Integrated DC-DC Digital Control Design
(November 2009)
Carlos João Oliveira Moreira
Instituto Superior Técnico
Lisbon, Portugal
Email: cmoreira@ieee.org

Abstract— Analog circuits have been dominating the switching power supplies control because of their simplicity and low implementation cost. However, they may not meet the increasingly stringent future processors requirements.

This study concentrates in the fact that sensing the output voltage (VMC) of a Buck converter a digital control can be built to simulate a CMC, and avoid disadvantages like sensitive and complex analog error amplifiers.

In this paper an algorithm is constructed and implemented in Matlab and HDL Verilog. A Buck and the necessary modules are modeled in Matlab and design using Cadence. The obtain results are within the specifications, such as steady state error less than 10mV for a load from 0.1A to 1A; Line transient from 2.5V to 3.3V with an overshoot less than 10% and an establishment-time less than 25us for a constant load of 0.1A to 1A; Load transient from 0.1A to 0.9A for a constant 2.5V input voltage with a overshoot less than 10% and establishment-time less than 30us.

This work is a humble contribution to help proving that digital control is a future solution to DC-DC converters control.

Index Terms—DC-DC Control, Buck Control, Digital Control, Voltage-Mode Control, Current-Mode Control

I. INTRODUCTION

The Power Management World is incorporating more and more digital ideas. Nowadays, due to recent circuit requirements it’s necessary to reduce the power consumption in systems such as DC-DC converters, but still improve converter performance, such as the voltage ripple and the establishment-time.

One solution is digital control, which can accommodate the main control methods, voltage-mode control (VMC) and current-mode control (CMC). This solution has a penalty that is the need of an ADC and a DAC. But still, it is believed that it can provide a power consumption reduction and performance optimization.

The main difference between analog and digital control is the quality and the amount of information available for the controller to make decisions regarding to the power stage operation. For example, in addition of just comparing the output voltage to a threshold, changes in this parameter value can be detected, stored and later reported back to a supervising system. If necessary, parameter values can be combined with other information in complex algorithms to perform even more sophisticated functions [1][2].

The most important digital control advantage is its flexibility, which makes it easy to reuse with programmable variations. Linear digital control methods are translated directly from those techniques, which are widely used in analog circuits, and provide comparable performance. Digital control implementation also offers the potential to provide much more powerful control functions such as predictive [3], adaptive [4], and multi-mode [5][6] controls that can improve static and dynamic performance. Also, full digital control of DC-DC converters offers the advantage that designers can easily retarget the DC-DC converter for a different technology without the redesign effort of complex analog modules.

Recent developments in digital control strategies for DC-DC are described in [7]. It states that as the cost of digital controllers decrease and the controller requirements of DC-DC converters increases, it is inevitable that digital control will assume a relevant part in the control. Although digital control still presents some drawbacks, their unique capabilities such as efficiency optimization, auto-tuning and nonlinear control will create a spot that cannot be filled by any analog controller.

II. DC-DC CONVERTER & CONTROL MODES

A. DC-DC Buck Converter

A Buck Converter is a magnetic switch-mode converters. In these converters, the energy is periodically stored into a magnetic field and then released. The power transfer is controlled by adjusting the ratio between the on-time, when the magnetic-field increase, and the switching cycle period, i.e. the Duty Cycle represent as D [8].

In this paper, the Step-Down magnetic switch-mode converter is explored where the output voltage is a fraction of the input voltage and with the same polarity, represented in Figure 1.

Summarizing, the main functions of a DC-DC converter are: Convert a DC input voltage into a DC output voltage; Regulate the DC output voltage against load and line variations; Reduce the AC voltage ripple on the DC output voltage below the required level; Protect the supplied systems and the input source from electromagnetic interference [9].

The Buck conduction mode duty cycle [8][9][10], is described by: \[ V_o = V_{in} \times D \] (1)

Equation (1) shows a linear relationship between the output voltage and the duty cycle, for a given input voltage. D cannot
be greater than one the output voltage is always less or equal to the input voltage. However, this equation does not take in count facts such as: Transistors, diodes and other passive components are not ideal, i.e. they are considered has lossless elements [9]. The input voltage is a clean dc voltage [9]. These assumptions can be far from reality and the real components imperfections can produce a negative effect on the converter operation.

The converter component values are also changing with time, temperature, pressure, and so forth. Concerning this, the control of the output voltage should be performed in a closed-loop manner using principles of negative feedback. Whether analog or digital, there are two ways to implement the control feedback loop of a DC-DC converter: Voltage-Mode Control (VMC) and Current-Mode Control (CMC).

B. Voltage-Mode Control

VMC is probably the most simple way to control a power supply. In essence, an error voltage is obtained from the difference between a reference voltage and the output voltage [8]. This difference is compared to a sawtooth signal with fixed frequency and amplitude. The output power transistors are driven by pulse widths controlled by this error signal. The higher the error voltage, the longer the switch is on.

According to [11], the major advantages of VMC are: 1. A single feedback loop is easier to design and analyze; 2. A large-amplitude ramp waveform provides good noise margin for stable modulation process; 3. A low-impedance power output provides better cross-regulation for multiple output supplies. According to [11], the major disadvantages of VMC are: 1. Any change in line or load must first be sensed as an output change and then corrected by the feedback loop, this usually means slow response; 2. The output filter adds two poles to the control loop requiring either a dominant-pole low frequency roll-off at the error amplifier or an added zero in the compensation.

C. Current-Mode Control

Current-Mode control is based on controlling the inductor current [12] [13].

According to [11], CMC advantages are: 1. Since inductor current rises with a slope determined by Vin-Vo, this waveform respond instantaneously to line voltage changes, eliminating both the delayed response and gain variation with input voltage changes; 2. Since the error amplifier is used to control an output current rather than voltage, the output inductor effect is minimized and the filter now places a single pole to the feedback loop, at least in the normal region of interest. This allows both simpler compensation and higher gain bandwidth over a comparable voltage-mode circuit; 3. Additional benefits with current-mode circuits include inherent pulse-by-pulse current limiting by merely clamping the control from the error amplifier.

According to [11], CMC disadvantages are: 1. Circuit analysis is more difficult because there are two feedback loops; 2. The control loop becomes unstable at duty cycles above 50%, unless slope compensation [8] is added to the control circuit; 3. Since the control modulation is based on a signal derived from output current, resonances in the power stages can insert noise into the control loop; 4. A particularly troublesome noise source is the leading edge current spike typically caused by transformer winding capacitance and output rectifier recovery current; 5. With the control loop forcing a current drive, load regulation is worse and coupled inductors are required to get acceptable cross-regulation with multiple outputs.

D. Comparison between VMC and CMC

Current-Mode Control and VMC offer drastically different dynamic behaviours. However, looking at a typical output converter waveform it’s not evident which control technique is implemented [14].

The VMC disadvantages are relatively significant and since all are attenuated with CMC, designers were highly motivated to consider this topology. Meanwhile, CMC comes with its unique problems which must be solved in the design process.

CMC will ease many of the limitations of VMC, it also contributes to a new set of challenges to the designer. However, with the knowledge gained from more recent developments in power control technology, a re-evaluation of VMC indicates that there were alternatives to correct its major weaknesses.

III. DC-DC DIGITAL CONTROL

A. Digital Control Theory

In order to avoid the CMC requirement of measuring the current, a VMC method was developed. The concept behind this control method is that monitoring the output voltage, the average inductor current can be estimated, and a CMC can be emulated. The objective is to obtain an expression to control the duty cycle, in the continuous conduction mode, imposed on the power switches. This control law goal is to manage the average inductor current and, indirectly the output voltage.

Figure 2 presents the converter filter and the load output. In a first phase some definitions are presented followed by the objective of each step that will guide the reader to the final control expression.

\[ i_{n-1} \] - Inductor current final value at switching cycle \( n-1 \);
\[ v_{n-1} \] - Output voltage final value at switching cycle \( n-1 \);
\[ D_n \] - Duty cycle value at switching cycle \( n \);
\[ \bar{v}_n \] - Average output voltage at switching cycle \( n \);
\[ \bar{i}_n \] - Average output current at switching cycle \( n \);

Figure 3 illustrates the definitions presented here:
In order to obtain the desired duty cycle \( D_{n+1} \) consists of three steps:

1. \( D_{n+1} \) is obtained from the required average inductor current, \( i_{n+1} \), and the estimated final inductor current of the earlier switching cycle, \( i_n \).

2. The required average inductor current, \( i_{n+1} \), is obtained from the estimated average inductor current of the earlier switching cycle \( i_n \), considering a constant load;

3. Also in this step, the estimated \( i_n \) is obtained from the estimated value of \( i_{n-1} \);

The contributions of \( i_{n-1} \) of 2.1 and 3 are cancelled, as will be explained later, in order to obtain the desired duty cycle value. Figure 4 illustrates these three steps of the methodology used.

(1.) In the first step, \( D_{n+1} \) is obtained from the average current \( i_{n+1} \) of the next switching cycle. Figure 5 shows the inductor current in a continuous current mode buck converter, as described in Section 2.1.2.

The average current \( i_{n+1} \) is obtained from the average inductor value \( I_1 \) when the switch pMOS is on and the average inductor value \( I_2 \) when the switch pMOS is off after N cycles as is represented in equation (2).

\[
i_{n+1} = \frac{I_1 + I_2}{2} \quad (2)
\]

The average inductor current during the rising slope, \( I_1 \), and the average inductor current during the falling slope, \( I_2 \), are obtained by simple analysis of the inductor current waveform which is expressed in (3) and in (4). \( S_{r'} \) denotes the average increasing current slope and \( S_{f'} \) denotes the average decreasing current slope in the following N cycles.

\[
i_1 = i_n + S_{r'}.D_n \frac{T}{2} \quad (3)
\]

\[
i_2 = i_n + S_{r'}.D_{n+1}.T . (1 - D_{n+1}) \left( N - \frac{1}{2} \right) \quad (4)
\]

Repeating (2) to (4), the average current \( i_{n+1} \) is obtained.

\[
i_{n+1} = i_n + \frac{3}{4} . S_{r'}.D_{n+1}.T . S_{f'} . (1 - D_{n+1}) . T \quad (5)
\]

The slopes \( S_{r'} \) and \( S_{f'} \) are approximated by (6) and joining (5) to (6) the desired current is obtain in (7).

\[
\begin{aligned}
S_{r'} &= \frac{V_{in} - \bar{v}_{n+1}}{L} \\
S_{f'} &= \frac{V_{in} - \bar{v}_{n+1}}{2L} \\
\bar{v}_{n+1} &= i_n + \frac{3}{4} . S_{r'}.D_{n+1}.T - \frac{S_{f'}}{4}. (1 - D_{n+1}).T \quad (6)
\end{aligned}
\]

Rearranging (7) to obtain the desired duty cycle function, equation (8) is obtained.

\[
D_{n+1} = \frac{2L}{3}. \frac{i_{n+1} - i_n + \bar{v}_{n+1}}{V_{in} - \bar{v}_{n+1}} \quad (8)
\]

(2.) In step 2, the average current of the next switching cycle \( i_{n+1} \), required to take the output voltage to the reference voltage in N switching cycles. Equation (9) describes the current flowing through the inductor of Figure 2. The average inductor current value required to drive the output voltage in N switching cycles with period T, is estimated by (10). Equation (10) does not represent the accurate value of the average current since the output voltage is affected by ripple. However, this estimation is expected to be acceptable for control proposes.

\[
i(t) = \frac{v_o(t)}{R} + \frac{dv_o(t)}{dt}.C \quad \Rightarrow \quad i_{n+1} = \frac{v_{n+1} + v_n + v_{n+1} - v_n}{2} + \frac{v_{n+1} - v_n}{2R}.C \quad (9)
\]

In the preceding cycle, the average inductor current is given by (11).

\[
i_n = \frac{v_n + v_{n-1} + v_{n} - v_{n-1}}{2R}.T . C \quad (10)
\]

Based on the preceding cycle, the \( I(2R) \) parameter can be estimated. This is done in order to obtain an expression with the load characteristic implied in the equation.
\[
\frac{1}{2R} = \left( i_n - \frac{v_n - v_{n-1}}{T}.C \right) \frac{1}{v_n + v_{n-1}} \quad (12)
\]

Combining (11) and (12), the average inductor current required to drive the output voltage to the reference voltage, in N cycles, is obtained. This current will force the output voltage to the reference voltage, based on the \(I/(2R)\) estimated value.

\[
\frac{1}{n+1} = \frac{v_n + V_{REF}}{v_n + V_{REF}} \left( i_{n+1} - \frac{v_n - v_{n-1}}{T}.C \right) + \frac{V_{REF} - v_n}{N}.T.C \quad (13)
\]

(2.1) The average current at switching cycle n is obtained in equation (14), using Figure 5 acknowledges with the average rising current value and the average falling current value. Equation (14) can be simplified into (15). In these equations \(S_n\) and \(S_f\) represent, the rising and falling slopes of the current in the preceding cycle respectively.

\[
i_n = i_{n-1} + D_n + I_2.\left( 1 - D_n \right)
\]

\[
\frac{1}{n+1} = i_{n-1} + \frac{S_n}{2}.D_n.T + S_n.D_n.T - \frac{S_f}{2}.T + S_f.D_n.T
\]

(14)

\[
i_n = i_{n-1} - \frac{S_n}{2}.D_n.T + S_n.D_n.T - \frac{S_f}{2}.T + S_f.D_n.T
\]

(15)

\[
\text{Figure 5 – Preceding switching cycle}
\]

In order to obtain the current slope \(S_n\) and \(S_f\), it is necessary to find out \(V_{1}\) and \(V_{2}\). See Figure 5. These voltages are estimated in (16) and (17). And the corresponding slopes are described by (18).

\[
\begin{align*}
V_1 &= v_{n-1} + \frac{v_n - v_{n-1}}{2}.D_n \quad (16) \\
V_2 &= \frac{v_n + v_{n-1}}{2} + \frac{v_n - v_{n-1}}{2}.D_n \quad (17)
\end{align*}
\]

\[
\begin{align*}
S_n &= \frac{V_{in}}{L} - \frac{V_{in} - V_{in-1}}{L} - \frac{v_n - v_{n-1}}{2 \times L} \\
S_f &= \frac{V_{in}}{L} + \frac{v_n + v_{n-1}}{2 \times L} + \frac{v_n - v_{n-1}}{2 \times L}
\end{align*}
\]

(18)

Joining (15) to (18), an estimative of the current in the preceding cycle is found.

\[
i_n = i_{n-1} + D_n + \frac{T}{2L} \left( 2D_n - \frac{3}{2}v_{n-1} - \frac{3}{2}v_n \right) + \frac{V_{REF} - v_n}{L} \left( 2D_n - \frac{3}{4}v_{n-1} + \frac{3}{4}v_n \right)
\]

\[
- \frac{v_n + v_{n-1}}{4L}.T \quad (19)
\]

(3.) Looking at Figure 5, the estimated inductor current \(i_n\) is presented in (20):

\[
i_n = i_{n-1} + D_n.T.V_{in} - v_n - (1 - D_n).T.\frac{V_{REF} - v_n}{L} \quad (20)
\]

Joining equation (8), (13), (19) and (20), and assuming \(v_n + v_{n+1} \approx 1\), the desired duty cycle expression is obtained in (21).

\[
d_{n+1} = \left( 2D_n - \frac{3}{2}v_n \right) - \frac{V_{REF} - v_n}{L} \left( 2D_n - \frac{3}{4}v_n \right) + \frac{\Delta V_{REF}}{L} \left( \frac{\Delta V_{REF}}{L} \right)
\]

(21)

In equation (21):

\[
\Delta V_1 = \frac{v_n + v_{n-1}}{2} \quad (22)
\]

\[
\Delta V_2 = v_{n-1} - v_n \quad (24)
\]

The term \((-3/2) \times \Delta V_{REF} \times D_n\) can be neglected because it makes no contribution to the transient regime. Due to the fast circuit response, \(V_1\) and \(V_{n+1}\) have so slight change that can be assumed as equal to the reference voltage, and so \(V_1\) + \(V_{n+1}\) is neglected because it is constant. The term \((-3/2) \times \Delta V_{REF} \times D_n\) in the transient response is approximated to \(-V_{in} \times D_n\).

Using the above approximations and rearranging equation (21) in order to make it similar to a typical control equation; equation (25) is obtained.

\[
d_n = \frac{-V_{in} \times D_n}{2} + \frac{2.L.C}{T^2} \left( \frac{3}{2} \times V_{in} + V_{REF} \right) (2.\Delta V_2) \quad (25)
\]

Identifying the constant terms present in equation (25), the final control equation obtained in this work is presented in (29):

\[
K_0 = \frac{V_{in}}{3 \times V_{in} + V_{REF}} \quad (26)
\]

\[
K_0 = \frac{2.L.C}{T^2} \left( \frac{3}{2} \times V_{in} + V_{REF} \right) \quad (27)
\]

\[
K_P = 2 \quad (28)
\]

\[
d_n = -K_D.D_{n-1}^2 + K_P \quad (29)
\]

B. DC-DC Buck Converter with Digital Control

In order to implement a digital control on a DC-DC Buck converter it is easily understandable that there are some hardware needs in order to obtain the digital signals for the control input and the analog control signal for the power block. Since the converter uses the output voltage in the VMC it’s necessary to obtain the difference between the reference voltage and the output sensed voltage. Among some solutions, one is to digitize only the error signal, i.e., the reference voltage minus the output voltage. The disadvantage of this
Introducing a \( L \) with an \( 10 \) \( \text{m} \Omega \) ESR. The \( L \) is going to increase much the transient \( \Delta \) necessary to control, \( K \). \( K \) gives a feedback about the solution \( 37 \) \( 1 \) \( 1 \) \( 1 \). With these values \( 6 \) \( 1 \) \( 1 \) \( 1 \), they were approximated to easily implementable \( t \) is necessary to define the values of each component in the DC-DC implementation. Beginning from the input voltage, \( V \), it's chosen the voltage \( 2.5 \) \( \text{V} \) with an internal resistor of \( 50 \) \( \text{m} \Omega \) in order to make this voltage source more realistic. The \( V \) is chosen to be \( 10 \) \( \text{µ} \)F with a \( 5 \) \( \text{m} \Omega \) ESR and an \( 1 \) \( \text{n} \)H ESL. The switch SW1 is a PMOS transistor and the switch SW2 is a NMOS transistor. Figure 6, represents an implementation of a DC-DC Converter with digital control, showing the blocks position as explain earlier.

![Figure 6 – Digital Buck converter main topology](image)

With these values, equation (29) coefficients can be calculated:

\[
K_0 = 1.86 \quad K_0 = 37.07 \quad K_p = 2
\]

Parameters \( K_0 \), \( K_D \) and \( K_P \) were computed for the DC-DC specs under implementation. However, analysing the values found for \( K_0 \) and \( K_D \), it is noted that they are not easily implemented in a digital control. Therefore, by simulation, they were approximated to easily implementable values.

\[
\begin{align*}
K_0 &= 1.31 \quad \text{turn by mul by 0.75} \quad K_0 &= 0.5 \\
K_0 &= 37.05 \quad \text{and approximate} \quad K_0 &= 16 \quad (30) \\
K_p &= 2 \quad K_p &= 1.5 \\
\end{align*}
\]

In order make equation (29) more generic, equation (31) is introduced, giving additional control to the control speed and output ripple. The new control coefficient, \( K_{\text{lin}} \), is a \( K_D \) and a \( K_P \) function expressed in (32) if implementing the equation equivalent to (29).

\[
D_n = -K_D \cdot D_{n-1}^2 + K_D \cdot (\Delta V(n) - \Delta V(n - 1)) + K_{\text{lin}} \cdot \Delta V(n) \quad (31)
\]

\[
K_{\text{lin}} = K_P \cdot (K_p - 1) \quad (32)
\]

The coefficient \( K_D \) is a differential gain that gives a feedback about the correlation between two cycles, this coefficient controls the output oscillation speed and convergence to the expected value. The coefficient \( K_{\text{lin}} \) is a proportional gain that gives a feedback about the solution convergence. If the voltage error is approximating zero this term also converges to zero. The coefficient \( K_D \) is a proportional coefficient to the quadratic previous duty cycle.

IV. DIGITAL CONTROL IMPLEMENTATION

After obtaining the desired duty cycle general function it is necessary to validate and optimize. In order validate and design the proposed control strategy, a methodology is presented in Figure 7. The DC-DC converter is implemented using two different main programs, the first is Mathworks Matlab and the second is Cadence Virtuso. Another program is used to help implementing the digital control, Xilinx ISE 10.1, and the software that synthesizes the Verilog control code to use in virtuoso is Design Vision from Synopsys.

![Figure 7 – Methodology for digital control validation](image)

A. Matlab Implementation

The Control Module only has one input port, \( \delta \text{ta} \text{V} \text{dig} \), and an output port, \( D \). However, for simulation purposes the output ports Int, Prop, Diff and Quad are also introduced. In order to implement the equation, there was necessary to make some modifications such as the saturation and the integration block inclusion.

In order to minimize the converter static error, an integration block is introduced that integrates the error voltage. This integration block is made by introducing a register regint and an adder with a saturation block. A control coefficient, \( K_i \), is also introduced weighting the integrator error voltage in the duty cycle calculus.

Looking at schematic of Figure 8, it can be seen that some saturation blocks were introduced. These blocks are needed to discretize and saturate the values that must be representable in the digital implementation. Additionally, saturation SAT0 and SAT2 were introduced because a number greater than one and lesser than minus one is going to increase much the transient time. Saturation SAT3 and SAT5 were introduced to limit the
maximum and minimum represented digital value. This is done to cancel the circuit representation overflow risk. SAT6 turn the control output value real duty cycle between one and zero. SAT7 was introduced in order to discretize the obtained value. However, in the digital implementation this saturation was not necessary because $K_o$ divides by two the multiplied value which is always represented in the digital word. This module has also a trigger that informs Matlab when to recomputed synchronous signals.

![Figure 8 – Buck Converter Control Block schematic implemented in Matlab](image)

Figure 8 presents the Buck converter with the necessary blocks to test it, such as the control block, the buck block, the ADC and others.

![Figure 9 – Buck Converter Control Block schematic implemented in Matlab](image)

Figure 9

### High-Level Design:

The implemented Matlab block is a good design start, because it can be easily translated into a block diagram that helps to design the high-level representation. Top module inputs are the error voltage, labelled deltavin, the reset and setup signals, the insetupvector, which selects the control coefficients, and the clock signal, clk. The top module output is the duty cycle, D.

Looking at Figure 9, the modules that are needed to design are:

- SBLOCKS0, SBLOCKS1, SBLOCKS2 and SBLOCKS3 – These modules have as inputs a 12-bit vector and a 3-bit select vector. This select vector enables to change the multiplying coefficient. The various modules have different multiplying coefficient scales;
- INTBLOCK – This module receives the integration value from the integration register and adds it to the error voltage that is multiplied by the integration coefficient. It also verifies if there is saturation;
- RCBLOCK – This module adds two 12-bit numbers and verifies if there is saturation. This module saturation values are minus two and two;
- RCOBLOCK – This module adds two 12-bit numbers and verifies if there is saturation. If it exists, the given saturation value is outputted. The saturation values are zero and one;
- SUBLOCK – This module subtracts one 12-bit number from other number with equal length and verify if there is saturation. If it exits the given saturation value is outputted. This module saturation values are zero and one;

With the constants express in (32). Another important definition in a digital circuit is the input, output and intermediate calculus resolution scale and limits, because these parameters will have a direct influence on the design. It is important to know that the circuit working frequency was specified as 2 MHz.

From the duty cycle signal definition it cannot be greater than one or lower than zero, so the output is between those limits. The input signal has 1 mV resolution because a better resolution would not influence the final result. The intermediate calculus as 1 mV precision and a scale limit of two and minus two. In order to standardize, the input value will also have the same scale limit that has the intermediate calculus, and so they are represented by a word with 12 bits fixed point. The output signal is limited between zero and one, so it has a 7-bit resolution with fixed point. The word representation and each bit ponderation are represented in Figure 10.

![Figure 10 – Digital word representation](image)

### Specification:
The control equation is already known, represented in (31)
• MULBIT – This module received a 6-bit input number and outputs the square input number. The output number has 12-bit length. It is necessary to say that this block does not receive a 7-bit number but a 6-bit number;
• AND_12bto1 and AND_7bto1 – These modules receive 12-bit, or 7-bit, operand and a 1-bit operand. They make a AND between each bit of the 12-bit operand and the 1-bit operand. These modules are an addition to the circuit in order to implement an instantaneous reset in some variables;
• SAT0 – This module is created to turn the input signal between one and minus one.

It is necessary to define some registers that are needed to the control module, they are all inserted in the top-module. Four 3-bit registers are required to save the coefficients KI, KO, KPLIN and KD, two 12-bit registers to save the earlier error voltage and the error voltage integration value, and a 7-bit register to save the early duty cycle. Figure 11, represents the high-level design. It is also the top control module illustration implemented in Verilog.

![Digital Control High Level Design](image)

**Low – Level Design**

• SAT0: This module is a saturation to saturate the input error voltage;
• SBLOCKS0, SBLOCKS1, SBLOCKS2 and SBLOCKS3: The SBLOCKS objective is to multiply an input number A by a coefficient that is selected with a setup vector. These coefficients are all two powers so the multiply can be seen as an arithmetic shift. This solution is better than implement a multiplier because it would imply a greater number of gates. The 2-bit multiplexer select is obtained from another multiplexer that chooses the select signal according to the 3-bit SEL signal. This is necessary because the saturation depends on the input signal and the multiplication coefficient. The multiple select signals are easy to obtain by combinational logic between the input signal and a given SEL signal;
• INTBLOCK, RCBLOCK and RCOBLOCK: These modules receive two inputs, A and B, and output the sum O. They also receive a reset signal and a reset number. The INTBLOCK interacts with an out-module register to simulate an integration.

Often, the solution that occupies less area and consumes less power is the better, but it must obey to the circuit speed requirement. In this case, the control working frequency is 2 MHz and for this is necessary to previse the speed necessary to the adders to accomplish the desired circuit working frequency. In this case, the RC solution is considered appropriate due to the circuit specifications. The main differences between each block are represented in Table 3;

• SUBLOCK: The SUBLOCK is a module that receives two positive numbers and subtracts the B input number from the A input number, and if necessary saturates the output. This module is implemented using a RC with a module that makes the two complements of the input B number. As the earlier modules, it also receives a reset signal and a reset number. The SUBLOCK multiplexer saturation limits are the representation limits;

• MULBIT: This module objective is to to calculate the power of two of an input number Din. This module input doesn’t use 7-bit because multiplying a 7-bit with a 7-bit number would generate a 14-bit number and our architecture is done with 12-bit. Mulbit receives the six more significant bits of the duty cycle. The multiplication can be made with three different approaches, using a multiplication algorithm, a look-up table or using combinational logic to generate the output word.

The multiplication algorithm is almost always based on the shift and adds operations, this means partial products are generated and added like when making a hand multiplication. These algorithms can be implemented sequentially, serially or in parallel. There are speed-up techniques that can be used in order to reduce the multiplication time.

Another way to implement a multiplication is using a look-up table. This is done using a memory that for each position saves the multiplication result. This is the fastest implementation because it only takes the memory access time to know the result. However, it's necessary to know all the possible results previously.

Other way to implement a multiplication is making a table with the multiplication results and, for each bit, to generate the combinational logic from the input operands. This, usually, is not a good implementation because it takes much design effort.

In the present case, the circuit needs to square a 6-bit number, so the best choice is a look-up table, because of its simplicity, or to use combinational logic generating each result bit, because it takes less logic than a multiplication circuit. The chosen implementation is the combinational logic. For each result bit it was generated the combinational logic based on the input value. These equations were obtain with the help of Karnaugh Minimizer Pro, a program to simplify and generate the Boolean expression of each function truth table;

• AND_12bto1 and AND_7bto1: These modules objectives are to implement an instantaneous reset in some variables. They implement a bit a bit AND with two given inputs one of 12-bit, or 7-bit, and the other with 1-bit, this is usually the inverted reset signal. This is implemented using twelve AND ports or seven AND ports respectively.

**C. Matlab/Modelsim Co-Implementation**

To validate the implemented Verilog code in a Buck converter simulation it is possible to use the early Matlab implementation with some modification. In order to use the
Verilog code with Matlab it is necessary to use the program Modelsim and a Matlab block called HDL co-simulation. This block interacts between a Modelsim library and Matlab. It is also necessary to insert some data type converters in order to convert the input and output HDL co-simulation values.

D. Cadence Implementation

The next step in the validation of the designed control consists on implementing the model in Cadence Virtuoso. The design models include some ideal components that are not this work main objective. It is necessary to implement the modules designed in Matlab and also a digital to analog converter. This new model allows the validation of the control with an electrical simulator. It’s necessary to implement the modules designed in Matlab and also it’s necessary to design a DAC.

An ADC is a block which converts continuous signals into a digital number, in a discrete range. This block converts an analog differential voltage, i.e. it receives a difference between the reference voltage and the output voltage. The differential voltage module is obtained in order to compare it to a voltage given by a multiplexer. The multiplexer has at its inputs twelve different voltages that go from zero to 64 mV. This value is large enough to the control system since the working frequency is high enough to accompany the converter, i.e. it is not important to know if the output voltage differs to much from the reference voltage but only if it differs in the presented range. If the error voltage is bigger than 64 mV the result is the equivalent, that is switch SW1 conduct in all the duty cycle or non-conduct in all the duty cycle, depending on if the error voltage is positive or negative.

Therefore, the requirements for the ADC are: Low power (in order to limit the global control consumption); high accuracy near the objective error voltage; high speed in order to evaluate the output voltage as close as possible of the end of the control cycle. These requirements drove the design of an ADC with the conversion characteristic presented in Figure 12 and the structure presented in Figure 13.

![Figure 13 - Analog to Digital Converter schematic implemented in Matlab](image)

The DAC was implemented using a topology called Current steering, i.e. that for each bit it is associated a given current value and if this bit is on the current associated to this bit flows through a resistor generating the correspondent analog voltage. If all the bits are with the logic one value, the DAC outputs its maximum voltage because the current flowing through the resistor has its maximum value. On the other hand, if all have zero, it outputs zero voltage because the current flowing through the resistor is zero. Figure 14 represents a schematic that illustrates the implemented DAC.

![Figure 14 - DAC illustration of Cadence implementation](image)

V. ANALYSIS AND RESULTS

A. Matlab Simulation

The first simulation step aims to confirm the obtained control law using Matlab. The coefficient values substitutions explained in the previous chapter were also simulated. Figure 15 presents a Matlab simulation that illustrates the output voltage and the error voltage with the original coefficient and with the implemented ones.

![Figure 15 – Simulation in Matlab with the calculated and approximated coefficients](image)
This figure shows that the new coefficients improve the control, decreasing the convergence time and also reduce the output voltage ripple. Combining these facts to the implementation simplification, the chosen implementation permits coefficients configuration that implement the values selected Section IV.

Several simulations were made to validate the new control law. In particularly, a simulation for a reference voltage of 1V requiring a 1A current load, as shown in Figure 16. This figure shows the duty cycle variation, the inductor current, the load current, the reference voltage, the output voltage, the “analog” error voltage and the discretized error voltage.

B. Matlab and Modelsim Co-simulation

This implementation is an intermediate implementation between the Matlab and the Cadence implementations. It serves to test the circuit recursively, but now, using it to control an almost ideal DC-DC converter.

C. HSIM Simulation

The implementation on Cadence was needed to generate the netlist for HSIM simulator. This simulation includes the devices models for a given technology, in this case AMS 0.35µ, and it is more realistic than the Matlab and Matlab/Modelsim co-simulation. It does not require the additional register as the co-simulation, using the circuit at is full performance. Figure 18 presents the duty cycle output voltage, error voltage and inductor current for a reference voltage of 1V and a load current of 1A.

Comparing the simulations with previous models, it can be seen that the control obtains an error voltage lower than 10mV for all the tests made. In fact, some of the obtained results are consequence of the non-linear ADC use. This ADC, when trying to digitize an error voltage of 17mV, will digitize the binary number 16mV or 24mV. This can have a negative effect in the control, but still the control gives good results due to the present controller. Having a better ADC, it is believed that it would decrease the error voltage and increase the circuit duty cycle stability.

D. Line and Load Transients

Line and load transient show how a power supply respond to abrupt changes in line voltage and load current.

In the transient simulations the important variables are the overshoot-peak-time, i.e. the time the circuit takes until it reaches the maximum peak value, the overshoot value and the establishment-time, i.e. the time the circuit takes to achieve a ripple below 1% of the reference voltage. For the present circuit, the transients analysed are the line variations between an input voltage of 2.5V to 3.3V and from 3.3V to 2.5V, for a reference voltage of 1V. Some load transients, which the maximum is from a load of 0.1A to 0.9A, and backwards, for a reference voltage of 1V.

A line transient is a sudden variation in the circuit input voltage. In this case a variation of 800 mV was considered. Figure 19 present the simulations for a line transient on the input voltage from 2.5V to 3.3V for the 1V reference voltage, and in Table 1 are presented the primary features of this simulation.
A load transient is a sudden variation in the output load. In this case the load changes transitions between 0.1A - 0.2A, 0.2A-0.5A and 0.1A-0.9A. These variations were chosen in order to analyse a light, medium and hard variation in the load. For a 1V reference voltage the load transients in HSIM are presented in Figure 20, and the primary features are presented in Table 2.

Analysing the 1V reference voltage transients, there are not many differences between the several models. The overshoots maximum voltages are always less than 5 \text{us} with an overshoot always less 15\% and a 1\% establishment-time less than 30 \text{us}. For the 1.5V simulation the overshoot-peak-time increases as the load transient is higher but not being bigger than 5 \text{us}, the overshoot increases as well, being smaller than 10\%, and the establishment-time increases as well.

Comparing the results for different reference voltages as the reference voltage rise, the overshoot load transition time increases, but not significantly, the overshoot decays and the 1\% establishment-time decreases.


table 2 - Primary features of Matlab, Matlab/Simulink Cosimulation and HSIM Load Transients when requiring a V_{ref} = 1V

<table>
<thead>
<tr>
<th>Vin(V)</th>
<th>ILoad(A)</th>
<th>MATLAB</th>
<th>HSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>1.5 us</td>
<td>1.1 us</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>1.3 us</td>
<td>1.5 us</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>1.3 us</td>
<td>2.5 us</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2</td>
<td>5 us</td>
<td>3 us</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>2.3 us</td>
<td>2.3 us</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>1.9 us</td>
<td>2.3 us</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>1.2%</td>
<td>0.93%</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>3.22%</td>
<td>3.4%</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2</td>
<td>2.87%</td>
<td>3%</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>11.5%</td>
<td>10%</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>9.46%</td>
<td>9.6%</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>1.7 us</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>1.1 us</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>6.4 us</td>
<td>6.3 us</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2</td>
<td>7.6 us</td>
<td>10 us</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>17.9 us</td>
<td>10 us</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>21.9 us</td>
<td>26 us</td>
</tr>
</tbody>
</table>

E. Line and Load Regulations

The power supply ability to maintain its output voltage, given changes in the input line voltage, it is called line regulation. Several simulations were made for different loads, 0.1A and 1A, to achieve an estimated line regulation curve for a constant reference voltage. The input voltage changes and several points were taken, the output voltage was measure and its mean value was obtained for a time window of 40\text{us} in the different models. Figure 21 the line regulations curves.

Analysing the line regulation for a reference voltage of 1V, it maintains the output voltage always with ripple less than 1\% which is a good result. The worst scenario is obtained when requiring a load current of 0.1A, but still, it maintains the overshoot peaks less than 5 mV.
The ability of an output channel to remain constant, given changes in the load, is called load regulation. These simulations were made for a reference voltage of 1V with an input voltage of 2.5V. In Figure 22 is presented the load regulation.

![Figure 22 - Matlab, Matlab/Simulink and HSIM Load Regulation for a reference voltage of 1V](image)

The different models present very similar and good results. The load regulation maximum error is 1 mV

VI. CONCLUSIONS

This MSc work main conclusion is that it is possible to design a digital control based on voltage sensing that emulates a current mode control with features such as:
- Steady state error voltage less than 10 mV for a load range of 0.1A to 1A;
- Line transition from 2.5V to 3.3V with overshoot less than 10% of the reference voltage and a establishment time less than 25 us for a constant load between 0.1A and 1A;
- Load transient from 0.1A to 0.9A for a constant input voltage of 2.5V with overshoot less than 10% and establishment time less than 30us;
- Line Regulation equal to 300 μV/V, for reference voltages 1V and 1.5V, with ripple less than 0.6% from 1.8V to 3.3V;
- Load Regulation equal to 0.2 μV/mA, for reference voltage of 1V and 1.5V from 0.1A to 1.1A with ripple less than 0.2%.

These results were obtained with the control implemented in Verilog HDL at RTL and tested with HSIM simulator, after a previous test with Matlab and co-simulation between Matlab and Simulink. They also obey to the circuit specifications.

The Power Management World may incorporate much more digital solutions in the next few years, and I hope that this work is humble but still a contribution to this fact.

FUTURE WORK

The world of digital control for DC-DC converters is huge and in constant expansion. Much can be done and a number of investigations prove this constant evolution. In this matter, there are some important things that can be done, such as a theoretical analysis of the control stability, an analysis in terms of small signal behaviour and a similar study in the discontinuous conduction mode.

The implemented analog DC-DC modules should be replaced by a more realistic model in order to obtain a full functional converter that can be implemented in silicon. The ADC should be analysed because it can produce a dreadful influence in the control behaviour, this means that it is believed that this type of control can work even better than the demonstrated but for that it is necessary a better ADC. The DAC and the PWM modules could also be replaced by a DPWM circuit.

Finally, to implement the obtained circuit using an FPGA is a low-cost way to prove the circuit functionality and it would be an interesting achievement and a practical way to evaluate the performance of the implemented digital control.

REFERENCES