Master's Thesis in
Electronic Engineering

# Study and Design of a High Resolution Digital PWM in a Digitally-Controlled Buck Converter for Automotive Applications


Politecnico di Torino

**Author:**
Marco COTTURONE

**Supervisors:**
Dr. Paolo Stefano CROVETTI
**Co-Supervisors:**
Eng. Matteo LANDINI
Eng. Paolo VILMERCATI

Academic Year 2022/2023

# Abstract

*Over one hundred electronic control units (ECUs) are needed for modern vehicles. Each ECU is supplied from the car battery by a front-end buck converter. In turn, back-end point-of-load (POL) buck converters, power the ECU's smart loads (MCU, CAN, I/Os) with low voltages and high currents required by each. One evident problem with high currents is the trace-induced voltage drop from the converter to the load. A point-of-load converter is a power supply DC-to-DC converter located as close to the load as possible to achieve proximity to power, minimising the effect of PCB resistance and parasitic inductance. These back-end voltage regulators must be compact, meet stringent safety regulations, function with low input voltages throughout a wide temperature range, and should generate very low Electromagnetic Interference (EMI) while demonstrating high precision, efficiency, and dependability.*

*Conventionally, DC-DC converters were controlled with analog approaches to lower power consumption and to make easy their implementation. Nowadays digital circuits are becoming less costly, encouraging interest in digital control. Digitally-controlled DC-DC switching converters offer several advantages, including programmability, reconfigurability, robustness to noise and the possibility to implement complex control algorithms.*

*In this thesis, a digitally controlled Buck converter characterised by an innovative digital pulse width modulation technique is proposed. The main differences between digital and analog small-signal modelling are introduced and a complete review of the digital compensator design process is reported. Quantization-induced limit cycle oscillations (LCOs) are described and Dyadic Dithering is proposed to raise the effective DPWM resolution, which is necessary for LCO-free operation, at low cost, without sacrificing DC accuracy and without having a negative impact on the ripple voltage. Thermometric dithering technique and* $\Sigma\Delta$ *modulation are compared to validate the strength of the Dyadic dithering.*

*A DPWM time resolution of 100ps is achieved combining three types of techniques: a 40 MHz (synchronous) counter-based modulator, a delay-line used to reach sub-clock resolution and the Dyadic Dithering modulator. Simulation results obtained in Simulink environment validate the presented approach showing noticeable loop performances.*

*Finally, a delay-line circuit is implemented and simulated in Cadence (Virtuoso) environment.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In order to meet the growing needs for small size, low power operation, and enhanced functionality, efficient power management is a crucial design component. Many electronics systems also need low multiple power rails and supply solutions that can handle both the over 100A requirements for application-specific integrated circuit (ASIC) voltage regulators and the few milliamps needed for standby supplies[14].

The power supply must meet strict specifications for low-voltage CPUs, digital signal processors (DSPs), and double data rate (DDR) memories. With advancements in direct current DC-to-DC regulator technology, complex power management technologies that have a significant impact on the whole solution are being implemented in response to these issues. Designing power management has historically been a difficult task for designers who are new to this field. It is now simpler to achieve these standards while achieving significant power reductions and maintaining alternating current (AC) performance thanks to key innovations in switching mode regulators. While choosing power management architecture and technology, a lot of aspects must be taken into account in order to produce the most efficient regulator design. The first design processes should include an examination of the requirements of the application, such as output control, high voltage range, extended temperature, and fast transient response in terms of required efficiency, system footprint limits, and component budget.

## 1.1 Switching Converters applied to Automotive Fields

A perfect example in which is possible to meet such problematics is the automotive field. Over the past ten years, the development of vehicle cockpit electronics has surged. In the past, self-contained systems like steering, braking, traction, and other safety devices, as well as entertainment technology and navigational aids, have developed into integrated infotainment systems that are increasingly layered with advanced driver assistance systems (ADAS). Particularly the latter have developed into the newest "must-have" for consumers and a point of uniqueness that aids automobile dealers in moving the customer up the pricing ladder. These systems are the first steps toward cars without rivers, so evolution does not end here[11].

For these reasons Point-of-load (POL) converters working at ever-lower voltages and larger currents are required by processors, logic components, memory devices, and interface circuits placed on a smart car. A point-of-load converter is a power supply DC-to-DC converter located as close to the load as possible to achieve proximity to power, minimising the effect of PCB resistance and parasitic inductance. The common power tree of an electronic control unit is shown in fig.1.1. The typical operating range for a car battery is 9 to 18 volts, with brief transients that can surpass 40 volts and fall as low as 5 volts DC. Flexible solutions supplied by secondary rail POL buck regulators that can deliver higher current power supply control at lower voltages are needed to meet the diverse load requirements of entry-level to premium cars.

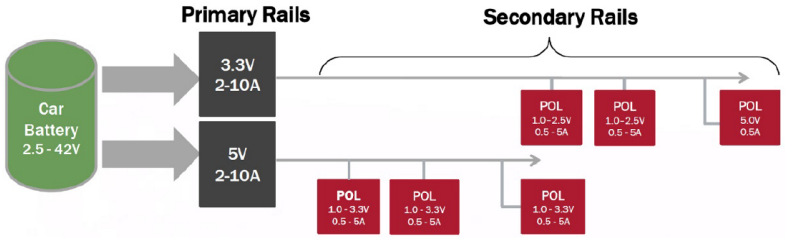



Figure 1.1: Automotive Power Management[11]

Traditionally, analog controllers have been used to regulate and achieve the appropriate output value from DC/DC converters. A high bandwidth, theoretically infinite resolution analog control system (ACS) operates in real time. Throughout the last several years, digital control has become the dominant option for DC/DC switching converters. As digital integrated circuits are becoming less expensive on a consistent basis, digital control technology is no longer prohibitively expensive[8]. The advantages of the digitally controlled DC-DC switching converters over their analog counterparts include flexibility to implement complex control strategies and algorithms, programmability to realize reconfigurable power systems, quick design time, high power conversion efficiency, robustness to noise, and insensitivity to component parameter drifts[2].

## 1.2 Digitally Controlled Buck Converters

Specialised hardware-based digital control loops are frequently favored at switching frequencies in the hundreds of kHz to MHz range. The structure in fig.1.2 serves as an example of this method. The control loop is digital and uses ad hoc, programmable A/D, DPWM, and compensator blocks to enable high-performance loop dynamic responses. A microcontroller is responsible for programmability, power management, and system interface[3].

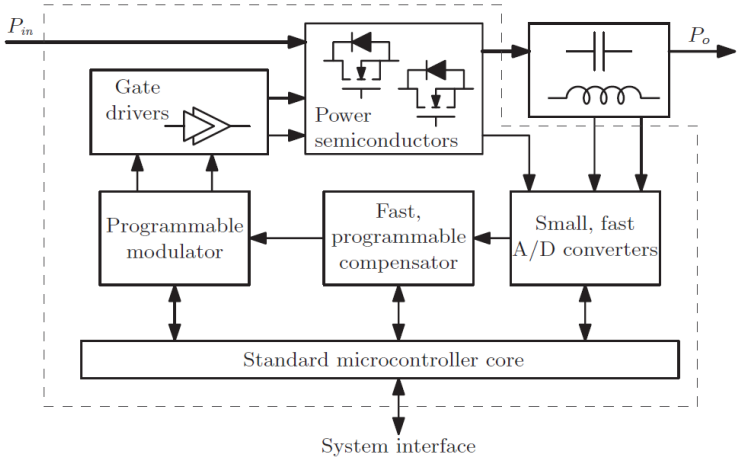



Figure 1.2: Digitally controlled DC-DC converter[3]

More in detail, the digital control loop can be represented as shows fig.1.3 [6]. For a voltage-mode converter it's sufficient to sense the output voltage and to compare it with the reference. The comparing can be performed both in digital, with a subtractor, or in analog with an OPAMP. The second case is that represented in the scheme, where a DAC is used to define the reference. The error voltage is firstly coverted in digital by an ADC, then is processed by the digital compensator, usually implemented by a PID filter. Finally the digital output of the PID is used as control signal by the DPWM to drive the power MOSFETs.

The red square indicates the digital part, it means that all the elements that it contains can be programmed and managed through a bus interface. For example, the PID gains can be set and adjusted whenever is needed at zero cost, changing the frequency response and regulation performances and just to underline one of the advantages of digital control, this operation cannot be done in an analog-controlled converter without replacing passive components.

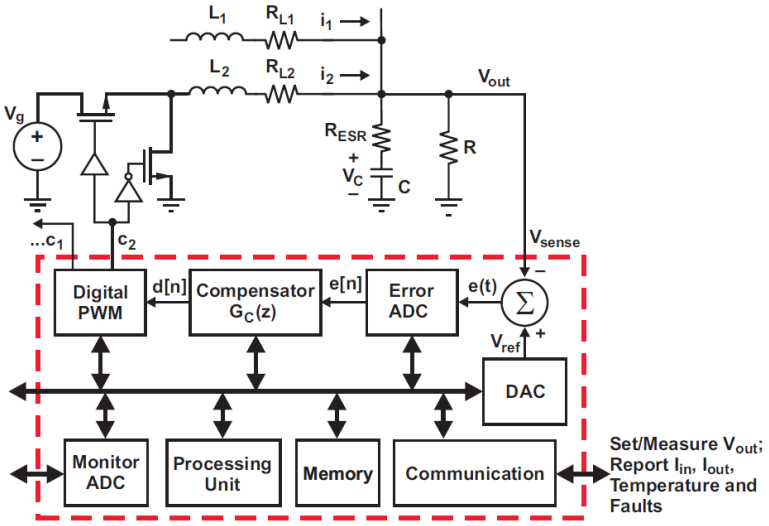



Figure 1.3: Digitally controlled DC-DC converter, more in detail[6]

The circuit is meant to be integrated with the STMicrolectronics' BCD (Bipolar-CMOS-DMOS) technology[1]. Thanks to this silicon processes, three different technologies can be included in a single chip, together with their advantages:

- **Bipolar**: the best for *analog* functions;
- **CMOS**: the best for *digital processing*;
- **DMOS**: for *power* and *high voltage* elements;

The use of different technologies together can provide many benefits, such as greater reliability, less electromagnetic interference, smaller chip areas, and the ability to address a wide range of applications in fields like power management, data acquisition, and power actuators.

## 1.3 Thesis Objectives

This thesis has as its goal the realisation of an innovative DC-DC converter based on digital control. The main target is to analyse the characteristics of this kind of control, maximising the performance and minimising the hardware cost. The main drawback of digital controlled converters is an undesired low-frequency limit cycle oscillation (LCOs) that are mainly caused by the limited digital pulse width modulation (DPWM) resolution, which needs to be sufficiently high. The innovative aspect of this converter is in its structure: the digital pulse-width modulation is implemented combining several techniques to obtain the desired resolution and avoid LCOs.

For a clear and complete explanation of the project realisation, this written report is organised as follows:

- Chapter 2 : A brief analysis of the Buck converter control loop, underling how it can be controlled digitally and the differences compared to the analog approach;
- Chapter 3 : Presentation of the circuit topology taken into account for the project. The circuit is modelled in Simulink environment and a description of all the blocks is provided, focusing on how each component has been designed and how the small-signal analysis of the system for its frequency compensation has been applied;
- Chapter 4 : Description of the simulations carried out in Simulink environment and of the optimisations made on the model based on the simulations results. Several techniques are proposed to increase the resolution of the converter and a comparison is provided to understand which one has been chosen for the design and why;
- Chapter 5 : A detailed description of the delay line and a possible transistor-level implementation are proposed, together with the wave forms and performances obtained by simulations.
- Conclusions are drawn. A final review of the future developments closes the thesis.

# Chapter 2

# Analysis of Buck Converters

A complete analysis of Buck converters is proposed in this chapter. In particular, all the theoretical aspects are illustrated starting from their basic structure and analog control techniques. After that, the digital control approach is presented and explained in details underlining all the differences compared with the common technique, why can be preferred in some applications and how it has to be considered during the design process.

## 2.1 The Buck converter: a brief review

The Buck or step-down converter is a switching DC-DC regulator, it converts a DC voltage to another lower DC voltage by using active switches, inductance and load capacitance. The regulator uses switching activity to charge and discharge the inductance with a fixed frequency (Pulse Width Modulation loop) or fixed time (Pulse Frequency Modulation loop) to obtain a target regulated voltage. In practice, the dc input voltage is "chopped" by the switching network, producing a reduced average output voltage through the L-C-R circuit, that transforms the square wave into a low-ripple dc output voltage by acting as a second-order low-pass filter[9]. The average output voltage is equal to the average voltage of the square wave since the average voltage across the inductor L is zero in the steady state. A pulse-width modulator is in charge of turning the switch on and off at the frequency $f_s = \frac{1}{T_s}$. It is possible to define the *duty cycle* as:

$$D = \frac{t_{on}}{Ts} = \frac{t_{on}}{t_{on} + t_{off}} = f_s t_{on} \tag{2.1}$$

$t_{on}$ is the time interval when the load is connected to the input voltage while $t_{off}$ to the reference. Depending on the inductor current's waveform, the buck converter can operate in either a continuous conduction mode (CCM) or a discontinuous conduction mode (DCM). In CCM, the inductor current flows throughout the entire cycle, but only for a portion of the cycle in DCM. Essentially, in DCM, it decreases to zero, stays there for a while, and then begins to rise. The critical mode is defined as operation at the CCM/DCM boundary. For CCM operation, the average value of the PWM voltage waveform is $V_O = DV_I$, which is independent of the load and depends on the duty cycle D. The duty cycle D might theoretically range from 0% to 100%. As a result, the output $V_O$ has a range from

0 to $V_I$. In actual use, the output voltage $V_O$ should be maintained at a set value while the dc input voltage $V_I$ changes across a predetermined range. When the dc voltage $V_I$ is raised, the duty cycle D is lowered to keep the product $DV_I$ constant, which represents the PWM voltage's average value. As opposed to this, if the input voltage $V_I$ is decreased, the duty cycle D is raised, resulting in a constant average value for the PWM signal. As a result, by adjusting the switch's on-duty cycle D, the quantity of energy delivered from the input voltage source $V_I$ to the load can be adjusted. The output power is constant if the output voltage and the load resistance (or the load current) are also constant. In order to transfer the same amount of energy, the switch on-time should be shortened as the voltage rises. The control circuit regulates the duty cycle D.

A simple circuit that shows a very simple structure of a buck is attached in fig.2.1.

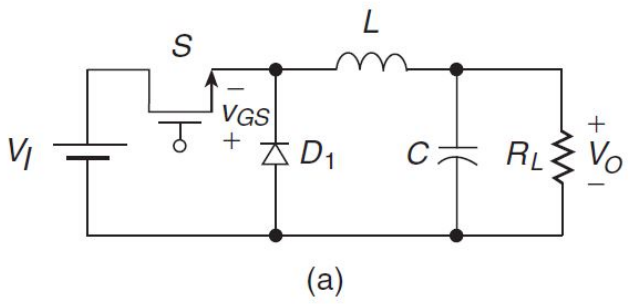
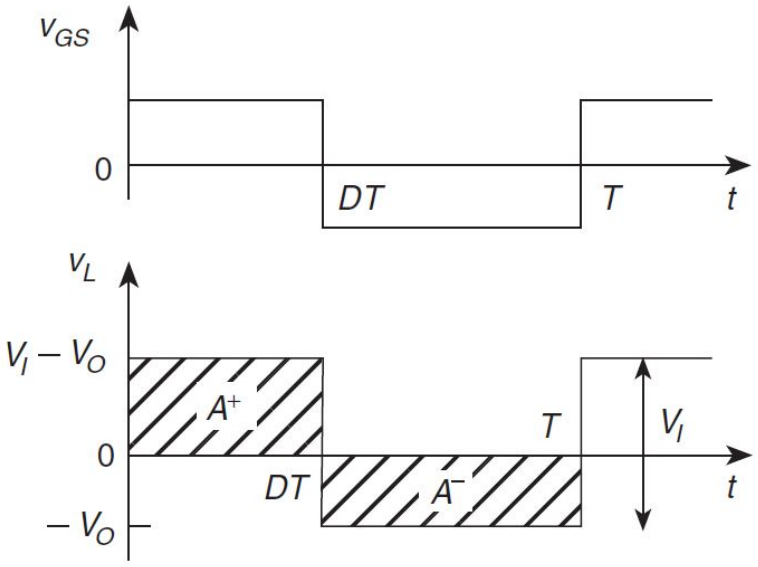



Figure 2.1: Basic Buck Structure[9]

In the study of the buck PWM converter, the following assumptions will be considered: all components are ideal elements, so switching losses are not considered; the converter is in steady state; the time constants of reactive components are substantially longer than the switching period $T_S = \frac{1}{f_s}$. In fig.2.2 the main waveforms in term of voltages and currents of the plant's components are shown.



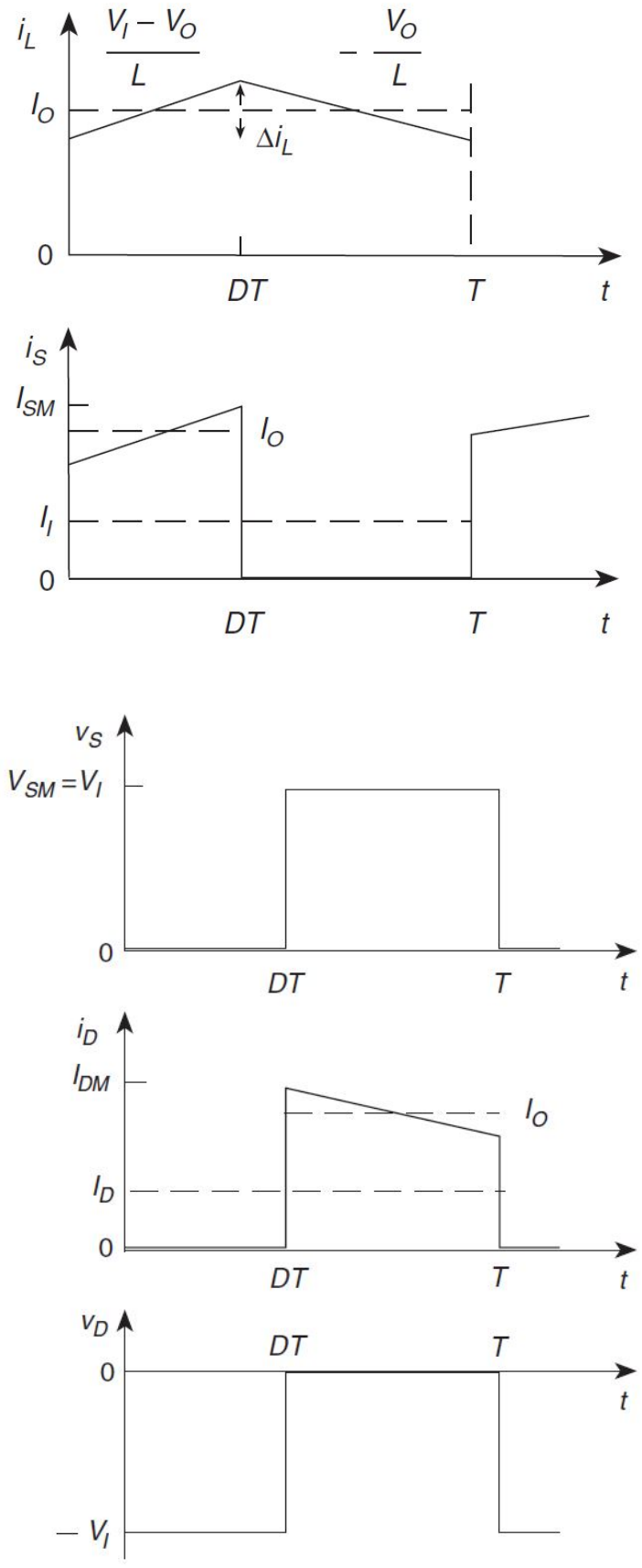



Figure 2.2: Components' waveforms[9]

The switch S is ON and the diode D1 is OFF for the duration of the time period $0 < t \leq$

$DT$. The diode is reverse biased when the switch is ON because the voltage across it is about $V_I$. Both the diode current and the voltage across the switch, $v_S$, are zero. The voltage across the inductor $L$ is given by:

$$v_L = V_I - V_O = L\frac{di_L}{dt}; \tag{2.2}$$

Consequently, the current flowing into the switch S and inductor $L$ is;

$$i_S = i_L = \frac{1}{L}\int_0^t v_L dt \, + \, i_L(0) \, = \frac{V_I - V_O}{L} \, \int_0^t dt \, + \, i_L(0) \, = \, \frac{V_I - V_O}{L} + \, i_L(0) \tag{2.3}$$

with $i_L(0)$ the initial inductor current, while its peak current becomes:

$$i_L(DT_s) = \frac{(V_I - V_O)DT_s}{L} + i_L(0) \tag{2.4}$$

and the peak-to-peak is:

$$\Delta i_L = i_L(DT_S) - i_L(0) = \frac{(V_I - V_O)DT_s}{L} = \frac{(V_I - V_O)D}{f_S L} = \frac{V_I D(1-D)}{f_S L} \tag{2.5}$$

The switch S is OFF and the diode D1 is ON for the duration of the time period $DT < t \leq T_S$. The switch is turned off at $t = DT_S$, and the diode is turned on because the inductor works as a current source. The switch current $i_S$ and the diode voltage $v_D$ are zero and the voltage across the inductor L is:

$$v_L = -V_O = L\frac{di_L}{dt} \tag{2.6}$$

The expression for the current flowing through the diode and inductor $L$ is:

$$i_D = i_L = \frac{1}{L}\int_{DT_S}^t v_L dt + i_L(DT_S) \, = -\frac{V_O}{L} \, \int_{DT_S}^t dt + i_L(DT_S) \, = -\, \frac{V_O}{L}(t - DT_S) + i_L(DT_S) \tag{2.7}$$

where $i_L(DT_S)$ is the initial condition of the current at $t = DT_S$. The inductor $L$'s peak-to-peak ripple current is:

$$\Delta i_L = i_L(DT_S) - i_L(T_S) = \frac{V_O T_S(1-D)}{L} = \frac{V_O(1-D)}{f_S L} \tag{2.8}$$

In fig.2.2 it's possible to see that in steady-state the average current is constant, so its ripple during On and Off intervals must be the same. From this consideration the DC voltage transfer function can be obtained:

$$(V_I - V_O)DT_S = V_O(1-D)T_S \tag{2.9}$$

$$V_O = DV_I \tag{2.10}$$

For an ideal converter, i.e. without losses, the input power is equal to the output one, thus:

$$M_{V_{DC}} = \frac{V_O}{V_I} = D \tag{2.11}$$

This ratio is always lower than 1, this is the reason why this type of converter is called, *step-down*, because the output voltage is inevitably smaller than the input one.

## 2.2 Control loop

In order to attain the appropriate output voltage regulation, switching mode voltage regulators rely on feedback loops. For instance, the goal of a typical dc-dc converter application is to keep the output voltage well regulated in presence of input voltage or load current changes. The foundation for feedback loop design methodologies based on frequency-domain concepts of loop gain, crossover frequency, phase margin, and gain margin is an accurate small-signal description of the converter control-to-output dynamics.

Switched-mode Power Converters basics have been structured on continous-time analysis and analog control design theory. Nowadays the implementation of power electronics is largely digital[3], for this reason is essential to establish a rigorous method to study and design power converters' digital feedback. This section starts with a review of continuous-time averaged modelling used in analog control and then provides two possible approaches to model digital loops of switched-mode power converters, the first is an approximation made starting from the continuous-time approach, not always feasible, the second tackles the loop directly in discrete-time presenting multiple advantages and better precision.

### 2.2.1 Continuous-Time Average Modelling

In this section the *average small-signal modelling* technique is reported, relating to the reference book [3], that explains the approach in detail to underline the limitations and approximations that it presents when used for digital-control design. This technique involves first averaging the converter behaviour over a switching period in order to soften the intrinsic discontinuities of the switching regulator, then the system is linearized near an operating point obtaining a linear time-invariant model, easier to compensate. The approach is, indeed, widely used for its simplicity and robustness, but it neglects some aspects that are fundamental for a discrete-time system. This section provides a review of the continuous-time analysis, underlining its principal assumptions and mathematical aspects.

To examine the fundamental concepts of the method, the voltage-mode control of a synchronous buck converter will be considered. The Buck converter is referred to be *synchronous* because of how the rectifying, or secondary, switch is implemented. Rather than using the typical free-wheeling diode as a passive rectifier, the Buck converter, depicted in fig.2.3, uses a controlled switch that is powered by the complementary version of the PWM signal $c'(t)$.

$$c'(t) = 1 - c(t) \tag{2.12}$$

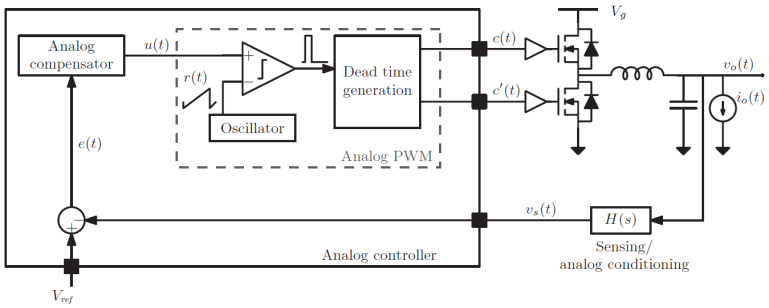



Figure 2.3: Analog control loop[3]

Synchronous rectification has the advantage of having a smaller voltage drop across the rectifier switch during conduction than a diode rectifier, which is necessary for controlling low output voltages. Additionally, the rectifier switch becomes bidirectional, ensuring CCM operation and converter controllability even in the absence of a load. Rather than using a resistance to depict the load, the loop in fig.2.3 uses an independent current source. For many digital loads in point-of-load applications, where the converter output current is dependent on the internal activity of the load and independent of the output voltage, this is an effective modelling option. The loop is based on a feedback of the regulated voltage $v_O(t)$. This voltage is first conditioned, it is scaled and filtered through $H(s)$, then is compared with a constant reference voltage $V_{ref}$. The error voltage, i.e. the difference between the two voltages, is processed by the analog compensator and transformed into a command signal $u(t)$. The latter is further compared with $r(t)$, the carrier of a trailing-edge pulse-width modulator that finally extrapolates the command signal for the power switches. The behaviour of these waveforms can be better appreciated in fig.2.4.

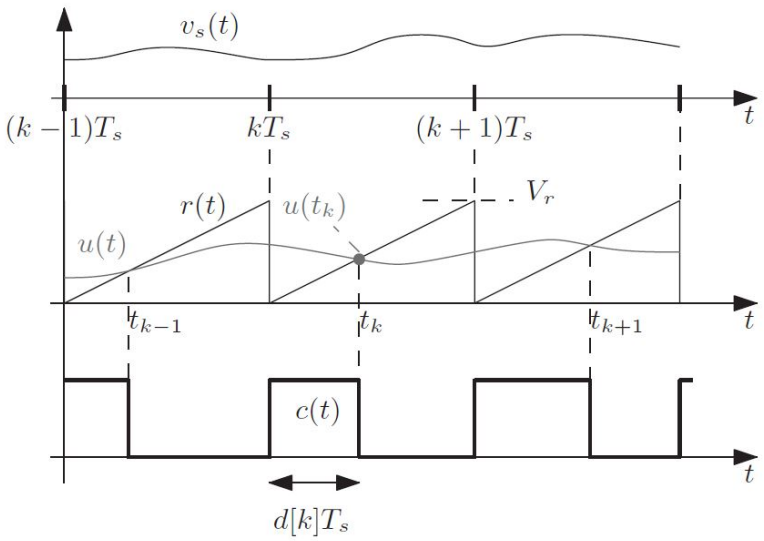



Figure 2.4: Analog control loop's waveforms[3]

Now the aim is to obtain a linear time-invariant system to construct a small-signal model of the buck. Let's focus the attention on the time-varying elements, the switches.

In fig.2.5 a simplified scheme is shown. It is possible to apply the moving average operator (2.15) to the time-varying quantities $v_x(t)$ and $i_g(t)$:

$$\bar{v}_x(t) \approx d(t)\bar{v}_g(t) \tag{2.13}$$

$$\bar{i}_g(t) \approx d(t)\bar{i}_L(t) \tag{2.14}$$

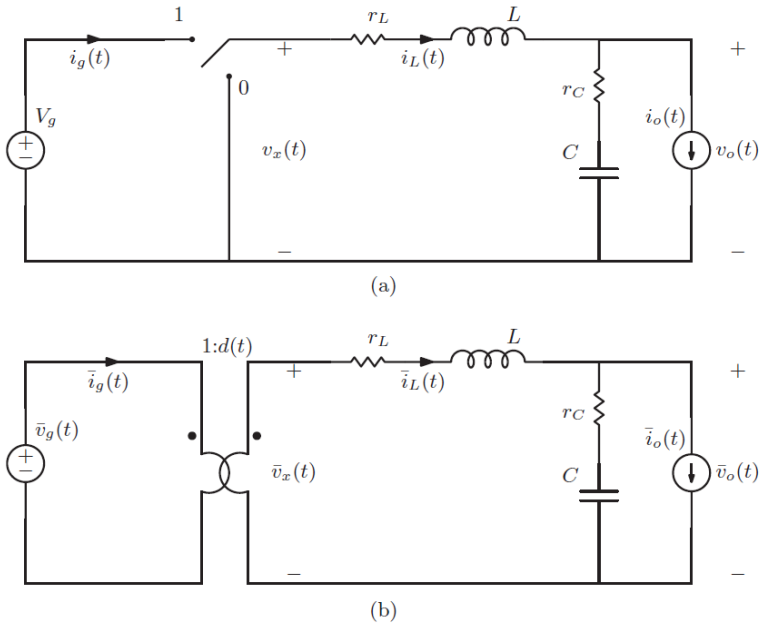



Figure 2.5: Averaging method[3]

$$x(t) \implies \bar{x}(t) = \frac{1}{T_S}\int_{t-T_S}^{t} x(\tau)d\tau \tag{2.15}$$

This approximation is acceptable just in case the two signals are uncorrelated or one of them is constant. In the proposed scenario the quantities $v_g(t)$ and $i_L(t)$ are essentially constant during a switching period $T_S$, so the approximation is justified. At this point the circuit can be represented, rather than with instant values, with average ones. De facto, implementing the reletions 2.16 the switching elements can be replaced by a trasformer, as depicted in fig.2.5. Now the model is time-invariant but still non-linear. A linearization around an operating point concludes the small-signal modelling. In fact now the previous average quantities can be expressed as:

$$\hat{\bar{v}}_x(t) \approx D\hat{\bar{v}}_g(t) + V_g\hat{d}(t) \tag{2.16}$$

$$\hat{\bar{i}}_g(t) \approx D\hat{\bar{i}}_L(t) + I_L\hat{d}(t) \tag{2.17}$$

Finally, the equivalent averaged small-signal model of the plant is obtained and represented in fig.2.6.

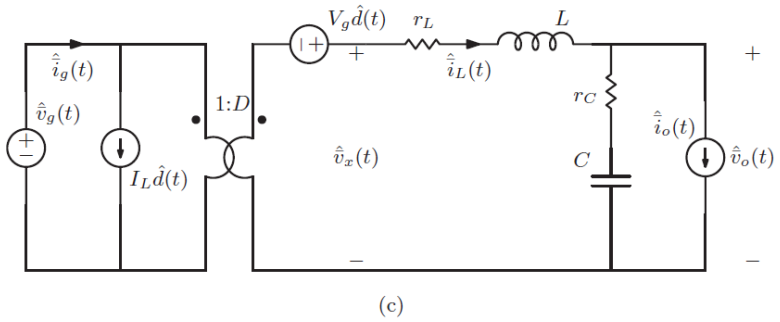



Figure 2.6: Small-signal model of the plant[3]

Looking at the circuit, a control-to-output transfer function $G_{vd}(s)$ of the plant can be extrapolated.

$$G_{vd}(s) = \frac{\hat{\hat{v}}_o(s)}{\hat{d}(s)}\Big|_{\hat{\hat{v}}_g=0,\hat{\hat{i}}_o=0} = V_g \frac{1 + sr_C C}{1 + s(r_C + r_L)C + s^2 LC} = G_{vd0} \frac{1 + \frac{s}{\omega_{ESR}}}{1 + \frac{s}{Q\omega_0} + \frac{s^2}{\omega_0^2}} \tag{2.18}$$

with

$$G_{vd0} = V_g \tag{2.19}$$

$$\omega_{ESR} = \frac{1}{r_C C} \tag{2.20}$$

$$\omega_0 = \frac{1}{\sqrt{LC}} \tag{2.21}$$

$$Q = \frac{1}{r_C + r_L}\sqrt{\frac{L}{C}} \tag{2.22}$$

Therefore, the behaviour of the converter's small signals is that of a second-order system with a resonant frequency and Q-factor ($\omega_0$, Q) and a real left half-plane (LHP) zero at $s = -\omega_{ESR}$. The output capacitor's equivalent series resistance (ESR) $r_C$ is where the zero comes from.

At this point it is important to include all the other contributions that impact on the closed-loop small-signal model of the converter. The aim of this study is to obtain the uncompensated loop gain in order to design the controller that has to guarantee the desired performances in term of stability and regulation capabilities. In fig.2.7 a small-signal model block diagram of the loop is depicted. It shows other contributions that have to be added to the plant's model obtained before (2.18). In this diagram the unknown is the compensator transfer function, so now $G_{PWM}(s)$ and $H(s)$ have to be defined.

For what concern $G_{PWM}(s)$ there are two type of pulse-width modulators:

- *Naturally sampled pulse width modulators* (NSPWMs) that elaborate continuous-time modulating signals. They are usually applied in analog controllers.
- *Uniformly sampled pulse width modulators* (USPWMs) that use a discrete-time modulating signal. In this technique the output is updated every switching period of PWM and, till the next one, is maintained constant.

For the analog control let's consider a NSPWM. As already seen in fig.2.4 and 2.3, at the k-th switching period the control signal $u(t)$ is compared with the PWM carrier, that is nothing but a triangular wave. The modulation is based on an analog comparator, that detects when a signal is smaller or greater than the other. In practice when $u(t)$ is greater than the carrier the output is high and vice versa. For this reason the duty cycle of a general PWM period is:

$$d[k] = \frac{u(t_k)}{V_r} \tag{2.23}$$

with $t_k$ the instant when the control signal intersects the carrier and the amplitude of the latter is $V_r$. Consequently, the duty cycle $d[k]$ during the $k-th$ switching cycle corresponds to a sampled version of the modulating signal $u(t)$. The lack of any delay between the modulator's natural sampling of $u(t)$ and the generation of the PWM modulated edge justifies, at least intuitively, the common practice in analog control modeling of treating the PWM as a simple gain block. Mathematically calling $\hat{d}$ and $\hat{u}$ the small signals of the control $u(t)$ and the output duty cycle $d(t)$ the gain of the PWM block is:

$$G_{PWM}(s) = \frac{\hat{d}}{\hat{u}} = \frac{1}{V_r} \tag{2.24}$$

That means that for the closed-loop dynamics, this block behaves like a simple constant, its behaviour is independent on the frequency. The behaviour of USPWMs is different and impacts on the frequency response of the closed-loop. This aspect is explained in detail in the section relating to digital control.

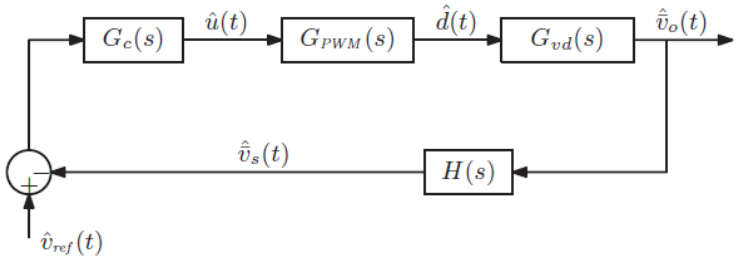



Figure 2.7: Small-signal model block diagram of the loop[3]

At this point the uncompensated loop gain become:

$$T_u(s) = G_{PWM}(s)\, G_{vd}(s)\, H(s) \tag{2.25}$$

Substituting:

$$T_u(s) = \frac{G_{vd0}}{V_r} \frac{1 + \frac{s}{\omega_{ESR}}}{1 + \frac{s}{Q\omega_0} + \frac{s^2}{\omega_0^2}} H(s) \tag{2.26}$$

Starting from 2.26, analog compensator design follows standard linear continuous-time control approach, with the target of conferring sufficient phase margins and an appropriate control bandwidth for the application.

### State-Space Averaging

Before entering in the compensator design theory, seems to be useful to mention the *State-Space Averaging* technique that permit to obtain a general method applicable to all type of switching converter to derive a small-signal model of the plant. In this thesis just the results of this approach are proposed, all the theoretical details can be found at [3].

Summarizing, in a switching converter two different topological states can be represented as linear set of state-space equations. Applying the moving average operator to both sides of the equations, repeating all the considerations made in the previous section about the admissibility of the operation, a large-signal state-space model is obtained. The same steps of linearization and perturbation are repeated also in this case. Finally, one can derive the *control transfer matrix*, which expresses how impacts a variation of the control command on the output and can be used for the compensator design:

$$\boxed{\boldsymbol{W}(s) = \frac{\hat{\boldsymbol{y}}(s)}{\hat{d}(s)}|_{\hat{\boldsymbol{v}}=0} = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{F} + \boldsymbol{G}}$$

### Continuous-time Compensator Design

The time of the analog compensator design has come. In fig.2.8 an example of controller is proposed. The network exploits external passive components to process the error voltage and provide the command signal $u(t)$ as output.

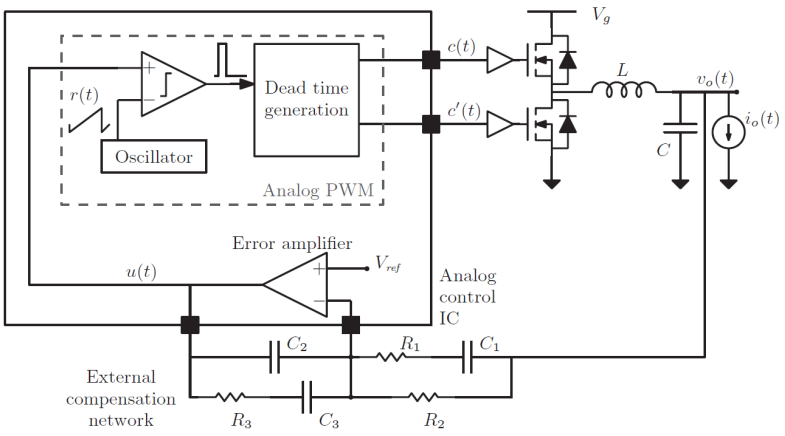



Figure 2.8: Example of analog compensation[3]

As design goal the crossover frequency is set to 1/10 of the converter switching frequency $f_c = \frac{f_s}{10}$, and the phase margin target is set at a desired value that is coherent with the specifications $\phi_m$, normally 50° is already enough for a good performance[3]. The proposed algorithm aims at reaching such phase margin adding phase to the uncompensated loop gain. The weakness of this approach is that one will be unable to subtract phase in case the transfer function to be modified has already sufficient margin. At this point, with the conditions just explained, a PD terms are commonly used for *lead* compensations, often in form of pole-zero pair:

$$G_{PD}(s) = G_{PD0}\frac{1+\frac{s}{\omega_z}}{1+\frac{s}{\omega_p}} \tag{2.27}$$

To place the zero and the pole is sufficient to considerate that the maximum phase boost given by the PD compensation occurs at

$$\omega_{max} = \sqrt{\omega_z \omega_p} \tag{2.28}$$

and in term of phase is

$$\angle G_{PD}(j\omega_{max}) = \arctan\left(\sqrt{\frac{\omega_p}{\omega_z}}\right) - \arctan\left(\sqrt{\frac{\omega_z}{\omega_p}}\right) = \frac{\pi}{2} - 2\arctan\left(\sqrt{\frac{\omega_z}{\omega_p}}\right) \tag{2.29}$$

Knowing that the phase boost has to be $\phi_m$ and that $\omega_c = \omega_{max} = \sqrt{\omega_p \omega_z}$ is possible to extract the frequency of the zero and the pole:

$$\omega_z = \omega_c\sqrt{\frac{1-\sin\theta}{1+\sin\theta}} \approx 2\pi \cdot f_{zero}$$
$$\omega_p = \omega_c\sqrt{\frac{1+\sin\theta}{1-\sin\theta}} \approx 2\pi \cdot f_{pole}$$

The low frequency gain $G_{PD0}$ can be obtained imposing unity loop gain at the desired crossover frequency $f_c$.

$$|T(j\omega_c)| = |T_u(j\omega_c)|G_{PD0}\sqrt{\frac{1+(\frac{\omega_c}{\omega_z})^2}{1+(\frac{\omega_c}{\omega_p})^2}} = 1 \tag{2.30}$$

Making explicit $G_{PD0}$

$$G_{PD0} = \frac{1}{|T_u(j\omega_c)|}\sqrt{\frac{1+(\frac{\omega_c}{\omega_p})^2}{1+(\frac{\omega_c}{\omega_z})^2}} \tag{2.31}$$

Now it is necessary to add an integral action to make zero the steady-state error that in practise means to guarantee an high value of dc gain to improve the regulation. For this reason a *lag* term is added to the compensator transfer function:

$$G_{PI}(s) = G_{PI\,\inf}(1+\frac{\omega_l}{s}) \tag{2.32}$$

The high frequency gain $G_{PI\infty}$ is set to one not to alter the *lead* term gain or its phase margin. Same for the $\omega_l$ that is chosen to be much smaller than the $\omega_c$. So, it can be expressed as:

$$\omega_l = \frac{\omega_c}{20} = 2\pi \cdot f_l \tag{2.33}$$

The complete compensator transfer function is therefore:

$$G_{PID}(s) = \underbrace{(1 + \frac{\omega_l}{s})}_{PI} \cdot \underbrace{G_{PD0}\frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}}_{PD} \tag{2.34}$$

This expression can be translated into a compensation circuit as that depicted in fig.2.8. Assuming that $C_3 \gg C_2$, $G_C(s)$ is:

$$G_{PID}(s) = -\frac{\hat{u}(s)}{\hat{\hat{v}}_O(s)} = \underbrace{(1 + \frac{1}{sR_3C_3})}_{PI} \cdot \underbrace{\frac{R_3}{R_2}\frac{1 + s(R_1 + R_2)C_1}{1 + sR_1C_1}}_{PD} \cdot \underbrace{\frac{1}{1 + sR_3C_2}}_{HFPole} \tag{2.35}$$

Note the presence of an HF Pole $\omega_{p2}$ used to reduce the gain of the compensator at high frequencies to minimize the effect of unwanted switching harmonics that come back from the feedback. A good design has $\omega_{p2} = 10\,\omega_c$ not to influence the phase margin.

### 2.2.2 The Digital Control Loop

Digital control of a switched-mode power converter differs from analog control in two aspects[3]:

1. **Sampling**. Inside the loop computations are made in discrete-time, a quantization of a continous-time signal is needed.
2. **Amplitude Quantization**. The controller is digital, for this reason the signal amplitude has to be quantized.

These operations are not found in analog control and impact on the behavior of the system. The first quantization introduces a delay into the loop and the second causes nonlinear effects that could impair the performance of static and dynamic regulation in converters that are managed digitally.

This section presents the characteristics and modelling of the principal elements of the digital control loop together with their intrinsic problems. The main blocks are shown in fig.2.9, in which a simplified structure is reported.

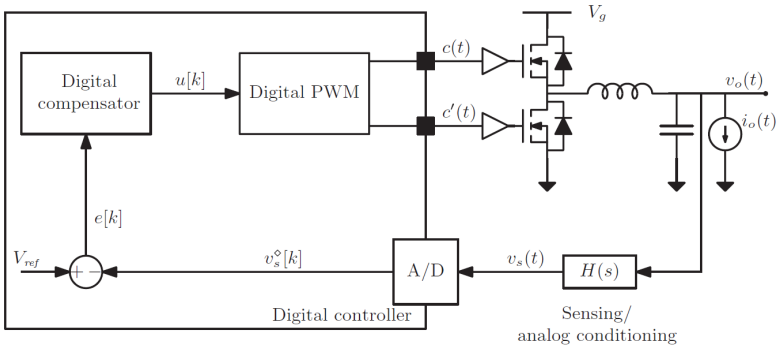



Figure 2.9: Simplified structure of a digital controlled buck[3]

### A/D Conversion

As in the analog control the output voltage is sensed by $H(s)$, but then the signal is converted in the digital domain, i.e. is sampled by an ADC. Truly speaking there are two ways of using the ADC: the first takes the output voltage and directly converts it into digital values; the second uses a subtractor, usually made with an operational amplifier (OPAMP), to compare the output voltage with the reference and then the ADC convert the error between them. The second option includes more area, given by the OPAMP and a DAC that provides the reference, but this approach achieves better performances in term of ADC resolution. In fact all the quantization steps of the ADC are used to obtain a digital error and improve the convertion resolution, in other words the ADC is able to appreciate smaller differences with the reference and enhance the resolution of the loop.

The ADC provides digital sequences every sampling period $T$. The simplest choices is to have:

$$\boxed{T = T_s}$$

in this why the sampling is *synchronized* with the switching action of the buck. The advantage is that the sampling instant occurs at the same position on the switching period.

The spectrum of the output voltage of a buck converter contains the DC component with some low-frequency contributions but also the high-frequency component from the switching action (the ripple) and its harmonics. The conditioning $H(s)$ can attenuate them but cannot completely eliminate their effect. For this reason a certain level of spectral aliasing is always present. The sampling period influences the level of aliasing in the loop. Let's suppose that $f_{sampling} = \frac{1}{T}$ is a *multiple* of $f_s$ (*multisampling*): the sampling action gives as output a spectrum that is periodic in frequency with $3f_s$, the Nyquist frequency ($\frac{f_{sampling}}{2}$) indicates the highest frequency visible to the digital system. But this mechanism involves that the DC component of the sampled signal is different from the input one, due to aliasing at low-frequency. Furthermore the switching frequency, i.e. the ripple is visible to the digital system and this happen every time the sampling frequency is higher than the switching one. In fig.2.10 is reported an example in which $f_{sampling} = 3f_s$. The digital controller must perform a filtering operation on this residual high-frequency component in the digital domain.

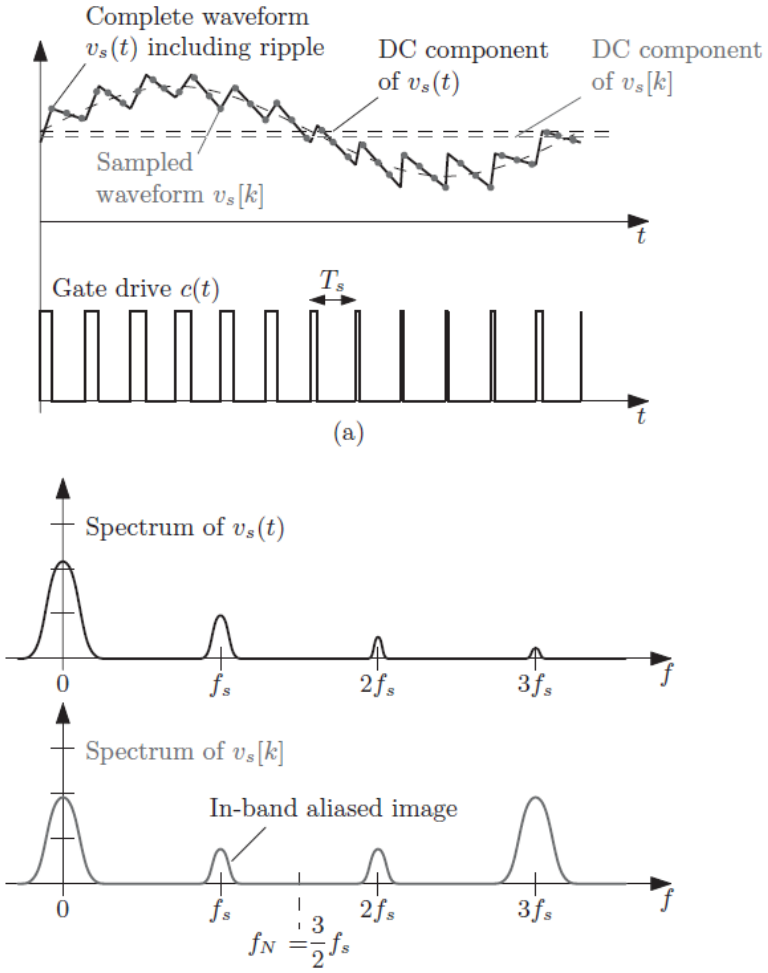



Figure 2.10: Sampling frequency higher than the switching frequency[3]

On the other hand, with $f_{sampling} = f_s$ only the DC is influenced by aliasing, there is no residual component at the switching frequency, as illustrated in fig.2.11.

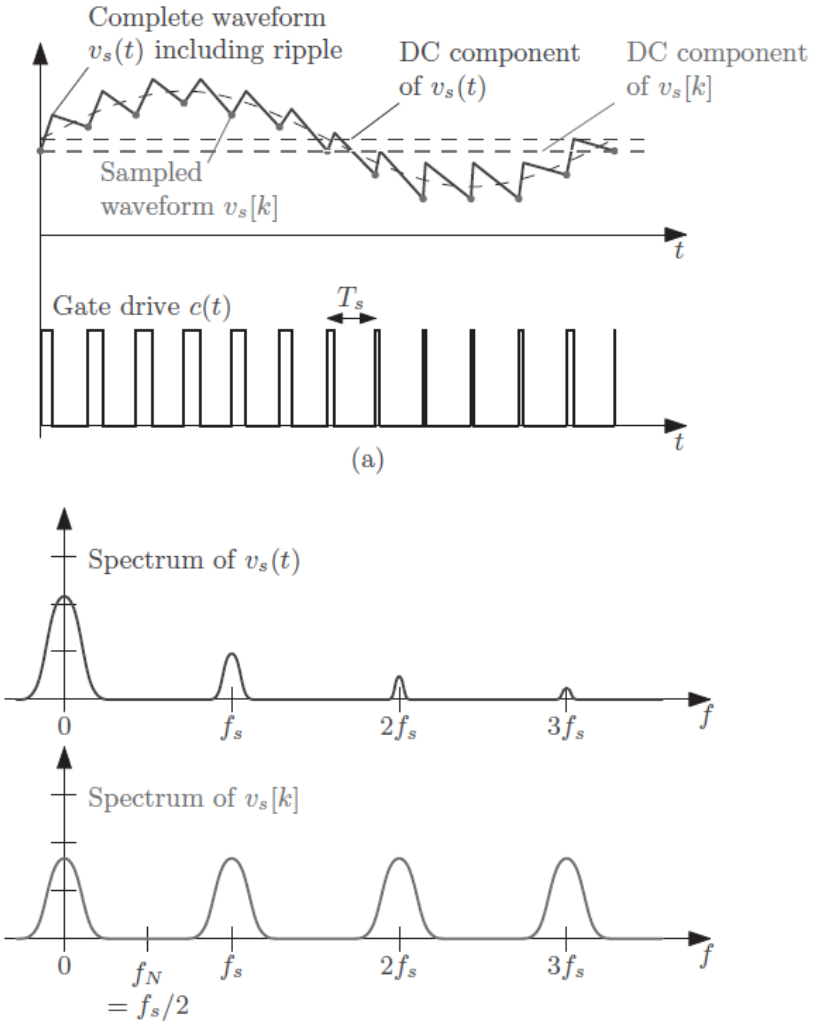



Figure 2.11: Synchronous sampling[3]

The drawback of this approach is that the steady-state is modified. Fig.2.12 make it more clear. The sampling action may corrupt the dc component, making it different from the input voltage.

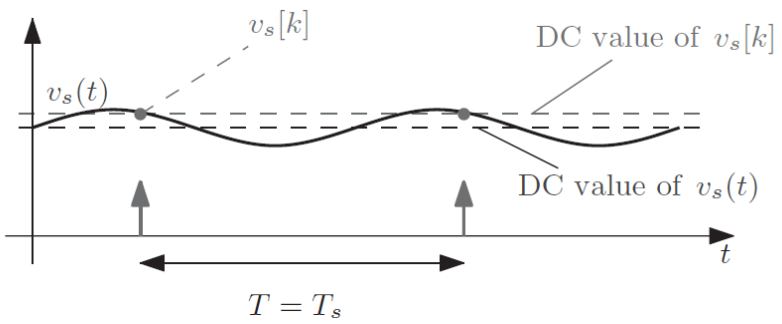



Figure 2.12: DC aliasing[3]

The design choice depends on multiple factors: multisampling is necessary whenever the switching ripple has a considerable amplitude in respect of the regulated voltage; the ADC throughput is also important, indeed it depends on the clock frequency, on its structure and least but not last on its cost (at fixed number of bit, faster ADCs are more expensive); a digital low-pass filter introduces delay, so it impacts on the loop gain performance. For these reasons multisampling cannot be used in all applications. In case of small ripple it is reasonable to use synchronous sampling.

### Digital Pulse Width Modulation

Once the digital compensator (explained later in this chapter), that takes $e[k]$ as input and, after a delay $t_{calc}$, generates the control command $u[k]$, has provided its output, the DPWM elaborates the control command $u[k]$ and produce a train of pulses with duty cycle $d[k]$ proportional to $u[k]$.

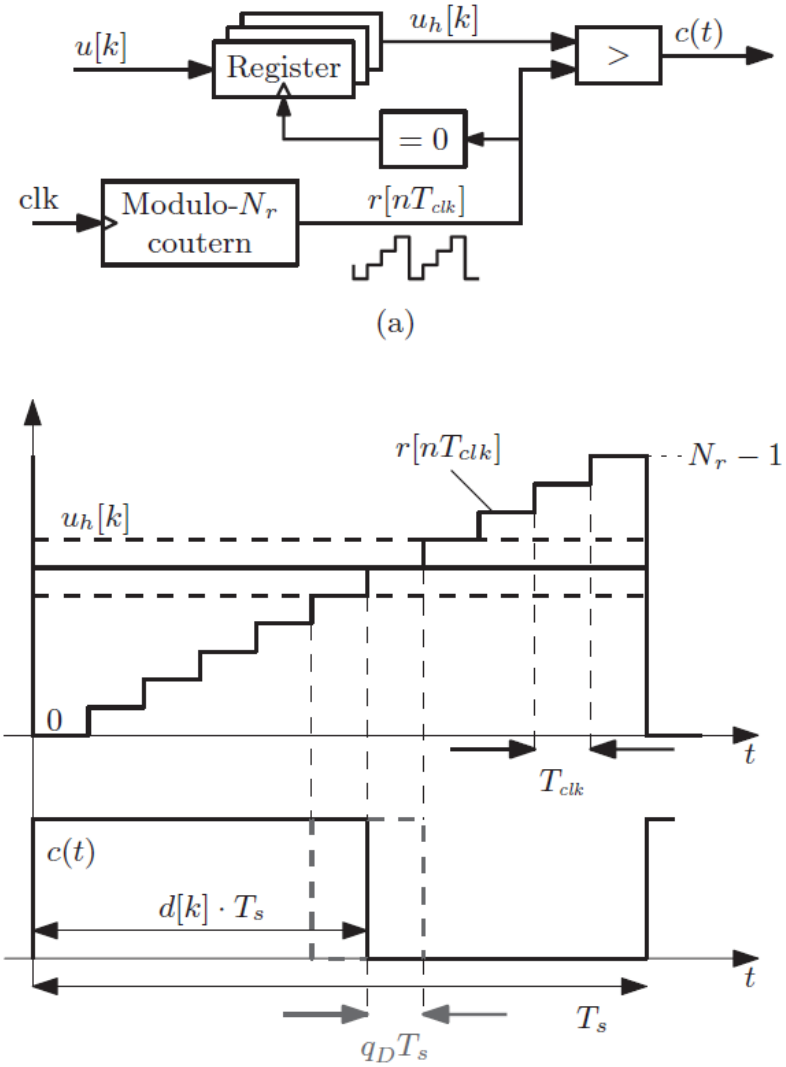



Figure 2.13: Counter Based DPWM[3]

In fig.2.13 a simple counter-based DPWM is illustrated. In practice, this structure is a digital implementation of the analog PWM, where there is a triangular waveform instead

of a free-running counter. In this case the ramp is quantized, with $N_r$ levels, clocked at a frequency $f_{clk}$. Every switching period the ramp completes its count:

$$T_s = N_r T_{clk} \tag{2.36}$$

During every switching period a value $u[k]$ is maintained constant into a register to be compared with the counter's output. When the two values are equal a pulse is generated:

$$d[k] = \frac{u[k]}{N_r} \tag{2.37}$$

The limitation of this approach is that the modulator can generate a limited set of duty cycles and their resolution is associated with the clock period:

$$\Delta t_{DPWM} = T_{clk}$$
$$q_{dutycycle} = \frac{Tclk}{T_s}$$

This resolution can be expressed as an equivalent number of bits:

$$n_{DPWM} = log_2\left(\frac{T_s}{\Delta t_{DPWM}}\right) = log_2\left(\frac{1}{q_{dutycycle}}\right) = log_2(N_r) \tag{2.38}$$

**Limit Cycles Oscillations**

Despite the above undeniable benefits, digital power converters' precision and transient reaction may be compromised by their intrinsic quantization, which may also result in undesirable low-frequency limit cycle oscillations (LCOs) that are absent from analog controlled converters. These problems are largely caused by the digital pulsewidth modulator's (DPWM) low precision, it is indeed fundamental to study the effect to establish how they can be avoided[4].

Supposing that the compensator designed makes the loop stable, it means that in steady-state the error voltage sampled by the ADC corresponds to its zero-bin, i.e. the interval of voltages that are traduced in digital domain as the number "0". But let's imagine just for a moment to have ideal ADC and DPWM with an infinite precision: their characteristics are straight lines within their ranges. The intersection of these lines gives a point (point A) that represent the equilibrium between the output of the DMPW and the input of the ADC, as showed in fig2.14. Anyway the output voltage is not equal to $V_{ref}$, an integral action is required to bring to infinite the DC loop gain, in this way the error goes to zero, as indicated by the point B in the figure.

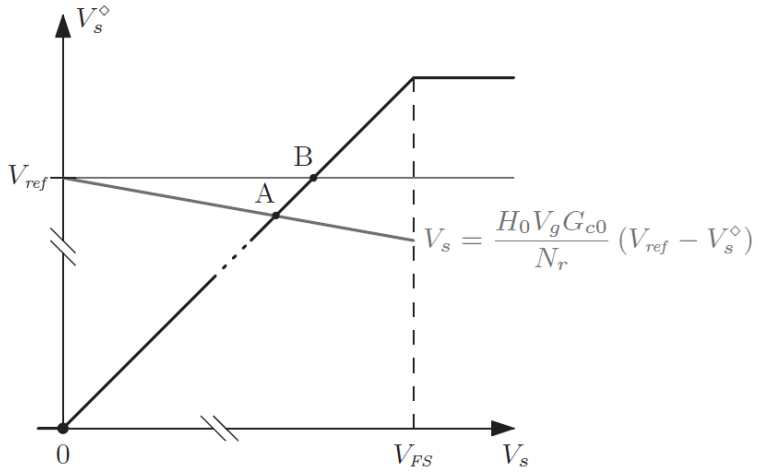



Figure 2.14: Equilibrium point between ideal ADC and DPWM[3]

The situation is different when the precision of ADC and DPWM is not infinite, their characteristics are quantized ramps, as the fig.2.15 represents. In this case the intersection between them, i.e. the point A, is not a stable point because it occurs on the vertical segment of the ADC's ramp. It means that the loop will continue to oscillate around two points. Adding an integral gain in the compensation the error can be brought to zero, so multiple points represent equilibrium for the loop. This condition is indicated in the figure with the B segment, but the stability is satisfied only if the orizontal step of the DPWM ramp is shorter then the ADC's one.

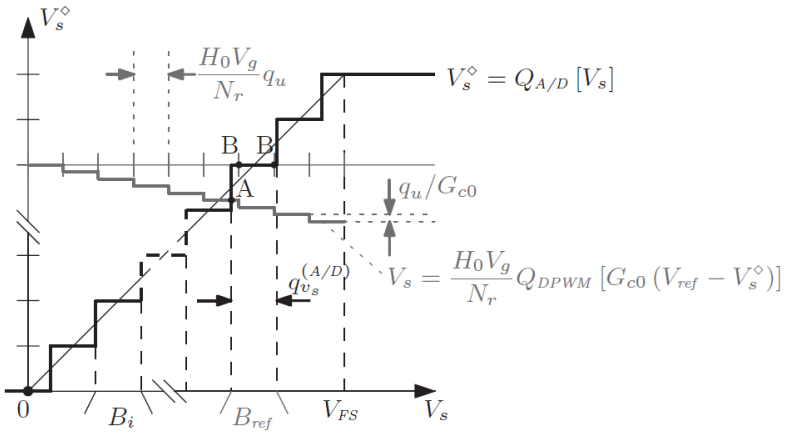



Figure 2.15: Equilibrium point between real ADC and DPWM[3]

This concept can be better appreciated in fig.2.16. The graph shows how the quantization step works. The vertical axis represents the output buck voltage, it is obtained by a quantized PWM, each segment indicates a level of $V_O$ obtainable by a step variation of the duty cycle. On the other hand, the same quantized $V_O$ is sampled by the ADC which has a limited number of bins. The example in the figure explains why LCO occur: none of the quantized output voltage fall in the zero-error bin of the ADC, so the loop oscillates

between the "001" bin and its negative correspondent, never reaching the stability[3]. It happens because the DPWM bin is larger than the ADC's one.

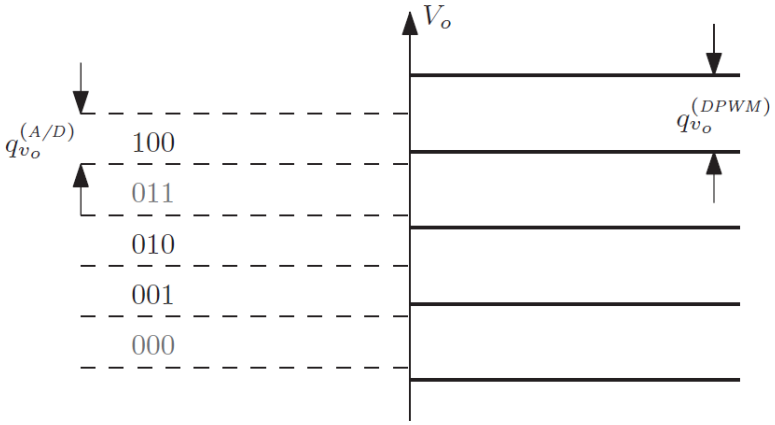



Figure 2.16: ADC and DPWM wrong step size[3]

If the ADC has the largest bin, the quantized voltage hits the zero-error condition, as shown in fig.2.17.

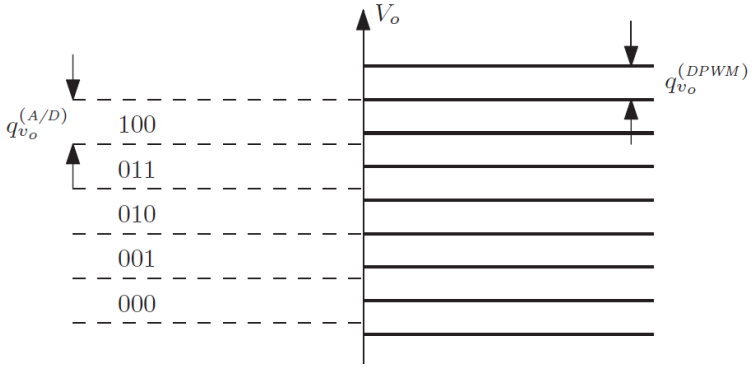



Figure 2.17: ADC and DPWM correct step size[3]

The above-mentioned considerations, together with the assumption that the compensator employs an integral action, $(Ki > 0)$, lead to a general no-limit-cycling condition.

$$\boxed{q_{v_O}^{(DPWM)} < q_{v_O}^{(A/D)}}$$

### Loop Delays

The existence of delays of various types, which influence the response and must be analyzed and taken into consideration in the design of digital control loops, is the key distinction between analog and digital control loops. As already mentioned, there are multiple types of delay inside a digital control loop, the first is the *control delay* intended as the time interval between the sampling event and the instant in which the modulator latches the new command value $u[k]$ in its register. The amount of this delay changes according to the structure of the used logic:

- In *hardware-based controllers* the integrated circuit is customised for the particular application. It is optimised to have fast A/D conversion and computational circuitry. In this case the control delay is usually a fraction of the switching period of the buck.
- In *software-based controllers* the structure is based on a µ$C$ running a control program. In this case the A/D conversion time combined with a logic delay is comparable to the switching period.

Furthermore, a *modulation delay* has to be added, because it impacts on the loop performance. This time corresponds to the interval between the update of the DPWM register an the actual update of the duty cycle. Long story short, in a trailing-edge DPWM this time is directly proportional to the duty cycle, so the higher the duty the higher the delay (fig.2.18)

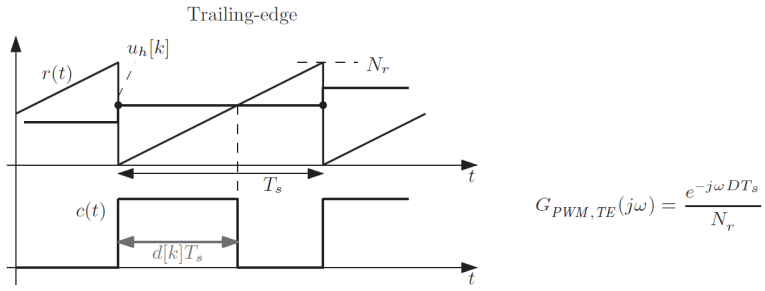



Figure 2.18: Modulation delay[3]

The total loop delay is the sum of the two contributions and it can be determined by inspection of the elements of the loop:

$$t_d = t_{ADC} + t_{calc} + t_{mod} = t_{cntrl} + t_{mod} \tag{2.39}$$

### 2.2.3 Discrete-Time Modelling

*Discrete-Time modelling techniques* offer a more accurate representation of the converter dynamics when controlled digitally, indeed they describe *sampled* converters waveforms behaviour without averaging steps in the process [3]. Furthermore they include in the model the fundamental aspect of loop delays, witch are not present in the continuous-time theory of analog controls.

The discrete-time approach, as the continous-time, starts with a state-space representation of the two topological states. Also in this case the details of the approach are left to the literature [3], the results are proposed below. The general steps of this approach are very intuitive:

1. In first step $\boldsymbol{x}$ at $[k+1]$, i.e. the sampled state vector, is expressed as a function of $\boldsymbol{x}$, the input vector $\boldsymbol{v}$ and the control input $u$ at time $k$:

$$\boldsymbol{x}[k+1] = \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{V}, u[k]) \tag{2.40}$$

2. The operating point of the converter $\boldsymbol{Q}$ is found using the equation of the first step, solved for constant value of the tate vector $\boldsymbol{x}[k+1] = \boldsymbol{x}[k] = \boldsymbol{X}$ and for a constant control input $u[k] = U$:

$$\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{V}, U) \tag{2.41}$$

So, $\boldsymbol{Q}$ is:

$$\boldsymbol{Q} = (\boldsymbol{X}, \boldsymbol{V}, U) \tag{2.42}$$

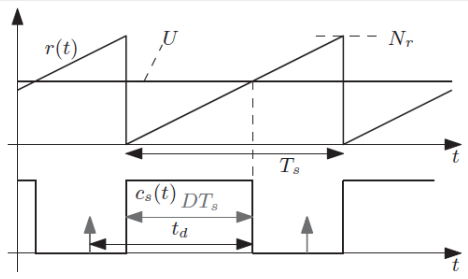
3. In the last step the state equation is *perturbed* and *linearized* around $\boldsymbol{Q}$, leading to a small-signal state-space description of the sampled dynamics. Finally, the small-signal control-to-output transfer matrix $\boldsymbol{W}(z)$ is:

$$\boxed{\boldsymbol{W}(z) = \frac{\hat{\boldsymbol{y}}(z)}{\hat{u}(z)} = \boldsymbol{\delta}(z\boldsymbol{I} - \Phi)^{-1}\gamma}$$

This approach can be extended to every type of converters, but just the buck will be analyzed in this thesis.

In order to obtain the transfer function is important to specify the modulating edge of the DPWM. The reason come from the different delay given by the various techniques. The simplest and the most common is the trailing edge modulation, whose equations is reported in fig.2.19.

**TABLE 3.1 Discrete-Time Small-Signal Model—Trailing-Edge Modulation**




Discrete-Time Model

$$\Phi = e^{\boldsymbol{A}_0(T_s - t_d)} e^{\boldsymbol{A}_1 D T_s} e^{\boldsymbol{A}_0(t_d - DT_s)},$$

$$\gamma = \frac{T_s}{N_r} e^{\boldsymbol{A}_0(T_s - t_d)} \boldsymbol{F}_{\downarrow},$$

$$\boldsymbol{\delta} = \boldsymbol{C}_0.$$

$$\boldsymbol{F}_{\downarrow} \triangleq \left(\boldsymbol{A}_1 \boldsymbol{X}_{\downarrow} + \boldsymbol{B}_1 \boldsymbol{V}\right) - \left(\boldsymbol{A}_0 \boldsymbol{X}_{\downarrow} + \boldsymbol{B}_0 \boldsymbol{V}\right),$$

$$\boldsymbol{X}_{\downarrow} = \left(\boldsymbol{I} - e^{\boldsymbol{A}_1 D T_s} e^{\boldsymbol{A}_0 D' T_s}\right)^{-1} \cdot$$

$$\left[-e^{\boldsymbol{A}_1 D T_s} \boldsymbol{A}_0^{-1} \left(\boldsymbol{I} - e^{\boldsymbol{A}_0 D' T_s}\right) \boldsymbol{B}_0 - \boldsymbol{A}_1^{-1} \left(\boldsymbol{I} - e^{\boldsymbol{A}_1 D T_s}\right) \boldsymbol{B}_1\right] \boldsymbol{V}.$$

Figure 2.19: Trailing edge equations for discrete-time modelling[3]

At this point some inputs are needed to complete the algorithm:

- $V_g$

- $I_O$
- $D = \frac{V_O}{V_g}$
- Total loop delay $t_d$
- Values of the various components of the plant

**Discrete-time Compensator Design**

There are two possible ways to design a discrete-time compensator:

- **Discretization of a continuous-time compensator**
- **Direct discrete-time design (p-mapping)**

The first is done starting from an averaged small-signal model of the plant and then obtaining a compensator transfer function, formulated in continuous-time, so using the analog common rules of loop stability and compensation. At this point, the continuous-time system can be discretized using a variety of methods to create a discrete-time system with comparable frequency response properties[3]. The simplest technique is the *Backward Euler* method, or *backward rectangular rule*, which is based on a rectangular approximation of the continuous-time integral operator:

$$\int_{(k-1)T_s}^{kT_s} x(\tau)\, d\tau \approx T_s x(kT_s) \tag{2.43}$$

In Z-transform domain, this discretization can be seen as a mapping from Laplace domain, defined by:

$$s \implies \frac{1 - z^{-1}}{T_s} \tag{2.44}$$

The PID transfer function become:

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad \frac{T_s K_i}{1 - z^{-1}} \quad + \quad \frac{K_d}{T_s}(1 - z^{-1}) \tag{2.45}$$

The other method is the *Tustin approach* or *trapezoidal rule*. The idea is to represent the integral over the sampling step with a trapezoidal approximation of the integrand:

$$\int_{(k-1)T_s}^{kT_s} x(\tau)\, d\tau \approx \frac{T_s}{2}(x(kT_s)) + x((k-1)T_s) \tag{2.46}$$

In Z-transform domain, this discretization can be seen as a mapping from Laplace domain, defined by:

$$s \implies \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{2.47}$$

The PID transfer function become:

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad K_i \frac{T_s}{2} \frac{1+z^{-1}}{1-z^{-1}} \quad + \quad K_d \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}} \tag{2.48}$$

The frequency responses of the continuous-time compensator an the discrete-time one essentially is the same throughout most of the frequency range and begin taking distance from one another just in vicinity of the Nyquist rate of the system, which is $\frac{f_s}{2}$[3].

Another problem to consider using this type of approach is that the averaged continuous-time small-signal model doesn't take into account the aspect of digital control that are not found in the analog one. As already mentioned, there are delays due to A/D convertion, internal computation time and PWM modulation that worsen the phase margin of the loop. For this reason is extremely important to add this amount of phase delay in the uncompensated loop transfer function before starting the compensator design. It can be performed just multiplying the transfer function by an exponential that includes that amount of delay ($t_d$):

$$H_{del}(s) = H(s)e^{-st_d} \tag{2.49}$$

This is a fundamental difference between the analog and the digital control, the latter's response is retarded. It starts to respond to the error variation only after it samples the input, computed the PID output and update the quantized duty cycle. Unfortunately, not always this delay is completely determined, it was already seen that it depends also on the output duty cycle or the throughput of the ADC. The value of $t_d$ is obtained considering the operating point of the buck and for this reason the added delay on $H_{del}$ is usually not optimized.

The second approach constructs directly the PID on the discrete-time small-signal model[3]. The objective is to determine the compensator z-domain transfer function $G_c(z)$ coherently with the system specifications of control bandwidth and stability margin. Multiple approaches exist to synthesise PID functions in the frequency domain. Here the proposed method is based on the *bilinear transform* and belongs to a *mapping-based* approach class. Firstly the uncompensated loop gain $T_u(z)$ is mapped into an equivalent continuous-time *p-domain*, where is possible to design the compensator with the common analog rules, then the synthesised transfer function is back-mapped into the z-domain and implemented in the digital controller. The approach is not affected by average modelling limitations because the design process starts and ends in the z-domain, the mapping is just a facility to better understand the choices made on pole and zero placement. The mapping can be represented as follow:

$$z(p) = \frac{1 + p\frac{T_s}{2}}{1 - p\frac{T_s}{2}} \tag{2.50}$$

It can be seen as an operator to move from the z-domain to the p-domain. The reverse operator is:

$$p(z) = \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}} \tag{2.51}$$

To be precise in the p-domain frequencies present a sort of distortion compared with normal frequencies (s-domain), generally called *frequency wrapping*. The relation between the two domains is:

$$\omega' = \frac{2}{T_s}\tan(\omega\frac{T_s}{2}) \tag{2.52}$$

and it has to be taken into account to obtain the right gains and pole and zero frequencies. At this point, neglecting all the other mathematical details, the transfer function

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad \frac{K_i}{1-z^{-1}} \quad + \quad K_d(1-z^{-1}) \tag{2.53}$$

is mapped into p-domain, becoming:

$$G'_{PID}(p) = \quad K_p \quad + \quad \frac{K_i}{T_s}\frac{1+\frac{p}{\omega_p}}{p} \quad + \quad K_d T_s \frac{p}{1+\frac{p}{\omega_p}} \tag{2.54}$$

$\omega_p$ come from the Nyquist frequency. Indeed, the Nyquist angular frequency $\omega_N = \omega_s/2 = \pi/T_s$ is represented in the z-plane by the $z = -1$ point, that is mapped to infinity by 3.49. As both the integral and the derivative terms have finite gains at the Nyquist rate in the z-domain, they must have finite gains at $\omega' = +\infty$ in the p-domain.

To design the PID transfer function in the p-domain the steps are exactly the same for the analog case. So firstly is necessary to chose a crossover frequency. In the example is $\omega_c$, that in order not to have frequency distortion, has to be *prewrapped*:

$$\omega'_c = \frac{2}{T_s}\tan(\omega_c\frac{T_s}{2}) \approx 2\pi \cdot f'_c \tag{2.55}$$

Then, $\omega_p$ has to be computed:

$$\omega_p = \frac{2}{T_s} \approx 2\pi \cdot f_p \tag{2.56}$$

Now, the phase and gain values at $\omega_c$ of $T_u(z)$ are required:

$$\left|T_u(e^{j\omega_c T_s})\right| = \left|T'_u(e^{j\omega'_c T_s})\right| \tag{2.57}$$

$$\angle T_u(e^{j\omega_c T_s}) = \angle T'_u(e^{j\omega'_c T_s}) \tag{2.58}$$

Following the analog rules:

$$G'_{PD0} = \frac{1}{|T'_u(j\omega'_c|}\sqrt{\frac{1+(\frac{\omega'_c}{\omega_p})^2}{1+(\frac{\omega'_c}{\omega_{PD}})^2}} \tag{2.59}$$

$$\omega_{PI} = 2\pi \cdot (\omega_c/20) \tag{2.60}$$

$$G'_{PI\infty} = 1 \tag{2.61}$$

Now the PID is completely defined in the p-domain, because now $G'_{PD0}$, $G'_{PI\,\inf}$, $\omega_{PI}$, $\omega_{PD}$ and $\omega_p$ have been computed.

$$G'_{PID}(p) = \underbrace{G'_{PI\infty}(1 + \frac{\omega_{PI}}{p})}_{PI}\underbrace{G'_{PD0}\frac{1 + \frac{p}{\omega_{PD}}}{1 + \frac{p}{\omega_p}}}_{PD} \tag{2.62}$$

From the last equation $K_p$, $K_i$ and $K_d$ can be obtained directly in the z-domain using:

$$K_p = G'_{PI\infty}G'_{PD0}(1 + \frac{\omega_{PI}}{\omega_{PD}} - \frac{2\omega_{PI}}{\omega_p})$$
$$K_i = 2G'_{PI\infty}G'_{PD0}\frac{\omega_{PI}}{\omega_p}$$
$$K_d = \frac{G'_{PI\infty}G'_{PD0}}{2}(1 - \frac{\omega_{PI}}{\omega_p})(\frac{\omega_p}{\omega_{PD}} - 1)$$

# Chapter 3

# Digitally Controlled Buck Converter: complete circuit analysis

Once the theory of digitally controlled buck converters has been proposed, it is possible to provide a complete review of the analysis made in this thesis project. Essentially, this chapter is structured in four parts:

- The first one will cover the topology description, focusing on the characteristics of each block of the model;
- The second is dedicated to the small-signal analysis of the system and its frequency compensation, made with the support of the Matlab environment;
- The third one will be devoted to the description of the simulations carried out in Simulink environment and to the optimisations made on the model based on the simulations results;
- The fourth part will provide a detailed description of the dithering techniques and a brief comparison with the $\Sigma\Delta$ modulation, which one has been chosen for the design and why.
- A final review will end up the chapter.

## 3.1 Specification Description

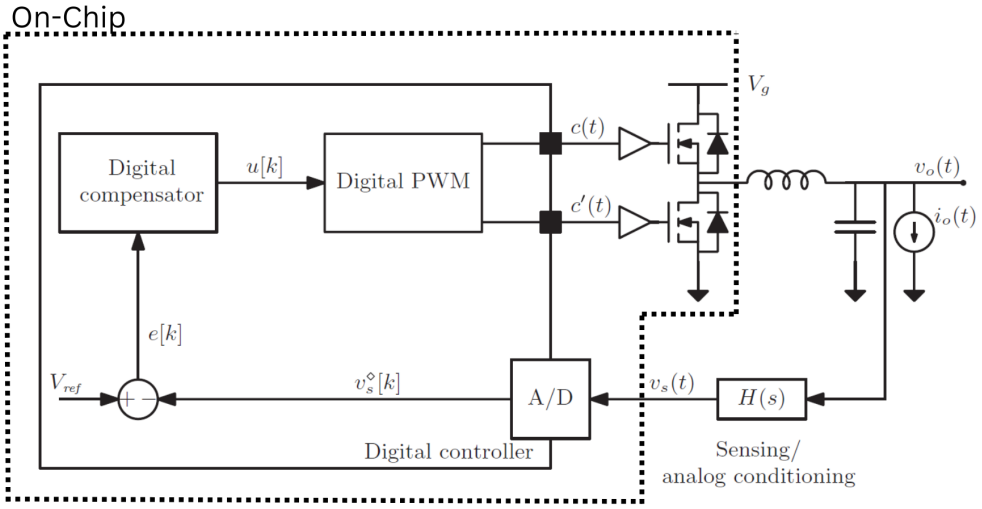



Figure 3.1: Integrated, digital Buck converter

In order to choose the values of the plant components is necessary to define the specification of the Buck under study. The latter is meant to be used after a primary rail converter that preregulates starting from a battery, for this reason the input voltage presents a limited range that can be $6\,V \pm 10\%$. The converter works as back-end point-of-load (POL) to supply different blocks of automotive control units, such as $\mu C$, interfaces or I/O, so the regulated voltage can be chosen to be in a range of $0.5 - 3.3\,V$. The control circuit, together with the power transistors, will be integrated with the STMicroelectronics' BCD technology. The fig.3.1 shows what is on-chip. The output current for the application can be reasonably set to a maximum of $4\,A$. It would be desirable to have the efficiency around 90% at $3.3\,V$, while for the minimum regulated voltage it is expected to be lower: the power dissipated by the circuit internally remains constant, no matter the output voltage, due to the conduction power of the MOSFETs. Their $R_{ON}$ is not particularly small, it is around $80\,\text{m}\Omega$, due to the fact that is preferable to have them integrated in the chip. The control loop is a voltage mode and the switching frequency is fixed at 2.4 MHz. This is a synchronous Buck in which the freewheeling action is performed by the low-side MOS. For what concerns the dynamic performances, a maximum response of 25 mV is requested for both input voltage and load current steps. For the static performance a maximum of 2% of the regulated voltage is desired. The table 3.1 summarises the specifications.

| SPECIFICATIONS | |
|---|---|
| **INPUT VOLTAGE** | $6\,V \pm 10\%$ |
| **OUTPUT VOLTAGE** | $0.5 - 3.3\,V$ |
| **OUTPUT CURRENT** | $0 - 4\,A$ |
| **EFFICIENCY** | Around 90 % at 3.3V |
| **BUCK CHARACTERISTICS** | |
| **LOOP MODE** | Voltage Mode |
| **SWITCHING FREQUENCY** | 2.4 MHz (fixed) |
| **DYNAMIC REGULATION** | ±25 mV (0 to 4A in 4µs) |
| **STATIC REGULATION** | 2% of regulated voltage |
| **Power MOS type** | Integrated |

Table 3.1: Specifications and main characteristics of the converter.

On the basis of this specification is possible to obtain a first evaluation of the plant components for a synchronous buck working in continuous conduction mode[7].

For the maximum switch current the value of D at the maximum input voltage condition is needed, this situation brings to the worst case. The result must be divided by the efficiency of the buck, $\eta$, (eq.3.1), the latter has to provide also the dissipated power that does not reach the load. It means that the converter has to transfer more power to lead to the desired output performances. The efficiency cannot be calculated in an easy way, an automatic Excel sheet owned by STMicroelecronics is used, containing all the formulas needed for the buck characterization. For both $0.5\,V$ and $3.3\,V$ of regulated voltage, an estimation of the efficiency is performed, including parameters such as conduction losses, switching losses of power MOS, low-side dead-time and recovery losses, output capacitance loss, loss due to MOS drivers and finally internal bias loss. All these quantities impact the output power provided by the buck, especially the conduction loss that in both cases constitutes about the 68% of the total power loss of the integrated circuit. This value impacts in heavier manner on the 0.5V of regulated voltage, the reason comes from the total output power provided in this case, that is at maximum 2 W. The dissipated power, considering all the contributions, is 2 W too, so the efficiency is around 50%. Instead, in the 3.3V case, the output power is higher, around 13.2 W, so, considering the power losses slightly higher than 2 W, the efficiency is near 86%. This value is slightly lower than the specification, but for the thesis purpose it is considered acceptable.

Now, a value of the current ripple in the inductor can be estimated. Normally it can be assumed to be 20% of the output current (3.2), a lower value leads to a bigger inductor that means higher cost and area. Knowing the relation between the current ripple and the inductor value (eq.3.3), a first value of inductance can be extrapolated (eq.3.4). Then we can include in the equation of the current ripple the real value of duty cycle, that considers the efficiency, to obtain a better sizing of the inductor. In this way the minimum value of inductance increases and finally the component is chosen between all the possible sizes provided by the standard series.

Regarding the output capacitor, 3.5 gives the minimum value to obtain the desired value of voltage ripple ($\Delta V_{OUT}$). Another source of ripple is the equivalent series resistance of

the capacitor, as reported in eq.3.6, a good choice is a capacitor with a low ESR[7]. In spite of the above considerations, the main that weights on the design choice of the output capacitor, is that given by dynamic considerations. During the worst load transient, i.e. from 0 to 4A, all the current to the load is not provided by the inductor, but by the output capacitor. In other words it looses charges and its voltage decreases, as well as the output voltage. The eq.3.7 expresses the minimum capacitance $C_{OUT(min),OS}$ that is needed to have a desired overshoot due to the transient, when a load regulation $\Delta I_{OUT}$ occurs.

$$D = \frac{V_{OUT}}{V_{IN(max)} \times \eta} \tag{3.1}$$

$$\Delta I_L = 0.2 \times I_{OUT(max)} \tag{3.2}$$

$$\Delta I_L = \frac{(V_{IN(max)} - V_{OUT}) \times D}{f_s \times L} \tag{3.3}$$

$$L_{min} = \frac{V_{OUT} \times (V_{IN} - V_{OUT})}{\Delta I_L \times f_s \times V_{IN}} \tag{3.4}$$

$$C_{OUT(min)} = \frac{\Delta I_L}{8 \times fs \times \Delta V_{OUT}} \tag{3.5}$$

$$\Delta V_{OUT(ESR)} = ESR \times \Delta I_L \tag{3.6}$$

$$C_{OUT(min),OS} = \frac{\Delta I_{OUT}^2 \times L}{2 \times V_{OUT} \times V_{OS}} \tag{3.7}$$

The table 3.2 reports all the found results for both $0.5\,V$ and $3.3\,V$.

| **BUCK COMPONENTS** | | |
|---|---|---|
| | **0.5 V** | **3.3 V** |
| **µ** | 47.9 % | 85.8 % |
| **D** (ideal) | 8 % | 55 % |
| **D** (real) | 17 % | 64 % |
| $\boldsymbol{\Delta I_L}$ | 0.8 A | 0.8 A |
| $\boldsymbol{L_{min}}$ | 238.7 nH | 773.4 nH |
| **L** | 510 nH | 1 µH |
| $\boldsymbol{C_{OUT(min)}}$ | 4.1 µF | 3.8 µF |
| $\boldsymbol{\Delta V_{OUT(ESR)}}$ | 7.8 mV | 7.2 mV |
| $\boldsymbol{C_{OUT(min),OS}}$ | 391.7 µF | 116.4 µF |
| **C** | 400 µF | 120 µF |

Table 3.2: Design of the Buck converter's components

## 3.2 Model-Based Analysis: Simulink Block Diagram

A first model-based analysis is carried out implementing the block diagram of the digitally controlled Buck converter already seen in the previous chapter. The model is structured in Simulink with the support of the Simscape environment, the latter used in particular for the plant representation.

### Simscape

Simscape enables to model and simulate multidomain physical systems, including electrical circuits, and provide more complex components and analysis capabilities than the standard Simulink blocks. In this case is fundamental to represent the plant behaviour through Simscape, indeed the aim of simulations is to appreciate changes on the output voltage due to electrical component response to the digital control. Using this kind of simulator improves the precision of the results. An example of a Simscape component is reported in 3.2. The parameters contained in the block's options permit to custom the component with high precision: one can choose MOS's $R_{DS\,on}$, $V_{th}$, $I_d$, parastic capacitances, body diode, temperature dependence and so on.

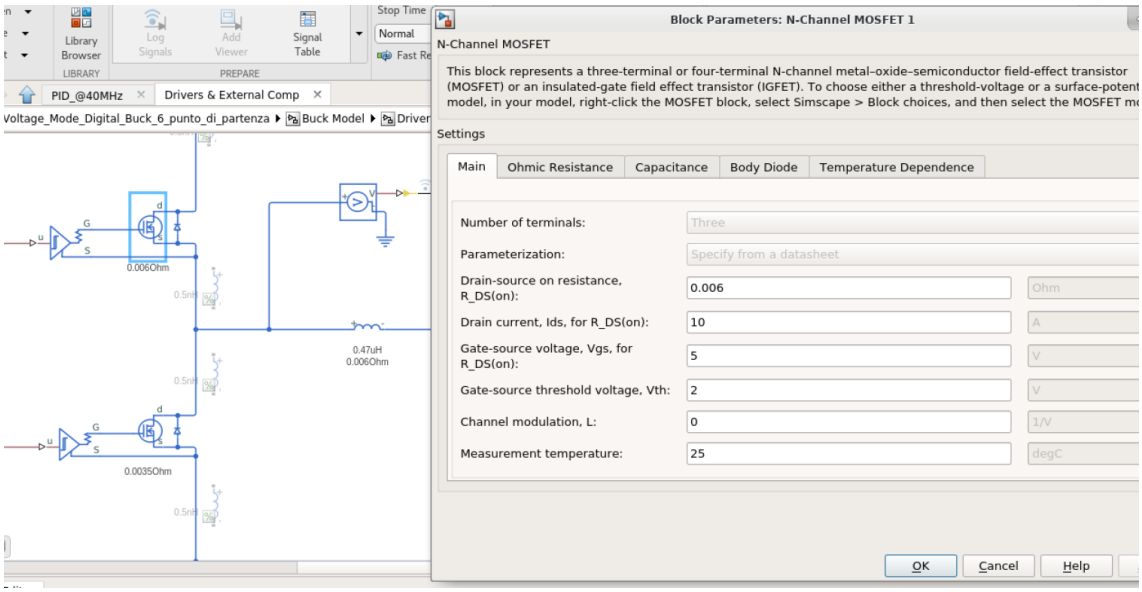



Figure 3.2: Simscape nMOS block

The model is structured in several blocks, each of them represents a fundamental element of the circuit. The scheme is shown in fig.3.3.

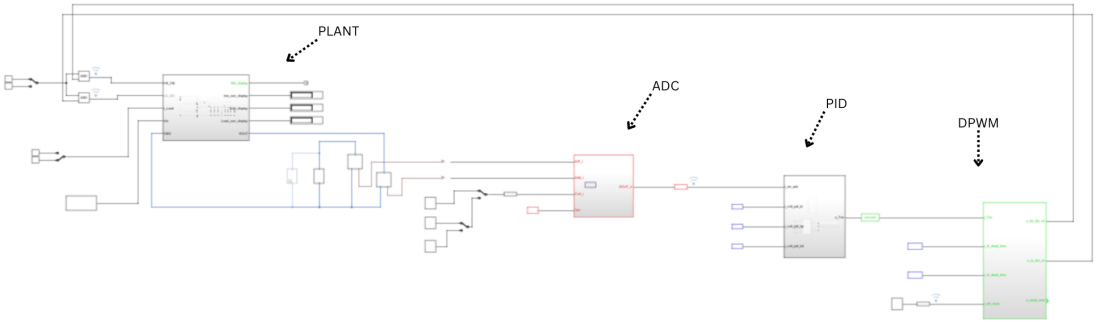



Figure 3.3: Basic Block Diagram of a Digitally Controlled Buck Converter

The main blocks can be distinguished, starting from the plant (fig.3.4), that contains all the components simulating their corresponding electrical behaviours through Simscape; the ADC block (fig.3.5) needed to convert the error voltage into digital value; the compensator block implemented through a digital Proportional-Integral-Derivative (PID) filter; the DPWM block used to modulate the duty cycle for the power MOSFETs. To better understand the circuit characteristics, all the blocks are explained in detail.

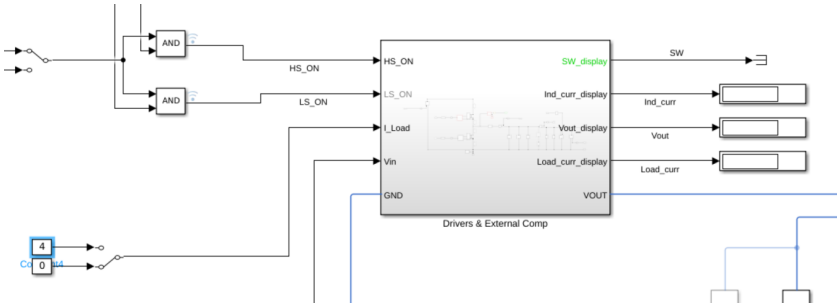



Figure 3.4: Plant block

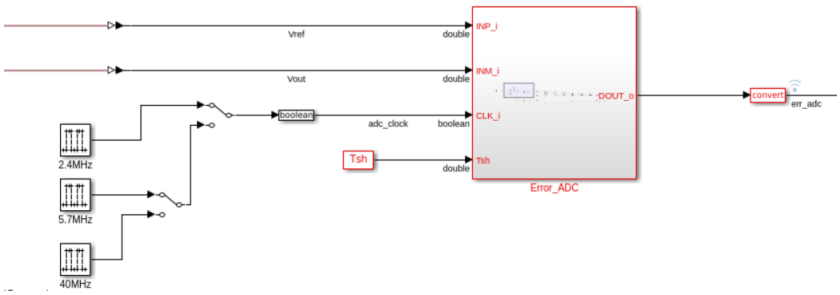



Figure 3.5: ADC Block

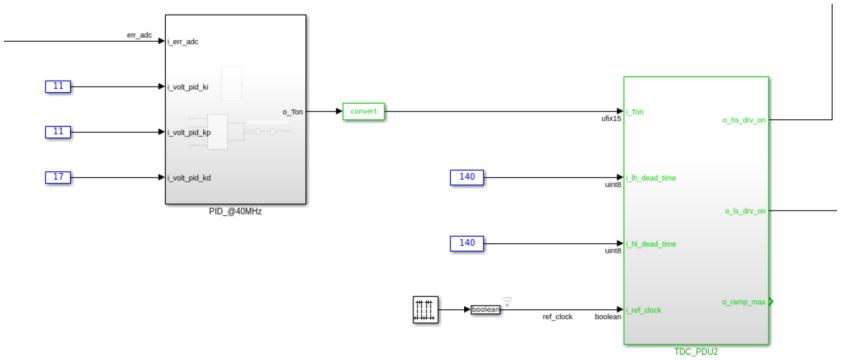



Figure 3.6: PID and PWM Block

### 3.2.1 Plant Block

In the model it is essential to properly represent the power circuitry that provides energy to the load. The plant, shown in fig.3.8, contains all these components, in particular power MOSFETs, gate drivers fig.3.9 and the inductance, the output capacitor and the load resistor that are indicated in fig.3.10 together with an adjustable current generator placed in parallel to the load to reproduce a variation on the current absorbed by the load. A voltage divider, reported in fig.3.7 is inserted at the output of the plant to adapt its dynamic range to that of the ADC. Probes measure voltages and currents in the plant circuit whenever is needed. They are recognisable by their electrical symbols placed inside a square, for example in fig.3.7 there are 2 voltage probes (with "V" inside the square), one measures the output voltage from the Plant and the other the voltage reference given by a voltage generator.

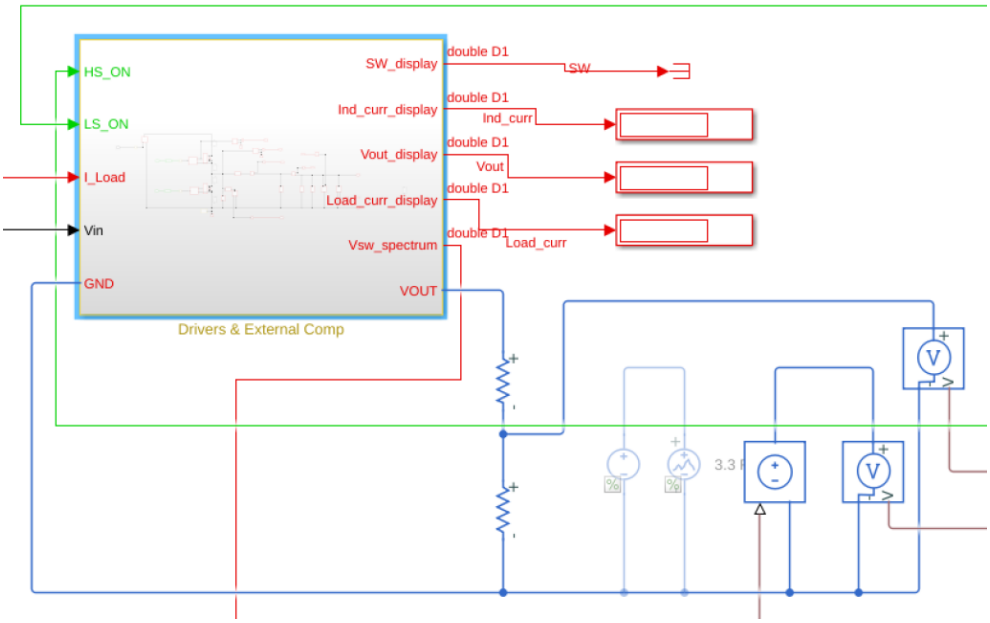



Figure 3.7: Voltage Divider

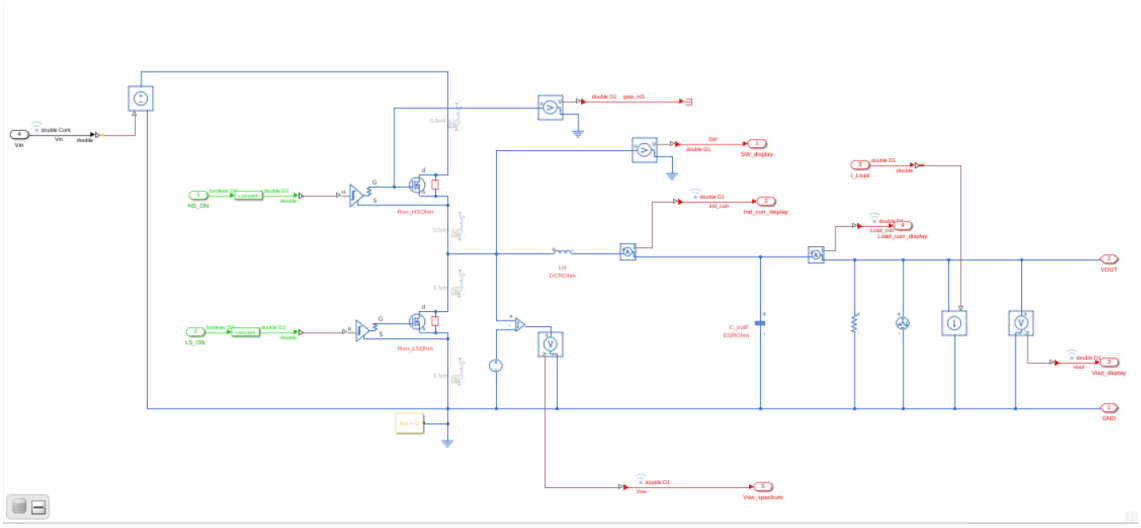


Figure 3.8: Model of the Plant Block

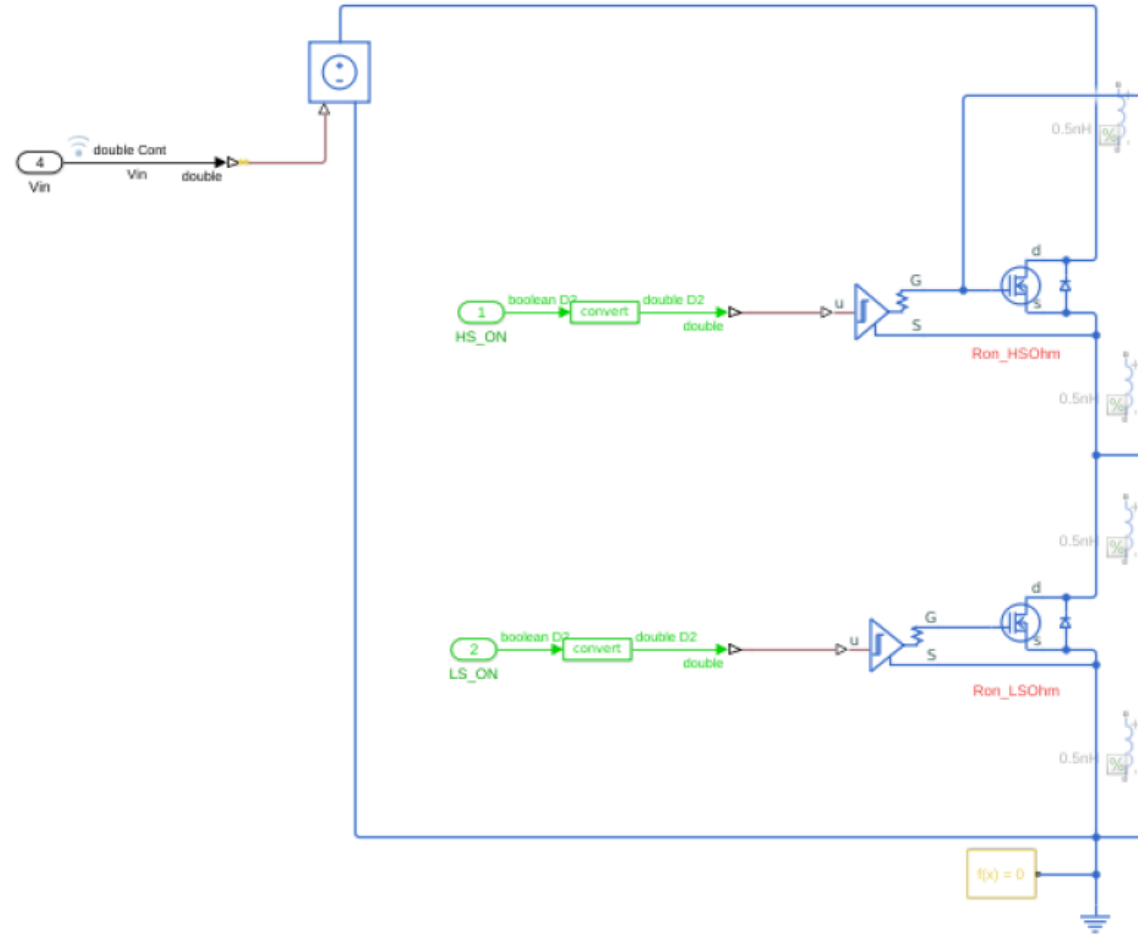



Figure 3.9: MOS and gate drivers model representation

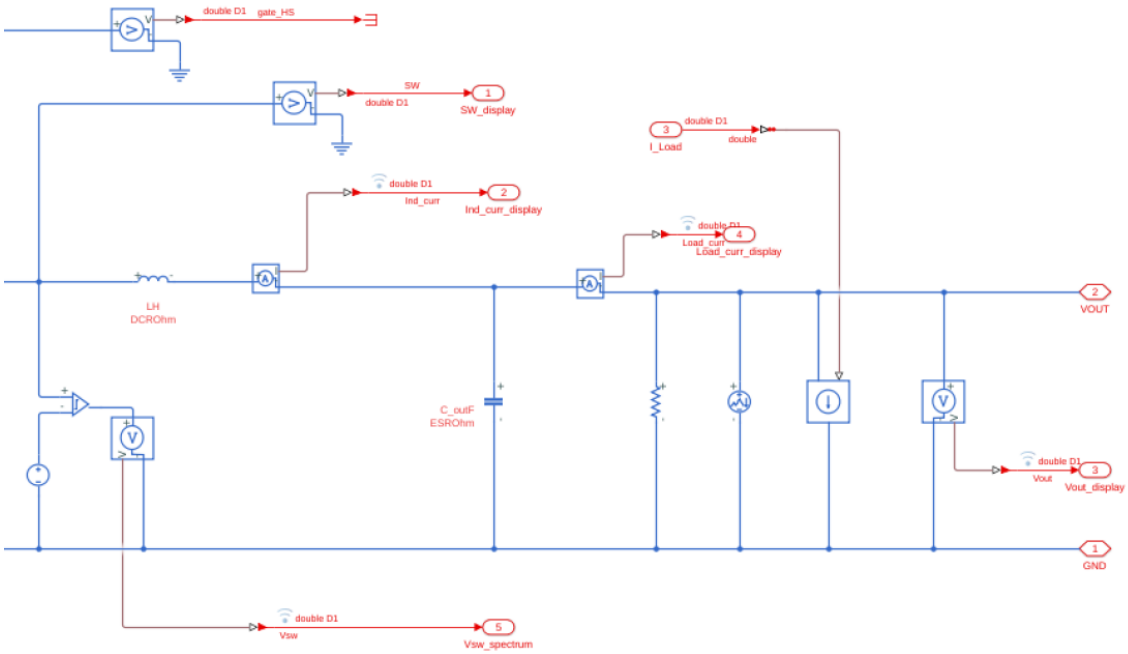



Figure 3.10: LC filter and load

### 3.2.2 ADC Block

First of all, a design choice has to be made on the structure of the control loop. In Chapter 2 the two possible solutions of ADC usage and their differences have been explained. Just to recap them, the first takes the output voltage and directly converts it into digital values; the second uses a subtractor, usually made with an operational amplifier (OPAMP), to compare the output voltage with the reference and then the ADC convert the error between them. Knowing that the second solution offers more accuracy on the static regulation, even if there is an higher cost of chip area, the structure of the loop is chosen to use a subtractor and an error ADC. So the second solution is preferred due to constraints in accuracy needs.

Now, before the digital control loop, it is fundamental to convert the error voltage, an analog quantity, into a digital one, to enable numerical manipulations performed by the compensator. The ADC design is a crucial step to obtain the desired output voltage resolution and not to face unwanted LCOs. At this point, the choice to make is the number of bit and the operating frequency (or sample per second) of the ADC. From the specifications is necessary to have a PWM working at 2.4 MHz, so the question is how many samples per PWM period are needed. For the very first try, the ADC is chosen to sample at a rate twice the PWM frequency, so it provides two samples every switching action of power MOSFETs. Technologically speaking is reasonable to have SAR (Successive-Approximation Register) ADC up to 40 MHz, so able to perform 40 millions of approximations per second.

It means that the sample rate depends on how many bits the ADC has, i.e. its resolution. This number, together with DPWM resolution, determines the static resolution of the regulated voltage. In practise when the ADC provides zero as output, the PID, after the

integral transient, maintains its actual value, which is taken by DPWM as input. So, the better the ADC "recognise" null voltages, the better the regulation is. Considering that the ADC has to convert the error voltages, that is supposed to be small compared to the regulated one, it is reasonable to use an ADC with a limited bipolar dynamic. Looking at the very first simulations and at the responses to load and line variations and ramp-up, a good value can be ±120 mV. This value has been chosen to guarantee the linearity of regulation, i.e. without clamping. Remembering the presence of the voltage divider, that has a gain of 20/55, it is possible to reduce the swing to ±43.6 mV. For the static resolution of the loop, as already said, the ADC plays a dominant role. For a 500 mV of regulated voltage it can be 0.5% or less (2.5 mV or less). The resolution of an ADC can be computed dividing the range by $2^{n_{bit}}$, i.e. the number of steps of its characteristic:

$$q_{ADC} = \frac{V_{FSR}}{2^{n_{bit}}} \tag{3.8}$$

$$= 7.5mV \quad \text{with } n_{bit} = 5 \tag{3.9}$$

$$= 3.75mV \quad \text{with } n_{bit} = 6 \tag{3.10}$$

$$= 1.875mV \quad \text{with } n_{bit} = 7 \tag{3.11}$$

$$= 0.938mV \quad \text{with } n_{bit} = 8 \tag{3.12}$$

Theoretically all solutions admitted are 7 and 8 bits, in that case $q_{ADC}$ is smaller than 2.5 mV. Choosing 7 as reference number of bits, means that, considering two samples per $T_{PWM}$, the approximations rate, i.e. the clock frequency, considering for the conversion 2 additional clock periods per sample, has to be at least $f_{clk} = (9 \times 2) \times 2.4\,\text{MHz} = 43.2\,\text{MHz}$. For simplicity the system clock at 40 MHz can be used to this purpose.

### 3.2.3 DPWM Block

This block is the true objective of this thesis. Generally the modulation is performed in digital at the same way of analog control loop, with a ramp. The difference is how it is created: the analog loop generates a continuous-time signal, whereas a digital loop works with discrete time, so the ramp is nothing but a free-running counter. The modulator compares the PID output with the count, when these values are equal turns off the high side MOS and, after a death time, turns on the low side. The clock frequency dictates the resolution of the modulator. The ramp is quantized by the clock period. The problem is clear in fig.3.11. In practise the PWM period is divided in $N_r = \frac{T_{PWM}}{T_{clk}}$ steps. The technology can provide up to $f_{clk} = 40\,\text{MHz}$, so in the most favourable case $N_r = \frac{417\,\text{ns}}{25\,\text{ns}} = 16.68$ that is 16 taking integer numbers.

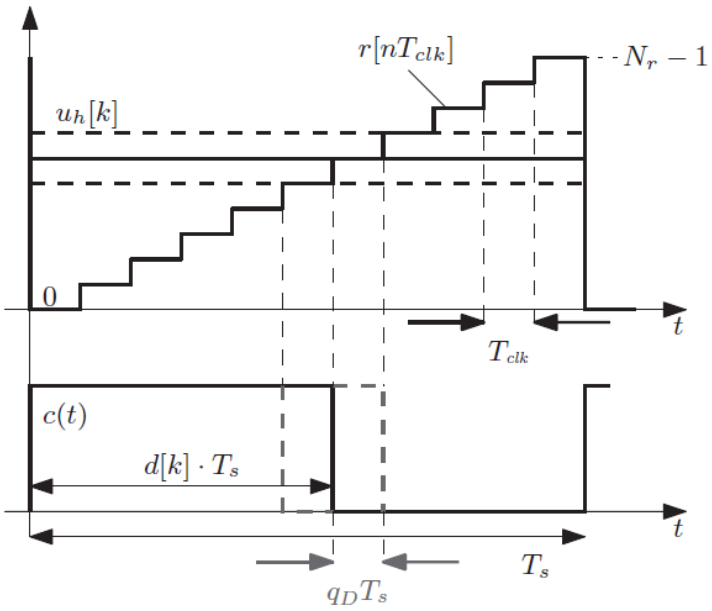



Figure 3.11: Digital ramp, quantization problem[3]

Now, the DPWM's resolution has to be set. Its value come from how precisely the modulator is capable to vary the duty cycle and so the average output voltage. In this design is not possible to increase too much the resolution, because the target is to implement the DPWM block without useless elements, it is fundamental not to use too much resources, i.e. power and area. In Chapter 2 all the theoretical details about the DPWM resolution have been proposed. A fundamental limitation comes from the considerations about the LCOs (2.2.2). The modulator resolution $q_{DPWM}$ has to be lower than the ADC resolution $q_{ADC}$, for this reason $100ps$ of DPWM time resolution is selected. Indeed, in term of voltage resolution it corresponds to $q_{DPWM}$, that can be obtained dividing the $Vin$ by the number of steps of the modulator that in turn is computed as the PWM period ($2.4MHz^{-1} \approx 417ns$) on the desired temporal resolution of the DPWM.

$$q_{DPWM} = \frac{V_{in}}{\#\text{number of step}} = \frac{6V}{\frac{417ns}{100ps}} = 1.439mV \tag{3.13}$$

It is obvious that is impossible to achieve such resolution just using the system clock, the ramp has to have $\approx 4167$ steps. In other words, the system clock would be of 10 GHz! The thesis aims to propose innovative techniques to improve the regulated voltage resolution without increasing the clock frequency. The DPWM block is designed to be compose by three part:

- Digital Ramp
- Delay Line
- Dithering

The Simulink block contains just an high-level representation of the DPWM: a ramp with a 100 ps temporal resolution is implemented to obtain an equivalent behaviour of the three techniques combined together. All the details of the delay line and dithering techniques are explained in the following chapters.

### 3.2.4 Compensator Block (PID)

The compensator block is very standard. The structure is composed by a parallel form of the PID's three gains:

- Proportional: the (digital) input sample is simply multiplied by $K_p$. It reacts on the present error and gives a response proportional to that;
- Integral: the sample is delayed by a sampling period, then multiplied by $K_i$ and finally the result is fed back to accumulate the all the previous samples.
- Derivative: the present sample is compared to the previous one, the difference between them is multiplied by a factor $K_d$. In practice it reacts to the variations of the error, the higher is the variation the bigger is the response. It is used to promptly react to a sudden event.

The compensator design is reduced to the choice of the three gain just mentioned. Obviously these values come from an accurate analysis to be done on frequency domain and that is fully described in the next sections. Nevertheless, a couple of considerations are presented here. One of the ADC design choices has been the number of samples per PWM period and that resulted to be 2. For this reason a fix has to be made on the structure of the PID. The switching action of the plant results on the output voltage as ripple that is sampled by the error ADC. The most urgent problem is now to filter out this ripple from the output of the PID, it can create oscillations or instability at the voltage output. The biggest contribution comes from the derivative term that usually has the gratest gain, an LSB of the error ADC output is enough to cause an higher compensator output variation. There are two ways to remove the most of it:

- Filters: a pair of digital integrator can be adopted. They are used as pre and post-filters to eliminate the ripple from the PID output. Obviously the cut-off frequency has to be coherent with the sampling frequency of the ADC.
- Delay on derivative branch: it can be sufficient to add a delay element in series with the original one just on the derivative branch. The reason comes from the double nature of the sampling action: the PID processes two samples per PWM period and the derivative branch senses the difference between two consecutive samples, so adding a delay means transform the comparing action from two consecutive sample to two alternating sample. The derivative action now compares the same points on every PWM period, eliminating the derivative contribution in steady state, it will emerge just in dynamic variations.

The first solution is not recommended in digital loop, because it slows down the dynamic response, due to the added double integral actions. The second can be used not to impact on the performance of the loop, the drawback is that the proportional gain is still running so it can be expected a small additional ripple, but it is expected to be much lower than that caused by the derivative. The scheme is attached in fig.3.12.

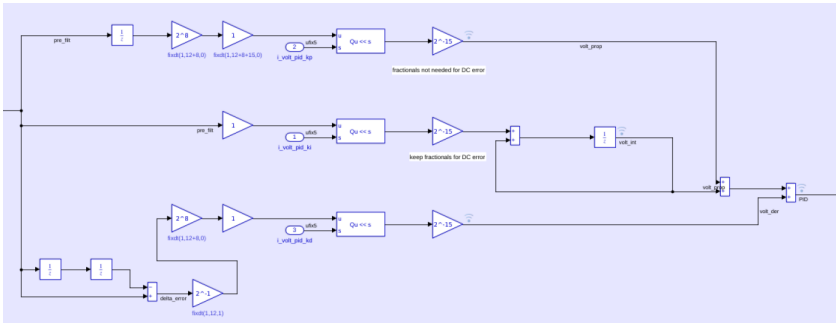


Figure 3.12: PID Simulink block diagram

## 3.3 Small-Signal Modelling

This step is fundamental to correctly design the compensator and its parameters in order to obtain particular static and dynamic performance, as the specifications expect. All the analysis is carried on in MATLAB environment both graphically and mathematically. The script implemented for the thesis project is reported in the Appendix A A. It includes all the code used for the small-signal analysis and the compensator design performed with both the continuous-time and the discrete-time approaches.

A first distinction has to be made during the analysis, in fact the system is thought to be versatile and reusable for multiple output voltage. The analysis is carried out for both the edges of the specification, i.e. 0.5V and 3.3V. Furthermore, as already explained in the previous chapter two possible ways can be taken into account to obtain a discrete time small-signal modelling. This section reports both of them and compares the results and differences.

### 3.3.1 Continuous-time modelling 2.2.1

The first approach aims at approximating the continuous time model to the discrete one. The small-signal model of the control loop is obtained starting from the plant, as already explained in the chapter, its transfer function in continuous time can be written as:

$$G_{vd}(s) = \frac{\hat{\tilde{v}}_o(s)}{\hat{d}(s)}\Big|_{\hat{\tilde{v}}_g=0,\hat{\tilde{i}}_o=0} = V_g \frac{1+sr_CC}{1+s(r_C+r_L)C+s^2LC} = G_{vd0}\frac{1+\frac{s}{\omega_{ESR}}}{1+\frac{s}{Q\omega_0}+\frac{s^2}{\omega_0^2}} \tag{3.14}$$

From the transfer function is possible to note that for the 0.5 V it has:

$$G_{vd0} = V_g = 6\,\text{V} \tag{3.15}$$

$$\omega_{ESR} = \frac{1}{r_C C} = (39.8\,\text{kHz}) \times 2\pi \tag{3.16}$$

$$\omega_0 = \frac{1}{\sqrt{LC}} = (11.1\,\text{kHz}) \times 2\pi \tag{3.17}$$

$$Q = \frac{1}{r_C + r_L}\sqrt{\frac{L}{C}} = 1.8 \tag{3.18}$$

and for 3.3 V:

$$G_{vd0} = V_g = 6\,\text{V} \tag{3.19}$$

$$\omega_{ESR} = \frac{1}{r_C C} = (133\,\text{kHz}) \times 2\pi \tag{3.20}$$

$$\omega_0 = \frac{1}{\sqrt{LC}} = (14.5\,\text{kHz}) \times 2\pi \tag{3.21}$$

$$Q = \frac{1}{r_C + r_L}\sqrt{\frac{L}{C}} = 4.56 \tag{3.22}$$

The Bode plots of the transfer function are:

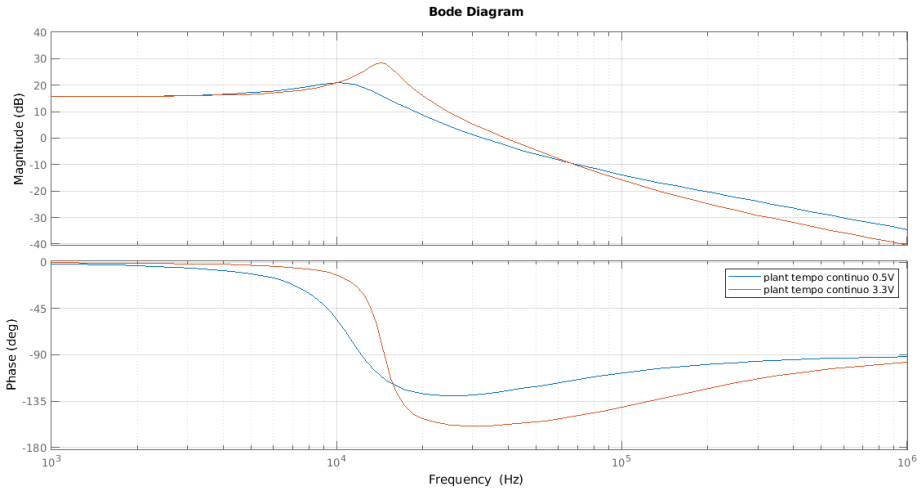



Figure 3.13: Plant's continuous time transfer function for 3.3V and 0.5V of regulated voltage

The characteristics are different from each other, for this reason the two compensators have to to behave differently to obtain specific performances, PID's gains, indeed, will differ a little. Following the first approach, it is necessary to add a phase delay in the transfer function of the plant to take into account the effect of time sampling and amplitude quantization of the digital control loop. Indeed, as already mentioned, the ADC introduces delay in the loop response and the DPWM delay in the modulation of the duty cycle. For the two edge cases, 3.3V and 0.5V, the phase delays are slightly different. The reason is the expected output duty cycle that is 8% for the lowest and 55% for the other, the modulation instant varies temporally between them. The delay is added manually in the

MATLAB script, multiplying the transfer function for a complex quantity, as reported in the following equation, in which $t_d$ is 600 ns for 3.3V and 450 ns for 0.5V:

$$H_{del}(s) = H(s)e^{-st_d} \tag{3.23}$$

It is interesting to see how the delay included in the model impacts on the frequency behaviour of the system.

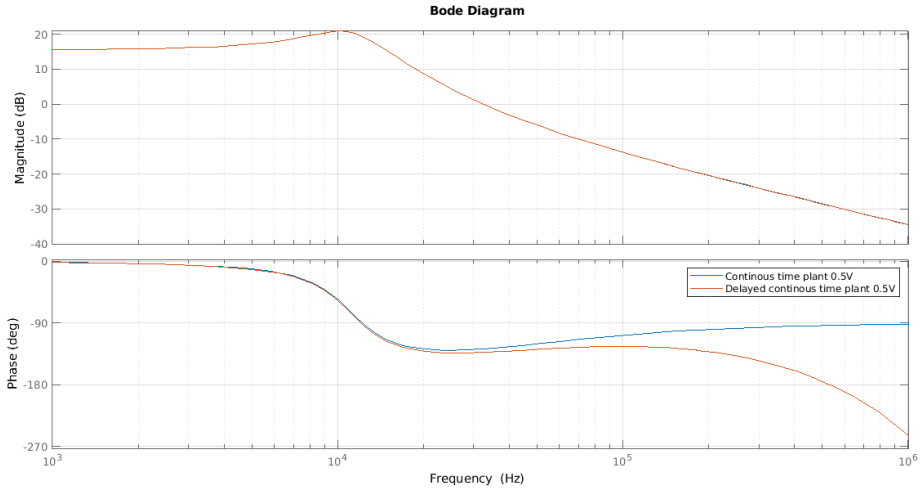



Figure 3.14: Effect of delay on continuous transfer function for 0.5V

In fig. 3.14 the delayed plant model for 0.5 V (but is the same for 3.3 V) shows a degradation of the phase shift for higher frequencies, on the other hand magnitude is not affected at all. This effect must be considered designing the PID, indeed the phase margin is strongly affected by delays, more stringent parameters are needed. After that, the gain of the voltage divider, the gain of the ADC and the gain of the PWM block have to be taken into account to calculate the uncompensated loop gain. The first one is easy to obtain just looking at the resistance values, the second is the reciprocal of the resolution of the ADC, i.e. the full scale range divided by the number of bins, the third is the reciprocal of the number of steps of the ramp implemented into the PWM block. Mathematically:

$$G_r = \frac{R1}{R1 + R2} = \frac{20}{55} \tag{3.24}$$

$$q_{ADC} = \frac{V_{FSR}}{2^{n_{bit}}} = \frac{0.086\,\mathrm{V}}{2^7} = 672\,\mu\mathrm{V} \tag{3.25}$$

$$\#number\ of\ steps = \frac{T_{PWM}}{q_{DPWM}} = \frac{416.7\,\mathrm{ns}}{100\,\mathrm{ps}} = 4167 \tag{3.26}$$

The final transfer function is:

$$H_u(s) = H_{del}(s)\,\frac{20}{55}\,672\,\mu\mathrm{V}^{-1}\,4167 \tag{3.27}$$

In fig.3.15 a plot of the delayed uncompensated loop is reported.

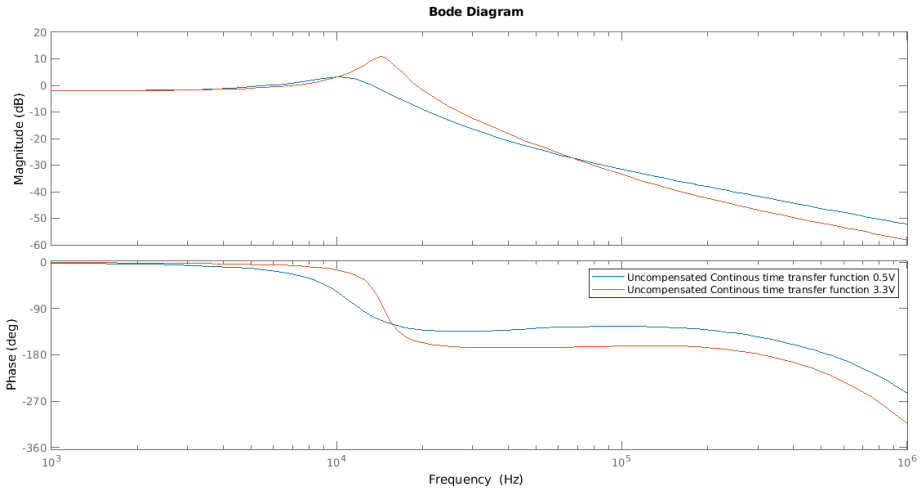



Figure 3.15: Uncompensated Continuous Time loop transfer function for 0.5 V and 3.3 V

At this point is possible to design the PID following the continuous-time common rules for buck converters compensation, already proposed in the second chapter. Indeed the first approach aims to obtain a continuous time representation of the compensator just to transform it to a discrete time one, using approximation techniques. The most used, as already seen in the previous chapter, are the *Backward Euler* and the *Tustin Approach*. Let's take as design example the case of 0.5 V of regulated voltage. As design goal, the crossover frequency is set to $f_c = 240\,\text{kHz}$, that is, 1/10 of the converter switching frequency, and the phase margin target is set at $\phi_m$=80°. Looking at the uncompensated loop transfer function, the phase is -134°, so it needs a phase boost of about $\theta$ =34°.

$$G_{PD}(s) = G_{PD0}\frac{1+\frac{s}{\omega_z}}{1+\frac{s}{\omega_p}} \tag{3.28}$$

The phase boost has to be $\theta$ =34° and $\omega_c = \omega_{max} = \sqrt{\omega_p\omega_z}$. Following the algorithm presented in Chapter 2[3], it is possible to extract the frequency of the zero and the pole:

$$\omega_z = \omega_c\sqrt{\frac{1-\sin\theta}{1+\sin\theta}} \approx 2\pi\cdot(128\,\text{kHz}) \tag{3.29}$$

$$\omega_p = \omega_c\sqrt{\frac{1+\sin\theta}{1-\sin\theta}} \approx 2\pi\cdot(451\,\text{kHz}) \tag{3.30}$$

Imposing,

$$|T(j\omega_c)| = |T_u(j\omega_c)|G_{PD0}\sqrt{\frac{1+(\frac{\omega_c}{\omega_z})^2}{1+(\frac{\omega_c}{\omega_p})^2}} = 1 \tag{3.31}$$

expliciting $G_{PD0}$ and looking at the uncompensated gain $T_u(j\omega_c) = 0.0178$, the value is found:

$$G_{PD0} = \frac{1}{|T_u(j\omega_c)|}\sqrt{\frac{1+(\frac{\omega_c}{\omega_p})^2}{1+(\frac{\omega_c}{\omega_z})^2}} \approx 30 \implies 29.5\,\text{dB} \tag{3.32}$$

The final step is adding an integral action to make zero the steady-state error:

$$G_{PI}(s) = G_{PI\,\text{inf}}(1+\frac{\omega_l}{s}) \tag{3.33}$$

The high frequency gain $G_{PI\infty}$ is set to one to not alter the *lead* term gain or its phase margin. Same for the $\omega_l$ that is chosen to be much smaller than the $\omega_c$. In this example:

$$\omega_l = \frac{\omega_c}{20} = 2\pi \cdot (12\,\text{kHz}) \tag{3.34}$$

The complete compensator transfer function is therefore:

$$G_{PID}(s) = \underbrace{(1+\frac{\omega_l}{s})}_{PI} \cdot \underbrace{G_{PD0}\frac{1+\frac{s}{\omega_z}}{1+\frac{s}{\omega_p}}}_{PD} \tag{3.35}$$

From the equation (3.35) is possible to recognise the *Proportional-Integral* and *Proportional-Derivative* terms in which the proportional part of the integral is set to one, so not reported in the expression. The transfer function can be graphically appreciated in 3.16.

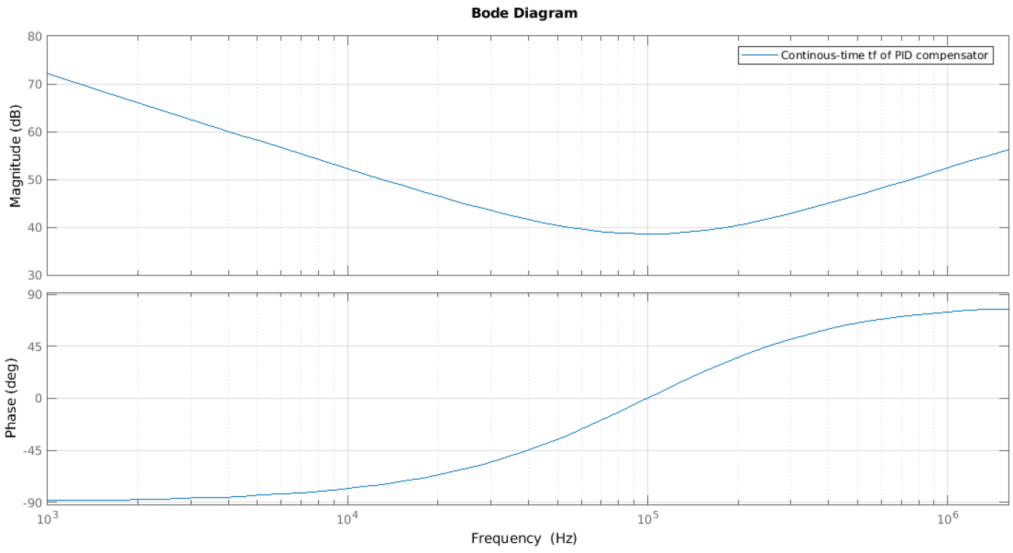



Figure 3.16: Continuous-time PID transfer function

At this point it is possible to obtain a more compact form for the PID expression, underlining the characteristic parameters $K_p$, $K_i$ and $K_d$ that are used in block diagrams like the one reported in 3.17, a parallel form of a PID compensator.

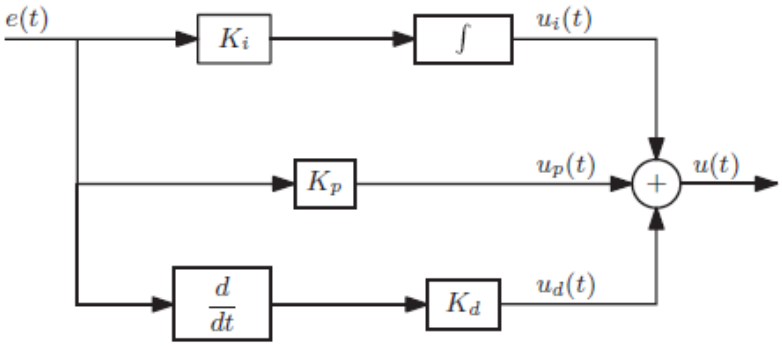



Figure 3.17: Parallel form PID block diagram[3]

In Laplace domain is:

$$G_{PID}(s) = \frac{\hat{u}(s)}{\hat{e}(s)} = K_p \quad + \quad \frac{K_i}{s} \quad + \quad sK_d \tag{3.36}$$

In the example the three terms are extrapolated using:

$$K_p = G_{PD0}(1 + \frac{\omega_l}{\omega_z} - \frac{2\omega_l}{\omega_p}) = 31.1$$
$$K_i = 2\,G_{PD0}\frac{\omega_l}{\omega_p} = 1.56\,\frac{\text{rad}}{\text{s}}$$
$$K_d = \frac{G_{PD0}}{2}(1 - \frac{\omega_l}{\omega_p})(\frac{\omega_l}{\omega_z} - 1) = 36.7\,(\frac{\text{rad}}{\text{s}})^{-1}$$

The next step aims to approximate the Continuous-time PID transfer function into a discrete one. The resulting block diagram is shown in 3.18.

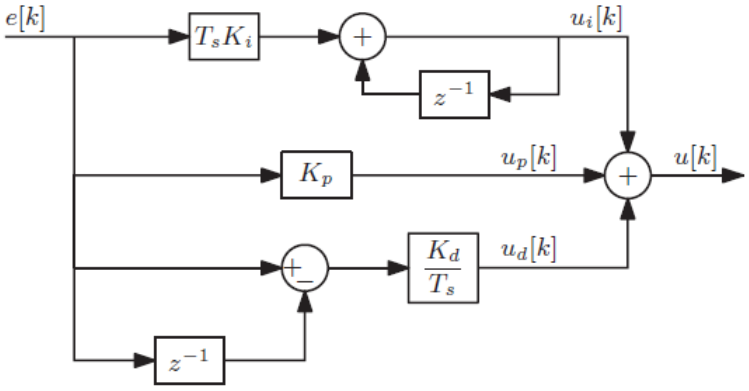



Figure 3.18: Discrete-time parallel form PID block diagram[3]

The simplest technique is the *Backward Euler* method, or *backward rectangular rule*, which is based on a rectangular approximation of the continuous-time integral operator:

$$\int_{(k-1)T_s}^{kT_s} x(\tau)\,d\tau \approx T_s x(kT_s) \tag{3.37}$$

In Z-transform domain, this discretization can be seen as a mapping from Laplace domain, defined by:

$$s \implies \frac{1 - z^{-1}}{T_s} \tag{3.38}$$

The PID transfer function becomes:

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad \frac{T_s K_i}{1 - z^{-1}} \quad + \quad \frac{K_d}{T_s}(1 - z^{-1}) \tag{3.39}$$

To appreciate the difference and analyse the correctness of the approximation both continuous-time and discrete one are reported in fig.3.19.

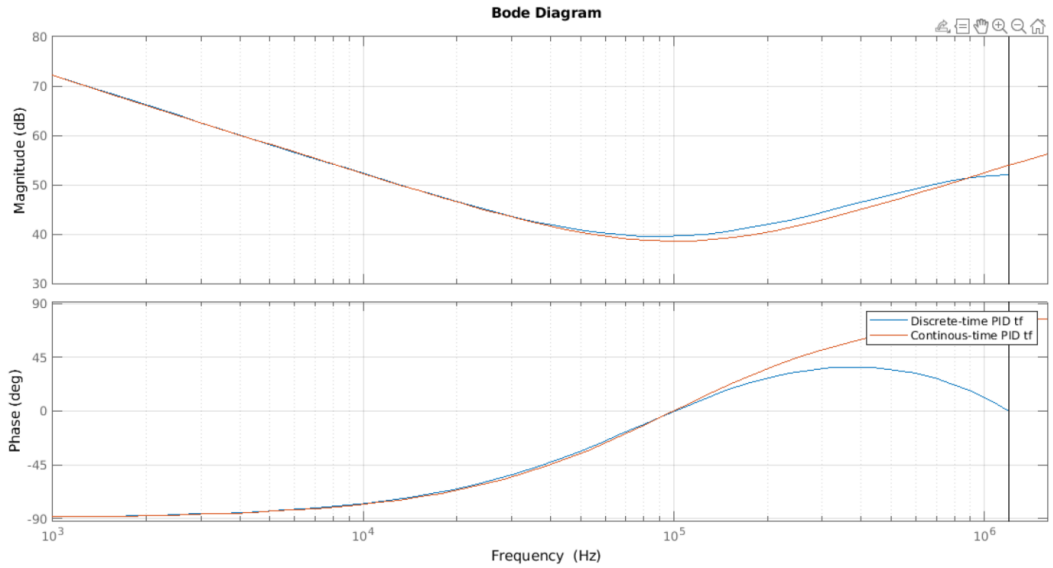



Figure 3.19: Continuous and discrete time PID transfer functions

From the graph is clear that for low frequencies the approximation is in very good agreement with the exact discrete-time analysis, but at higher frequencies both magnitude and phase differ from the continuous case. Even if the error is not so huge for the magnitude, for the phase the approximation fails and it leads to another degradation of the loop phase margin. The system is less robust in term of stability.

The other method is the *Tustin approach* or *trapezoidal rule*. The idea is to represent the integral over the sampling step with a trapezoidal approximation of the integrand:

$$\int_{(k-1)T_s}^{kT_s} x(\tau)\, d\tau \approx \frac{T_s}{2}(x(kT_s)) + x((k-1)T_s) \tag{3.40}$$

In Z-transform domain, this discretization can be seen as a mapping from Laplace domain, defined by:

$$s \implies \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{3.41}$$

The PID transfer function become:

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad K_i \frac{T_s}{2} \frac{1+z^{-1}}{1-z^{-1}} \quad + \quad K_d \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}} \tag{3.42}$$

In fig.3.20 an overall comparison between continuous-time, discrete-time Euler and discrete-time Tustin approaches is reported.

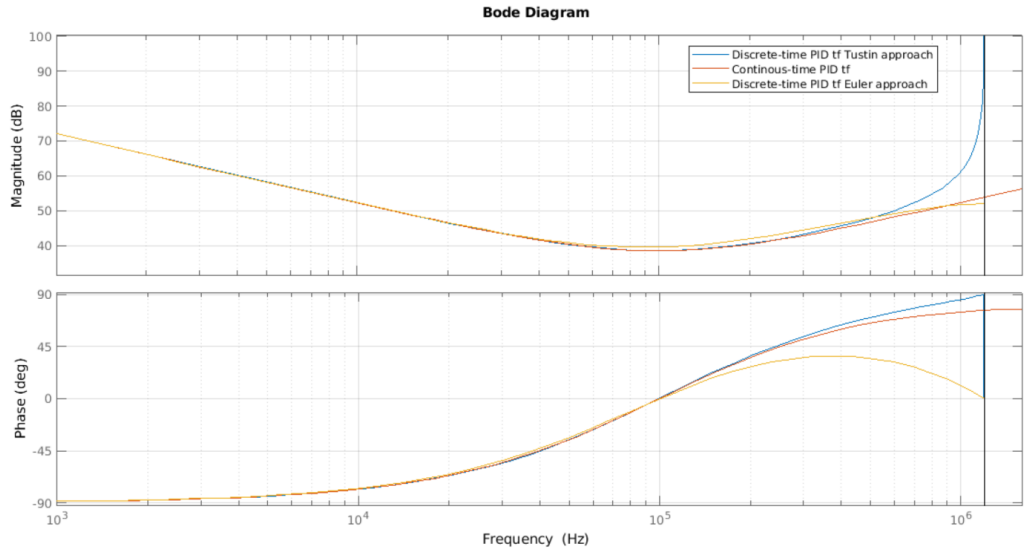



Figure 3.20: Continuous and discrete time PID transfer functions with Euler and Tustin approaches

This time just the approximation is better for the phase, which is not so different, but the magnitude is strongly altered, in particular near the Nyquist frequency. At this point seems to be interesting to see how this approximation impacts on the closed loop transfer function. The fig.3.21 shows how the closed loop is affected by discrete time approximations. Both of cases shows differences on both phase and magnitude that come out also in time domain, looking at the regolated voltage dynamic performances.

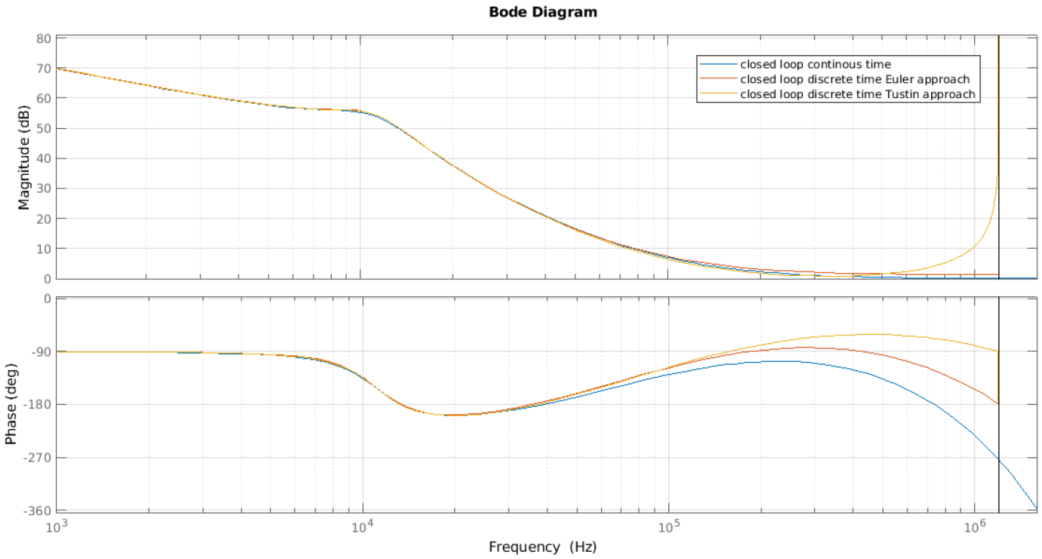



Figure 3.21: Closed loop transfer function using continuous and discrete approaches (Euler and Tustin)

### 3.3.2 Discrete-Time Modelling

In this section a discrete-time model of the plant is extrapolated using the state-space approach, already introduced in Chapter 2. The results are shown below, neglecting the intermediate steps of the approach.

A trailing edge modulation is assumed for the DPWM, from the specifications the parameters of the converters are already known and also the total loop delay $t_d$ has been computed:

$$V_g = 6\,\mathrm{V} \tag{3.43}$$

$$I_o = 4\,\mathrm{A} \tag{3.44}$$

$$D = \frac{V_o}{V_g} = \frac{0.5\,\mathrm{V}}{6\,\mathrm{V}} = 0.1 \tag{3.45}$$

$$t_d = t_{cntrl} + t_{ADC} + t_{DPWM} = 500\,\mathrm{ns} \tag{3.46}$$

At this point, from the theory, the small-signal control-to-output transfer function matrix $\boldsymbol{W}(z)$ in z-domain is:

$$\boldsymbol{W}(z) = \frac{\hat{\boldsymbol{y}}(z)}{\hat{u}(z)} = \boldsymbol{\delta}(z\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{\gamma} \tag{3.47}$$

For the purpose of this example is sufficient to extract the voltage control-to-output matrix $G_{vu}(z)$:

$$\boldsymbol{W}(z) = \begin{bmatrix} G_{iu}(z) = \frac{\hat{i_L}(z)}{\hat{u}(z)} \\ G_{vu}(z) = \frac{\hat{v_o}(z)}{\hat{u}(z)} \end{bmatrix} \tag{3.48}$$

The resulting transfer function is shown in fig.3.22 superimposed to the continuous case. The two function are very similar but the phase is much worse in the continuous case, in which the modulation delay imposed manually (outside the model) is overestimated compared to the discrete case. For this reason a digital compensator designed on the basis of an approximated plant transfer function requires a more stringent stability considerations. After this analysis is clear how choosing a discrete-time modelling is advantageous to correctly and efficiently design a digital control. Indeed, from now on, the proposed thesis project is carried on using discrete-time modelling.

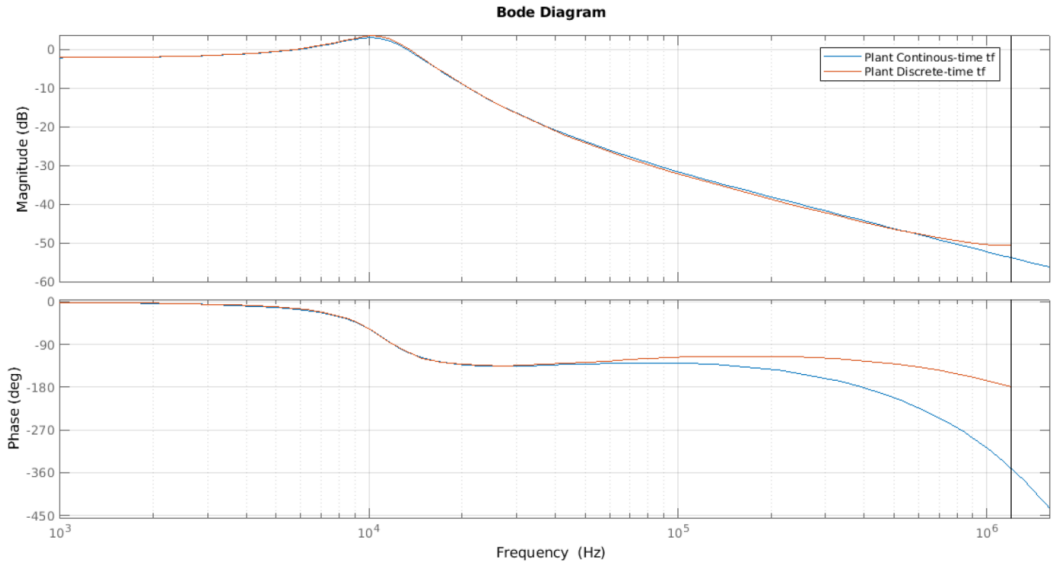



Figure 3.22: Plant discrete-time transfer function)

### 3.3.3 PID Design

The aim is to obtain the PID z-domain transfer function $G_c(z)$ that respect the specifications. Just to recall what already explained in the previous chapter, the proposed method is based on the *bilinear transform* and belongs to a *mapping-based* approach class. Firstly the uncompensated loop gain $T_u(z)$ is mapped into an equivalent continuous-time *p-domain*, where is possible to design the compensator with the common analog rules, then the synthesized transfer function is back-mapped into the z-domain and implemented in the digital controller. The mapping can be represented as follow:

$$z(p) = \frac{1 + p\frac{T_s}{2}}{1 - p\frac{T_s}{2}} \tag{3.49}$$

To be precise in the p-domain frequencies present a sort of distorsion compared with normal frequencies (s-domain), generally called *frequency wrapping*. The relation between the two domains is:

$$\omega' = \frac{2}{T_s}\tan(\omega\frac{T_s}{2}) \tag{3.50}$$

Firstly, the PID expression must be mapped in p-domain. So, $G_{PID}$

$$G_{PID}(z) = \frac{\hat{u}(z)}{\hat{e}(z)} = \quad K_p \quad + \quad \frac{K_i}{1 - z^{-1}} \quad + \quad K_d(1 - z^{-1}) \tag{3.51}$$

becomes:

$$G'_{PID}(p) = \quad K_p \quad + \quad \frac{K_i}{T_s} \frac{1 + \frac{p}{\omega_p}}{p} \quad + \quad K_d T_s \frac{p}{1 + \frac{p}{\omega_p}} \tag{3.52}$$

To design the PID transfer function in the p-domain the steps are exactly the same for the analog case. So firstly is necessary to chose a crossover frequency. In the example is $\omega_c = 240KHz$, that in order not to have frequency distortion, has to be *prewrapped*:

$$\omega'_c = \frac{2}{T_s} \tan(\omega_c \frac{T_s}{2}) \approx 2\pi \cdot (248.22\,\text{kHz}) \tag{3.53}$$

Then, $\omega_p$ has to be computed:

$$\omega_p = \frac{2}{T_s} \approx 2\pi \cdot (763.94\,\text{kHz}) \tag{3.54}$$

Now, the phase and gain values at $\omega_c$ of $T_u(z)$ are required:

$$\left|T_u(e^{j\omega_c T_s})\right| = \left|T'_u(e^{j\omega'_c T_s})\right| \approx -40\,dB$$
$$\angle T_u(e^{j\omega_c T_s}) = \angle T'_u(e^{j\omega'_c T_s}) \approx -118°$$

Following the analog rules:

$$\omega_{PD} = \omega'_c \sqrt{\frac{1 - \sin\theta}{1 + \sin\theta}} = 180\,\text{kHz} \tag{3.55}$$

$$G'_{PD0} = \frac{1}{|T'_u(j\omega'_c|} \sqrt{\frac{1 + (\frac{\omega'_c}{\omega_p})^2}{1 + (\frac{\omega'_c}{\omega_{PD}})^2}} = 62 \tag{3.56}$$

$$\omega_{PI} = 2\pi \cdot (\omega_c/20) = 2\pi \times (12\,\text{kHz}) \tag{3.57}$$

$$G'_{PI\infty} = 1 \tag{3.58}$$

Now the PID is completely defined in the p-domain:

$$G'_{PD0} = 62$$
$$G'_{PI\,\text{inf}} = 1$$
$$\omega_{PI} = 2\pi \times (12\,\text{kHz})$$
$$\omega_{PD} = 2\pi \times (180\,\text{kHz})$$

$$K_p = 64 \quad K_i = 2 \quad K_d = 99$$

In fig.3.23 it is possible to appreciate the result graphically and in fig.3.24 the loop gain function.

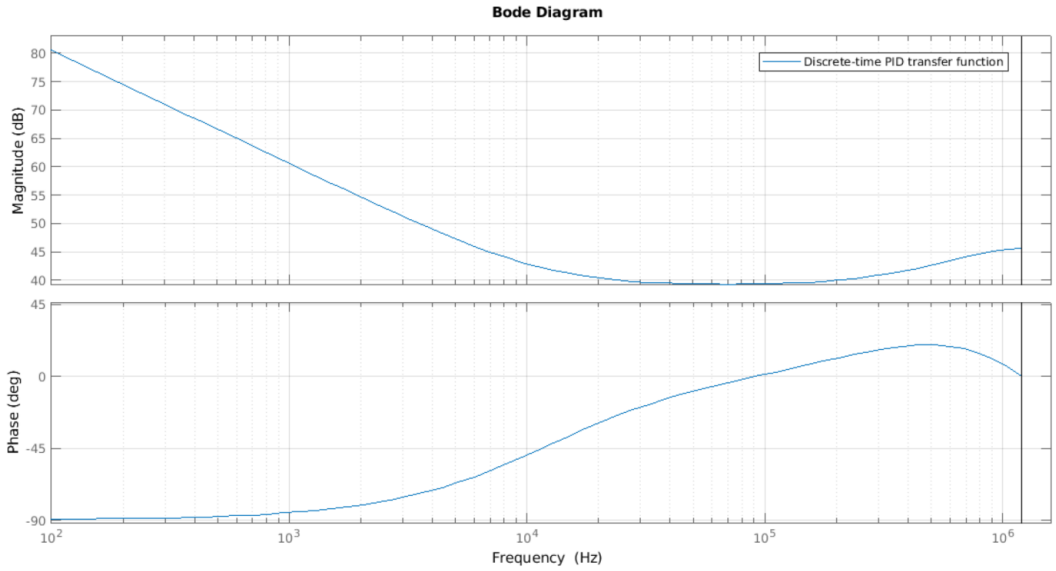



Figure 3.23: Discrete time p-mapped PID

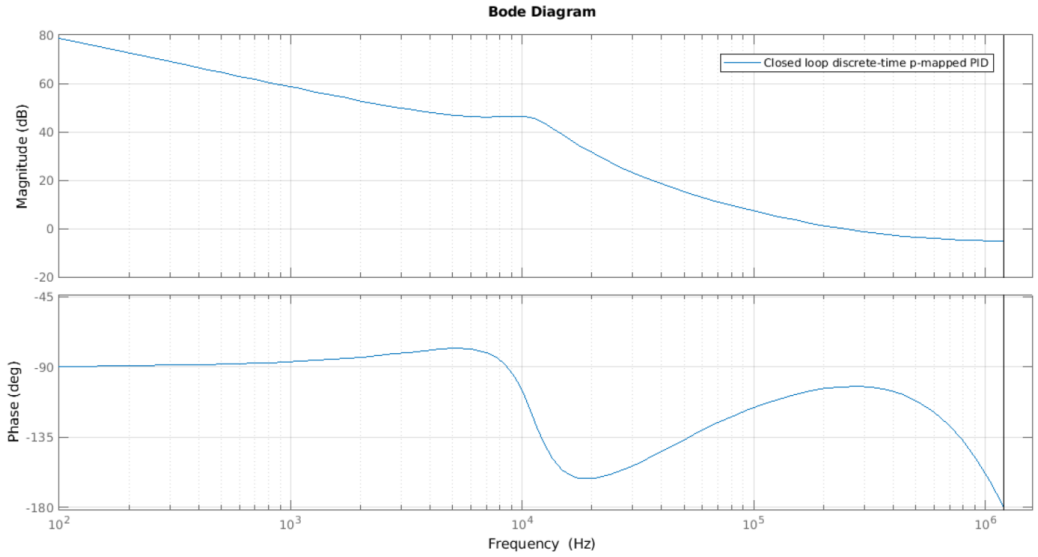



Figure 3.24: Loop gain transfer function with p-mapped PID)

The same steps can be repeated for the 3.3V of regulated voltage:

Following the analog rules:

$$\omega_{PD} = \omega_c' \sqrt{\frac{1-\sin\theta}{1+\sin\theta}} = 110\,\text{kHz} \tag{3.59}$$

$$G'_{PD0} = \frac{1}{|T'_u(j\omega'_c|} \sqrt{\frac{1+(\frac{\omega'_c}{\omega_p})^2}{1+(\frac{\omega'_c}{\omega_{PD}})^2}} = 85 \tag{3.60}$$

$$\omega_{PI} = 2\pi \cdot (\omega_c/20) = 2\pi \times (12\,\text{kHz}) \tag{3.61}$$

$$G'_{PI\infty} = 1 \tag{3.62}$$

Now the PID is completely defined in the p-domain:

$$G'_{PD0} = 85$$
$$G'_{PI\,\mathrm{inf}} = 1$$
$$\omega_{PI} = 2\pi \times (12\,\mathrm{kHz})$$
$$\omega_{PD} = 2\pi \times (180\,\mathrm{kHz})$$

$$K_p = 91 \quad K_i = 2.7 \quad K_d = 249$$

These values will be used as reference for the PID gains in the following Chapters. From the theory they represent the most accurate compensator design among the presented approaches.

# Chapter 4

# Simulink simulations

Once the PID design is concluded is possible to study the loop performance. In order to study the stability and dynamic response of the controller, a set of inputs are needed. All the input changes are organised as follows:

- **SOFT START**: $0\,V \implies V_O$ at $0\,s \implies \frac{V_o}{33\frac{mV}{\mu s}}$
- **LINE VOLTAGE**: $6\,V \implies 7\,V$ at $150\mu s \implies 151\mu s$
- **LOAD CURRENT**: $0\,A \implies 4\,A$ at $250\mu s \implies 254\mu s$
- **LINE VOLTAGE**: $7\,V \implies 5\,V$ at $350\mu s \implies 351\mu s$
- **LOAD CURRENT**: $4\,A \implies 0\,A$ at $500\mu s \implies 504\mu s$
- **LINE VOLTAGE**: $5\,V \implies 6\,V$ at $600\mu s \implies 601\mu s$

in which the "SOFT START" refers to an initial condition during which the reference voltage is given to the system slowly to avoid inrush current at the start and damaging to the components. In this condition the output voltage follows the reference, so it increases slowly with a slope indicated in the table till the normal operating point, i.e. $V_O$. The others inputs are simply changes on the line voltage, $V_{in}$, or in the current absorbed by the load, $I_O$. Summarising the input changes, the loop has to respond to line voltage steps of at most $2\frac{V}{\mu s}$ and load current steps of $1\frac{A}{\mu s}$.

Regarding other design choices, it is important to underline that the ADC provides two samples every switching period, the DPWM block is modelled at high level for the very first simulations, maintaining the resolution selected in the previous chapter, i.e. 100 ps, so the dithering and the delay line are not made explicit. Its internal structure is just a digital ramp (counter) with an higher number of steps, so more accurate. This choice has been made just to study the loop stability. As mentioned before all the simulations on the model are carried on in Simulink with a fixed-step solver and automatic step size detection.

## 4.1 Basic Block Diagram

At this point the Simulink block diagram is very simple, the innovative technique is still implicit. The model represents just a high abstraction level behaviour of the DPWM. This step is needed to verify the correctness of the design choices and test the loop stability and dynamic performances in the best case. The first simulations are presented in fig.4.1 and 4.2 and last 700µ$s$. In the table 4.1 the PID parameters used in the simulations are shown. For these simulations the PID gain values are not chosen to test the loop performances, but just to verify that the system is stable, after this check, the parameters are taken from the small-signal analysis of the previous chapters.

| | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| **0.5V** | 64 | 2 | 32 |
| **3.3V** | 16 | 0.25 | 128 |

To examine the loop behaviour, just the most meaningful waveforms are reported in the graphs, i.e. the output voltage, the error voltage, the PID digital output, the inductor current and some input signals to indicates when a step comes, such as line voltage and load current.

The first thing that is clear from the figures is that in both cases the error voltages never reach zero, they continue to oscillate around it. In static condition it is important that the loop follows the reference, otherwise it fails, oscillating and worsening the static regulation. Zooming on the problem, in fig.4.3 and in fig.4.4, is possible to observe that the oscillations are always $\pm(1 \text{ or } 2)LSB$ of ADC dynamic. The fluctuation can come out from two possible effects, the first are the already well-known LCOs, the other is the double sampling of the ADC. Actually it works at twice the PWM frequency, it can be a problem when the switching ripple of the regulated voltage is not negligible than the ADC resolution, giving two different digital value as output. Let's analyse the problem.


Figure 4.1: First simulation 3.3V

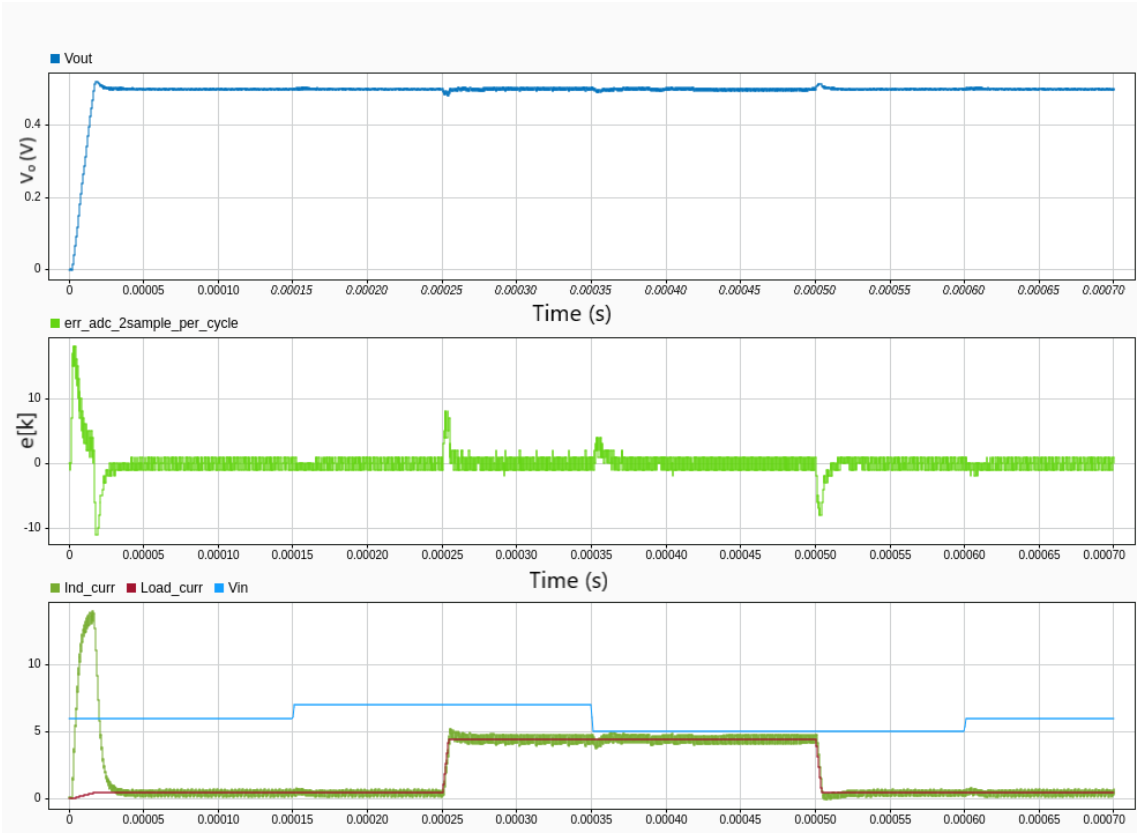



Figure 4.2: First simulation 0.5V

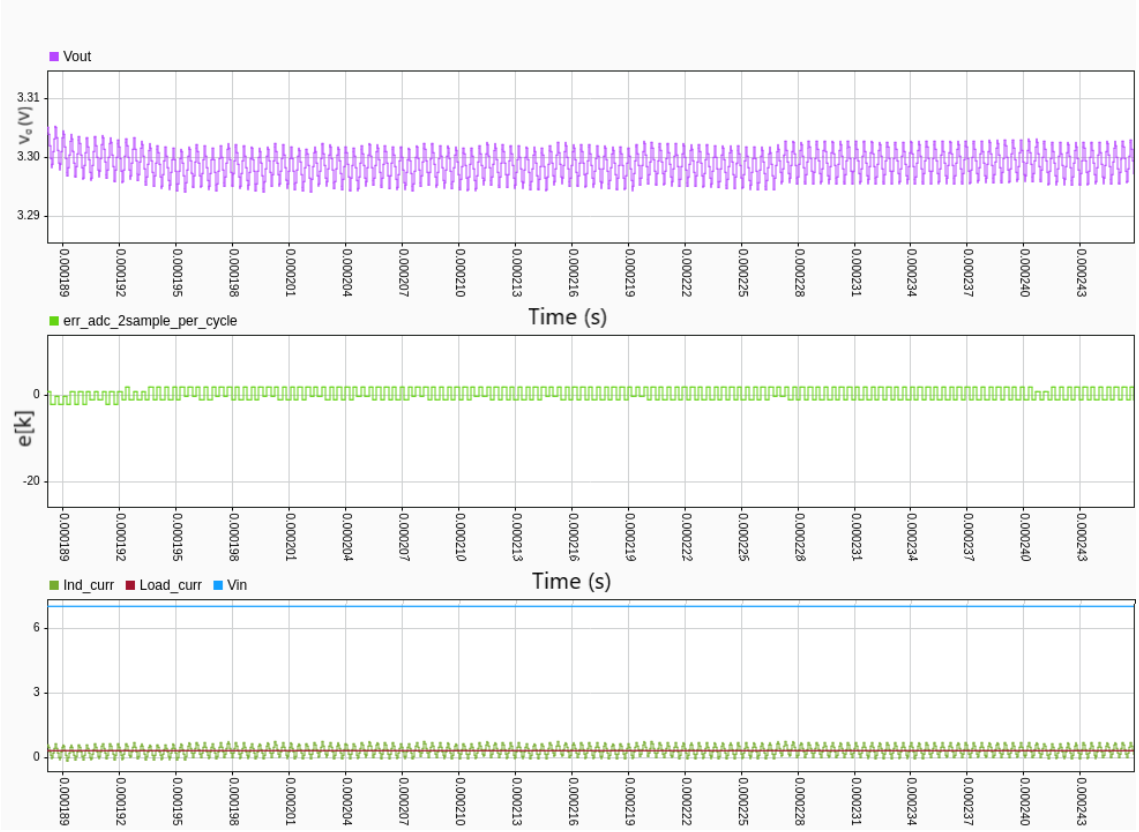



Figure 4.3: Static conditions oscillations 3.3V

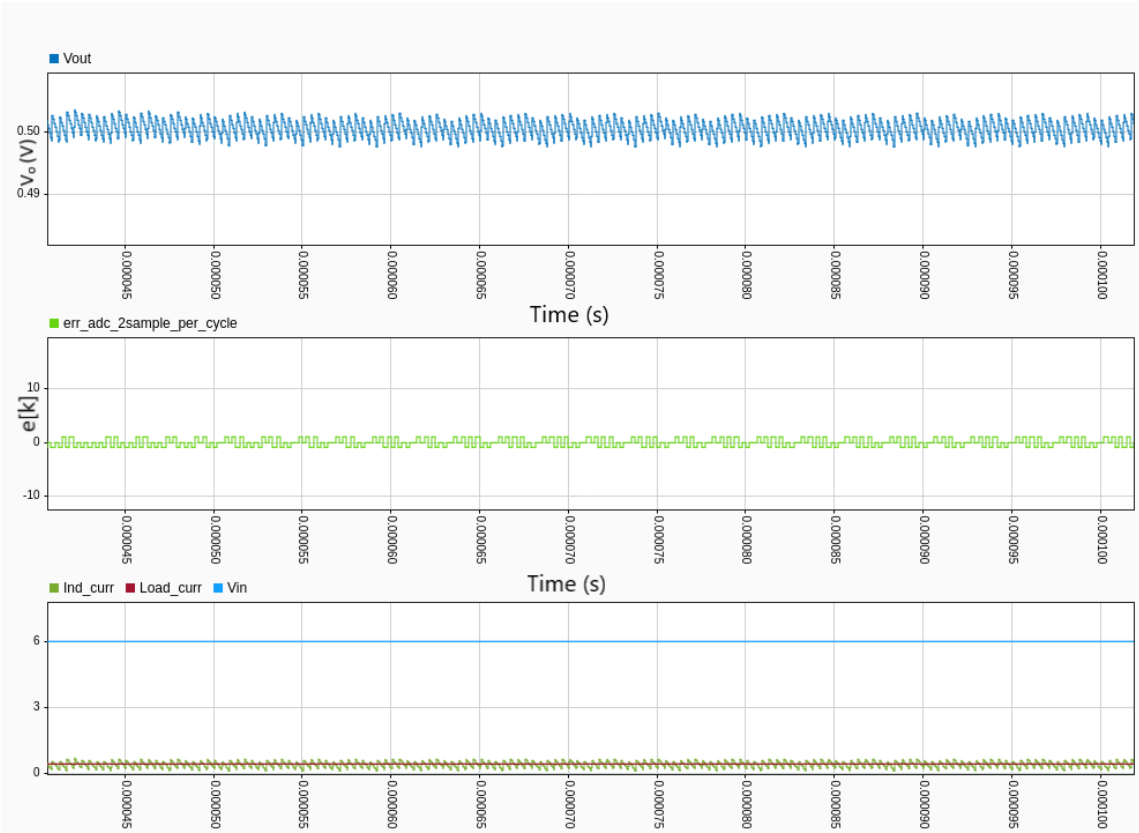



Figure 4.4: Static conditions oscillations 0.5V

Looking at the behaviour of the error, the phenomenon seems to be identical to the common LCOs in all aspects but the frequency. In general this kind of fluctuations exhibits a period larger than the PWM, but in this case the output of the error ADC moves with a lower one. Mathematically, as mentioned before, with an ADC of 7 bits and a 100 ps of DPWM resolution, LCOs shouldn't appear and that's because the LCO absence condition (2.2.2) is satisfied, even if the margin is small, i.e. just 436 mV of difference between ADC an DPWM resolutions (eq. 3.8 and 3.13).

Let's now check how the double sampling impacts on the error voltage. A zoom on the switching action, proposed in fig.4.5, allows to better understand the phenomenon dynamic. The graph shows not only the signals but also two vertical cursors that permit to note when the sampling instants occur. The output voltage is relatively much different in these cases, there are about 5 mV between these two points, more than the error ADC resolution ($\approx 1.8$ mV). For this reason the digital value changes during one PWM period, causing the oscillations seen in fig4.3. Furthermore the amplitude of PID output variations is coherent with the proportional gain of the compensator, for example in the case of 4.5 the cursors' values differ by $1LSB - (-1LSB) = 2LSB$ and the PID output is $\Delta_{PID} = 2483 - 2452 = 31$. Knowing the proportional gain $K_p = 16$ and $\Delta_{err} = 2LSB$, $\Delta_{PID}$ is:

$$\Delta_{PID} = K_p \cdot \Delta_{err} = 16 \cdot 2 = 32LSB \tag{4.1}$$

The value is indeed consistent with the PID action. This demonstrates the cause of the oscillations.

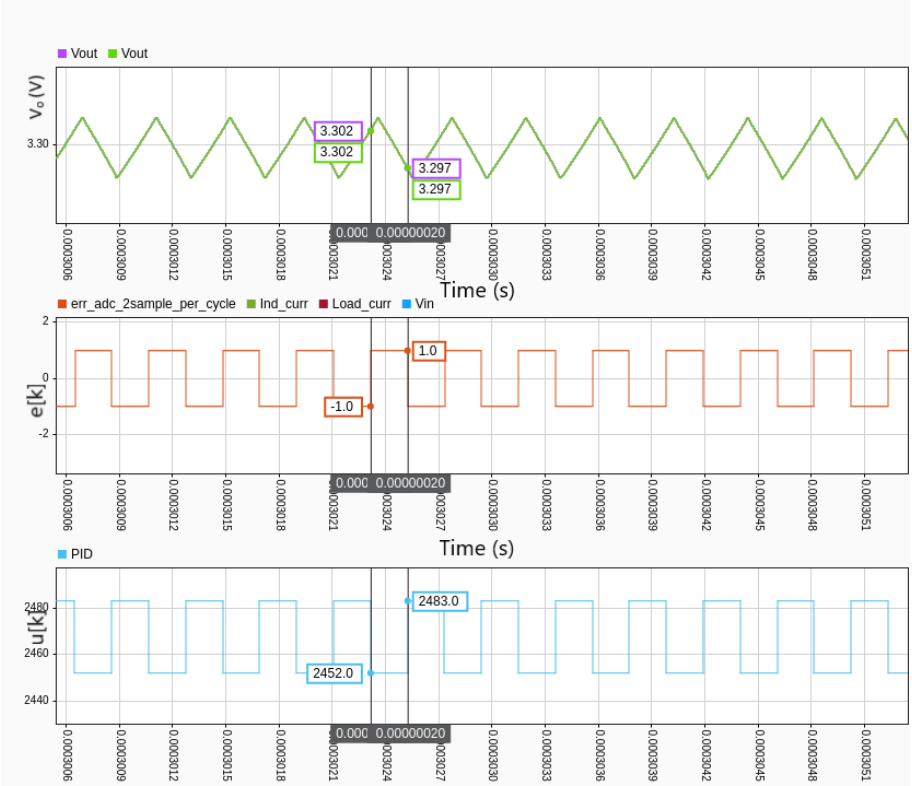



Figure 4.5: Zoom on the double sampling

In view of these considerations, it seems to be reasonable to simplify the model choosing a single sampling approach, i.e. one sample each PWM period. With this choice a pair

of changes have to be done in some other blocks: in the PID block, leaving all the gains unchanged, the second delay on the derivative branch in the explicit representation of the compensator is useless, so it can be removed from the model as is depicted in fig.4.6; in the same way the second delay in the ADC block. In order to verify the correctness of the changes the loop has to be simulated again. The results are shown in fig.4.7 and fig.4.8.

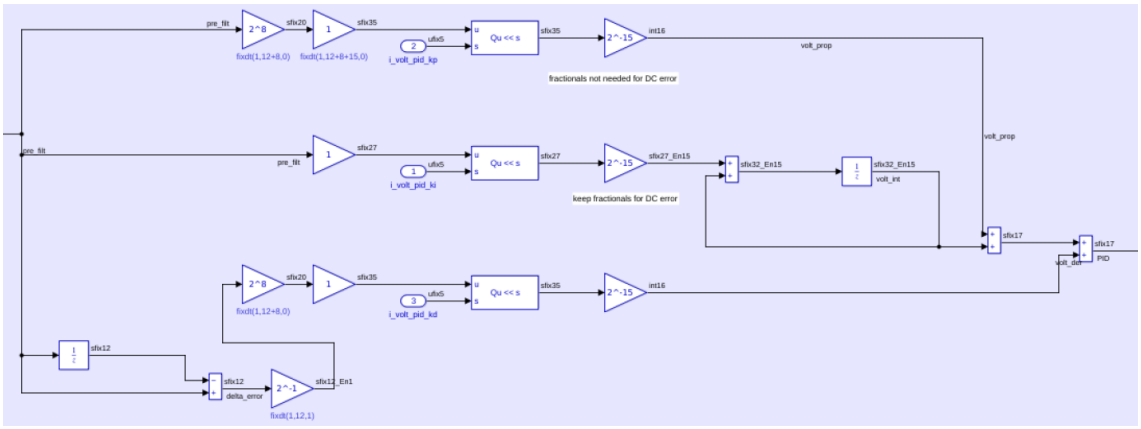



Figure 4.6: New PID Simulink block diagram

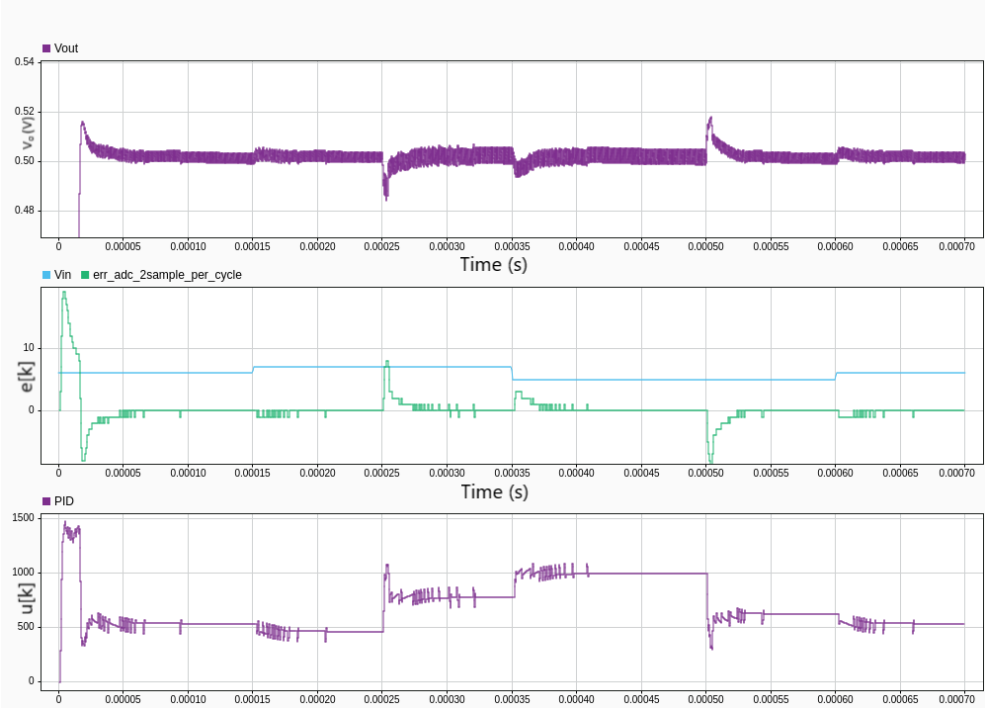



Figure 4.7: 0.5V single sampling

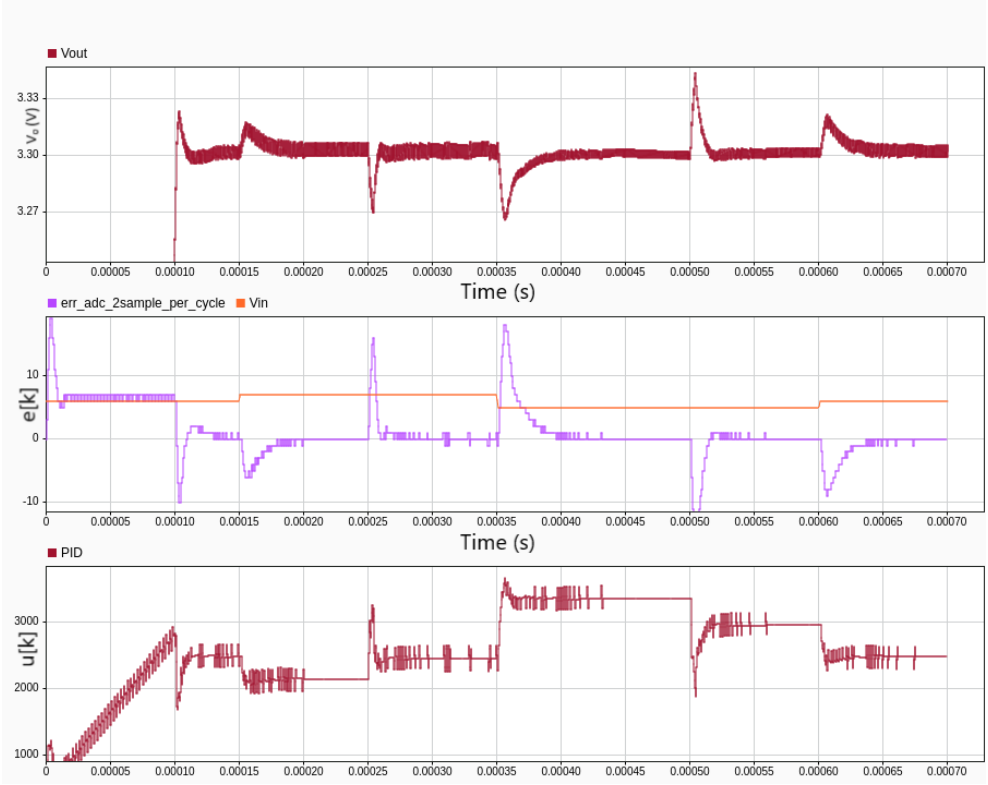



Figure 4.8: 3.3V single sampling

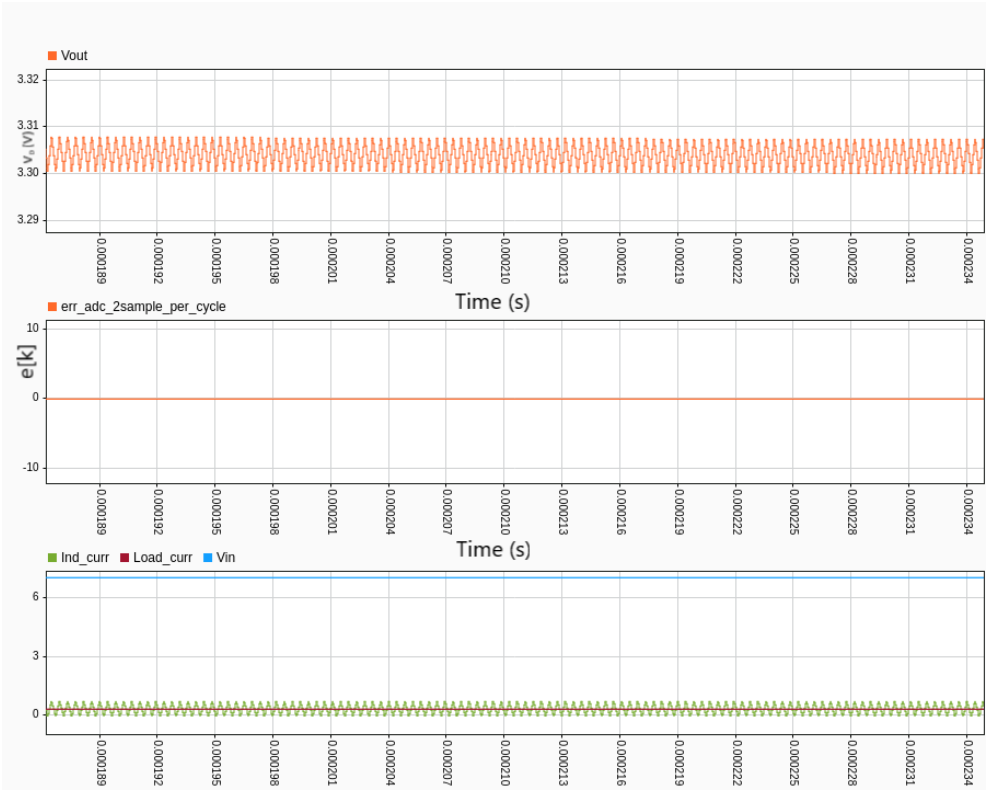



Figure 4.9: No LCO, steady-state conditions, 3.3V

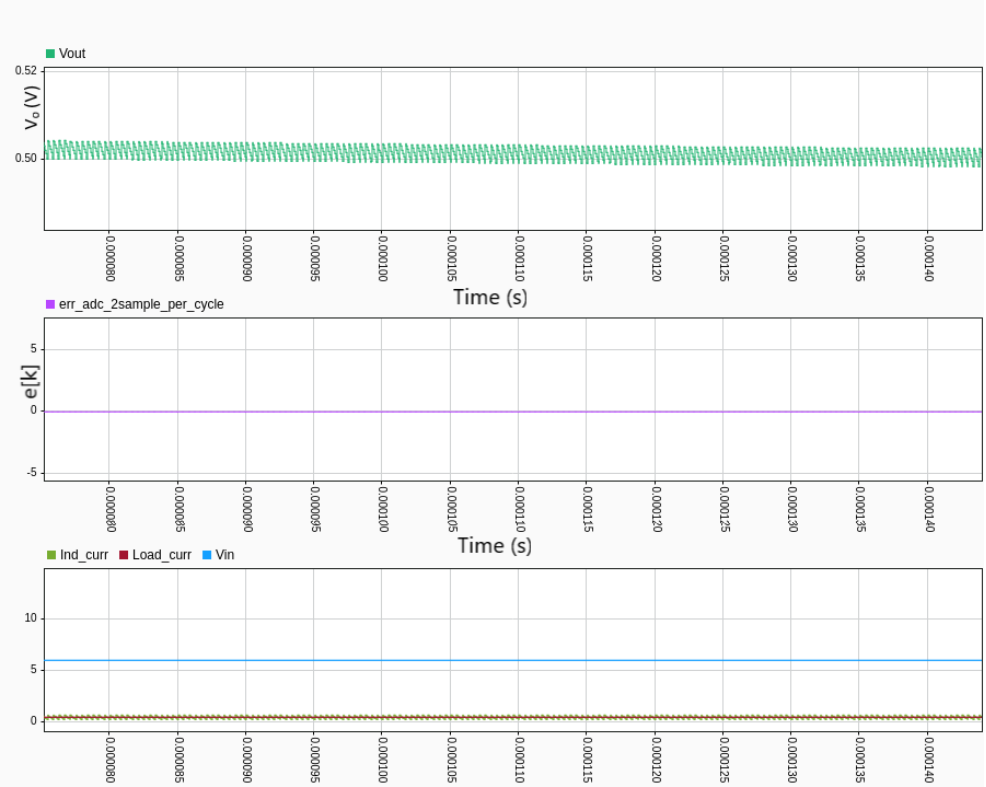



Figure 4.10: No LCO, steady-state conditions, 0.5V

The oscillations, neglecting transients, disappear, so the problem can be considered solved. To better appreciate the results, a focus on the steady state zones is plotted in fig.4.9 and 4.10.

Just to be sure, one can ask what will happen if the LCOs absence condition is violated (2.2.2). The answer can be obtained increasing intentionally the ADC resolution till a value smaller than the DPWM one. For example let's take 8 bit, letting the other parameters unvaried. Now, mathematically is:

$$q_{ADC} = \frac{V_{FSR}}{2^{n_{bit}}} = \frac{240\,\text{mV}}{2^8} = 0.938\,\text{mV} \quad \text{with } n_{bit} = 8 \tag{4.2}$$

$$q_{DPWM} = \frac{V_{in}}{\#\text{number of step}} = \frac{6\,\text{V}}{\frac{417\,\text{ns}}{100ps}} = 1.439\,\text{mV} \tag{4.3}$$

With this ADC, $q_{ADC} < q_{DPWM}$ and LCO has to come out in the output voltage. Restarting the simulation, fluctuations are clearly evident (fig.4.11 and 4.12) and their amplitude changes according with the input quantities. Indeed, in both cases, with 5 V input voltage the oscillations decrease, but the theory explains that: referring to the 3.13 the DPWM resolution is proportional to $V_{in}$, so the fluctuation is relaxed and at the same way with $V_{in} = 7\,\text{V}$ its amplitude is much higher.

These results shows how important is to pay attention to LCOs, they impacts in a huge manner to the static regulation, but the design choice made in the previous sections reaches to completely eliminate the oscillations and to confer to the loop better performance.

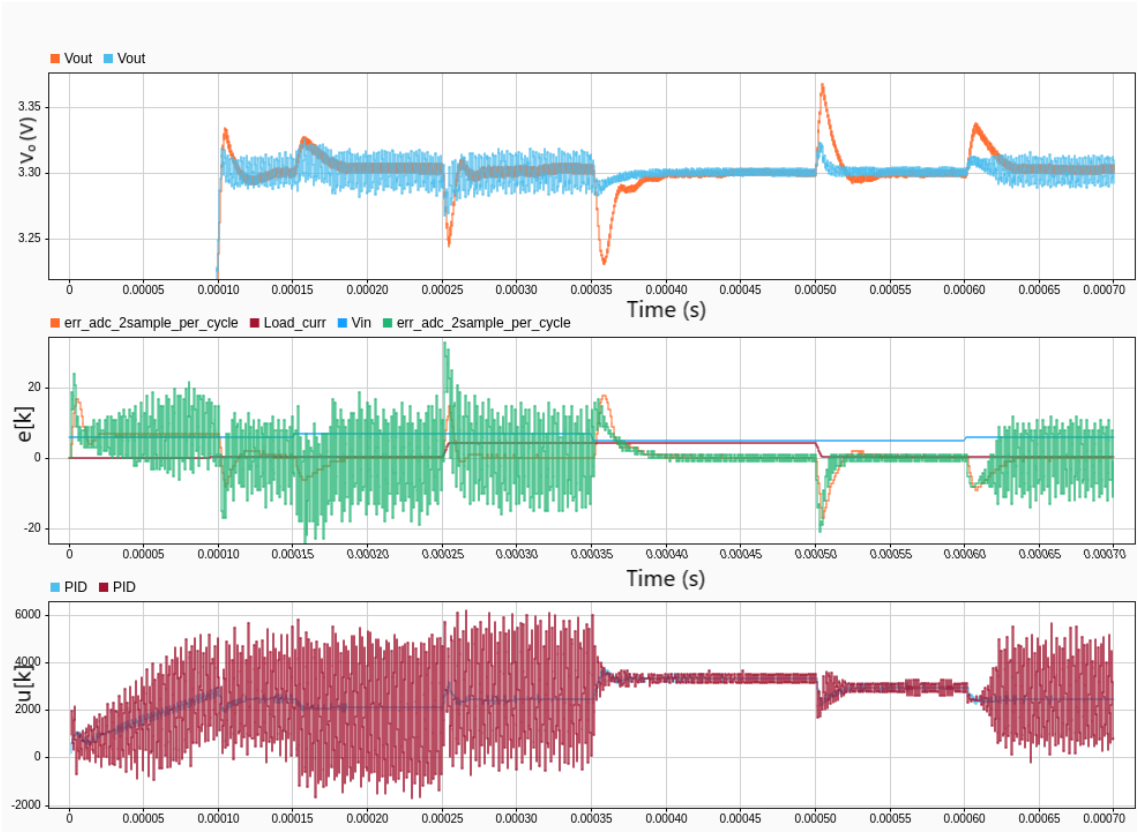



Figure 4.11: LCOs with 8 bit ADC, 3.3V of regulated voltage

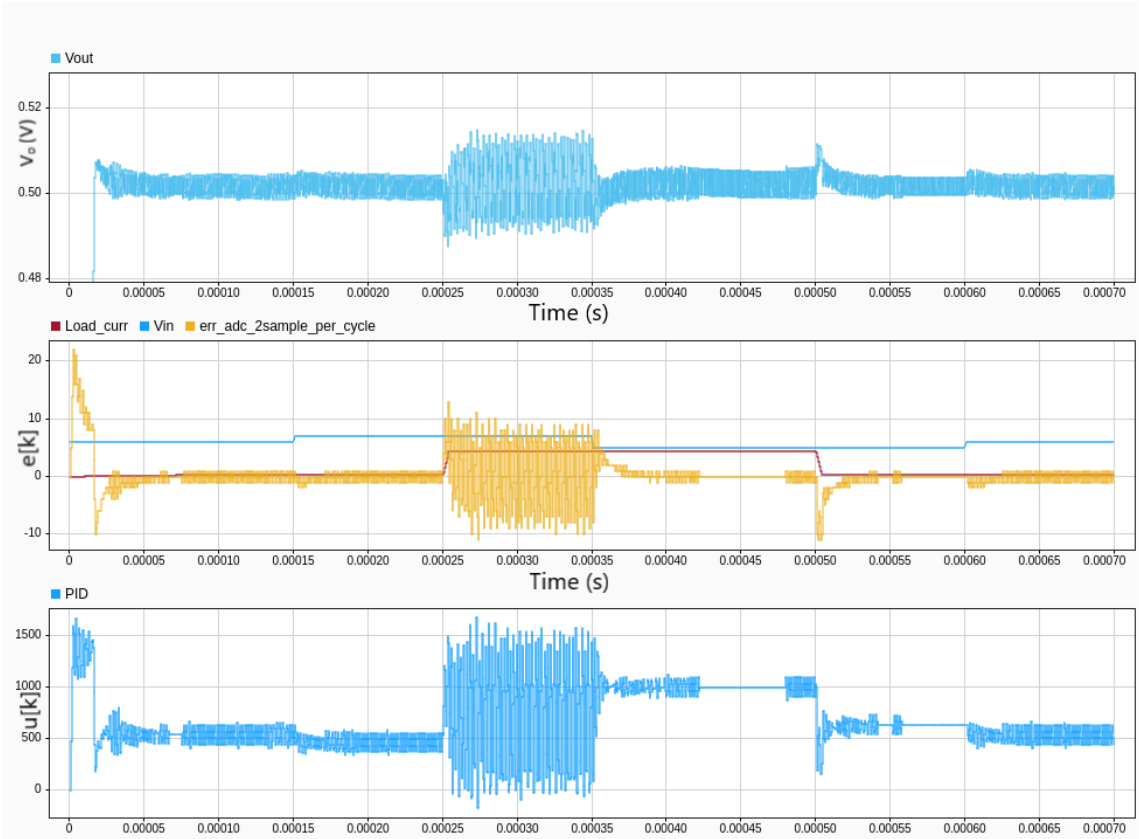



Figure 4.12: LCOs with 8 bit ADC, 0.5V of regulated voltage

At this point it is possible to focus on the performance of the loop. Just with a first arrangement of PID parameters, the system reaches the stability and the dynamic specification is respected for 0.5 V of regulated voltage. 3.3 V needs an adjustment of PID parameters, essentially it presents on the regulated voltage some peaks that depart from 3.3V by 50 mV (fig.4.8). It's time to use the values similar to that found in section 3.3.3, i.e. $K_p = 91$, $K_i = 2.67$ and $k_d = 249$. The result is shown in fig.4.13.

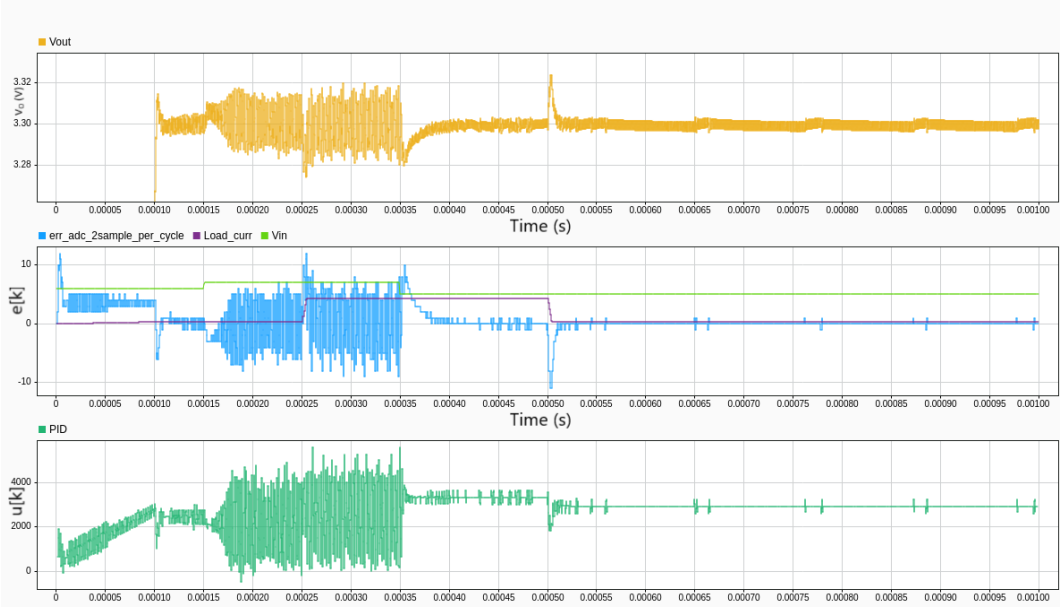



Figure 4.13: 3.3V, PID parameter found with the formulas


Figure 4.14: 3.3V, improved performance

It is clear that the parameteres need an adjustment. The regulated voltage shows instability, in particular in correspondence of high input voltage and high load current. The PID is changed several times to obtain an acceptable performance. The simulations show that the parameter that the most influences the behavior of the loop is the derivative gain. Indeed $K_d$ seems to be estimated higher than the loop needs, in all the attempts its value had to be lowered by a 40%. Fig.4.14 shows the PID performance with a $K_d = 144$ instead of 249.

The simulation is more stable in every load and line voltage conditions. Now the peaks depart just by 30 mV, the amplitude has been reduced by 40%. The problem are some other oscillations that come up in steady state, some of them die very quickly, others seem to remain. The phenomenon is a type of instability given by some constraints that are not included in the small-signal model during the PID design. For this reason it can be useful to know why they come up and how to eliminate them. To better study the effect, the inputs that causes fluctuations are maintained constant till the end of the simulation, the latter is proposed in fig.4.15. The effect does not seem to be temporary, indeed it lasts till the end of the simulation.

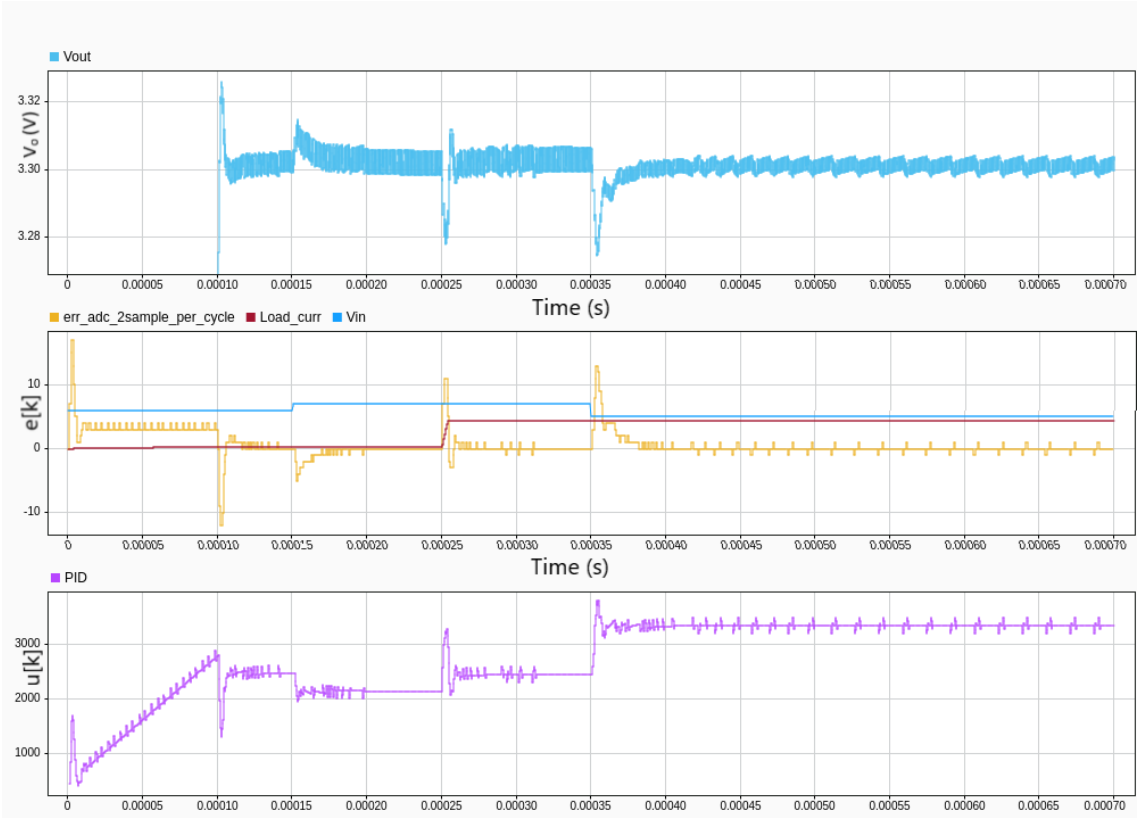



Figure 4.15: 3.3V, durable oscillations

It seems a sort of undamped or slightly damped oscillation coming from the controller that appears to be unable to quickly cut out the fluctuation. To confirm the hypothesis, it is enough to vary the values of the PID parameters, in particular it is necessary to relax the performance not to induce oscillations. In order to avoid lowering performances, the most logical option is to alter just the integral gain. As already seen in Chapter 2, the proportional and derivative gains are responsible of the lead compensation. The integral gain represent the lag term of the transfer function and it impacts majorly on the lower frequencies guaranteeing an high value of dc gain to improve the regulation in steady-state. As an example, the value can be reduced from 3 to approximately 1.5, so its half. In fig.4.16 there are still some oscillations in the regulated voltage, but it is easy to see that they are more strongly damped than the previous case. It is possible to emphasise the improvement further decreasing the gain, one can choose 0.83 and see what happens. Launching the simulation, reported in fig.4.17, the oscillations completely disappeared. Restoring the initial load and line steps, some oscillations are present in the response of the system, but observing their tendency (in fig.4.18), it's clear that they are just a dying transient that takes some time to reach the perfect stability. Furthermore, in spite of changes in the PID parameters, performances remain the same of 4.14, but without the annoying fluctuations.


Figure 4.16: 3.3V, damped oscillation

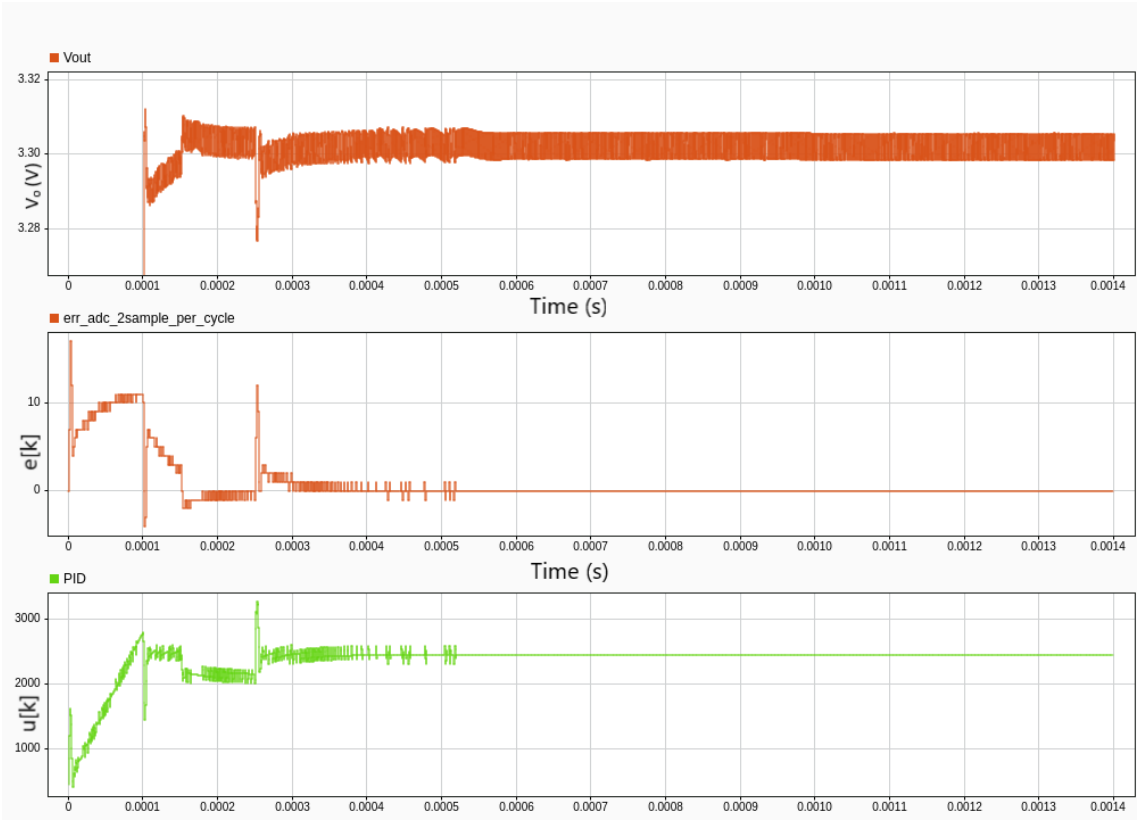



Figure 4.17: 3.3V, oscillations disappear

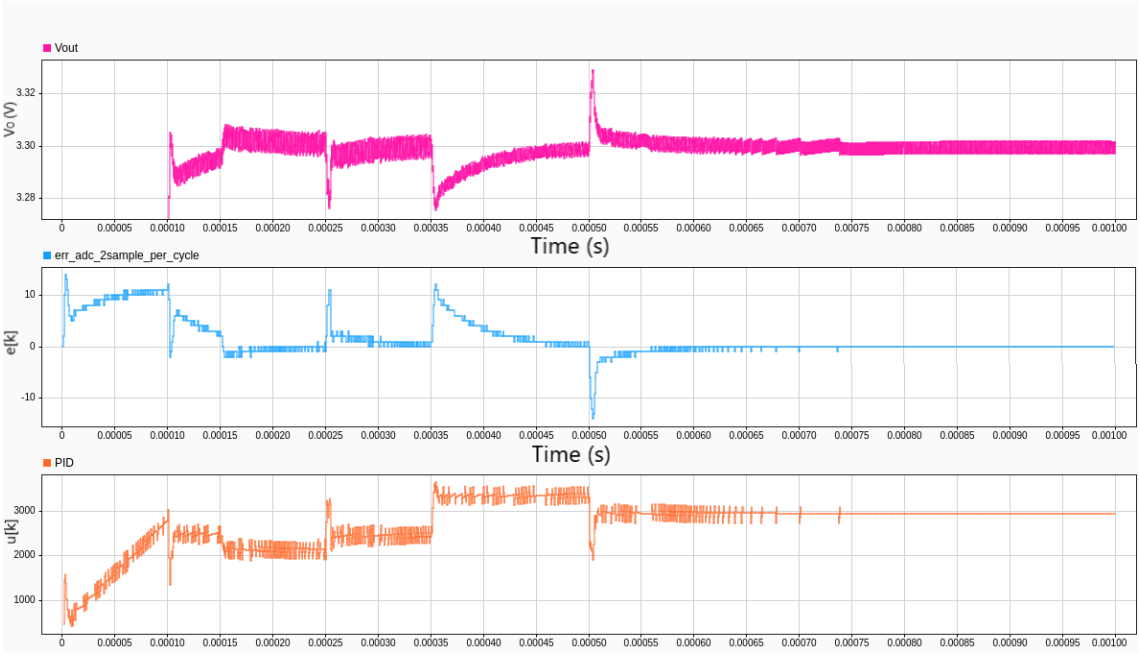



Figure 4.18: 3.3V, modified PID, overall simulation

From the above considerations, it is evident that the PID values provided in Chapter 2 need to be tuned to obtain a very stable voltage regulation. This simulations bring to the attention non-linear aspects that the theory does not take into account. In particular to

eliminate all the oscillations the derivative term has to be almost halved and the integral one slightly moved of few points when needed.

## 4.2 High-resolution DPWM Techniques

In this section the high-resolution DPWM techniques are turned explicit implementing block diagrams that represent their behaviour. In order to design the block and obtain the best performance in terms of dynamic transients and spectral envelope of the output voltage, a comparison of all techniques is proposed. Firstly the conventional *Thermometric* dithering technique is introduced, then the *Dyadic* dithering technique is proposed and the advantages are highlighted. To complete the analysis the first-order and second-order *Sigma-Delta (*$\Sigma\Delta$*)* techniques are included in the design process and a final recap of all techniques is reported.

### 4.2.1 Dithering

Digital approaches that vary the duty cycle of one least significant bit (LSB) through predetermined dithering patterns have been developed in order to manage the average duty cycle with a sub-LSB resolution in order to improve the DPWM resolution without raising the digital clock rate. Dithering is cheap and easy to perform digitally, but it can regrettably contribute noise at switching frequency subharmonics that the converter output filter cannot effectively reject, which typically results in significant subswitching frequency output ripple [4].

#### Thermometric Dithering

The conventional dithering technique is certainly the *Thermometric*. As reported in fig.4.19, the digitally quantized duty cycle is $\frac{(n+1)}{2^N}$ in the first $m$ switching periods of a $2M$ dithering pattern and $\frac{n}{2^N}$ in the remaining periods, so that an average duty cycle $\frac{(n \cdot 2M+m)}{2^{N+M}}$ quantized over $N+M$ bits can be achieved, thus increasing the effective DPWM resolution by M bits. As mentioned in Section I, this technique may introduce noise at switching frequency subharmonics.

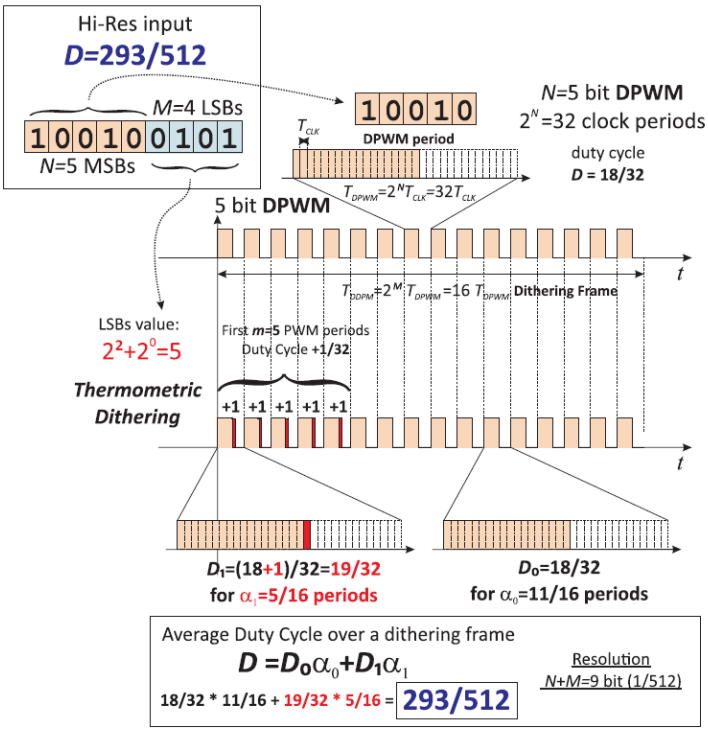



Figure 4.19: High-resolution DPWM by thermometric duty cycle dithering over $2^M$ switching periods[4]

In order to check the validity of this technique, a model has to be structured and simulated. For this test a 4 bit dithering is used and the previous DPWM block resolution is reduced by the same number to balance the overall resolution. This approach can be represented easly in Simulink, it is sufficient to compare a 4 bits free-running counter, that increments every pwm period, with the 4 LSBs provided by the PID. Until the count is greater than LSBs, the block adds one LSB to the MSBs coming from the PID, otherwise it let the PID's value as it is, without adding anything. In this way a higher resolution can be obtained averaging the whole duty cycle during 16 pwm periods.

In fig.4.20 it can be observed that, without input steps, the output voltage is very stable. It means that the approach is able to obtain such a resolution to satisfy the LCO absence condition, without increasing the system clock frequency. The drawback is an increased ripple that comes from the average operation between two level of consecutive quantized duty cycle. In fig.4.21 it can be better understood how it works and better observe how the ripple acts. In the top of the figure the regulated voltage is shown where a low-frequency oscillation is superimposed to the normal switching activity. Its period seems to be something more than 5 µs. In the bottom of the figure the dithering ramp is represented with the value of PID LSBs, that is used as comparing term. The ramp, as already seen, is obtained from a 4bit free-counter, for this reason the LSB used as input for the PWM block is changed every 16 PWM period. This is why the oscillation has such a period, i.e. one sixteenth of the switching period that, is $f_{low} = \frac{f_s}{16} = \frac{2.4\,\text{MHz}}{16} = 150\,\text{kHz}$, that corresponds to a period of $T_{low} = (150\,\text{kHz})^{-1} = 6.67\,\mu\text{s}$. It is consistent with the period previously observed in fig.4.21. Every time the ramp value exceeds the 4 PID LSBs the dithering

circuit provides, instead of zero, one at the output that is added to the MSBs of the PID's output. It is important to underline that this process is useful only in steady-state, during dynamic variations of the input a LSB added to the other 8 MSBs doesn't change the performance. Nevertheless the technique works, in fact the error voltage sampled by the ADC is fixed to zero, so the system reaches the stability.

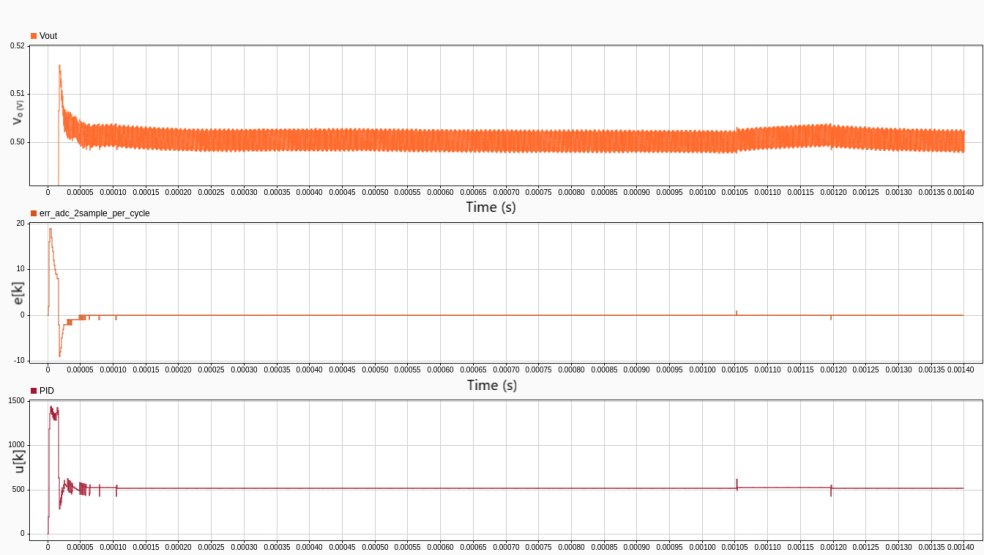



Figure 4.20: Thermometric Dithering simulation

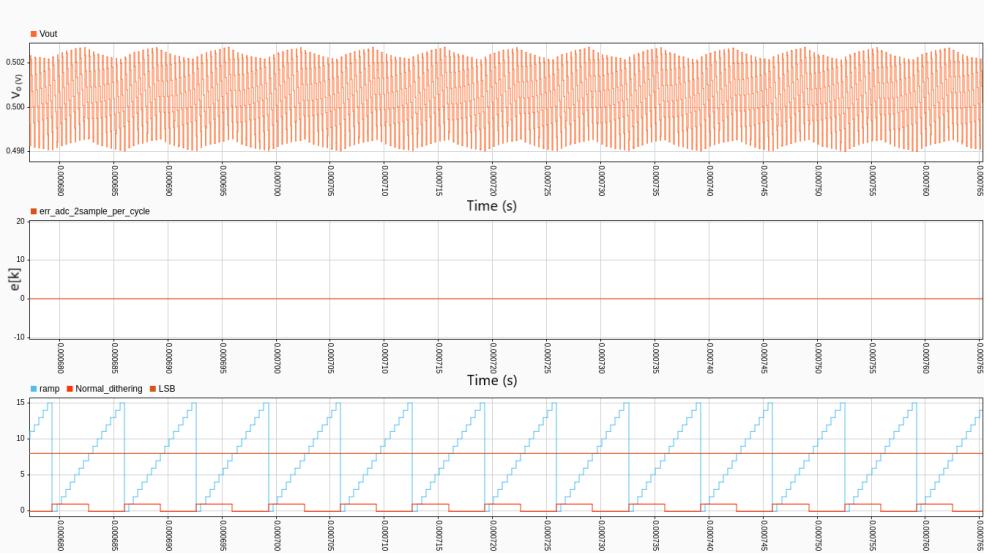



Figure 4.21: Thermometric Dithering ripple oscillations

Now, it is important to quantify this oscillation in term of voltage. In fig.4.22 there is a zoom of the ripple, where its amplitude can be appreciated. The trend of this oscillation seems to be a sort of exponential, consistent with the capacitive behaviour of the load. The amplitude is approximately 600 µV, considering 1.875 mV of ADC resolution, that is a contribution of the static load regulation, the oscillation amplitude is not so small.

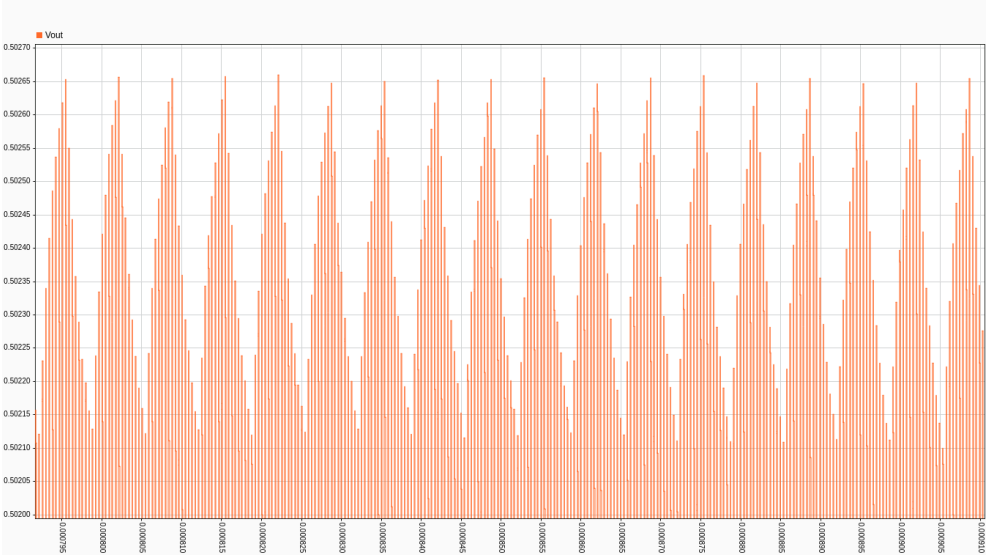



Figure 4.22: Thermometric Dithering ripple oscillations zoom

The last consideration regards the spectrum of the output voltage. Analysing the FFT of the output it is possible study the band occupation of the Buck and if such oscillations are a problem for the emissions of the system that must be lower that certain values in determined bands. The resulting spectrum is attached in fig.4.23. From the graph it is easy to recognise the main frequency component at 2.4 MHz that represents the PWM switching frequency and its harmonics at multiples of the fundamental, but there are also *subharmonics*. In particular, neglecting components at very low frequencies that are of low resolution, there is a peak at 150 kHz, i.e. the dithering oscillation frequency, and its harmonics. Obviously the amplitude of switching action component is not of interest, it is an intrinsic effect of the Buck converter, but is different for the dithering component. Its peak power is $-55dBm$, the *Dyadic Dithering* aims to reduce this power or in other terms reduce the amplitude of the oscillations.

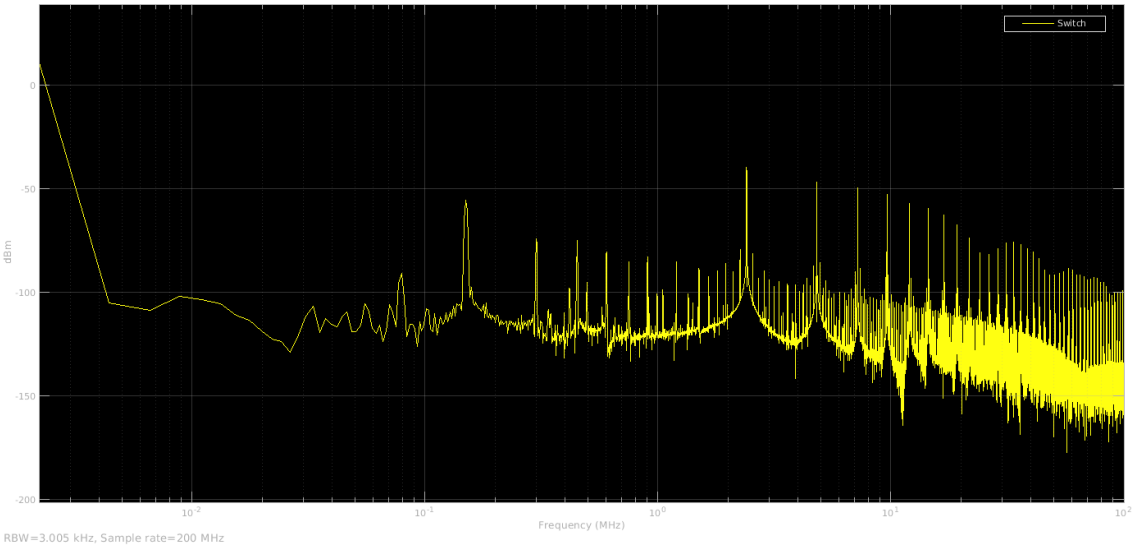



Figure 4.23: Thermometric Dithering spectrum

### Dyadic Dithering

The *Dyadic Dithering* technique offers a more symmetric pattern of pulses compared to the thermometric one that can reduce the oscillation amplitude[4]. The idea is proposed in fig.4.24. It's clear how the additional pulses are better distributed over multiple PWM periods, this modulation permits to avoid abrupt peaks on the ripple and to take under control the average switching action. Truly speaking its implementation is very simple, it is sufficient to use a priority multiplexer and a n-bit binary counter as selection signal, it depends on how many bits of dithering one want to use.

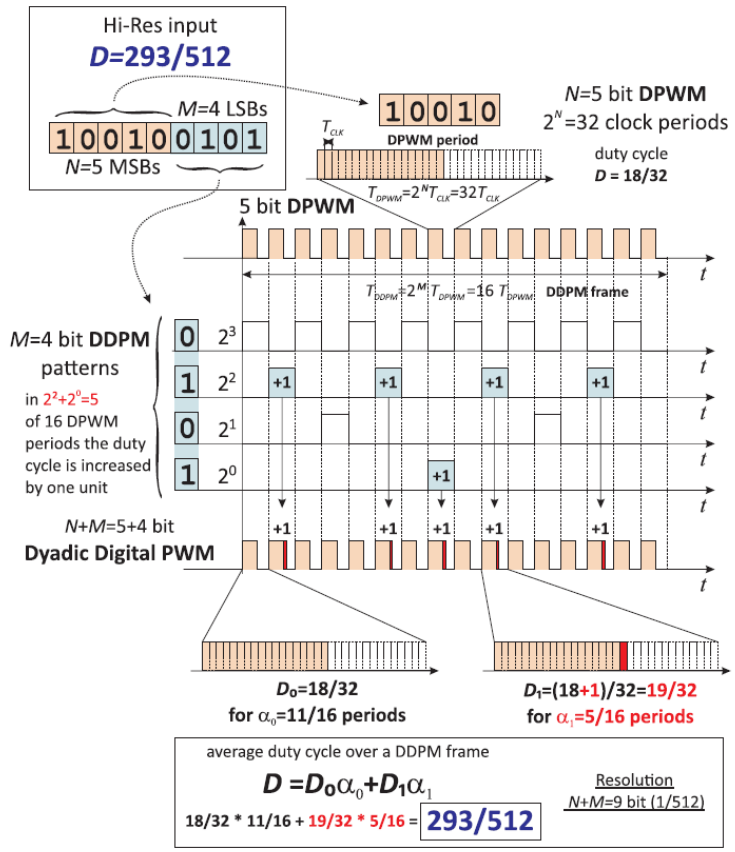



Figure 4.24: Dyadic Dithering basic idea[4]

The circuit automatically provides as input a more symmetric sequence of pulses and consequently less oscillations on the regulated voltage. The high-level structure of the dithering technique is shown in fig.4.25, where is underlined how it combines with the counter-based DPWM. Indeed, it processes only a part of the word coming from the PID, i.e. a certain number of LSBs, instead the remaining MSBs act normal. The dithering adds one LSB to the MSBs only when needed.

In Simulink the behaviour of the technique is represented through the model attached in fig.4.26. The structure is more complicated but, in case of 4 bit of dithering, it simply uses 4 dyadic basis signals obtained by a 4 bit free-running counter (16 PWM periods) and puts these in AND with the 4 PID LSBs, finally puts all AND's outputs in OR to obtain the output.

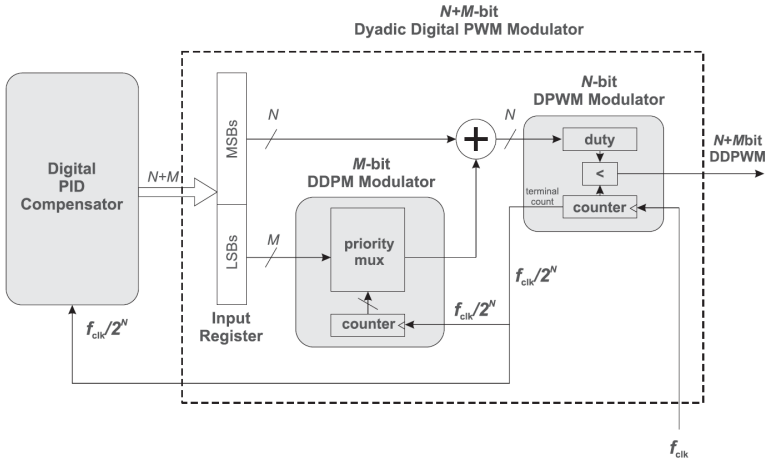



Figure 4.25: Dyadic Dithering high-level block diagram[4]

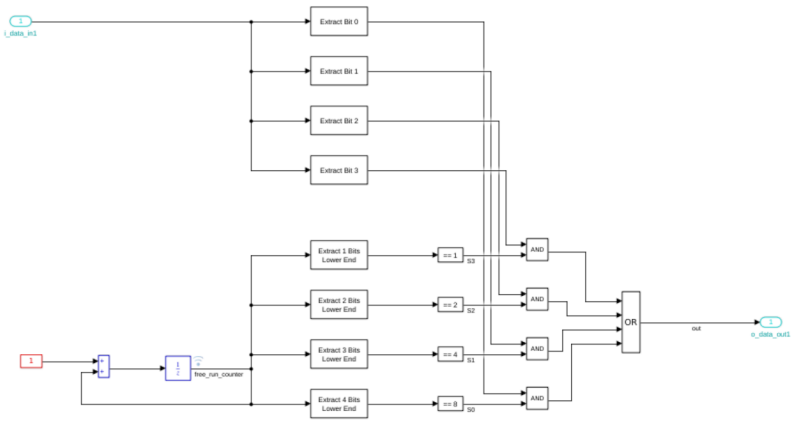



Figure 4.26: Dyadic Dithering block diagram

Now it's possible to measure the performance of the proposed technique. The analysis is made more deeply to make a comparison between multiple depth of dithering, i.e. how many bits are used. After that a design choice is made considering a trade-off between the regulated voltage oscillations and expected area saving. In other words the more is the depth of dithering the less is the number of steps used for the counter-based PWM and delay line, but on the other hand the more is the depth of dithering the higher is the amplitude of oscillations due to the average action. The analysed cases are, 3, 4, 5, 6 and 7 bits. Obviously the overall resolution of the DPWM is not changed, an increment of bits of the dithering corresponds to a decrement of the same entity of the number of bits on the ramp used for the PWM modulation.

Regarding 3 bit of dithering, all the simulation for 0.5V of regulated voltage are reported in fig.4.27 and 4.28. Unfortunately in this case the dither does not appear, in fact the 4 PID's LSBs are all zero, so there is no oscillation.

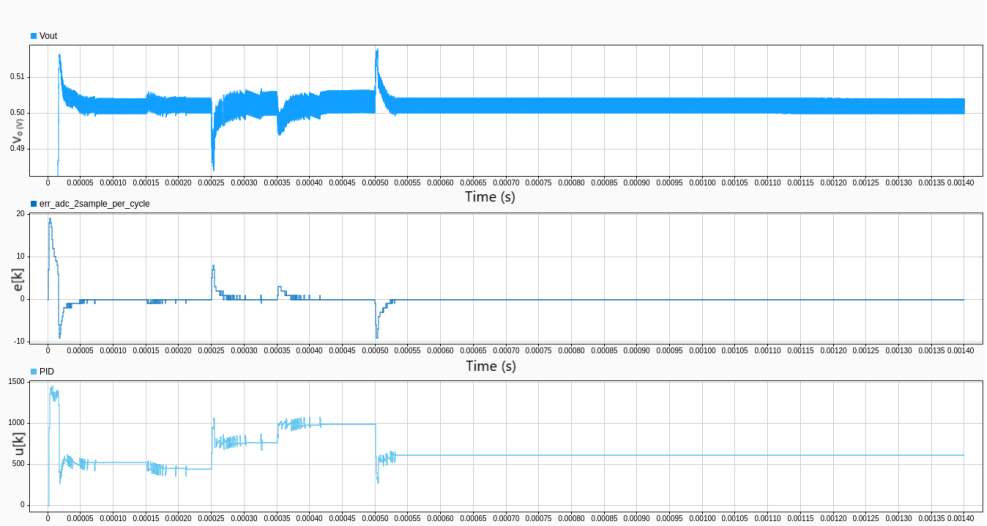



Figure 4.27: 0.5V, 3 bit Dyadic Dithering

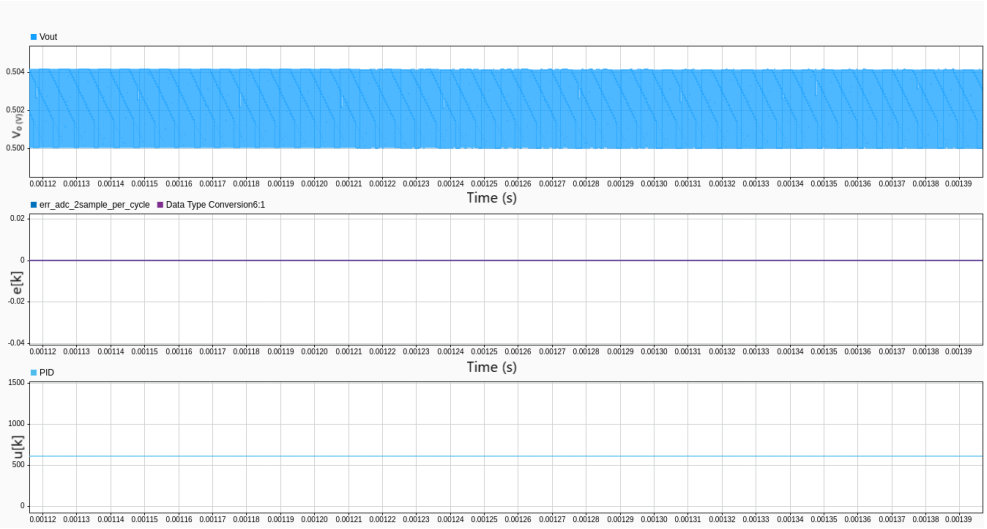



Figure 4.28: 0.5V, 3 bit Dyadic Dithering oscillations

Things change for 3.3V, the simulations are attached in fig.4.29, 4.30 and 4.31. In the first plot the validity of the technique can be appreciated: actually the regulated voltage, after transients, is fixed, the error voltage is stable at zero as well as the PID output. The second offers a view of the used dithering pattern and it is clear that the pulses are better distributed within the PWM period than the thermometric method. The third image shows a zoom on the oscillations enabling a measure of their amplitude, that is 50 µV.

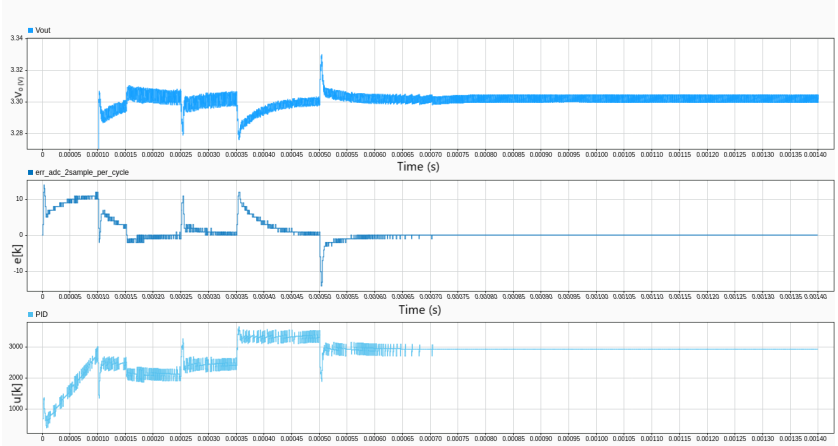



Figure 4.29: 3.3V, 3 bit Dyadic Dithering

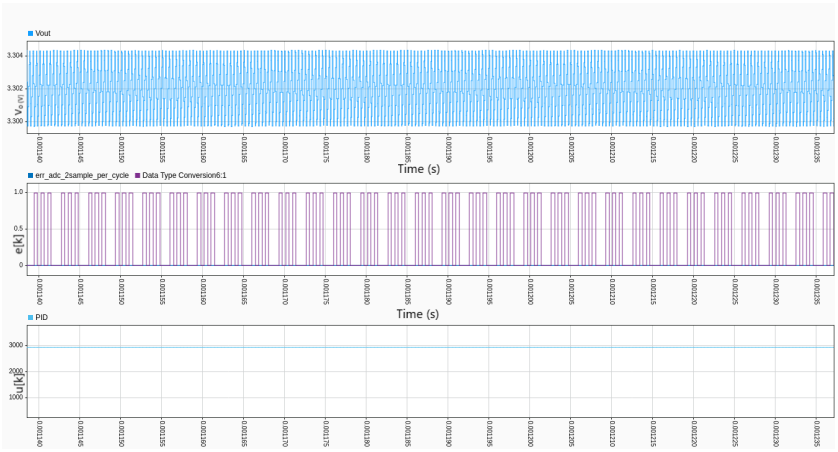



Figure 4.30: 3.3V, 3 bit Dyadic Dithering oscillations

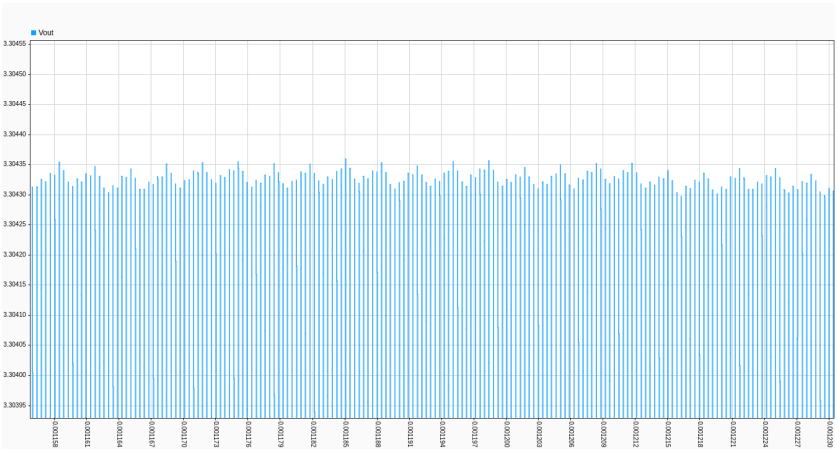



Figure 4.31: 3.3V, 3 bit Dyadic Dithering zoom on oscillations

The same analysis is made for 4 bits of dithering. In fig.4.32, 4.33 and 4.34 all the simulations are reported for 0.5V of regulated voltage. The situation is the same of 3 bits of depth, there's just a difference, the amplitude of oscillations, that is now 75 µV.


Figure 4.32: 0.5V, 4 bit Dyadic Dithering

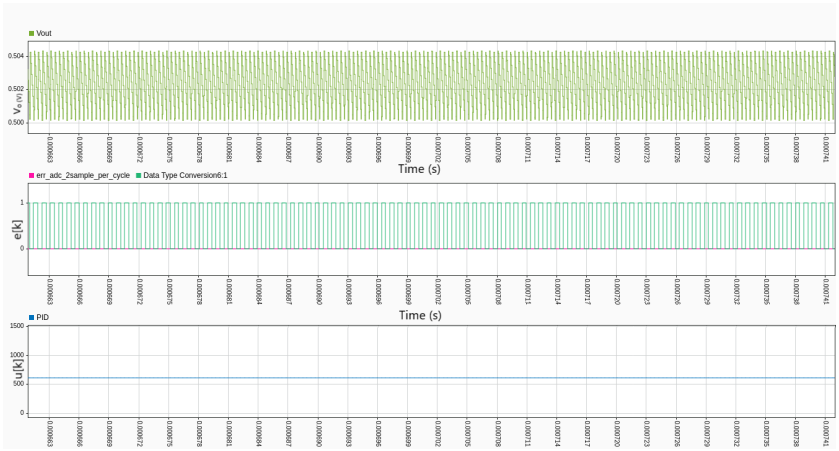



Figure 4.33: 0.5V, 4 bit Dyadic Dithering oscillations

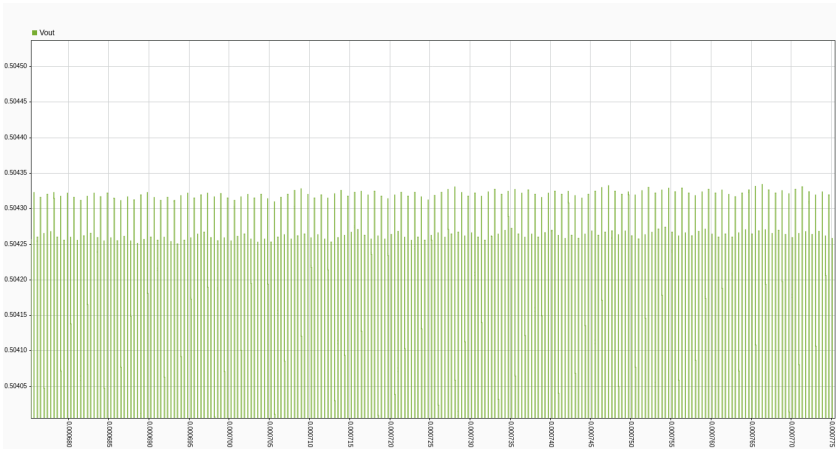



Figure 4.34: 0.5V, 4 bit Dyadic Dithering zoom on oscillations

Same considerations for 3.3V of regulated voltage. All the simulations are shown in fig.4.35, 4.36 and 4.37. As already said, the only difference is in the amplitude, that is 100 µV.

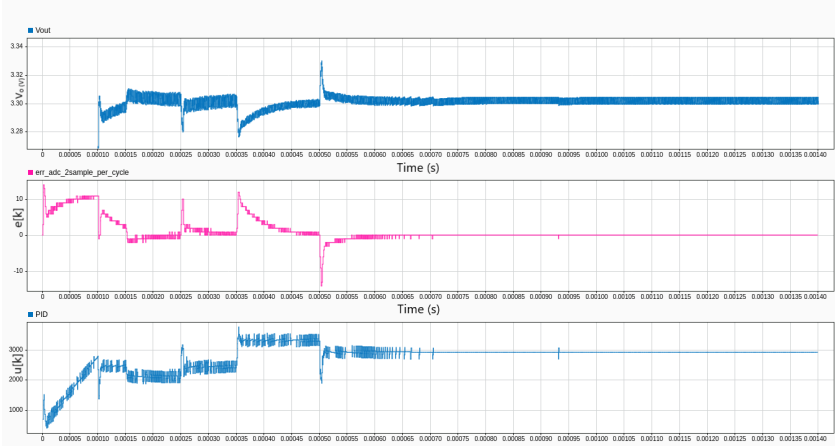



Figure 4.35: 3.3V, 4 bit Dyadic Dithering

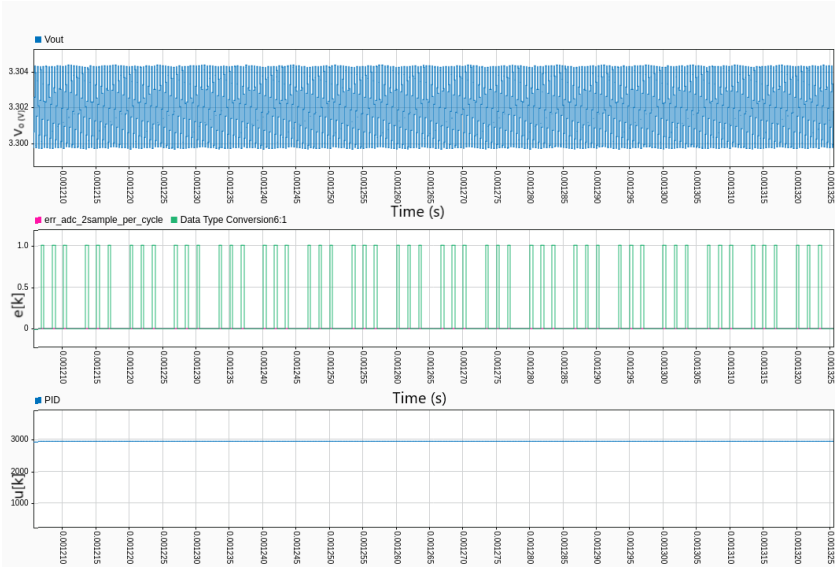



Figure 4.36: 3.3V, 4 bit Dyadic Dithering oscillations

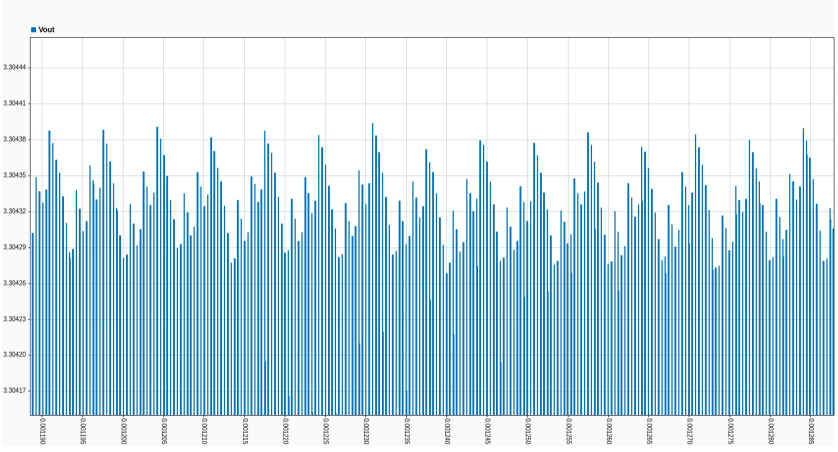



Figure 4.37: 3.3V, 4 bit Dyadic Dithering zoom on oscillations

The same analysis is made for 5 bits of dithering. In fig.4.38, 4.39 and 4.40 all the

simulations are reported for 0.5V of regulated voltage. Now the amplitude is 200 µV.

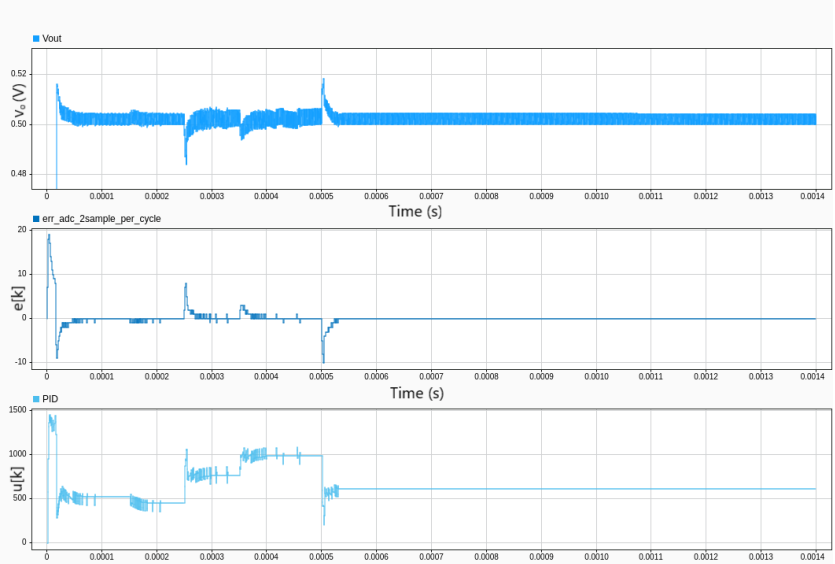



Figure 4.38: 0.5V, 5 bit Dyadic Dithering

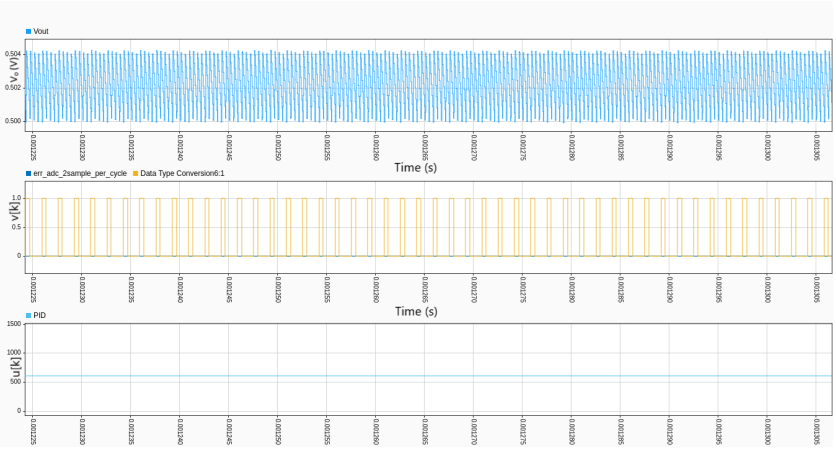



Figure 4.39: 0.5V, 5 bit Dyadic Dithering oscillations

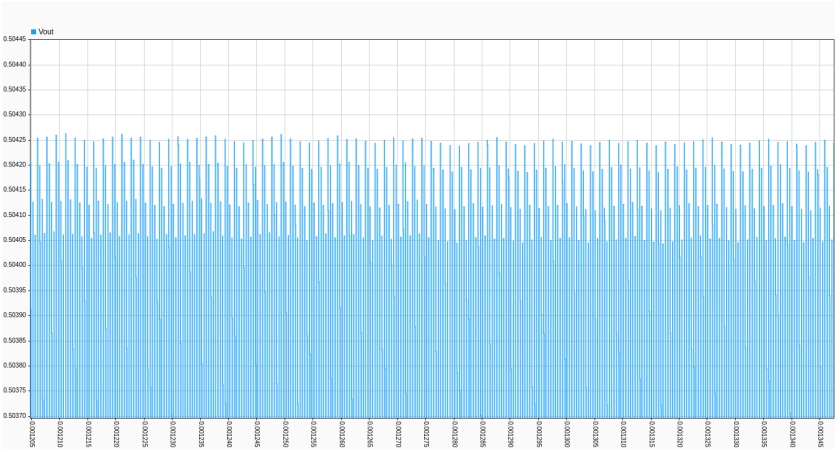



Figure 4.40: 0.5V, 5 bit Dyadic Dithering zoom on oscillations

Same considerations for 3.3V of regulated voltage. All the simulations are attached in fig.4.41, 4.42 and 4.43. As already said the only difference is the amplitude, that is 200 µV.

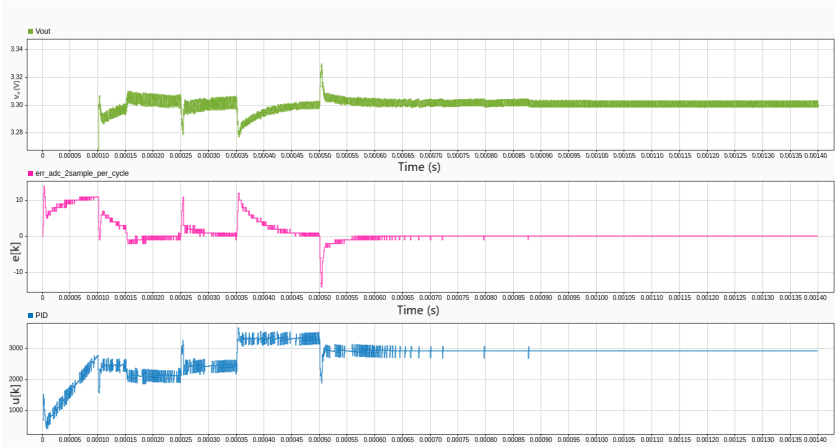



Figure 4.41: 3.3V, 5 bit Dyadic Dithering

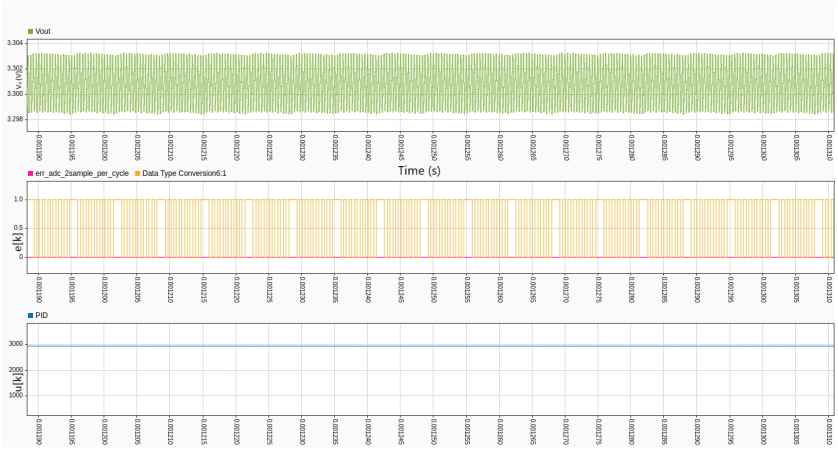



Figure 4.42: 3.3V, 5 bit Dyadic Dithering oscillations

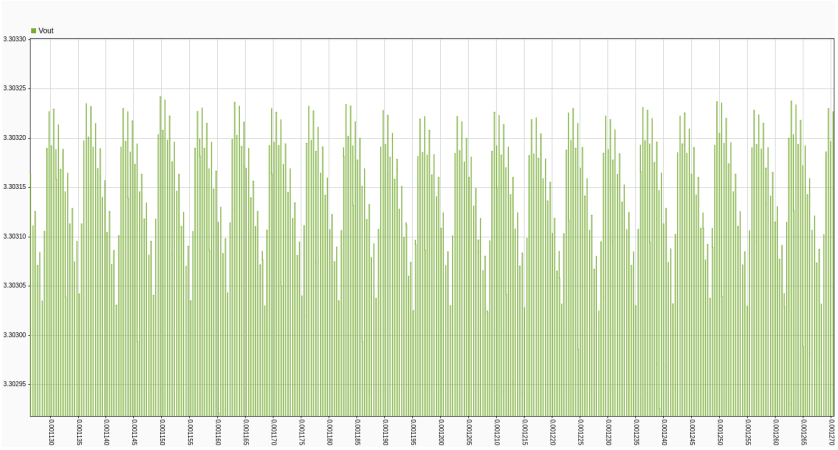



Figure 4.43: 3.3V, 5 bit Dyadic Dithering zoom on oscillations

Regarding 6 bits of depth, the situation is slightly different because, as shown by

fig. 4.44, 4.45 and 4.46 for the 0.5V regulated voltage, the steady-state condition is not guaranteed, sometimes the ADC provides non-zero values. The second figure proposes a focus on the instants when PID output changes. It is possible to observe how a variation on the pattern provided by the dithering block gives rise to differences in the oscillations. The third figure provides a zoom of these oscillations, their amplitude is around 600 µV.

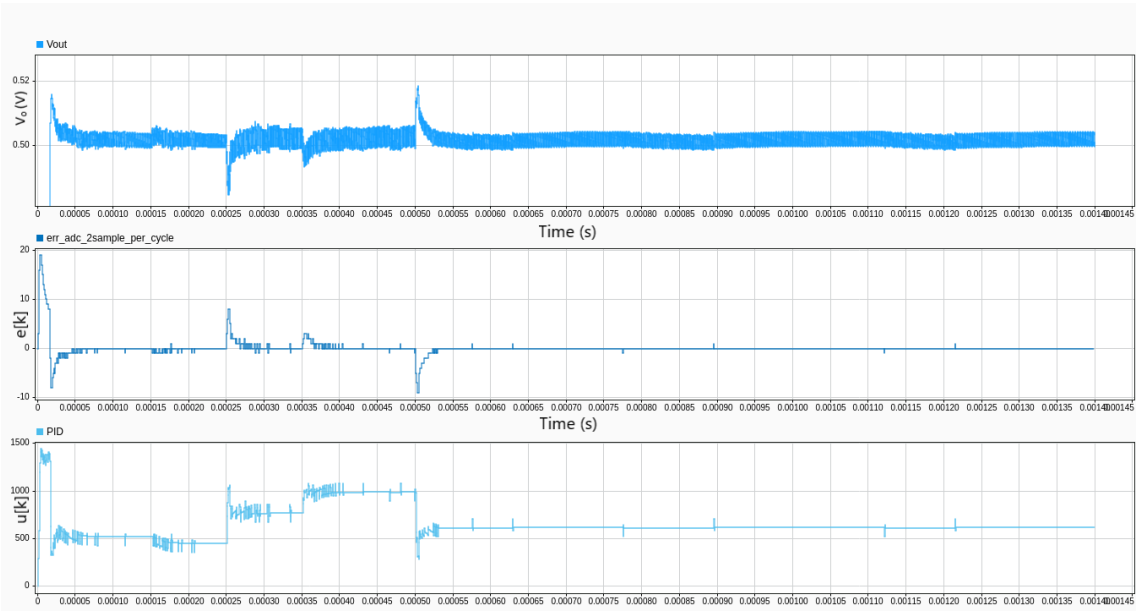



Figure 4.44: 0.5V, 6 bit Dyadic Dithering

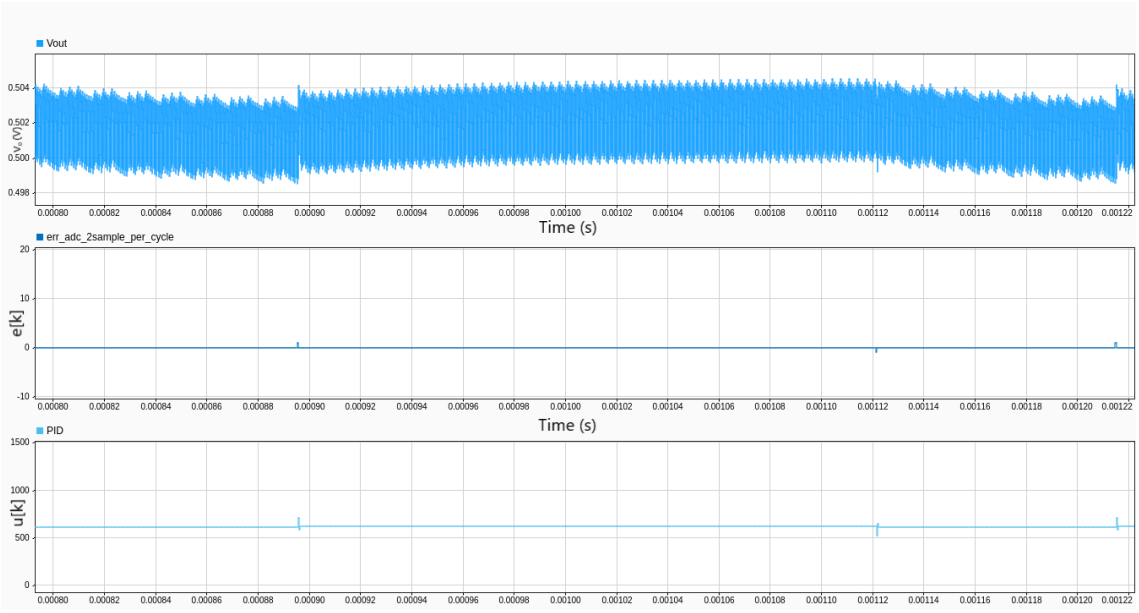



Figure 4.45: 0.5V, 6 bit Dyadic Dithering oscillations

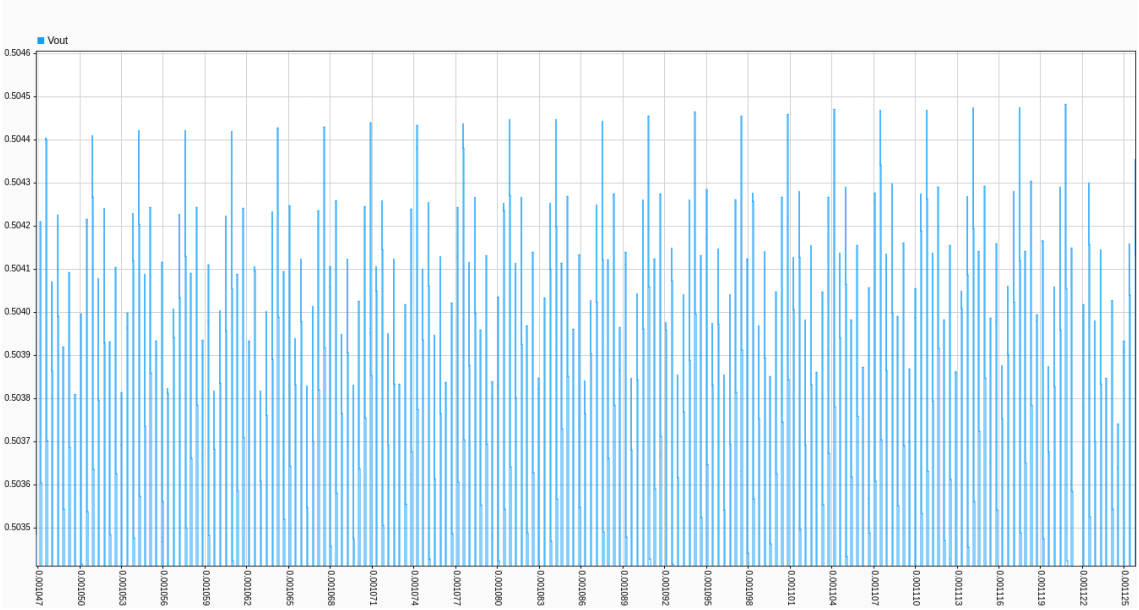



Figure 4.46: 0.5V, 6 bit Dyadic Dithering zoom on oscillations

Concerning 3.3V of regulated voltage, the results are better, in fact the steady-state condition (2.2.2) is satisfied. In fig.4.47 the simulation shows that the error voltage is stably at zero, but contrarily to the cases with a lower number of bit, the oscillations are visible. In fig.4.48 and 4.49 a better view is proposed. Their amplitude is approximately 550 µV, so the same as thermometric ones. Once again it can be noted that this value is not so negligible with respect to the ADC resolution, it's a significant contribution.

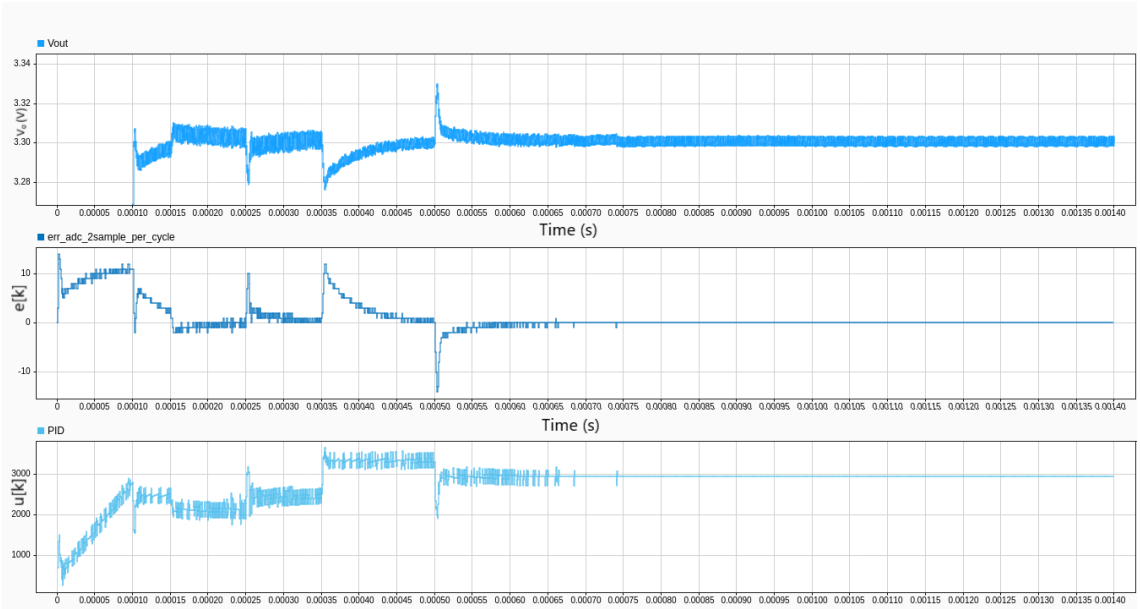



Figure 4.47: 3.3V, 6 bit Dyadic Dithering


Figure 4.48: 3.3V, 6 bit Dyadic Dithering oscillations

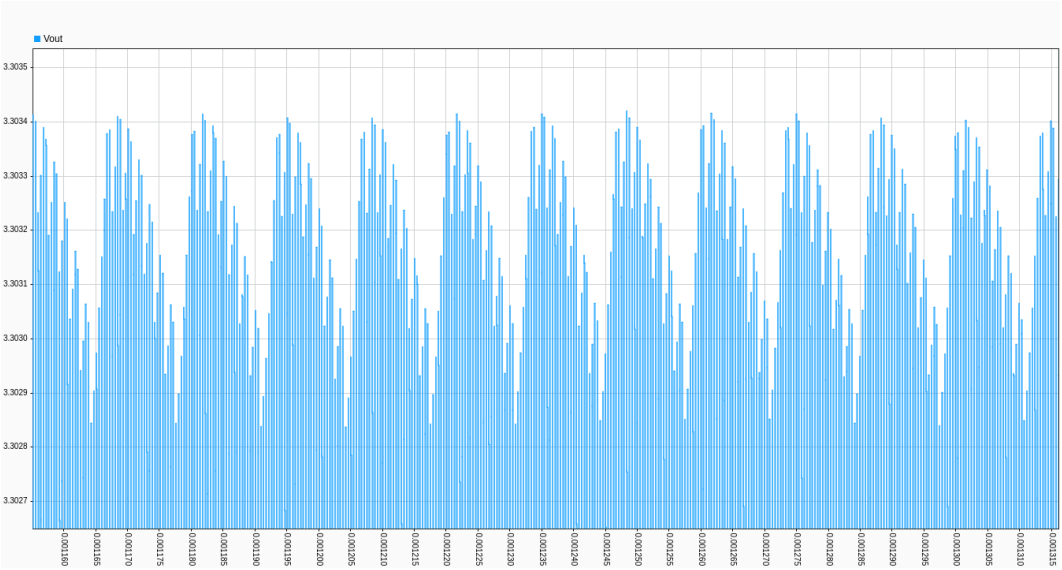



Figure 4.49: 3.3V, 6 bit Dyadic Dithering zoom on oscillations

Just to emphasise the problem, the 7 bit depth case is reported. In the fig.4.50, 4.51, 4.52 and 4.53 it's clear how the regulated voltage is out of control. The error assumes non-zero values in a chaotic manner and the amplitude of these oscillations reach voltage higher than 5 mV, that it is not acceptable at all.

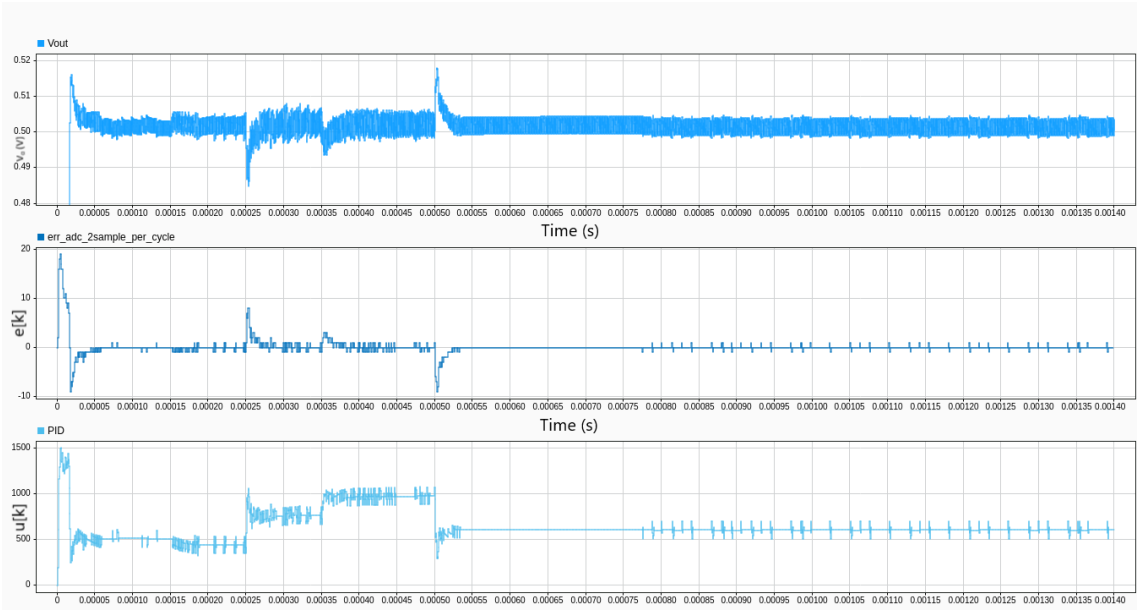



Figure 4.50: 0.5V, 7 bit Dyadic Dithering

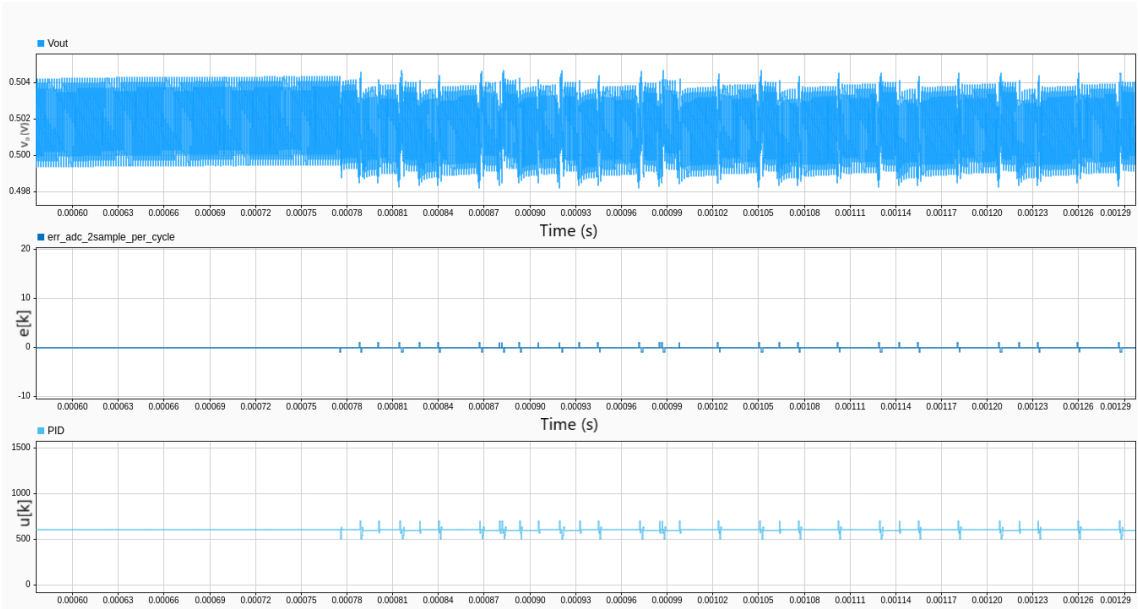



Figure 4.51: 0.5V, 7 bit Dyadic Dithering oscillations

| | Dyadic | Dyadic | Thermometric |
|---|---|---|---|
| num. of bits | $\Delta V_{0.5V}$ | $\Delta V_{3.3V}$ | $\Delta V_{0.5V}$ |
| 3 | $10\mu V$ | $50\mu V$ | |
| 4 | $75\mu V$ | $100\mu V$ | $600\mu V$ |
| 5 | $200\mu V$ | $200\mu V$ | |
| 6 | $600\mu V$ | $550\mu V$ | |
| 7 | $6mV$ | $7mV$ | |

Table 4.1: Dithering comparison

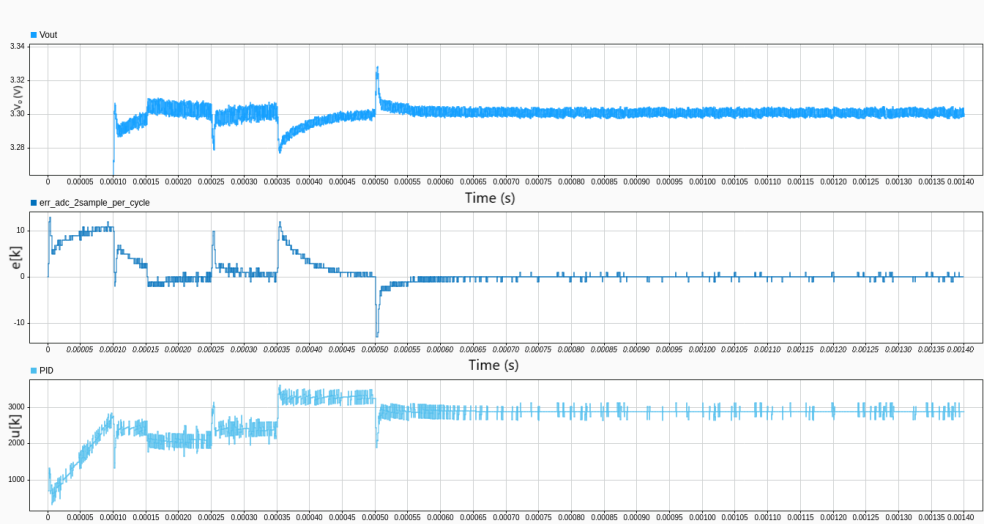



Figure 4.52: 3.3V, 7 bit Dyadic Dithering

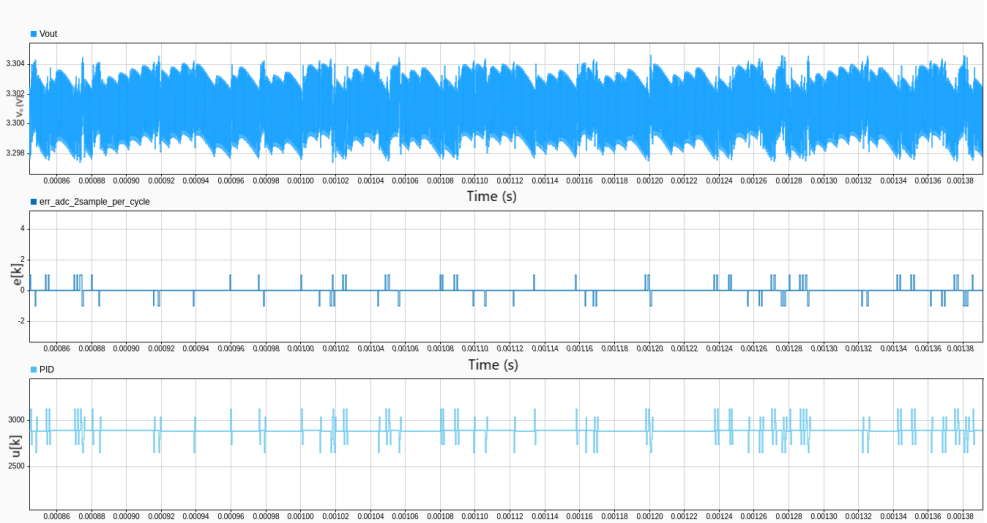



Figure 4.53: 3.3V, 7 bit Dyadic Dithering oscillations

The table 4.1 summarises the results of this analysis, showing the ripple amplitude found during the simulations.

In order to choose a feasible number of bits, a maximum oscillation amplitude has to be selected. A reasonable value can be a tenth of the main contribution of the static

regulation, i.e. the ADC resolution that is $1.875mV$. Examining the results a 4 bit dyadic dithering seems to be affordable. Now, comparing the dyadic and the thermometric dithering is clear that the first offers advantages in respect to the other: the amplitude of the oscillations. With the same dithering depth, i.e. 4 bits, the dyadic shows a ripple 6 times lower than the alternative technique. The phenomenon can be studied also in the frequency domain. A spectrum of the thermometric method has been already proposed, in fig.4.54 that of the dyadic technique is reported. Once again a peak at 2.4 MHz with its harmonics come from the switching action of the Buck converter and at lower frequency the dithering peak emerges at 150 kHz. The difference between this case and the thermometric one, is that the peak due to dithering actions has a lower amplitude. Its power is $-70\,dBm$ that is $15\,dB$ lower than before. This result is consistent with the time domain simulations in which the oscillation of the dyadic technique presents an amplitude (100 µV) 6 times smaller than the thermometric (600 µV). Finally, taking in consideration all the analysis done, the dyadic dithering is preferred to the thermometric, both in term of in term of band occupation and amplitude of oscillation, i.e. static regulation.

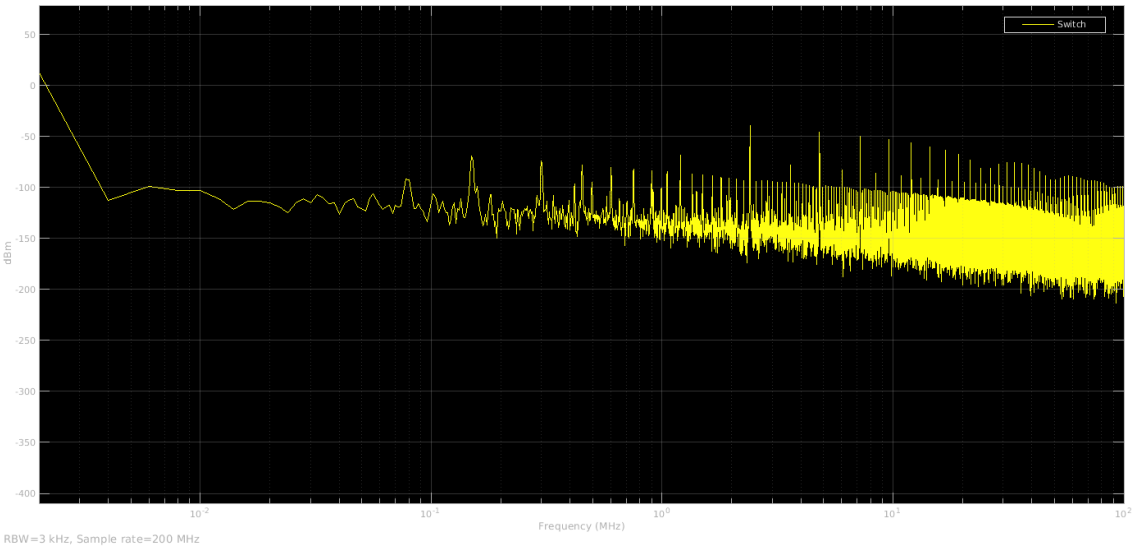



Figure 4.54: Dyadic Dithering spectral envelope

### 4.2.2 Sigma-Delta (ΣΔ)

In this section a brief review of the ΣΔ modulation is proposed without entering in the mathematical details. It is just an overview of the alternative technique used to achieve higher resolutions, similarly to the dithering techniques. Indeed this type of modulator permits through a negative feedback to obtain a more precise value of the duty cycle, averaged on multiple PWM periods. The principle is the same of the dithering but in this case there is not a periodic pattern of the duty cycle, it changes in a chaotic way till it reaches an average steady-state condition.

After that, a comparison between dithering and ΣΔ modulations is proposed in order to choose the best design option. In the analysis both the first and the second order ΣΔ are used and simulated in the Simulink model, then, in order to conclude the comparison with the dyadic dithering technique, the cases with 5, 6 and 7 bits of modulation are proposed.

Digital pulse width modulators (DPWM) built on multi-bit sigma-delta are used in the management of DC-DC converters to obtain high resolution and thus high output voltage accuracy at switching rates up to multiple MHz improving low-area and low-power design in comparison to pulse width modulators that rely only on a basic counter[10][12]. The sigma-delta modulator and the core DPWM (counter-based) make up the two stages of the traditional sigma-delta based DPWM. The modulator consists of a loop filter $H(z)$ and a B-bit quantizer, which may be nothing more than a truncator that eliminates a number of B LSBs. The filter has a transfer function of $H(z) = 1 - NTF(z)$ where NTF is the transfer function of the modulator, and it handles the quantization noise, which is the disparity between the quantizer's input and output. The quantization noise is shaped by the filter so that the majority of the noise power is transferred to high frequencies. At the cost of added intricacy, higher order modulators shift more quantization noise to higher frequencies. The equations proposed above, indicates the first and second order modulator transfer functions:

$$H_1(z) = 1 - z^{-1} \tag{4.4}$$

$$H_2(z) = (1 - z^{-1})^2 \tag{4.5}$$

More in detail, the structure of a $\Sigma\Delta$ modulator can be represented as in fig.4.55. This modulator is based on a negative feedback, it means that if the loop gain is high, the output value will be equal to the input because the error $x(n) - y(n)$ will be very small.

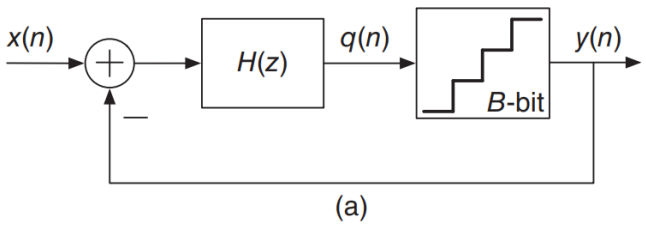



Figure 4.55: Basic structure of a first order Sigma-Delta modulator[12]

Sigma-delta modulators produce low frequency tones at specific duty cycles, which can cause significant output voltage oscillations if their frequency corresponds with the converter's corner frequency. This is a drawback of conventional sigma-delta DPWMs. A sigma-delta based DPWM produces an idle tone at the minimum frequency of:

$$f_{mintone} = \frac{f_{SW}}{2^{(N_H - N_{CORE})}} \tag{4.6}$$

where $N_H$ is the resolution of the total duty cycle and $N_{CORE}$ is the resolution in input just to the core , so less precise. These idle tones occur when the converter is in steady state that can be significantly reduced by using a second or third order rather than a first order sigma-delta modulator.

### First-order ΣΔ

The structure of a first-order ΣΔ modulator is the same of 4.55, in which a single integrator block is interposed between the input and the truncator. In Simulink its representation is very simple, its structure is attached in fig.4.56. The output is subtracted to the input ("PID" in the figure) to realize the negative feedback, then the integrator is implemented in the same way of the PID, i.e. with an adder that accumulate the past values through a delay block. Finally the truncator extracts the first 11 bits (eliminating the remaining LSBs) obtaining the output. The modulator acts on 4 bits, indeed as one can see from 4.56 the word coming from the PID, i.e. the input, is on 15 unsigned bits on fixed-point representation, instead the output is on 11 unsigned bits on fixed-point representation.

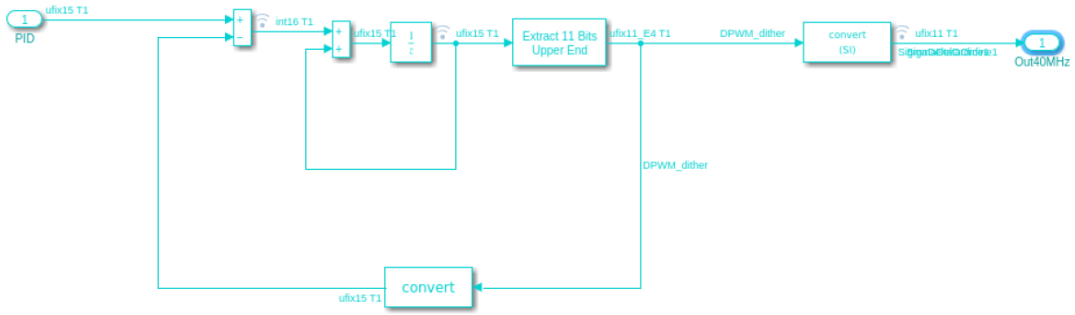



Figure 4.56: First order modulator, Simulink block diagram

The simulations are carried for both the 0.5V and the 3.3V cases. For the first one the simulation is reported in fig.4.57, fig4.58 and fig.4.59.

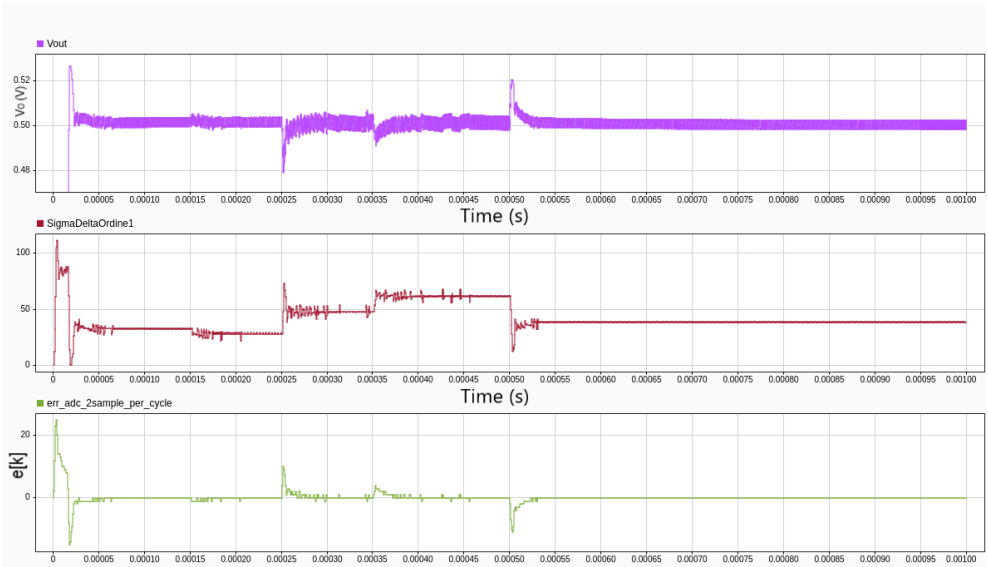



Figure 4.57

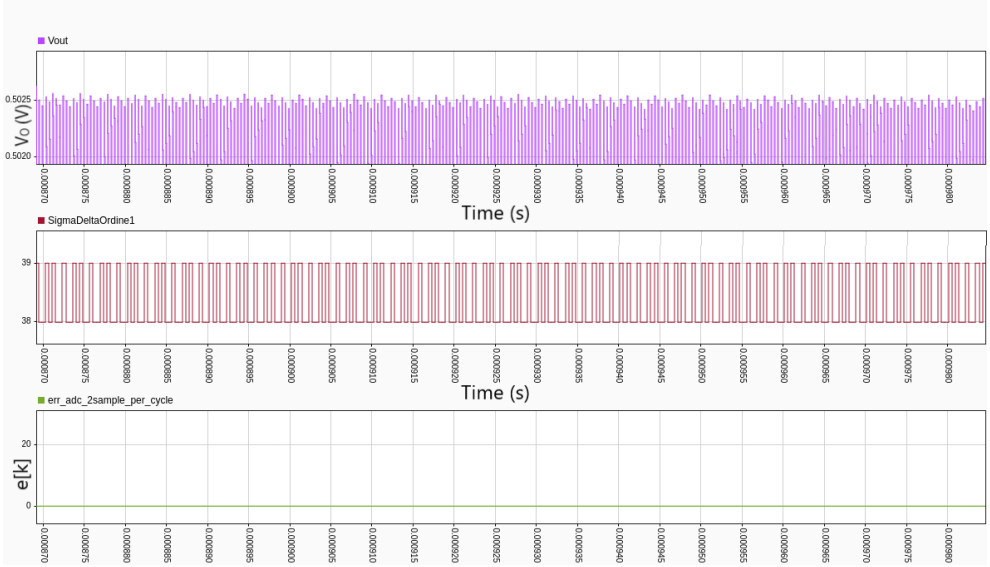



Figure 4.58

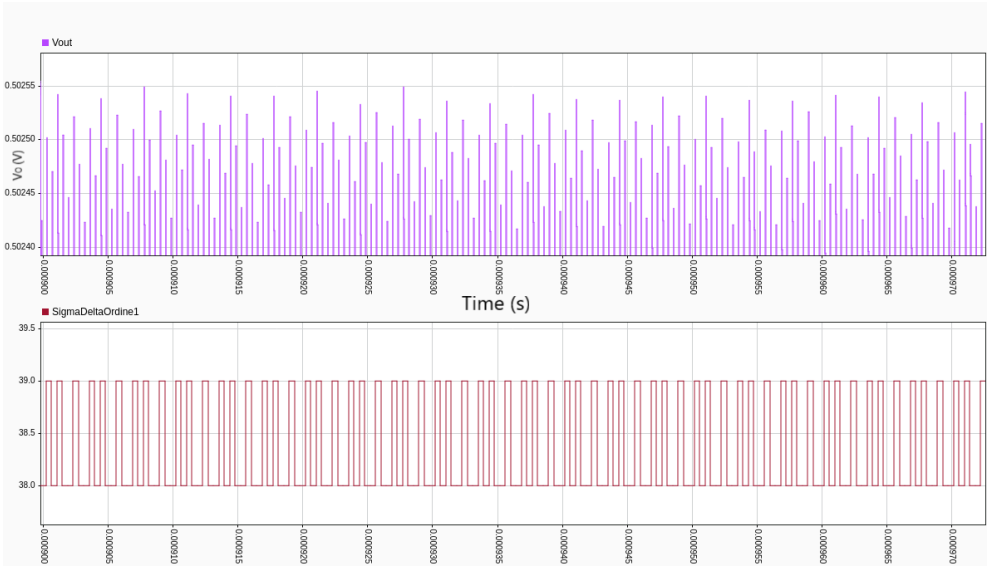



Figure 4.59: First-order modulator, Simulink simulation, 0.5V of regulated voltage

The simulation shows a well defined steady-state, with small ripple. The proposed method seems to be a valid alternative to the dyadic dithering, indeed the amplitude of the oscillations coming from the $\Sigma\Delta$ modulation are more or less of 150 mV, a little bit higher than the dyadic but nevertheless small. These fluctuations can be appreciated also in the frequency-domain. In fig.4.60 the spectrum of the simulation can be appreciated. This figure is very similar to that observed for dyadic dithering: the dominant component is found around 150kHz, i.e. $\frac{f_{SW}}{2^{N_{bit}}} = \frac{2.4MHz}{16} = 150\,\text{kHz}$ (eq.4.6), but other high-frequency components are even more powerful. They can be seen also on the time-domain simulation, in which is clear how the regulated voltage is more nervous compared to the dyadic one. In the latter's spectrum is dominant the component at 150 kHz, but the high-frequencies ones decay, they present less power than the dominant.

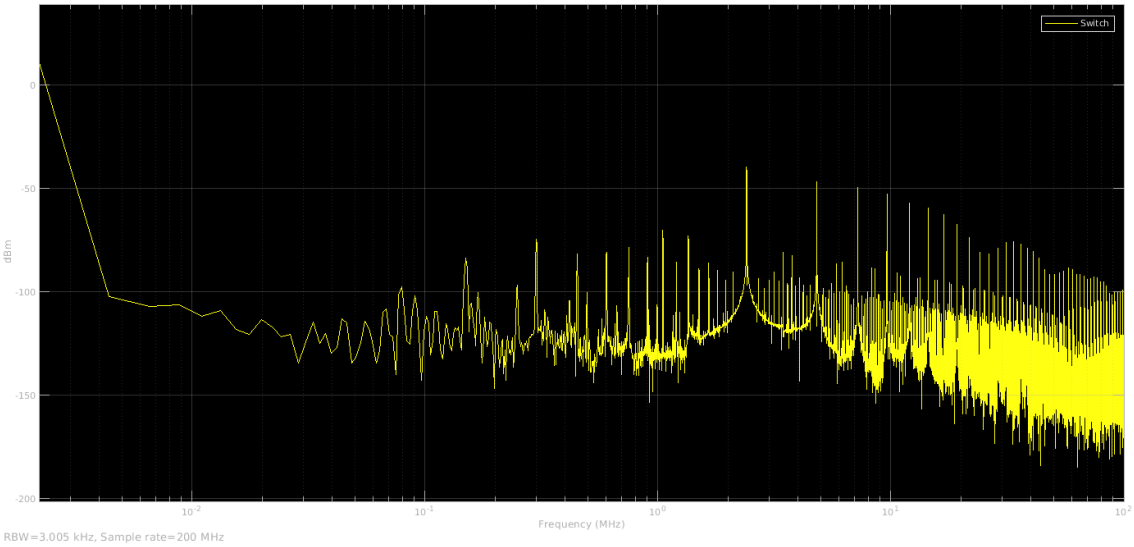



Figure 4.60: Spectrum of the first-order ΣΔ

Regarding the dynamic performance, neglecting the soft start, the peak of the responses doesn't exceed 20 mV. In the dyadic dithering the peak was a bit smaller than 20 mV, so the overall performances are worse by little, but still valid.

For the 3.3V case the simulations are proposed in fig.4.61, fig.4.62 and fig.4.63.

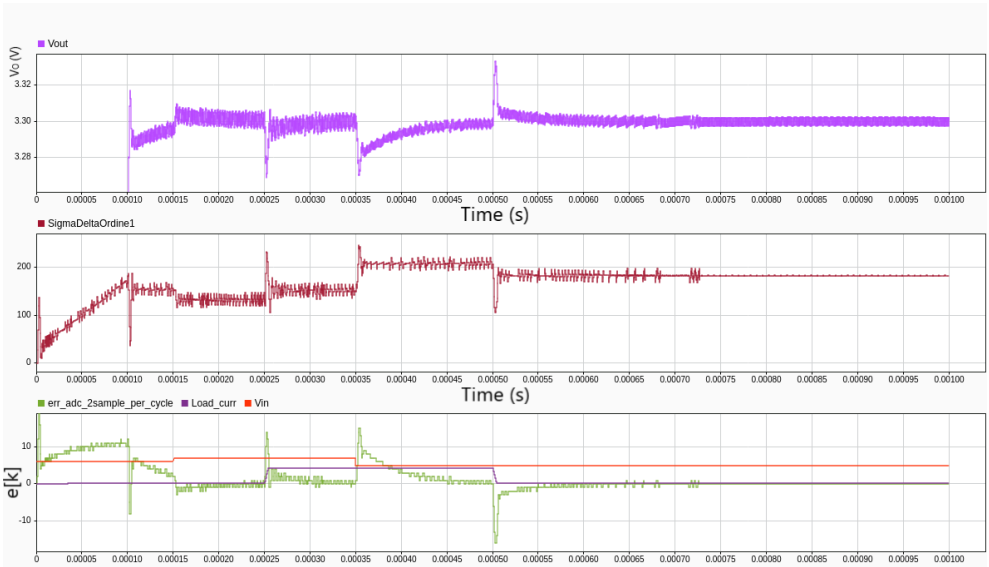



Figure 4.61

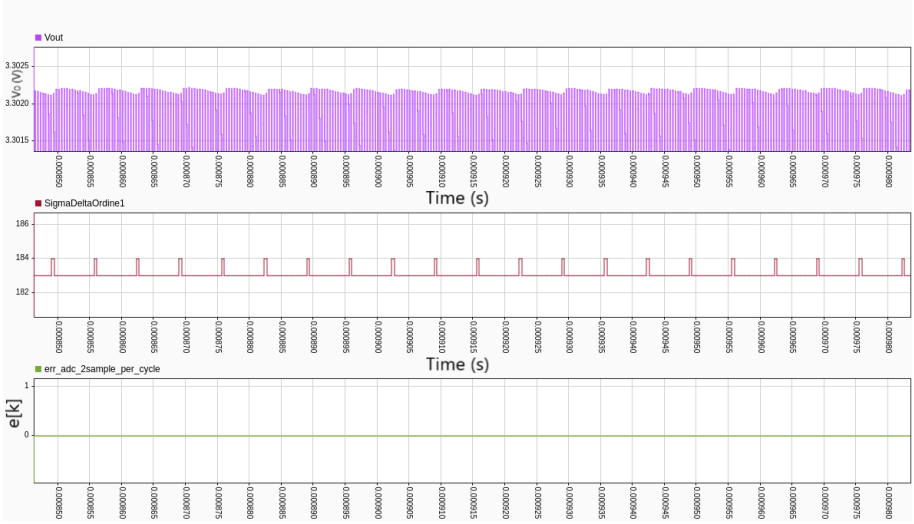



Figure 4.62

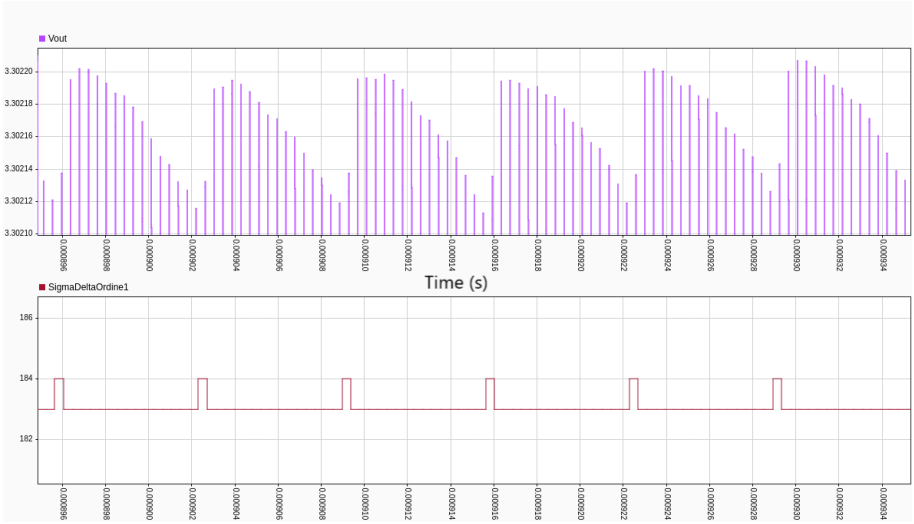



Figure 4.63: First-order modulator, Simulink simulation, 0.5V of regulated voltage

From the graphs it is clear that the method works well but a kind of oscillation, i.e. idle tone, appear in steady-state. Applying a zoom it is easier to appreciate them. They appear in a periodic manner and their amplitude is 100 µV. The dynamic performance is 30 mV of peak response, so totally comparable to the dyadic dithering.

### Second-order ΣΔ

As already mentioned the second-order is used to attenuate the idle tones coming from the modulation. There are various types of structure that can be implemented in the loop, in this study the structure called "Noise–Coupled Structure" is proposed, its block diagram together with its transfer function is shown in fig.4.64. It is composed by a first order modulator in the form of error-feedback and a second external loop that introduces the second integral action.

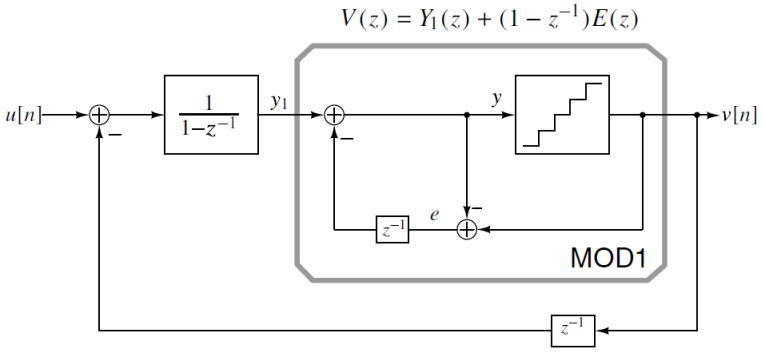



Figure 4.64: Second-order ΣΔ modulator: Noise-Coupled Structure[12]

Also in this case the simulations are performed for both voltages. Starting with 0.5V, the simulations are reported in fig.4.65, fig.4.66 and fig.4.67.

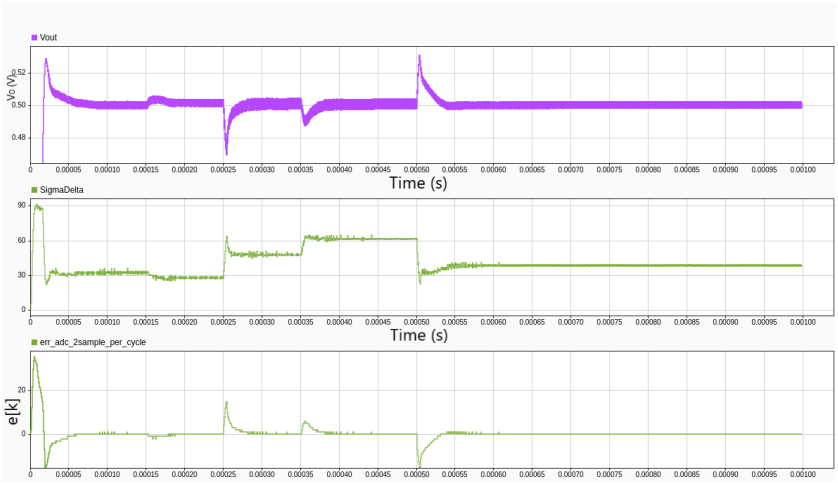



Figure 4.65


Figure 4.66

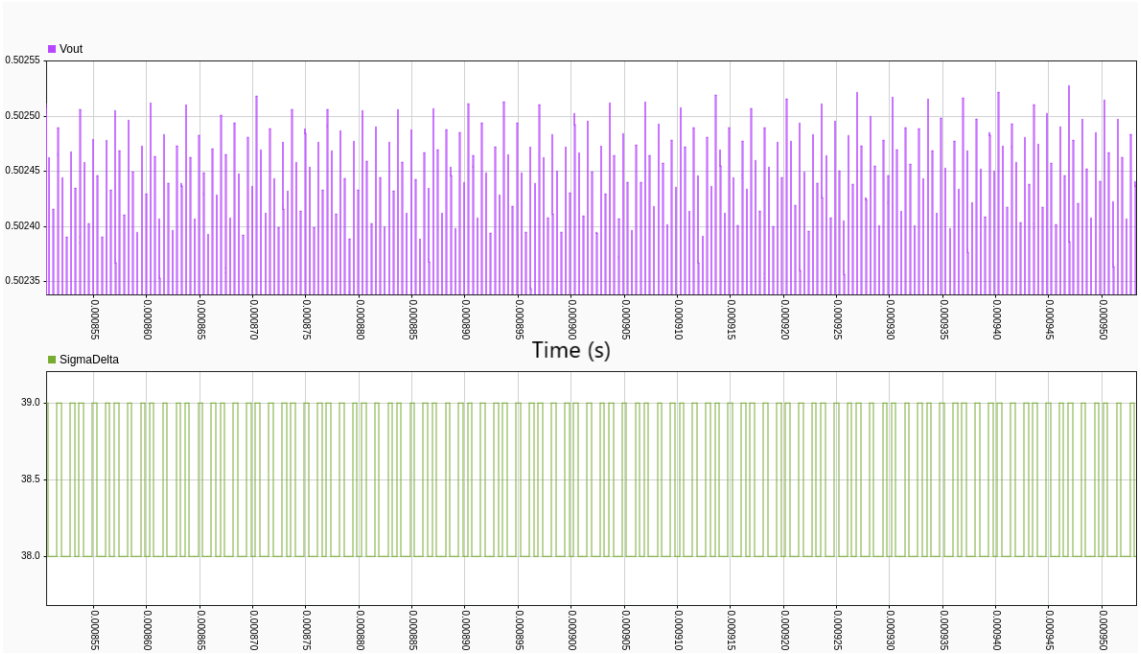



Figure 4.67: Second-order modulator, Simulink simulation, 0.5V of regulated voltage

The static regulation is more or less equal to that of dyadic dithering, the ripple's amplitude is 100 µV. In the frequency-domain, as represented in fig.4.68, the situation seems to be better than both the dyadic and first-order ΣΔ techniques, the component at 150 kHz is more attenuated. The low-frequency components present low power, so the second-order modulator is able to push all the ripple at higher frequencies.

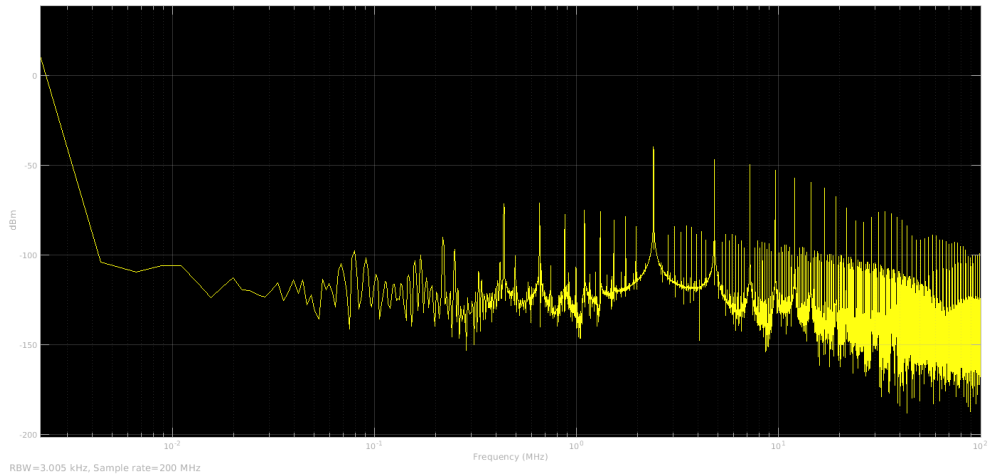



Figure 4.68: Spectrum of the second-order ΣΔ

The main drawback is the dynamic regulation, in fact the peak of the load and line response is 30 mV, so higher than the previous cases (more or less 20 mV of peak). For what concern the 3.3V of regulated voltage the simulations are attached in fig.4.69, fig.4.70 and fig.4.71

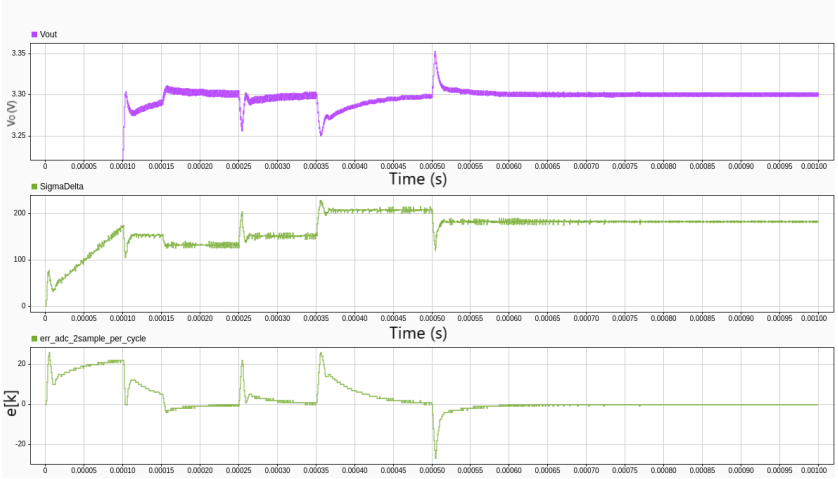



Figure 4.69

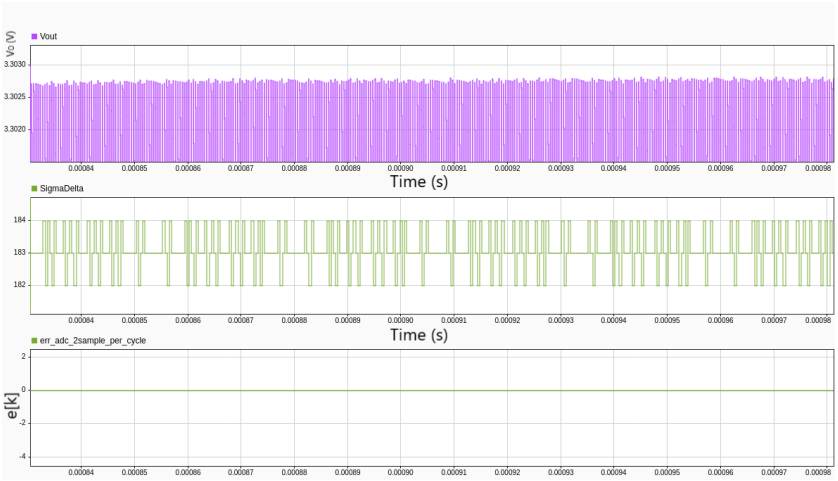



Figure 4.70

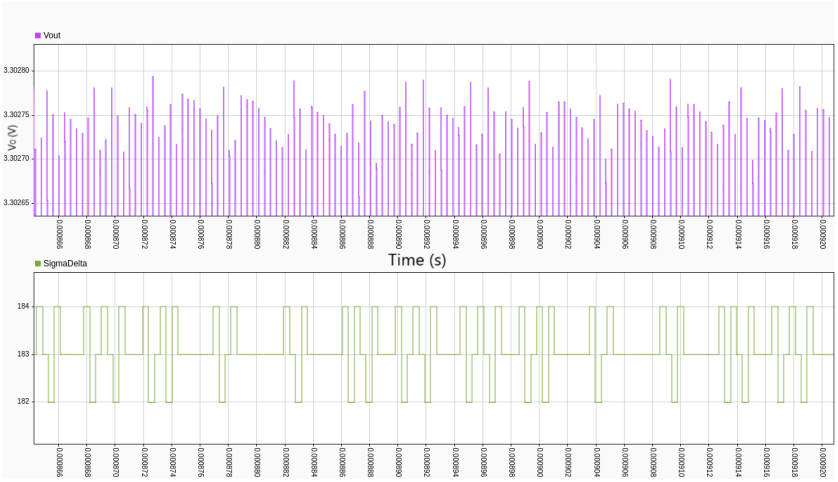



Figure 4.71: Second-order modulator, Simulink simulation, 3.3V of regulated voltage

Also in this case the loop regulates pretty well, the ripple's amplitude is again 100 µV. As happened in the case of 0.5V, the step response to the dynamic variations are worse

by far than both the dyadic dithering and the first-order ΣΔ modulator. Indeed the peak is now 50 mV, that is more than the 50% higher compared to the others techniques.

Finally, the cases with 5, 6 and 7 bits of modulation are reported below. The simulations are carried just for the 3.3V of regulated voltage. In fig.4.72 and fig.4.73 the 5 bits modulator is proposed. The first image shows how the regulation is reached, the error goes to zero in a stable way. The drawback also in this case is the dynamic performances, the responses are around 50 mV. The second image shows the ripple amplitude, that reaches 200 µV and the ΣΔ modulation pattern.

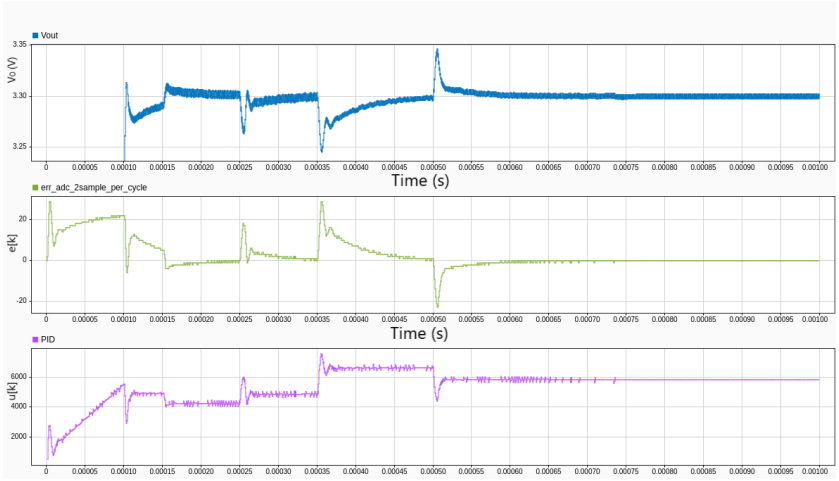



Figure 4.72: Second-order modulator, Simulink simulation, 3.3V of regulated voltage and 5 bits of modulation

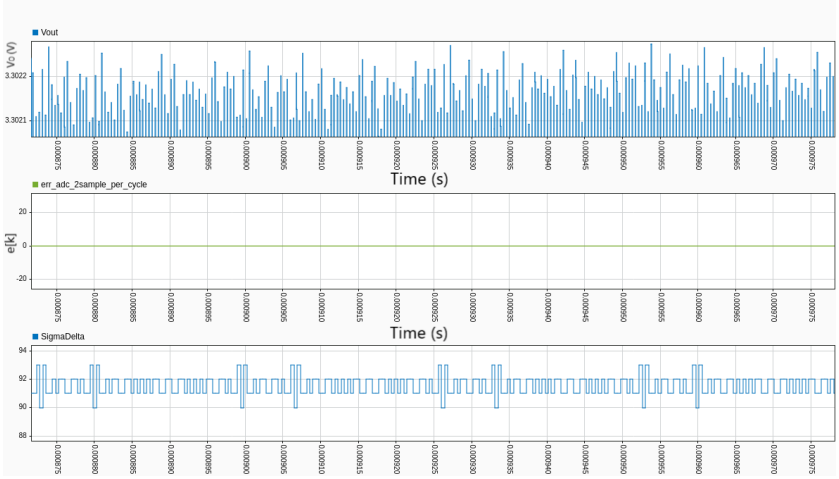



Figure 4.73: Second-order modulator, Simulink simulation, 3.3V of regulated voltage and 5 bits of modulation, zoom on the ripple

In case of 6 bits of modulation the situation is more or less the same, as it is possible to see in fig.4.74 and fig.4.75. So, also in this case the loop works well, following the reference successfully, with the error stably to zero, but, again, the performances are not better than the dyadic dithering. The second figure reports the ripple voltage and the pattern. The amplitude of the ripple seems to be lower then the previous case, showing just 100 µV, but

this is just a random event. The pattern of the modulator, indeed, is not impacting in a heavy manner on the loop compared with the fig.4.73 in which the modulator modify its output more frequently. In fact the ripple amplitude is function of the particular pattern provided by the modulator. This happens for the ΣΔ modulator as well as for the dyadic dithering.

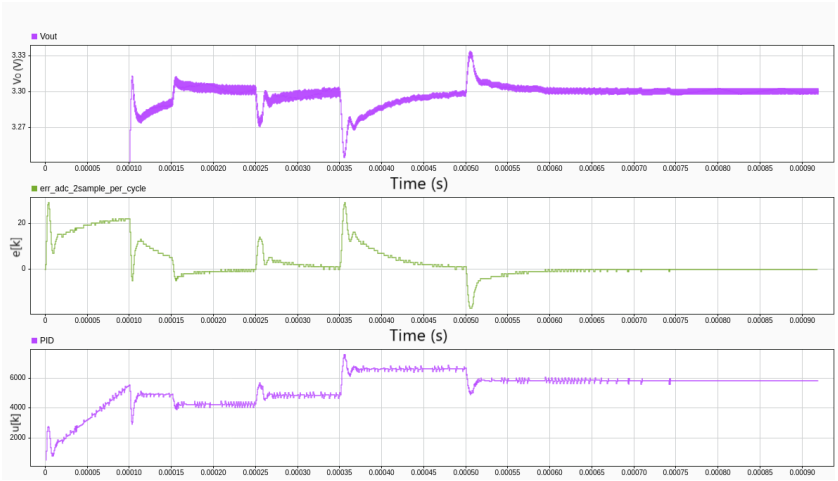



Figure 4.74: Second-order modulator, Simulink simulation, 3.3V of regulated voltage and 6 bits of modulation

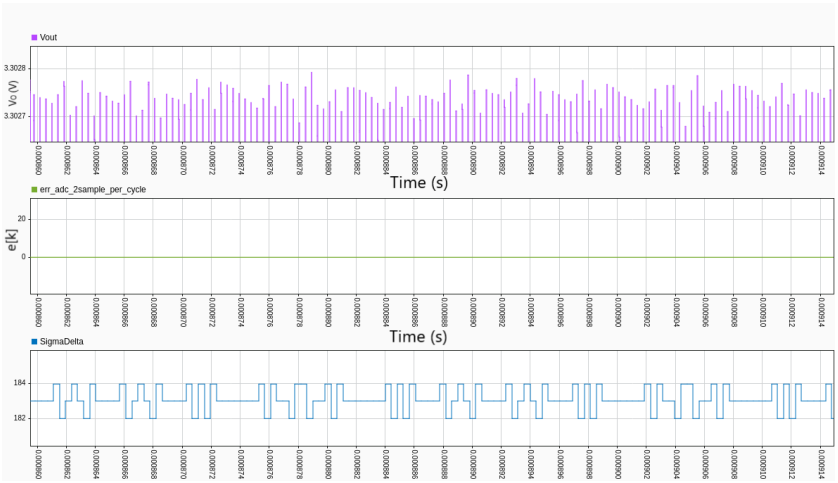



Figure 4.75: Second-order modulator, Simulink simulation, 3.3V of regulated voltage and 6 bits of modulation, zoom on the ripple

The last simulation, proposed in fig.4.76, regards the 7 bits of modulation. It is clear that the regulation fails, showing great oscillations at low frequency. Their amplitude is of several mV, so much higher than the other cases. Actually, it is the same behaviour of dyadic dithering, whose simulations shows high oscillation for 7 bits. For this reason the comparison wil be done only till 6 bits of modulation.

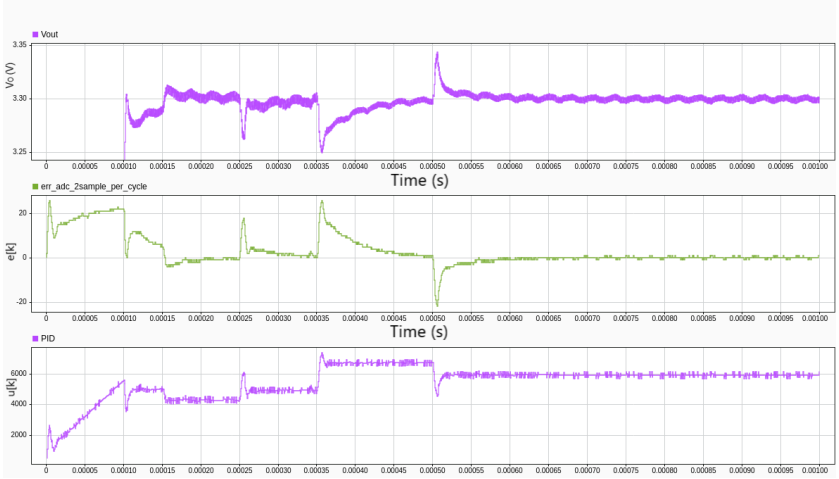



Figure 4.76: Second-order modulator, Simulink simulation, 3.3V of regulated voltage and 7 bits of modulation

### 4.2.3 Design Choice

At this point, a final analysis has to be made in order to make a decision on the right technique to implement. The comparison will be done fixing the number of modulator's bits to 4 that, as already said in the dyadic dithering section, is a good trade-off between ripple oscillation and hardware complexity and neglecting the thermometric dithering, already excluded in the previous sections. For what concern the others, all show interesting features and point of weakness:

- **Static Regulation**: the dyadic dithering and the second-order $\Sigma\Delta$ modulator present the best performance, both show a ripple of 100 µV. On the other hand the first-order is characterised by a behaviour more nervous and a ripple up to 150 µV;
- **Dynamic Regulation**: the dyadic dithering and the first-order modulator have the fastest response with a peak of 30 mV facing a load regulation. The second-order modulator seems to be delayed compared to the other, its response to the same load regulation is 50 mV and this happens for 4 bits, but also for 5 and 6 bits. This difference can be attribute to the addition of another integrator in the modulator, it slow down the response because it introduces an additional phase delay to the transfer function of the loop.
- **Output spectrum**: in the frequency domain the three techniques behave differently, indeed, neglecting the switching frequency and its harmonics, the dyadic one presents a low-frequency component exactly at 150 kHz (one sixteenth of $f_{SW}$) but its harmonics have lower amplitude; the first-order modulator has the same low-frequency component but its harmonics shows higher amplitude, it means that the modulation ripple oscillates faster, carrying power at higher frequency; the last technique, the second-order modulator presents the best frequency conduct, indeed the

low-frequency components are particularly weak, all the ripple coming from the modulation action is pushed at higher frequency, near the switching one. The fig.4.77, proposes a comparison between all the spectra.

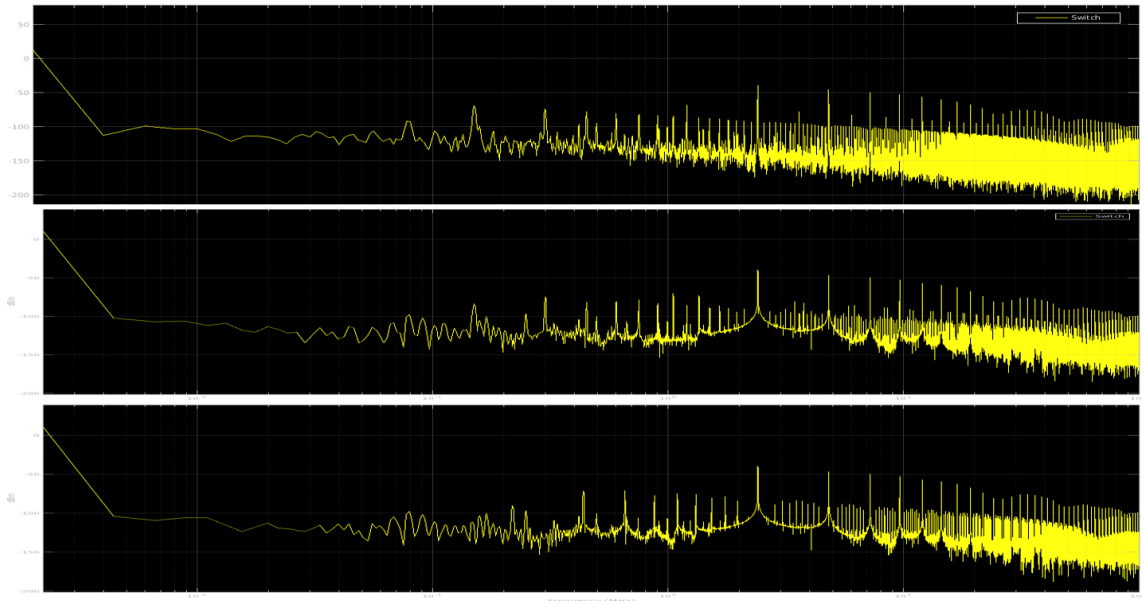



Figure 4.77: Spectral comparison of all proposed techniques. Respectively: Dyadic Dithering, First-order Sigma-Delta and Second-order Sigma-Delta

In conclusion, the Dyadic Dithering seems to be the most affordable choice: the simulations shows the best trade-off between regulation performances and frequency behaviour. At this point, once that the dyadic dithering is selected to be inserted in the thesis loop, the system is simulated in more general cases, in which the input line voltage and the load current are changed with smaller steps, in order to verify that all these conditions are stable. Starting with the line voltage variations, in fig.4.78 is proposed the simulation for the 3.3 V case and in 4.79 for the 0.5 V. In both cases the loop follows the reference very well, the stable condition is reached every time. Obviously the peaks of the dynamic regulation are smaller because the line voltage steps are smaller too.

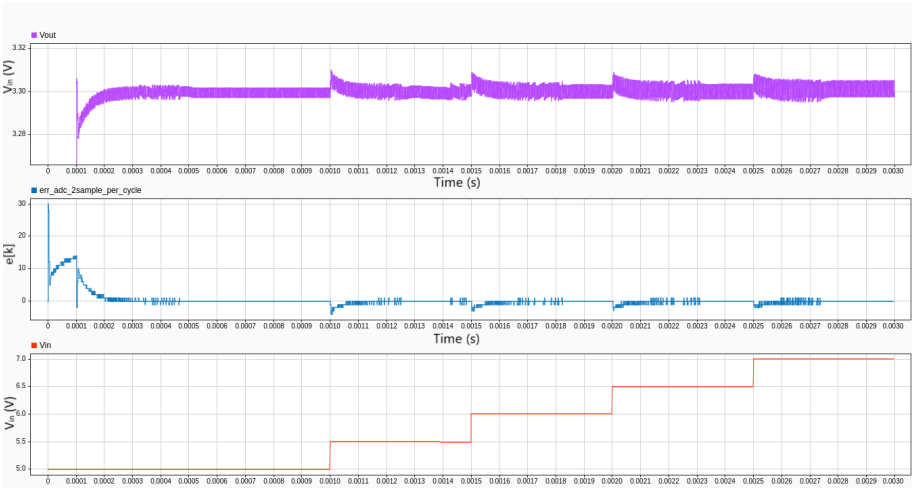



Figure 4.78: Line voltage regulation, 3.3V


Figure 4.79: Line voltage regulation, 0.5V

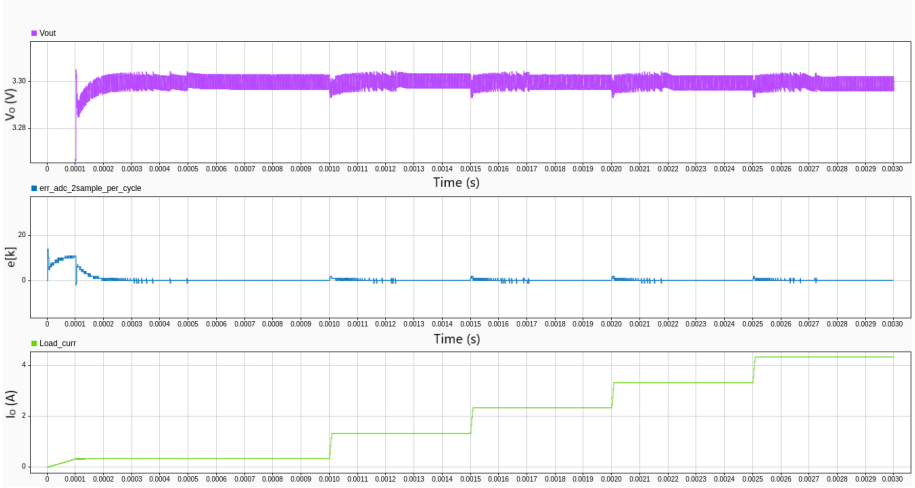



Figure 4.80: Load current regulation, 3.3V


Figure 4.81: Load current regulation, 0.5V

The last simulations of this section regards the load current variations, in fig.4.78 is pro-

posed the simulation for the 3.3 V case and in 4.79 for the 0.5 V. All the considerations made on the line regulation can be applied to the load current one. The loop works in all the simulated cases.

# Chapter 5

# Delay Line: Transistor-level Design

In this chapter a complete overview of the *Delay Line* technique is provided. Essentially, this overview is structured in three parts: the first one will cover the topology description, focusing on the characteristics of this approach, how it works in the DC-DC converter's loop and in particular how it can cooperate with other techniques to obtain a better resolution. The second one will be devoted to the description of the transistor-level design of the delay line used in this thesis project in Cadence Virtuoso environment. The third part will provide simulations of the circuit and a description of its performance in various conditions.

## 5.1 Delay Line Principles

In the previous chapter the novel *Dyadic Dithering* technique was proposed to control the average duty cycle with a sub-clock resolution. This is not the only one, in this chapter is indeed presented the *Delay Line* technique. The basic goal of these systems is to replace the very high-frequency clock with a tapped string of delay cells, known as a delay line, to achieve high-resolution time quantization. In fig.5.1 a basic structure of a delay line is attached.

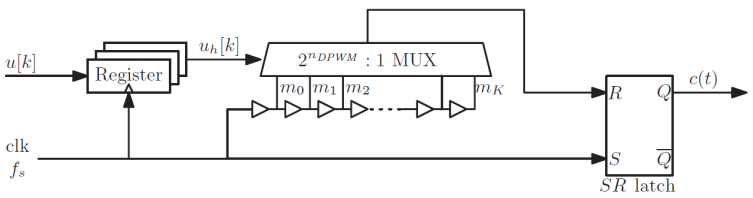



Figure 5.1: Delay Line basic structure[3]

The output latch is set at the beginning of a switching period by a clock signal with a frequency equal to the switching frequency, $f_{clk} = f_s$. The output at tap $m_k$ is delayed with respect to the output at tap $m_k - 1$ by a cell delay of $\Delta t_{DPWM} = t_c$ thanks to the

propagation of the identical clock signal across a delay line. The output signal is reset using a digital $2^{n_{DPWM}} : 1$ multiplexer by the latched digital control command $u_h$ chooses that tap. In fig.5.2 the basic idea can be better appreciated, it is clear that a cell delay ($t_c$) rather than the clock period ($T_{clk}$) determines the time resolution in the output signal. As a result, the switching frequency and the clock frequency are equal.

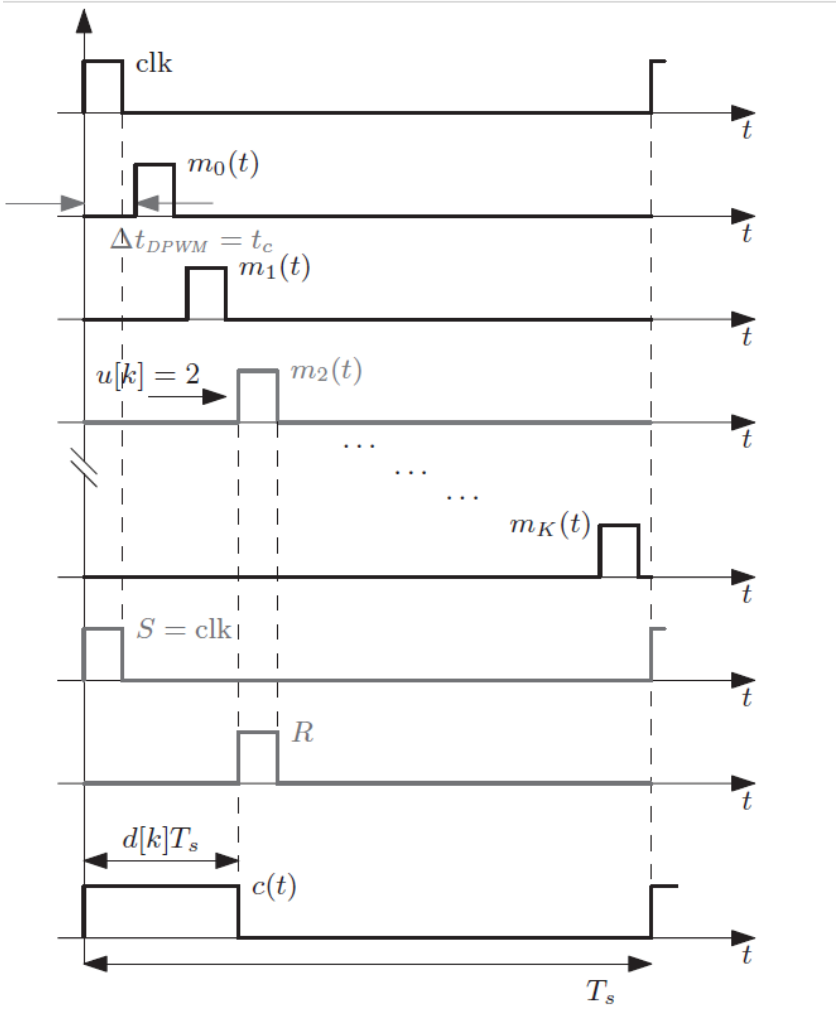



Figure 5.2: Delay Line basic idea[3]

Unfortunately, the overall delay may be either too short or too long in comparison to the intended switching period ($T_s$), for example due to variations on the environmental conditions, such as temperature, and changing the switching frequency. These issues are practical concerns. The problem can be solved in two ways:

- close the delay line into a self-oscillating ring that serves as a clock generator;
- exploit a delay-locked loop (DLL) that follows the clock period regulating $t_c$

Another drawback is that the length of the delay line and the dimension of the output multiplexer grows exponentially with the number of bits $n_{DPWM}$. For this reason DPWM

usually uses hybrid structures, i.e. a structure where the counter and delay-line techniques are combined, as showed in fig.5.3.

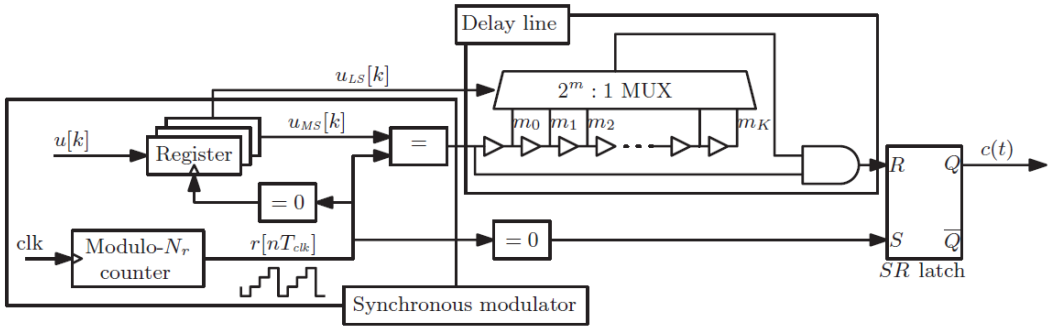



Figure 5.3: Hybrid Delay Line basic structure[3]

The control signal $u[k]$ has $n_{DPWM}$ bits, but the m MSBs are used for the counter-based part (synchronous modulator) and the remaining $n_{DPWM} - m$ for the multiplexer selection of the delay-line. In practise when the counter is initiated to zero every switching period, the output latch of the PWM signal is set, then the counter reaches the value that correspond to the m MSBs of the control signal $u[k]$ and a pulse is generated and used, instead as reset of the latch, as input of the delay line that further delays it by a number of cell corresponding to the value in input at the multiplexer selection. So the resulting duty cycle is given by:

$$d[k] = (u_{MSB} + \frac{u_{LSB}}{2^{n_{DPWM-m}}})\frac{T_{clk}}{T_s} \tag{5.1}$$

with $u_{MSB}$ the value provided to the counter, given by the MSBs coming from the digital compensator, and $u_{LSB}$ the value provided to the delay-line, given by the LSBs coming from the digital compensator. The behaviour of the hybrid structure can be better appreciated in fig.5.4.

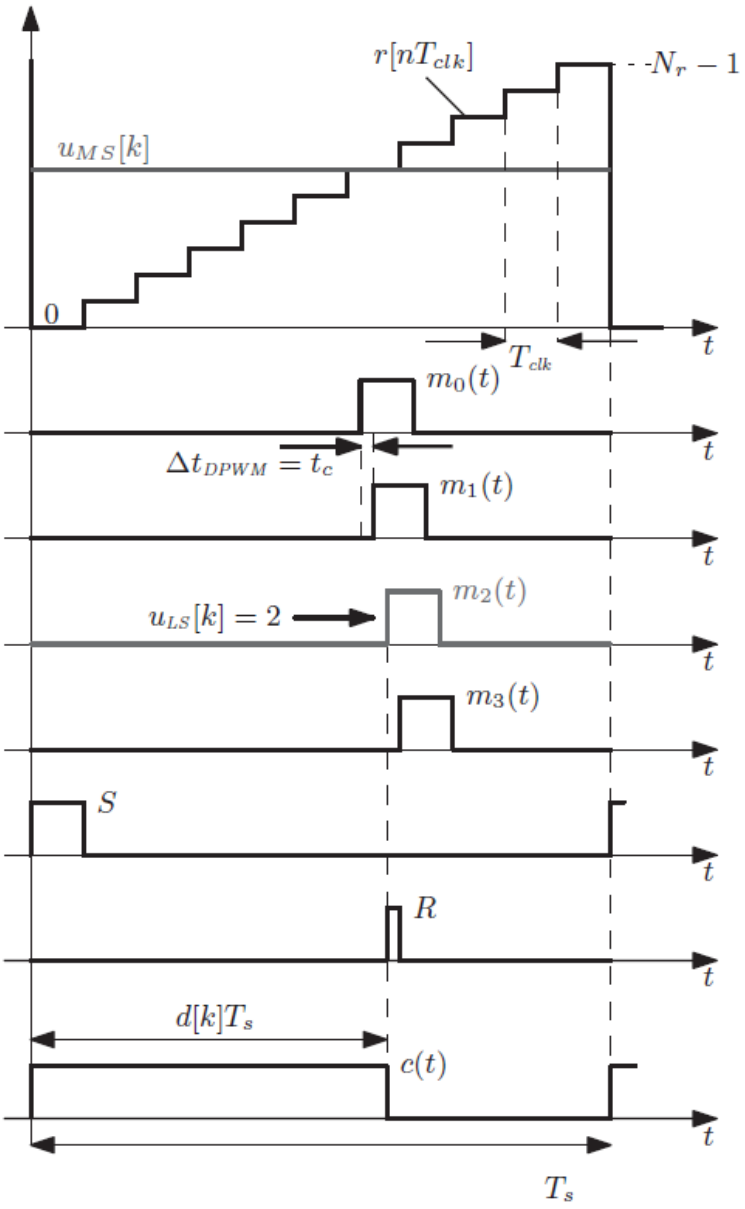



Figure 5.4: Hybrid Delay Line basic idea[3]

It is clear that there must be a trade-off between the delay-line DPWM and the counter-based DPWM or in other words, the clock rate. A larger $n_{DPWM} - m$ implies a longer delay line and a larger multiplexer, while the required clock rate is reduced. If the clock frequency is fixed, then the trade-off is focused between the number of cell delays instantiated in the delay line and the area occupied, power consumption and timing constraints. These problems suggests that the resolution of DPWM has a technology limit, make it as long as one wishes is impossible. This thesis project aims to break this limit combining three elements to define the DPWM resolution.

## 5.2 Hybrid DPWM combined with Dithering technique

As the title of this section may suggest, the innovative DPWM structure proposed in this thesis project is composed by an Hybrid modulator that cooperates with a Dithering

modulator, already examined in the previous chapters. The structure is attached in fig.5.5. The block diagram shows a first dithering modulation, already seen in fig.4.25, whose output is used as input for the Hybrid DPWM, that in turn is composed by a first counter-based modulator and a successive delay line which improve further the resolution.

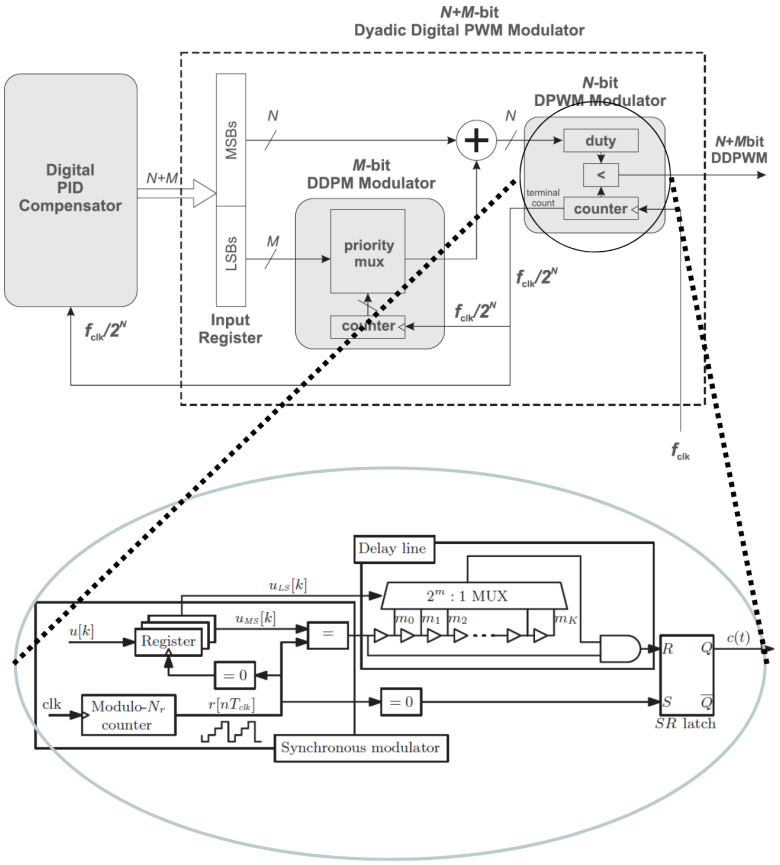



Figure 5.5: Innovative Hybrid DPWM structure[3][4]

In practise the word coming from the PID is divided in three sub-words, which are used for each techniques. In the thesis project the needed DPWM resolution is of 100ps, knowing the switching period $T_s \approx 416.7\,\text{ns}$, the number of the equivalent steps is $\frac{416.7\,\text{ns}}{100\,\text{ps}} = 4167$ that is a little more of $2^{n_D PWM} = 2^{12} = 4096$, the remaining steps are incorporated into the counter-based DPWM's count. For this reason the word coming from the PID can be divided in 4 bit to the dithering, 4 bit to the delay line and the remaining bits to the counter. In particular the delay line needs 16 delay cells that decompose the clock period $T_{CLK} = 25\,\text{ns}$ in 16 taps of $T_{delaytap} = T_c = \frac{25\,\text{ns}}{16} = 1.56\,\text{ns} \approx 1.6\,\text{ns}$. The contribution of the dithering further improves the performance obtaining an equivalent steady-state resolution of $q_{DPWM} = \frac{T_c}{2^4} = \frac{1.6\,\text{ns}}{16} = 100\,\text{ps}$.

## 5.3 Transistor-level Design of the Delay Line in Cadence Virtuoso Environment

This section aims to propose a possible transistor-level design of the delay line composed of 16 cells. The circuit provided in this thesis project is obtained starting from an existing

project of the automotive division of STMicroelectronics, properly modified to respect the resolution specification.

The circuit structure is very standard, in fig.5.6 it is possible to observe the high-level behaviour of the system. The delay line consists of a sequence of delay stages that progressively add an increasing delay to the an input signal. Each delay stage could, for example, be a basic logic gate that sends an input signal to its output with a delay determined by the logic gate's properties. The delay line terminals are linked to distinct inputs of the multiplexer that generates the delay circuit output. The programmable delay circuit is simply designed to provide a desired delay by providing input control data to the multiplexer, which causes it to send the selected input tap to its output. The chosen delay is used to reset the output latch, that control the PWM signal, as already showed in fig.5.5.

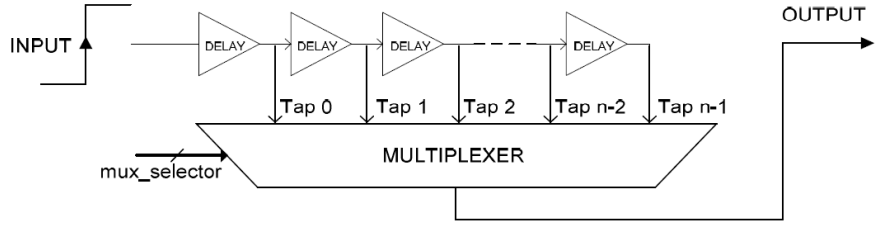



Figure 5.6: Delay Line high-level structure [5]

A linear configurable delay circuit generates a delay that is a multiple of a specified unit delay. If all delay line elements had the same unit delay, the overall delay supplied by the programmable delay circuit would be a linear function of the number of delay elements.

The peculiarity of the proposed circuit is that it provides two different programmable delays. Indeed, it is feasible to have two output signals by introducing a second multiplexer. Obviously, the addition of a second input delay selection data is required, as showed in fig.5.7. The purpose of this choice is that the circuit is meant to modulate multiple phases of the buck, so it is reasonable to save area reusing the same delay line. It means that each channel control an output signal.

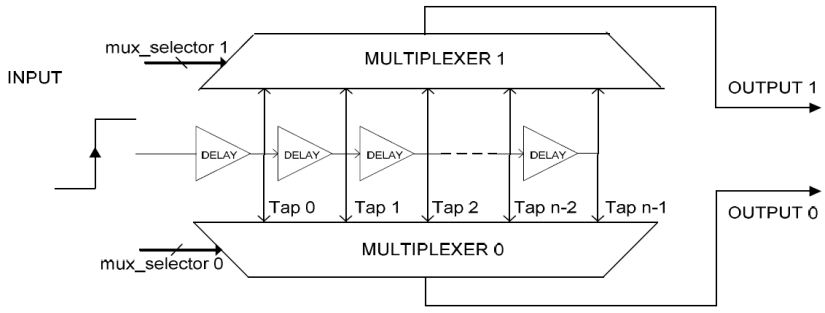



Figure 5.7: Delay Line high-level structure with two output [5]

As mentioned in the previous sections, due to disturbances or changes in environmental quantities, the overall delay may be either too short or too long in comparison to the intended switching period ($T_s$). The problem in this case is resolved using a DLL (Delay Locked Loop). The fig.5.8 offers a schematic of the DLL. The system clock is routed through a delay chain of non-inverting voltage-controlled delay elements before being passed into the input of a phase detector, where its phase is compared to the phase of the next clock. The Phase Detector generates the charge and discharge signals for a charge pump, which regulates the voltage command of all delay components. As a consequence of this type of loop, when the control voltage is steady, each of the N elements in the delay chain adds a controlled delay that is a 1/N portion of the clock. It is feasible to carry out the output of a chosen delay element of the chain and thus a digitally selected phase of the clock using a N to 1 multiplexer.

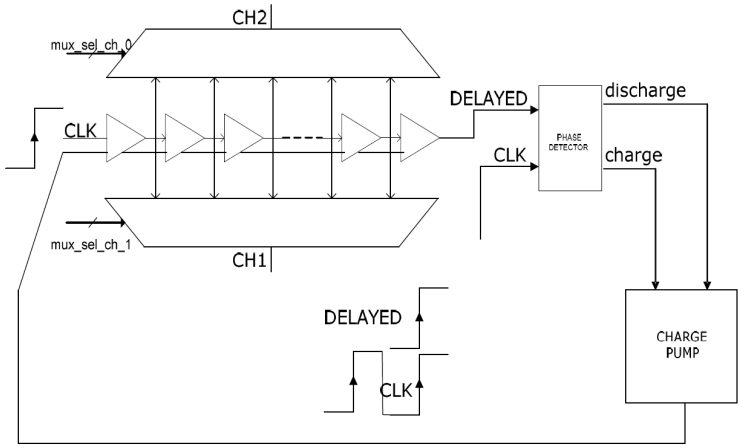



Figure 5.8: Delay Line: Delay Locked Loop (DLL) [5]

More in detail the circuit processes the system clock and feeds it to the Phase Detector (PD) via two separate pathways, as attached in fig.5.9. The first route is the delay chain, which is made up of eight groups of sixteen delay elements each plus two additional blocks (First and Last); the second path is made up of only these two special blocks (and is known as the "del_line_fast_path"). The PD sends charge/discharge impulses to the charge pump and a circuit that senses loop lock ("Loop_LK").

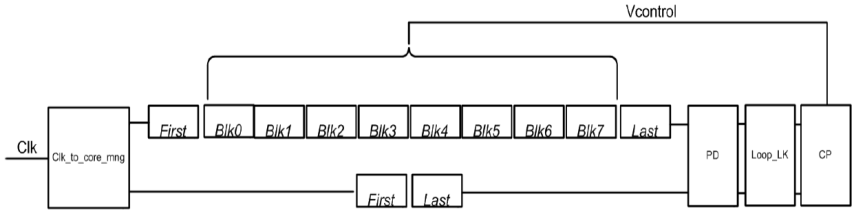



Figure 5.9: Delay Locked Loop (DLL): how it works [5]

Now, it's time to explain how the single delay element is implemented. In fig.5.10 the transistor-level circuit is proposed. This circuit architecture allows fast delays without cross-conduction current and using a simple principle of operation.

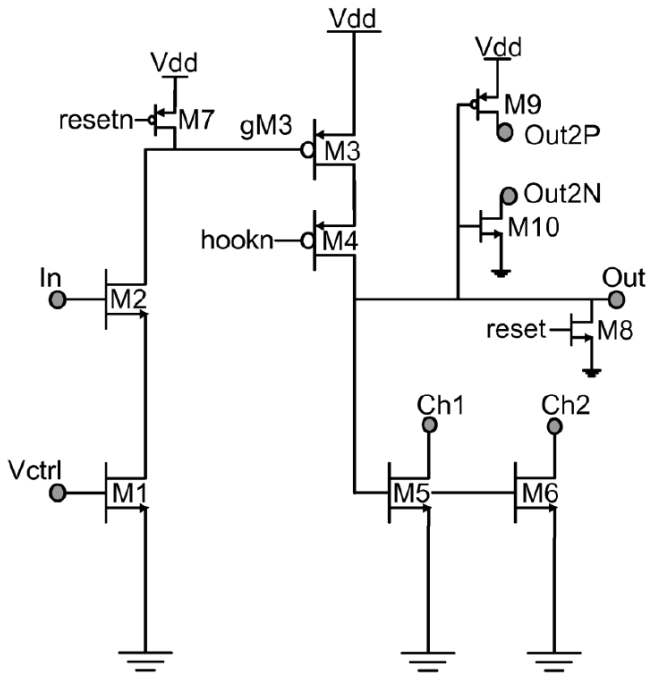



Figure 5.10: Single Delay Element of the Delay Line [5]

The operation of the elementary cell is proposed reporting exactly the reference manual's words in order to explain as clearly as possible: "Starting from a reset condition, in which M7 M4 M8 are ON, the delay cell goes in a floating condition just before the input signal arrives (M7 and M8 are switched OFF). The input signal rising edge turns on M2 that discharge M3 gate using a tunable current fixed by Vctrl gate voltage of M1. Because of M7 is off there is no cross-conduction current and the delay time of the stage is determined only by the current of M1-M2 and the capacitance on the gate of M3; this gives advantages on both energy consumption and short delay time. The same principle, except of current control, is applied to the second inverting stage that turn on also M5 and M6 output transistor activating the input of the two channel output multiplexers. After the exit of rising edge from the Out port of the element and before the next input arrives the cell has to be properly reset. In order to avoid cross-conduction current once more, we start turning off M4, then with some disoverlap between them, M8 and M7 and after these, turning on again M4, coming back to the reset condition. M9 and M10 are normally used with Out2N shorted to ground and Out2P shorted to Vdd acting as a little capacitive load. "[5].

The ResetMan block menages all timings for setting and resetting delay element transistors, shown in fig.5.11. This block is implemented with common cross-conductive standard logic cells; thus It will be used to handle signals for pages of 16 delay elements in order to reduce its power impact.

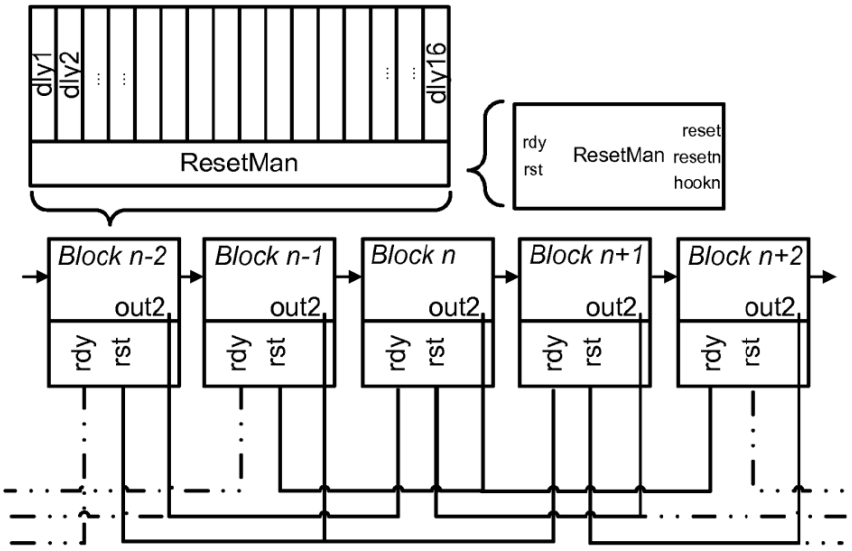



Figure 5.11: ResetMan Block [5]

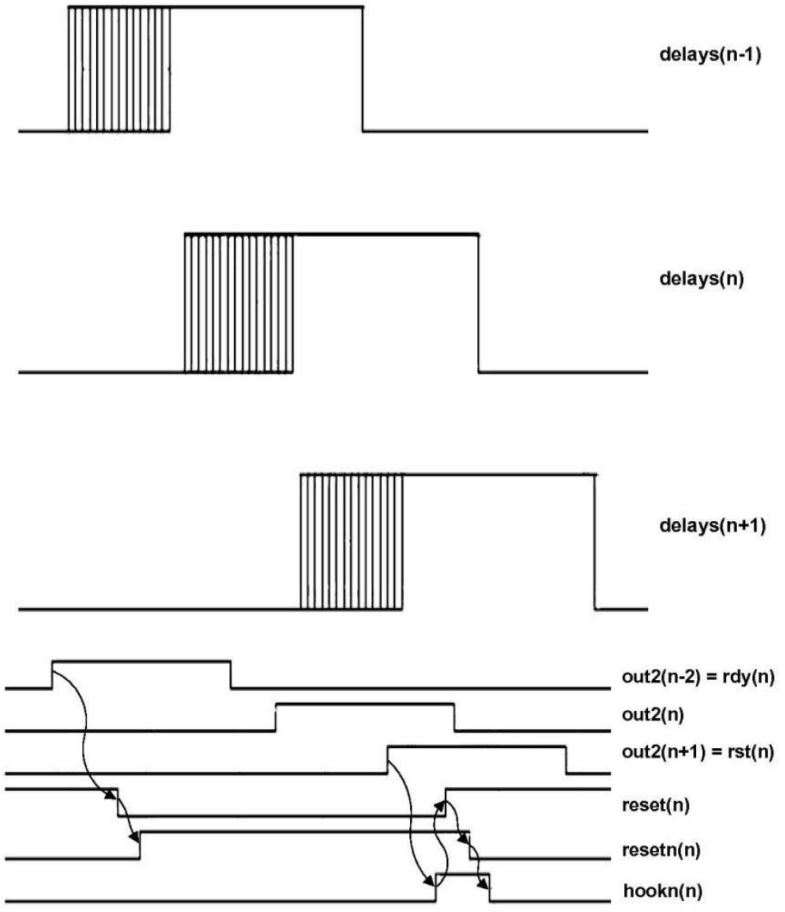



Figure 5.12: Timing ResetMan Block [5]

Each page is formed by 15 delay elements like these described in the previous paragraph and a 16th in which Out2N and Out2P (fig.5.10) are connected together forming an inverter

that is used to verify if the output signal has passed through the page. This Out2 signal is fed to block n-2 as a "go to reset" signal and to block n+1 as a "go to floating state" signal. In fig.5.12 the timing relation of the ResetMan block are reported.

Blocks 1, 2 and block 8 receive respectively their Rdy and Rst signals from two other special blocks that realize correct timings through some regular buffers. In such a way the whole delay line has only 10 little cross-conductive blocks and 128 fast low-power delay elements.

In fig.5.13 the schematic of the Phase Detector is provided. The falling edges of IN0 and IN1 coming from the fast path and the delay chain, are converted into pulses (P1R and P2R in the figure, below ), using proper falling edge detectors. Each of them sets its output, CHG for IN0 and DISCH for IN1.

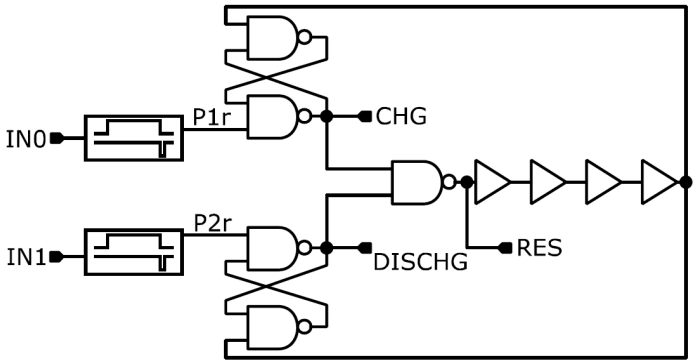



Figure 5.13: Phase Detector Schematic [5]

In fig.5.13 the schematic of the Charge Pump is provided. The control loop is locked when after a charge and a discharge action of the charge pump the net amount of charge is zero. If we indicate with $T_{chg}$, $T_{disch}$, $I_{chg}$ and $I_{disch}$ respectively the charge and discharge pulse width and the charge and discharge current, the lock condition is:

$$T_{chg}I_{chg} - T_{disch}I_{disch} = 0 \tag{5.2}$$

Ideally charge and discharge currents are equal, and condition is met when $T_{chg} = T_{disch}$, when clock edge and delay chain output edge are in phase. Unfortunately charge and discharge current suffers of mismatch that causes a phase error when lock condition is met.

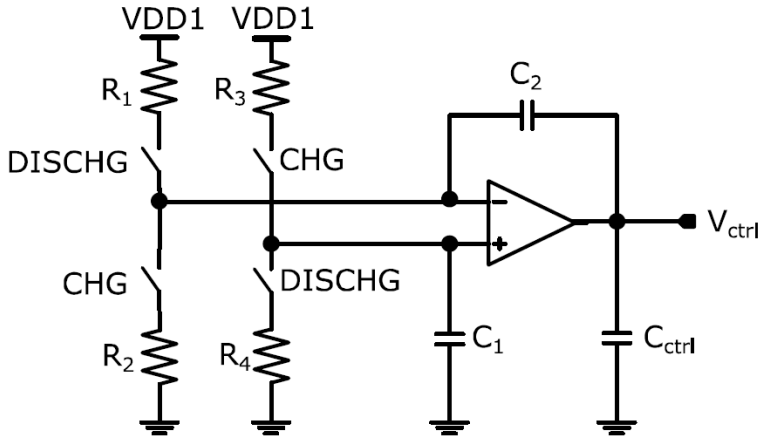



Figure 5.14: Charge Pump Schematic [5]

The proposed charge pump architecture has a very good current matching, based on resistors and capacitors values: this results in accuracy boost. Resistors R1 and R2 acts as charge/discharge current generators on C2 (and minus terminal of the operational amplifier), resistors R3 and R4 acts as charge/discharge current generators on C1 (and plus terminal of the operational amplifier). The control node will raise when charge switch are closed and will fall when switch dischg are closed. Equations for the steady state condition are:

$$T_{chg}I_{chg,R2} - T_{disch}I_{disch,R1} = 0$$
$$T_{chg}I_{chg,R3} - T_{disch}I_{disch,R4} = 0$$
$$T_{chg}(I_{chg,R2} + I_{chg,R3}) - T_{disch}(I_{chg,R1} + I_{chg,R4}) = 0$$

that results in:

$$I_{chg,R2} = I_{disch,R1}$$
$$I_{chg,R3} = I_{disch,R4}$$
$$T_{chg} = T_{disch}$$

Residual mismatches causes are resistors and capacitor mismatches and operational amplifier non-infinite gain: that is at least one order of magnitude better than in traditional charge pumps, where matching are related to current generator matching.

## 5.4 Modified Circuit and Simulation Results

The original cell was part of a dual loop multi-phase buck converter for Automotive CPU voltage regulation. It uses 4 delay lines with 2 channel each, to regulate at most 8 phases simultaneously. For a single DPWM modulation the circuit implemented by STMicroelectronics presents the behaviour attached in fig.5.15. The value of the multiplexer's selection is changed twice each PWM period, in order to delay both the rising and the falling edge

of the input synchronous (with the system clock) PWM. It's important to note is that the PDU presents an intrinsic delay independent from what value of the selection is in input. This is due to the retards of the logic inside the circuit.

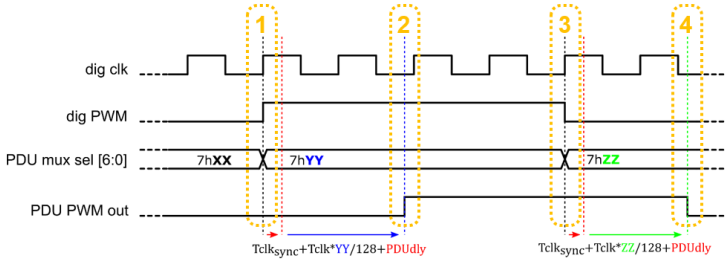



Figure 5.15: DPWM modulation of the original PDU [13]

The delay line symbol in Cadence Virtuoso environment is shown in fig.5.16. Some pins can be neglected, but the most relevant can be recognised, such as the input clock, the two multiplexers' selectors, the outputs (chA and chB) and the input control commands, composed by 2 bits. The latter set the mode of the delay line, indeed, depending on the value of these bits, with the same selector it is possible to set or reset chA or chB. It means that both inputs can acts on the same output channel. This function is exploited to adapt the original cell to the purposes of the thesis. The working principle is very simple: the loop requires a single modulation on the falling edge of the pwm, so we can use the cell, to generate a temporally fixed rising edge with one multiplexer's selector and a programmed falling edge with the other. The rising modulation is used just to include in the modulation of the two edges the intrinsic delay of the PDU.

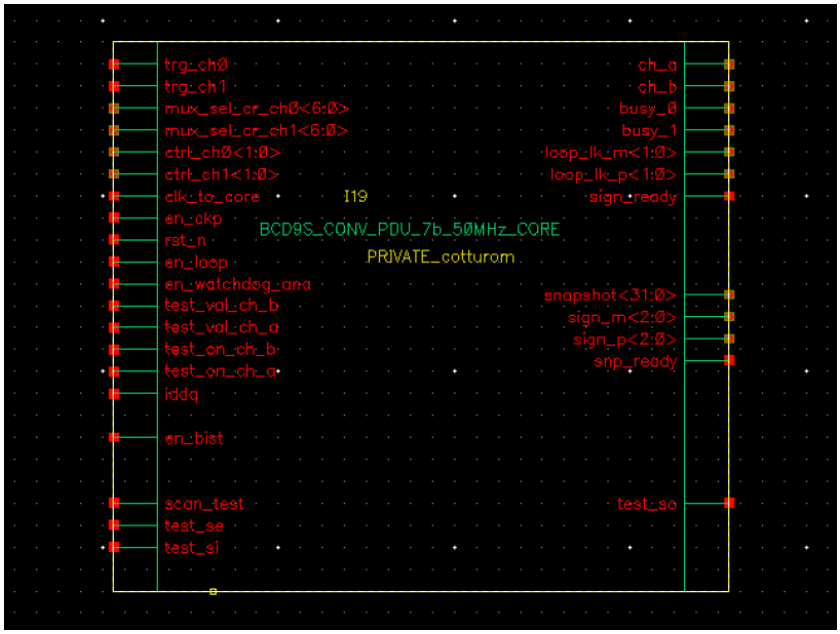



Figure 5.16: Delay Line cell of STMicroelectronics

The internal structure of the PDU is not of concern for the thesis' aim, but the fundamental delay cell may be interesting to mention. In fig.5.17 there is the transistor-level

circuit already depicted in fig.5.10, the only difference is that the transistors M3 and M4 are implemented with multiple branches instead of one to obtain the correct delay from the cell. The timing of the signals of the delay cell is guaranteed by the ResetMan Block (5.11) that manages all the signals of a bunch of 16 delay cells.

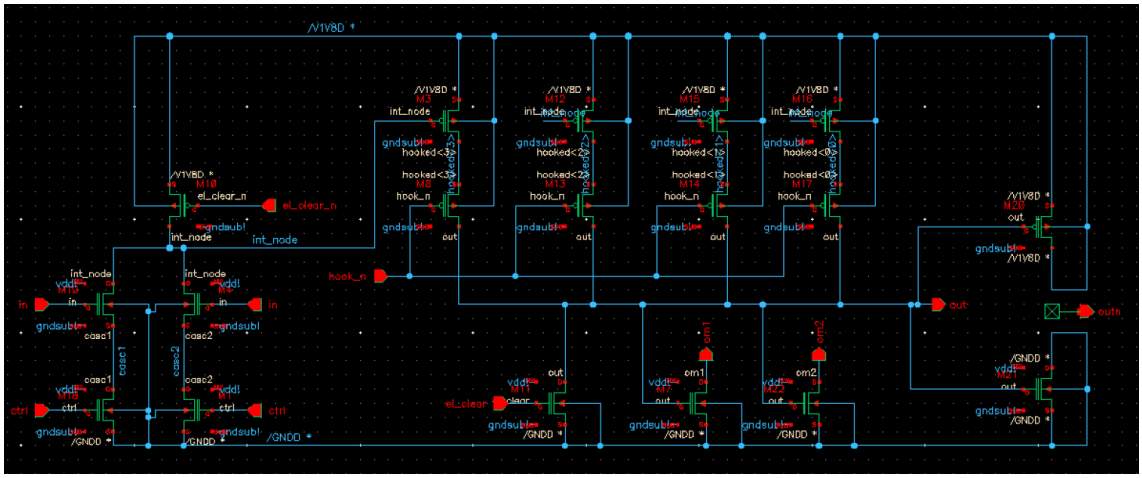



Figure 5.17: Single delay cell, transistor-level circuit

An overview of the first simulation is proposed in fig.5.18. The inputs start to change just after 10 µs, this is due to the warm-up time of the PDU, that is around 6 µs. Also the presence of another type of signal never mentioned before, the "trg_chx". This signal permits to trigger the delay action, when this pulse comes the PDU starts the propagation of the input commands and after a certain internal retard acts on the output channel. In the simulation, as already mentioned, only the CHA is used, while CHB remains idle.

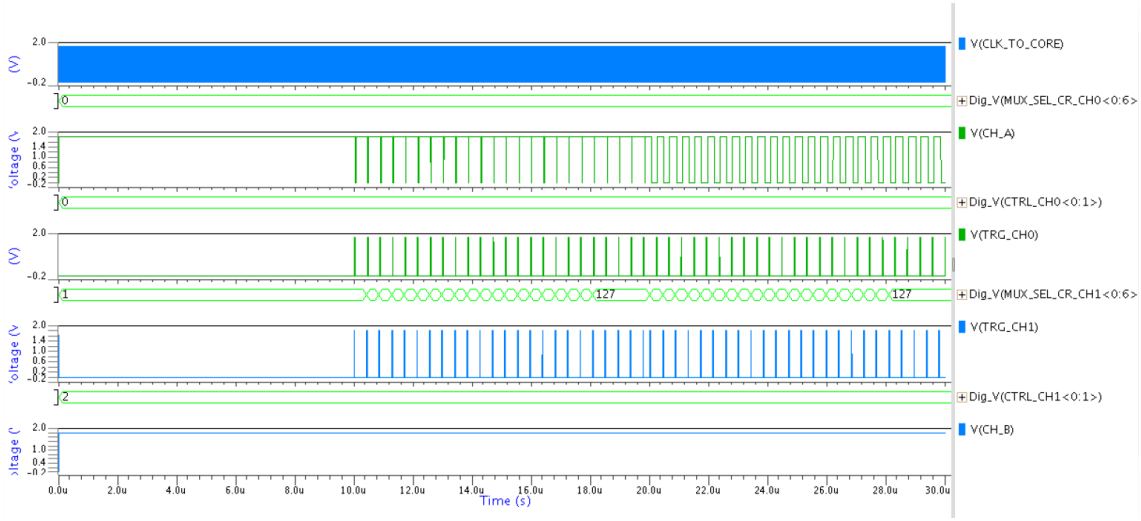



Figure 5.18: Overview of the first simulation

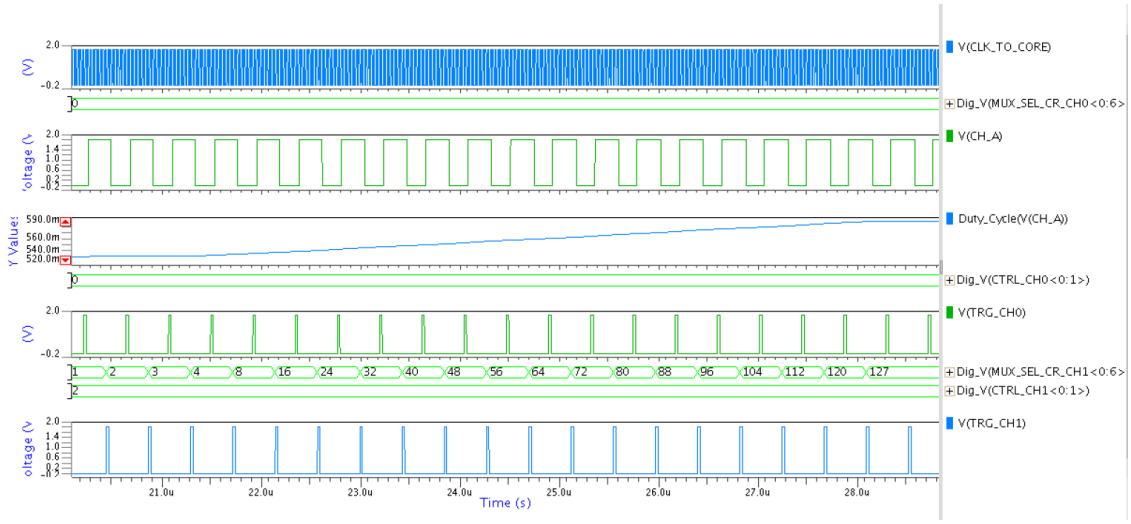



Figure 5.19: Low Duty Cycle simulation

Entering in the details of the simulation, some values in input are tested to verify the functionality of the PDU. In fig.5.19 a duty cycle of 50% is used as input of the delay line, the values of the multiplexed selection are moved from 1 to 127 in order to study how the Ton varies. In the results a signal is added to see graphically how the duty cycle changes (blu signal), it's scale is reported in mV but it's a dimensionless quantity that goes from 0 to 1, the simulator acts like that. Its values increases according with the digital input number of the selector ("mux_sel_cr_ch1"). These input, neglecting the first 4 words, changes with a step of 8 LSB. This is not a case, in fact the thesis circuit require 16 steps of delay line ($step\,size = \frac{128}{16} = 8$), for this reason 3 LSBs of the input selection are always fixed to zero, because are not useful.

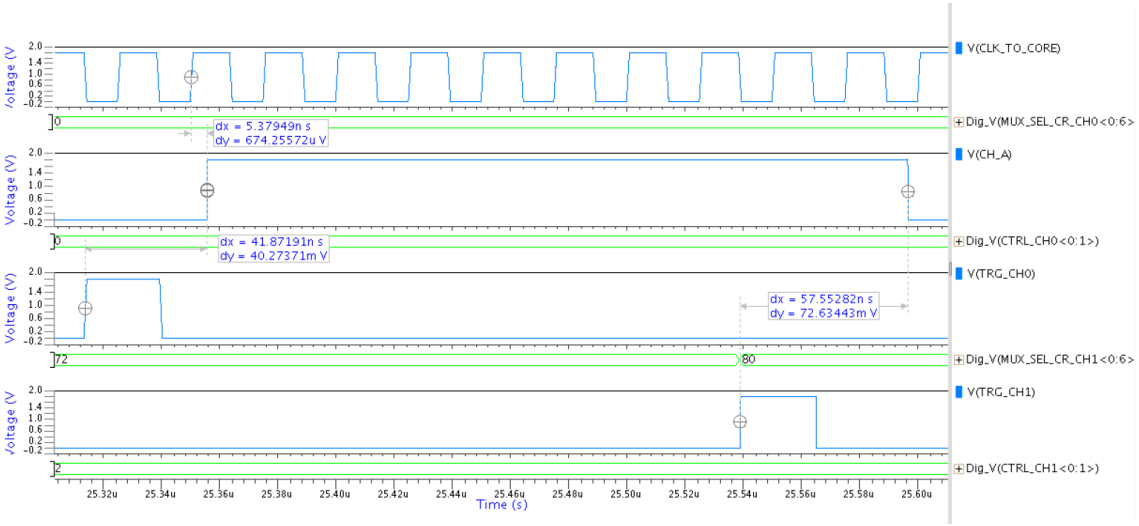



Figure 5.20: Single PWM modulation

The fig.5.20 proposes a zoom of a single PWM. There are three intervals of time highlighted in the scheme, the first two shows the intrinsic retard between the edge of the trigger and the actual rising edge of the PWM signal. They present a clock period and a half of delay plus an additive one of about 5.4 ns, that come out in correspondence of

the input value 0, so not due to the delay taps of the delay line. It can be considered as a time offset, equal to 41.8 ns.

Looking at the last highlighted time interval, the delay is much higher, it's around 57.5 ns. To verify the proper operation of the PDU it must be coherent with the input value of the multiplexer, i.e. 80 in that case. Now, considering the step size and the time offset the result can be obtain using:

$$\Delta t_{stepsize} = \frac{T_{clk}}{128} = 195\,\text{ps}$$
$$num.\,of\,taps = \frac{(\Delta t_{falling\,edge} - \Delta t_{offset})}{\Delta t_{step\,size}} = \frac{57,5\,\text{ns} - 41.8\,\text{ns}}{195\,\text{ps}} = 80$$

Just for completeness another example is provided in fig.5.21, here the input duty cycle is the highest possible coming from the synchronous counter-based modulator.

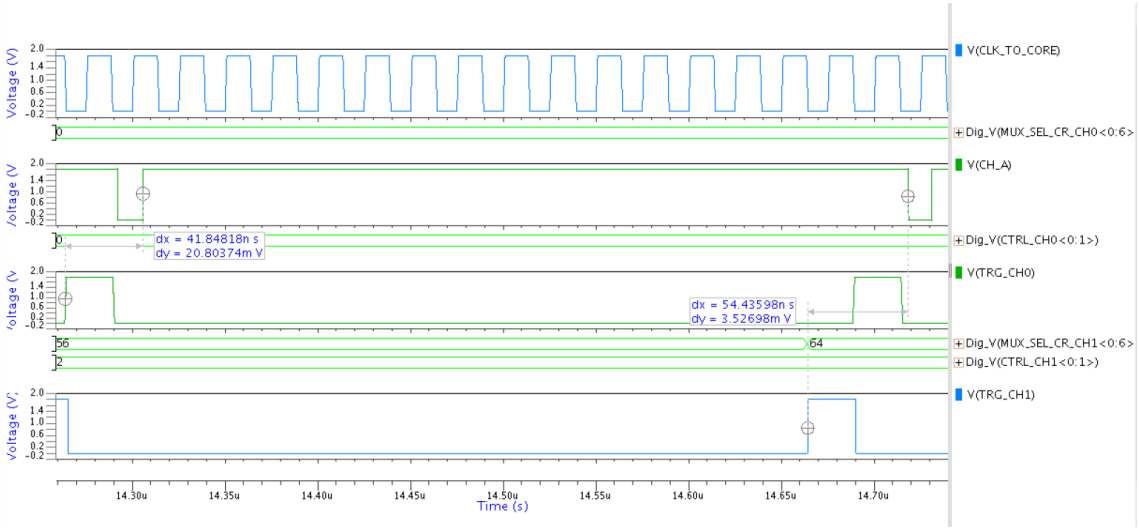



Figure 5.21: Simulation with an high Duty Cycle as input

Also in this case the simulation is coherent, showing the capabilities of the PDU. The computations are the same of the previous case, with the inputs changed. The PDU acts as expected, the delay introduced by the circuit is proportional to the input digital value. This kind of operation can be extended to other duty cycle values and to all the other selection's digital value of the multiplexer, not showed in the simulation.

In conclusion, the proposed circuit is adaptable to the specifications. There is no need to change the transistor-level design of the delay line. It is sufficient to fix the input LSB and choose only the delay taps of interest. It is already possible to integrate the block in the loop and use it for future possible measurements.

# Chapter 6

# Conclusion and Future Works

From what described in the previous chapters, the following conclusions can be drawn: the architecture of the digital control loop taken into account for this project has a great potential due to the combination between configurability and precision. These characteristics are ensured by a configurable PID that can be programmed thought a series bus interface and by the proposed method to determine its parameter; the combination of three different techniques obtains a DPWM resolution of 0.024% on the Duty Cycle; The proposed structures of the dyadic dithering and the delay line offer a concrete example of how to implement the circuit. This aspect also represents the main critical issue for what concern the chip area: the proposed structure is not meant to the thesis circuit, its application needed more complexity. Finally, the design strategy chosen for this project suggests how it is possible to introduce a new control structure that permits to adapt the chip to several scenarios, avoiding the redesign of the entire control. With the proposed loop it is possible to minimise the time-to-market and the design costs, two aspects of the production process that cannot be ignored.

## 6.1 Future Works

The conclusions drawn in the previous section offer some hints for future works: first of all, an ad hoc delay line for the thesis circuit is needed to minimise the chip area. As already said the circuit proposed in Chapter 5 is oversized for the application, a delay chain with only 16 cells is enough. Another aspect worthy of a future work is a deeper study on the compensator design. From the simulations it is clear that a non-linear effect appears from the quantized nature of the loop.

The diffusion on silicon of the proposed control loop in a **Test Chip** will offer the possibility to measure practically how it works and the performance that it presents. The ideal scenario for the test is the integration of two types of control, digital and analog, on the same chip, with the possibility to switch between them. In this way it is easy to compare their behavior, understanding if the digital loop is a valid alternative to the common used analog loop, also on a practical level. STMicroelectronics is already moving in this way, programming to integrate both controls in a chip. In the next mouths the first results will be available to extrapolate the first conclusions. The loop will be very similar to the thesis

proposal, some changes will be applied, in particular on the ADC structure: the available devices are not conform to the specifications, for this reason it will have an input range much higher, 1.8 v, with 12 bits of resolution, but it will still be a SAR. It means that the voltage divider that acts on the output voltage will be changed accordingly. Also the working frequency will be changed, due to the operating frequency of the ADC: it needs 17 clock periods to perform the conversion, so the switching frequency will be at most 2 MHz, or smaller.

# Appendix A

# MATLAB code

```
%The following data are needed to obtain the plant component
%tranfer function and the PID gains
%%%%%%%%%%
rC=ESR;                          %capcitor series resistance
C=C_out;                         %output capcitor
L=L_value;                       %inductor value
rL=DCR;                          %DC Resistance
D=0.5;                           %expected ideal duty cycle
Dprime=1-D;
fs=2.4e6;                        %switching frequency
Ts=fs^-1;                        %switching period
Vg=6;                            %Vin
Io=4;                            %output current
td=600e-9;                       %total delay of the digital
                                 %control loop

%voltage divider gain
Gp=0.3636363636;
%ADC gain (full scale Range divided by 2^nbit)
Gadc=(6.72e-4)^-1;
%DPWM gain (equivalent modulator time resolution, dithering
%included, divided by Ts)
Gdpwm=2.4e-4;

wc =1.5e6;                       %desired crossover frequency
   (rad/s)
mphi = (pi/180)*80;              %desired phase margin
%%%%%%%%%

%Code from the book:
```

```
%("Digital Control of High-Frequency Switched-Mode Power
   Converters")

%Discrete-time transfer function of the plant:

%%%%%%%%%
%First, define the converter state-space
%matrices:
A1 = [ -(rC+rL)/L -1/L; 1/C 0 ];
A0 = A1;
b1 = [ 1/L rC/L; 0 -1/C ];
b0 = [ 0 rC/L; 0 -1/C];
c1 = [1 0; rC 1];
c0 = c1;

%Next, evaluate X
A1i = A1^-1;
A0i = A0^-1;
Xdown = ((eye(2)-expm(A1*D*Ts)*expm(A0*Dprime*Ts))^-1)*...
(-expm(A1*D*Ts)*A0i*(eye(2)-expm(A0*Dprime*Ts))*b0+...
-A1i*(eye(2)-expm(A1*D*Ts))*b1)*[Vg;Io];

%Then, construct small-signal model matrices   ,   , and
Phi = expm(A0*(Ts-td))*expm(A1*D*Ts)*expm(A0*(td-D*Ts));
gamma = expm(A0*(Ts-td))*((A1-A0)*Xdown + (b1-b0)*[Vg;Io])*Ts
   ;
delta = c0;

%Finally, extract the control-to-output transfer functions
   Gvu(z) and Giu(z) by converting
%the state-space representation into Matlab  transfer
   function objects.
%First, build a state-space object usingmethod ss and then
   the transfer function objects
%using tf:
sys_i = ss(Phi,gamma,delta(1,:),0,Ts);
Giuz = tf(sys_i);
sys = ss(Phi,gamma,delta(2,:),0,Ts);
%%%%%%%%%%

%Discrete-time transfer function of the plant with the gains
   of the loop included
Gvuz = tf(sys*Gp*(Gadc)*Gdpwm);

figure(1)
```

```
 hh=bodeplot(Gvuz, {0 1e7});
 pp = getoptions(hh);
 pp.FreqUnits = 'Hz';
 setoptions(hh,pp);
 grid on
 legend('Plant dicrete-time tranfer function')

%Plant transfer function obtained by manual computation (
   continous time):
buck_cont=tf(Vg*[rC*C 1], [L*C*(Rload+rC)/(Rload+rL) rC*C+(
   Rload*rL/(Rload+rL))*C+L/(Rload+rL) 1]);

%continous-time plant transfer function with the total delay
   of the loop included.
s=tf('s');
buck_rit=buck_cont*exp(-s*td);
%delayed continous-time plant transfer function with the
   gains of the loop included
buck=buck_rit_0_5*Gp*Gadc*Gdpwm;

%comparison between the continous-time plant transfer
   function and the delayed one
figure(2)
h=bodeplot(buck_cont, {0 6.3e6});
p = getoptions(h);
p.FreqUnits = 'Hz';
setoptions(h,p);
hold on
a=bodeplot(buck_rit, {0 6.3e6});
p = getoptions(h);
p.FreqUnits = 'Hz';
setoptions(a,p);
grid on
hold off
legend('Continous time plant', 'Delayed continous time plant'
   )

%comparison between delayed continous-time plant transfer
   function and the
%discrete-time one
figure(3)
h=bodeplot(buck, {0 6.3e6});
p = getoptions(h);
p.FreqUnits = 'Hz';
setoptions(h,p);
```

```
hold on
a=bodeplot(Gvuz, {0 6.3e6});
p = getoptions(h);
p.FreqUnits = 'Hz';
setoptions(a,p);
grid on
hold off
legend('Continous time plant', 'Delayed continous time plant'
   )

%Continuos-time PID design
%%%%%%%%
[mm,pp]=bode(buck, {wc,wc+0.1});
TuWc=mm(1);
phi=phi*pi/180;
W_z=wc*sqrt((1-sin(phi))/(1+sin(phi)));
W_p=wc*sqrt((1+sin(phi))/(1-sin(phi)));

GPD0=TuWc^-1*sqrt((1+(wc/W_p)^2)/(1+(wc/W_z)^2));

%polo dato dall'azione integrale
W_l=wc/20;

%definisco la funzione del PID ed estraggo i parametri da
   mettere in
%simulink
GPID=tf((1+W_l/s)*GPD0*(1+s/Wz)/(1+s/Wp));
[p, i, d, Tf]=piddata(GPID)
%%%%%%%%%

Tuz_new=Gvuz;


%m and p are the magnitude and the phase of the plant
   computed at the desired crossover frequency
[m,p] = bode(Tuz_new,wc);


%Code from the book ("Digital Control of High-Frequency
   Switched-Mode Power Converters")
%Discrete-time transfer function of the PID compensator, p-
   mapping method:

%%%%%%%%%%%%
p = (pi/180)*p;
```

```
% Prewarping on wc
wcp = (2/Ts)*tan(wc*Ts/2);
% PD Design
wp = 2*pi*fs/pi;
pw = atan(wcp/wp);
wPD = 1/(tan(-pi+mphi-p+pw)/wcp);
GPD0 = sqrt(1+(wcp/wp)^2)/(m*(sqrt(1+(wcp/wPD)^2)));
% PI zero and high-frequency gain
wPI = wc/20;
GPIinf = 1;
% Proportional, Integral and Derivative Gains
Kp = GPIinf*GPD0*(1+wPI/wPD-2*wPI/wp)
Ki = 2*GPIinf*GPD0*wPI/wp
Kd = GPIinf*GPD0/2*(1-wPI/wp)*(wp/wPD-1)
% PID Transfer function
z = tf('z',Ts);
Gcz = Kp + Ki/(1-z^-1) + Kd*(1-z^-1);
%%%%%%%%%%%%


%MATLAB provides an automatic mehod to tune the PID
   compensator strating from a
%certain transfer function:

Options = pidtuneOptions('PhaseMargin',mphi, 'DesignFocus','
   balanced');
C0 = pid(1,1,1,'Ts',Ts,'IFormula','BackwardEuler','DFormula',
    'BackwardEuler');
[auto_pid, info]=pidtune(Tuz_new,C0, wc, Options)
                    %discrete-time
[auto_pid_s, info_s]=pidtune(buck,'PID', wc, Options)
            %continuous-time

%comparison between the loop gains with different PIDs
figure (1)
hh=bodeplot(Tuz_new*Gcz, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);

grid on
hold on
hh=bodeplot(Tuz_new*auto_pid, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
```

```
setoptions(hh,pp);

hh=bodeplot(buck*auto_pid_s, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);

hh=bodeplot(buck*GPID, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);
hold off
legend('book approach (p-mapping) discrete-time', 'MATLAB
   automatic approach discrete-time', 'MATLAB automatic
   approach continuous-time', 'book approach continuous-time'
   );

%just PID functions
figure(3)
hh=bodeplot(Gcz, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);
grid on
hold on

hh=bodeplot(auto_pid, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);

hh=bodeplot(auto_pid_s, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);

hh=bodeplot(GPID, {0 1e7});
pp = getoptions(hh);
pp.FreqUnits = 'Hz';
setoptions(hh,pp);
hold off
legend('book approach (p-mapping) discrete-time', 'MATLAB
   automatic approach discrete-time', 'MATLAB automatic
   approach continuous-time', 'book approach continuous-time'
   );
```

```
%SUMMARY OF ALL THE PID GAINS for the discrete-time
   approaches
%Table
rownames={'Kp','Ki','Kd'};
columsnames={'LIBRO_d', 'MATLAB_d','LIBRO_c', 'MATLAB_c'};

LIBRO_d=[Kp; Ki; Kd];
MATLAB_d=[auto_pid.kp; auto_pid.ki/fs; auto_pid.kd*fs];
LIBRO_c=[p; i; d];
MATLAB_c=[auto_pid_s.kp; auto_pid_s.ki/fs; auto_pid_s.kd*fs];
table(LIBRO_d, MATLAB_d, LIBRO_c, MATLAB_c,'RowNames',
   rownames)
```

# Bibliography


[1] *BCD: Bipolar-CMOS-DMOS*. URL: https://www.st.com/content/st_com/en/about/innovation---technology/BCD.html.

[2] Nan Chen et al. "A Digital Controller IC for High-Frequency DC-DC Switching Converters". In: *IEEE TRANSACTIONS ON POWER ELECTRONICS* (2019).

[3] Luca Corradini et al. *Digital Control of High-Frequency Switched-Mode Power Converters*. IEEE Press Series on Power and Energy Systems. IEEE Press, 2015. ISBN: 9781118935101.

[4] Paolo S. Crovetti et al. "Limit Cycle-Free Digitally Controlled DC-DC Converters based on Dyadic Digital PWM". In: *IEEE Transactions on Power Electronics* (2020), pp. 1–1. DOI: 10.1109/tpel.2020.2978696. URL: https://doi.org/10.1109%5C%2Ftpel.2020.2978696.

[5] Juri Giovannone and Andrea Gambero. "BCD9S_CONV_PDU_7b_50MHz_1V8. USER MANUAL". In: Castelletto, Italy, STMicroelectronics, 2021.

[6] Mark Hagen and Vahid Yousefzadeh. "Applying Digital Technology to PWM Control-Loop Designs". In: (2008).

[7] Brigitte Hauke. "Basic Calculation of a Buck Converter's Power Stage. Low Power DC-DC Applications". In: (2011). Revised in 2015.

[8] Dheeraj Kumar Palwalia Kartik Sharma. "Design of Digital PID Controller for Voltage Mode Control of DC-DC Converters". In: *IEEE TRANSACTIONS ON POWER ELECTRONICS* (2017).

[9] Marian K. Kazimierczuk. *Pulse-width Modulated DC–DC Power Converters*. Dayton, Ohio, USA: John Wiley  Sons, 2008. ISBN: 9780470773017.

[10] James Mooney et al. "Dithered multi-bit sigma-delta modulator based DPWM for DC-DC converters". In: *2015 IEEE Applied Power Electronics Conference and Exposition (APEC)*. 2015, pp. 2835–2839. DOI: 10.1109/APEC.2015.7104752.

[11] "Powering Automotive Cockpit Electronics. Automotive POL Regulator Requirements". In: *RENESAS* ().

[12] Jose M. de la Rosa. *Sigma-Delta Converters: Practical Design Guide, 2nd Edition*. Wiley-IEEE Press, 2018. ISBN: 9781119275763.

[13] STMicroelectronics. *UR6Y Loop Description*. Castelletto, Italy.

[14] "Switching Regulator Advances: Compact, Efficient Power Management Design. Switching Regulators White Paper". In: *Rohm* (2015).