. Prodic, D. Maksimovic, "Design of a digital PID regulator based on
% Look-Up tables for control of high-frequency DC-DC converters", IEEE
% COMPEL 2002, page(s): 18-22
%
% http://www.istri.polito.it/~meola/zeropole.m

% Input Parameters:
%
% fo = resonant frequency of the converter [Hz]
% fo = 1/(2*pi*(L*C)^2)
% fs = switching frequency [Hz]
% fc = desired cross-over frequency [Hz]
% Gvdz = control to output transfer function of the converter affected by
% loop delays and DPWM and ADC gains (Z domain)
%
% Output Parameters:
% Gcz = digital controller transfer function (discrete domain)
% alpha, beta, gamma = digital PID coefficients
% Ki = Ki coefficient of the digital PID (required for LCO analysis)

z1place=0.7; % location of the first zero with respect to fo
z2place=0.9; % location of the second zero with respect to fo

wcgoal=2*pi*fc;
wz1=2*pi*z1place*fo;
wz2=2*pi*z2place*fo;
wzn=sqrt(wz1*wz2);
Q=wzn/(wz1+wz2);
apid=wzn/(2*Q)
bpid=wzn
r=exp(-apid*(1/fs));

z=tf('z',1/fs);
Gcztemp=(z^2-2*r*cos(bpid*(1/fs))*z+r^2)/(z*(z-1));
Ttemp=Gvdz*Gcztemp;
[Gcztemp_mag,Gcztemp_ph]=bode(Ttemp,wcgoal);

Gcz=(1/Gcztemp_mag)*Gcztemp;

% extraction of digital PID compensator coefficients
[Gcz_num,Gcz_den]=tfdata(Gcz,'v');

% general digital compensator transfer function
% Gcz=(a1z^2+a2z+a3)/(b1z^2+b2z+b3)
alpha=Gcz_num(1); beta=Gcz_num(2); gamma=Gcz_num(3);
Ki=alpha+beta+gamma;

Function copecPID.m

function [Gczrnd,alpharnd,betarnd,gammarnd,e_alpha_rnd,e_beta_rnd,e_gamma_rnd]=
copecPID(alpha,beta,omega,fs,RegBandAmp,qad,ndpwm)

% [Gczrnd,alpharnd,betarnd,gammarnd,e_alpha_rnd,e_beta_rnd,e_gamma_rnd]=cop
% ecPID(alpha,beta,omega,fs,RegBandAmp,qad,ndpwm)
%
% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
%
% Effect of coefficient round-off in digital PID compensators using the
% structure:
% Gcz(z)=(alpha+beta*z^-1+gamma*z^-2)/(1-z^-1)
%
% which is the stucture presented in:
% A. Prodic, D. Maksimovic, "Design of a digital PID regulator based on
% Look-Up tables for control of high-frequency DC-DC converters", IEEE
% COMPEL 2002, page(s): 18-22
%
% Digital PID compensator with rounded coefficients has the structure
%
% Gcz(z)=(alpharnd+betarnd*z^-1+gammarnd*z^-2)/(1-z^-1)
%
%-------------------------------------------------------------------------
% Input Parameters:
% 
% alpha,beta,gamma = digital PID compensator parameters
% fs = switching frequency of the converter [Hz]
% RegBandAmp = number of bins used to code the output voltage:
% if the error is in the interval [-esat,esat] and qad is the
% quantization step of the ADC at the output of the converter then
% RegBandAmp=esat/qad
% qad = quantization step of the ADC at the output of the converter [V]
% ndpwm = number of bits of the DPWM
% 
% Output Parameters:
% 
% Gczrnd = digital PID controller transfer function (Z domain) with
% rounded coefficients
% alpharnd,betarnd,gammarnd = rounded PID coefficients
% e_alpha_rnd,e_beta_rnd,e_gamma_rnd = decimal words to store into LUTs
% 
%-------------------------------------------------------------------------
% QUANTIZATION EFFECTS ON PID COMPENSATOR IN CCM
%-------------------------------------------------------------------------

for i = 1:1:RegBandAmp
    e_alpha(i,1)=alpha*i*qad*2^ndpwm;
    e_beta(i,1)=beta*i*qad*2^ndpwm;
    e_gamma(i,1)=gamma*i*qad*2^ndpwm;
end
%Coefficient stored into LUT - binary format
e_alpha_bin=dec2bin(abs(e_alpha));
e_beta_bin=dec2bin(abs(e_beta));
e_gamma_bin=dec2bin(abs(e_gamma));

%Coefficient stored into LUT - decimal format
e_alpha_rnd=sign(e_alpha).*bin2dec(e_alpha_bin)/(2^ndpwm);
e_beta_rnd=sign(e_beta).*bin2dec(e_beta_bin)/(2^ndpwm);
e_gamma_rnd=sign(e_gamma).*bin2dec(e_gamma_bin)/(2^ndpwm);

%rounded coefficients
for i = 1:1:RegBandAmp
alpharnd=e_alpha_rnd(i,1)/(i*qad);
betarnd=e_beta_rnd(i,1)/(i*qad);
gammarnd=e_gamma_rnd(i,1)/(i*qad);
end

z=tf('z',1/fs);
Gczrnd=(alpharnd*z^2+betarnd*z+gammarnd)/(z^2-z);

Function PI-DCM.m

function [Gcz,Kp,Ki]=PI_DCM(fs,fc,Gvdz)

% [Gcz,Kp,Ki]=PI_DCM(fs,fc,Gvdz)
%
% Author: Marco Meola
% University of Trieste
%---------------------------------------------------------------
% Simple digital PI compensator for a buck converter working in DCM
%
%--------------------------------------------------------

% Input Parameters:
%
% fs = switching frequency [Hz]
% fc = desired cross-over frequency [Hz]
% Gvdz = control to output transfer function of the converter affected by
% loop delays and DPWM and ADC gains (discrete domain)
%
% Output Parameters:
%
% Gcz = digital controller transfer function (discrete domain)
% Kp,Ki = PI compensator parameters

wcgoal=2*pi*fc;
% Extract gain, pole and zero values from Gvdz
[zz,pz,kz]=zpkdata(zpk(Gvdz),’v’);
zlf=pz(1);
z=tf(’z’,1/fs);
Gcztemp=(z-zlf)/(z-1);
Tunc=Gcztemp*Gvdz;
[Tuncmag,Tuncph]=bode(Tunc,wcgoal);
wkDCM=(1/Tuncmag);
Gcz=wkDCM*Gcztemp;
Kp=zlf*wkDCM;
Ki=wkDCM*(1-zlf);

Function StandardPI.m

function [Gczrnd,Kprnd,Kirnd,ephex,eihex]=standardPI(Kp,Ki,fs,RegBandAmp,qad,ndpwm)

% [Gczrnd,Kprnd,Kirnd,ephex,eihex]=standardPI(Kp,Ki,fs,RegBandAmp,qad,ndpwm)
%
% Author: Marco Meola
% University of Trieste
%
%-------------------------------------------------------------------------
% Effect of coefficient round-off in digital PI compensators using the
% structure:
% Gcz(z)=Kp+Ki/(1-z^-1)
%
% Digital PI compensator with rounded coefficients has the structure
% Gczrnd(z)=Kprnd+Kirnd/(1-z^-1)
%
%-------------------------------------------------------------------------
% Input Parameters:
%
% Kp,Ki = digital PI compensator parameters
% fs = switching frequency of the converter [Hz]
% RegBandAmp = number of bins used to code the output voltage:
% if the error is in the interval [-esat,esat] and qad is the
% quantization step of the ADC at the output of the converter, then
% RegBandAmp=esat/qad
% qad = quantization step of the ADC at the output of the converter [V]
% ndpwm = number of bits of the DPWM
%
% Output Parameters:
%
% Gczrnd = digital PID controller transfer function (Z domain) with
% rounded coefficients
% Kprnd,Kirnd = rounded PI coefficients
% ephex,eihex = hexadecimal words to store into LUTs

for i = 1:1:RegBandAmp
    ep(i,1)=Kp*i*qad*2^-ndpwm;
    ei(i,1)=Ki*i*qad*2^-ndpwm;
end

%Coefficient stored into LUT - binary format
epbin=dec2bin(abs(ep));
eibin=dec2bin(abs(ei));

ephex=dec2hex(bin2dec(epbin));
eihex=dec2hex(bin2dec(eibin));

%Coefficient stored into LUT - decimal format
eprnd=sign(ep).*bin2dec(epbin)/(2^-ndpwm);
eirnd=sign(ei).*bin2dec(eibin)/(2^-ndpwm);

for i = 1:1:RegBandAmp
    Kprnd=eprnd(i,1)/(i*qad);
    Kirnd=eirnd(i,1)/(i*qad);
end
end

z=tf('z',1/fs);
Gczrnd=((Kprnd+Kirnd)*z-Kprnd)/(z-1);