AUTO-COMPENSATION METHOD FOR A BUCK CONVERTER

Toma PATARAU Radu ETZ Dorin PETREUS
Technical University of Cluj-Napoca, Cluj-Napoca, Romania
26-28 G. Baritiu Street, 400027 Phone +40264202348, E-mail: toma.patarau@ael.utcluj.ro

Abstract: The paper proposes a method to compute the passive components values of a synchronous buck converter. The method is implemented on a step down converter using a DSC(digital signal controller) F28027 from Texas Instruments. A third order IIR filter is used for compensation and the coefficients are computed automatically based on the components determined values. The algorithms for computing the inductance, capacitance and capacitor parasitic resistance are presented in detail with flow charts and simulations. The software is then transposed in code composer and the algorithm is tested on the experimental setup. The algorithm once implemented doesn't need recalibration of the compensator coefficients made by an operator in case of a hardware change in the power circuit.

Keywords: *auto-compensation, digital control, step down converter.*

I. INTRODUCTION

Over the past decade, control systems gained a very important role in the evolution of our society as well as in the development of modern technology. Practically, every daily activity of our lives is affected by some type of control system.

Over the past few years, in electrical engineering field, research focused its attention on the numerous benefits that digital control can offer when referring to dc-dc power converters: immunity to noise, programmable and low sensitivity to ageing and environmental factors.

Auto-compensation is a technique that specifies a way to determine automatically the parameters of the power circuits and once these parameters have been calculated, the control coefficients can themselves be automatically computed using digital circuits. The major achievement is that the loop is stable even if the parameters have a large variation.

By using auto-compensation techniques, the compensation coefficients can be calculated taking into account the values for the phase margin and for the bandwidth. To achieve this purpose, a specified frequency is injected in the control loop or nonlinearity is added/amplified so that in the output voltage appears limit cycle oscillations [1]-[5]. In [1], after the limit cycle oscillations (LCO) effect is obtained, it is amplified by replacing the digital compensator with a PI configuration and so the frequency and the amplitude of the resulting LCO are measured. This information is used to design a proper PID.

Auto-compensation can be also accomplished by introducing a relay into the control loop, [6], [7]. It will cause LCO at the output that will oscillate at the resonant frequency of the circuit. This frequency is measured and based on it the first zero of the PID compensator is placed. The phase margin and crossover frequency requirements are respected because the PID controller is passed through a low-pass filter. The next zero is then placed until the output oscillates at the crossover frequency. The gain of the compensator is then set by using the information stated above and after the tuning process is over, the relay is turned off allowing normal loop operation. As shown above, only the frequency is monitored and measured at this tuning method contrary to articles [1]-[5] that monitor the amplitude too. This is the major advantage of this auto-tuning method.

Other methods have been developed, that do not introduce oscillations at the converter output [8]-[11].

In [12] an adaptive tuning system is implemented on a digitally controlled converter that is able to measure the crossover frequency and the phase margin using the stability margin monitor. The accuracy of the stability margin monitor has been tested on a buck converter because of the fact that the transition from DCM to CCM or vice versa can result in a decrease in loop the stability margins. Methods to resolve this problem were developed.

In [13], another auto-compensation algorithm is developed based on frequency response measurement. The algorithm is implemented on a DC-DC converter and controlled by a digital PID compensator. The parameters of the PID controller are determined by respecting the specified stability margins, integral no-limit-cycling criteria and on the other hand by ensuring operation at the crossover-frequency and high gain at low frequencies.

This paper proposes a method that computes the compensator coefficients for a step down converter based on the passive components values. These components values are determined using a DSC using the methods described in section II. The advantages of the proposed method are: i) it can be implemented on any second order system, ii) doesn't need any compensator being designed by the user, iii) low cost implementation. The algorithm is implemented in C language and tested in Psim. In section III the experimental results are presented and in section IV the conclusions are drawn.

II. COMPONENTS DETERMINATION

The auto compensation method proposed is based on computing the parameters involved in the compensation algorithm. The DC-DC converter is a voltage mode control step down converter. The control algorithm is a digital IIR

Manuscript received May 4, 2012; revised May 22, 2012

filter implemented on a DSC. The coefficients of the filter are computed based on the angular frequencies of the poles and zeros. The angular frequencies are correlated with the values of the step down converter's passive components as presented in [14]. By using the proposed method, the user needs to specify only the information of the reference voltage. The step down converter must have a measure point for the output voltage, one for the input voltage and a current sensor for the inductor current. These three values are fed to the input of the analog to digital converter (ADC) to compute the values of the parasitic capacitor resistance (ESR), the capacitance of the output capacitor and the inductance of the output inductor.

Based on the determined values the angular frequencies are computed and the IIR filter coefficients are updated.

In the next section the methods used to determine the values of the passive components are explained. Also for each method a software flowchart is presented.

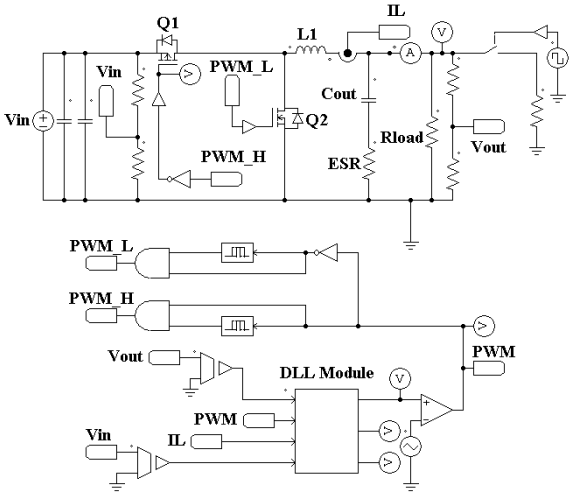

Figure 1. Synchronous step down converter schematic with digital controller.

The algorithms implemented use the same schematic presented in figure 1. The power topology is unchanged but the DLL block that simulates a generic digital controller has different functionality based on the C software implemented for each passive component computation.

A. Computing the Inductance

The inductance can be determined by computing the voltage on the inductor (1) during the transistor OFF state in continuous conduction mode (CCM).

$$U_L = L \cdot \frac{di_L}{dt} \tag{1}$$

During the OFF state the value of the inductor voltage is equal to the output voltage, and di_L is approximated as the difference between the sampled values of the inductor current *(2)*.

$$L = \frac{V_{out} \cdot (1 - dutyV) \cdot T_S}{I_{L2} - I_{L1}} \tag{2}$$

where dutyV is the duty cycle of the pulse width modulated (PWM) MOSFET driving signal, T_s the switching frequency and I_{L1}, I_{L2} the inductor current samples.

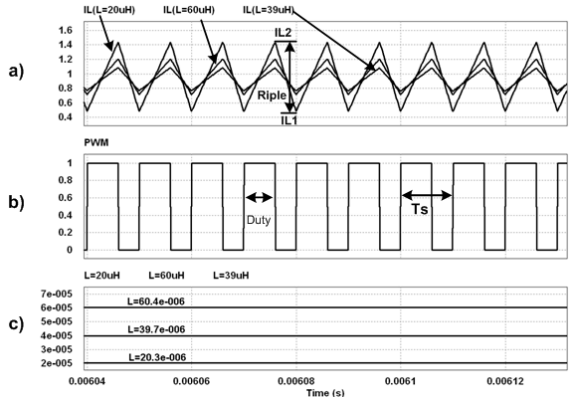

Figure 2. Waveforms for inductance computation.

During the time needed to determine the inductance, the duty cycle is maintained to a constant value equal to 50%. I_{L1} is sampled at the beginning of a switching period, T, and I_{L2} at DT where D is the duty cycle. In figure 2 a) the ripple of the inductor current is presented in three cases: when the inductance has a value of 20uH, 39uH and 60uH. As presented in figure 2 the values of the inductor ripple are sampled based on the duty cycle value, figure 2 b), and using *(2)* the inductance is computed and presented in figure 2 c). It can be observed that the computed values are very close to the values of the passive components considered.

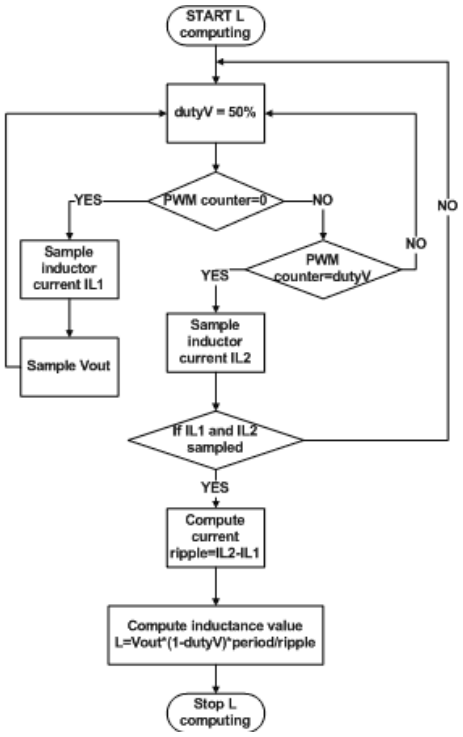

Figure 3. Software flowchart for the inductance computation.

Figure 3 presents the flowchart of the software used in the DLL block used to compute the inductance value. The algorithm first initializes the duty cycle to a known value and keeps it unchanged during the time needed to determine the inductance. The samples of the inductor current are measured at specific time periods based on the PWM state.

When all the unknown variables, I_{L1}, I_{L2}, V_{out}, are sampled the current ripple is computed and the inductor value is updated based on *(2)*. In the simulation software the value of the inductor was fed to one of the DLL's output for debug purposes.

B. Computing the ESR

The ESR is also determined using the sampled values of the output voltage and inductor current.

The output voltage ripple has two components one determined by the value of the capacitor and one determined by the ESR. In most practical situations, because the value of the ESR is large enough, the voltage ripple is entirely determined by its value and can be expressed as *(3)*.

The information about the current ripple was computed and used to obtain the value of the ESR.

$$\Delta V_{out} = ESR \cdot \Delta I_L \tag{3}$$

$$ESR = \frac{\Delta V_{out} \cdot L}{(V_{in} - V_{out}) \cdot Duty \cdot T_s} \tag{4}$$

where ΔI_L is the inductor current ripple, ΔV_{out} is the output voltage ripple and T_s the switching period.

Both equations (*3*) and (*4*) can be used to determine the value of the ESR. Equation (*4*) has the advantage that it can determine the ESR without measuring the ripple of the inductor current.

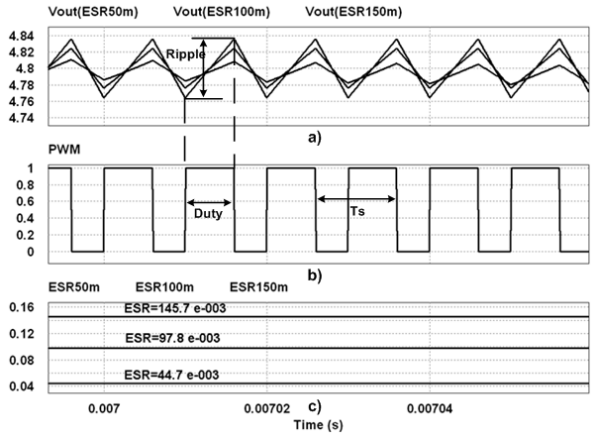

Figure 4. Waveforms for ESR computation.

The output voltage ripple is computed from the samples from the ADC triggered by the PWM counter value. The voltage is sampled at the same time as the inductor current. In figure 4 a), the waveforms for the output voltage are presented in three cases for an ESR of 50mΩ, 100mΩ and 150mΩ. The computed ESR is presented in figure 4 c). The same schematic from figure 1 is used and the flowchart from figure 5 is implemented in the DLL.

The ESR value influences the response of the step down converter by modifying the position of the zero thus it is important to determine its value online because the value specified by the capacitor supplier may change significantly being influenced by the environmental factors and the production process.

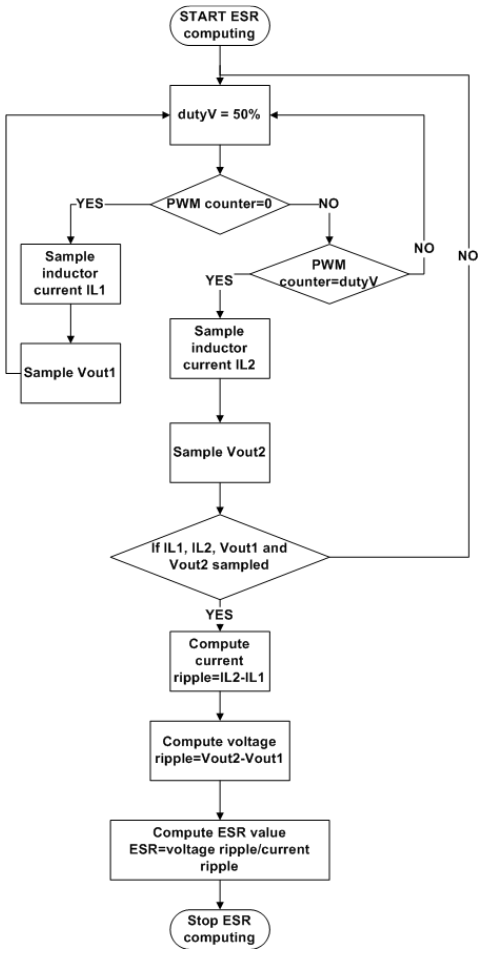

Figure 5. Software flowchart for the ESR computation.

C. Computing the Capacitance

The capacitance of the output capacitor influences the zero and the pole of the step down converter and its value is important in order to compute the compensator coefficients.

The proposed method is based on the oscillatory response of the converter. A step is introduced in the duty cycle (from 20% to 80%) to trigger the oscillatory response. Because the converter is a second order system the response of the converter can be used to determine the output capacitor value by measuring the frequency of the oscillations obtained, figure 6.

The switching frequency is maintained unchanged but the duty cycle is varied between two values in order for the LC filter to oscillate, (*5*).

$$f_{LC} = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}} \tag{5}$$

The duty cycle is kept to a low value for a couple of milliseconds and then it is changed to a high value. The

output voltage is monitored and extracted from the reference corresponding to that duty cycle. The result is a signal like the one in figure 6 where T_d is the damped resonance period and per is half of T_d.

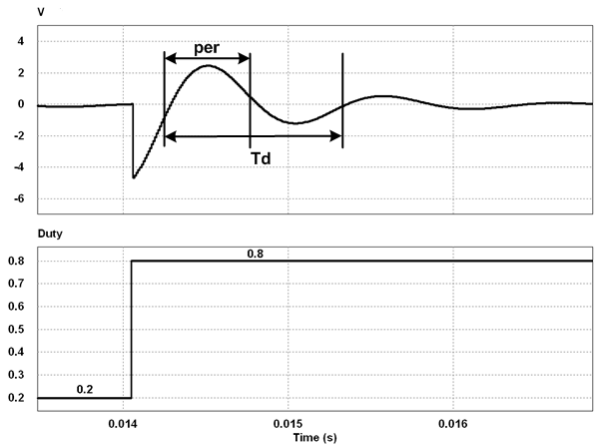

Figure 6. Output voltage at duty cycle change from 20% to 80%.

The value "*per*", figure 6, is computed in the DLL block. The output inductance L, the output capacitance C and the ESR form a RLC series circuit.

The damping factor, *(6)*, is influenced by this three values and the damped resonance frequency ω_d, *(9)* depends on this value and on the value of the undamped resonance frequency ω_0 *(7)*. By substituting *(6)*, *(7)* and *(8)* in *(9)* and by rearranging the terms the capacitance is computed using *(10)*.

$$\zeta = \frac{ESR}{2} \cdot \sqrt{\frac{C}{L}} \tag{6}$$

$$\omega_0 = \frac{1}{\sqrt{L \cdot C}} \tag{7}$$

$$f_d = \frac{1}{T_d} = \frac{1}{2 \cdot per} \tag{8}$$

$$\omega_d = \omega_0 \cdot \sqrt{1 - \zeta^2} \tag{9}$$

$$C = \frac{1}{\frac{\pi^2}{per^2} \cdot L + \frac{ESR^2}{4 \cdot L}} \tag{10}$$

Figure 7 presents the waveforms when the duty cycle is changed from 20% to 80% for three cases when the capacitance is 200uF, 680uF and 1500uF and the values of the capacitance computed in the DLL block using the relation from *(10)*. One can observe that the value of per in figure 6 changes in figure 7 depending on the capacitance value as described in equations 6-9.

It can be observed that the values computed are similar to the values of the capacitors considered.

From (*10*) it can be observed that the capacitance value is dependent on priori knowing the ESR value and the inductance value.

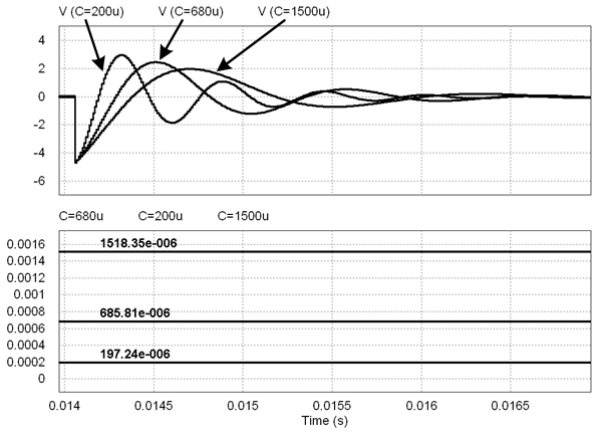

Figure 7. Waveforms for capacitance computation.

Because of this dependence, the software flowchart implemented in the DLL, figure 8, depends on the other two flow charts presented in figure 3 and figure 5.

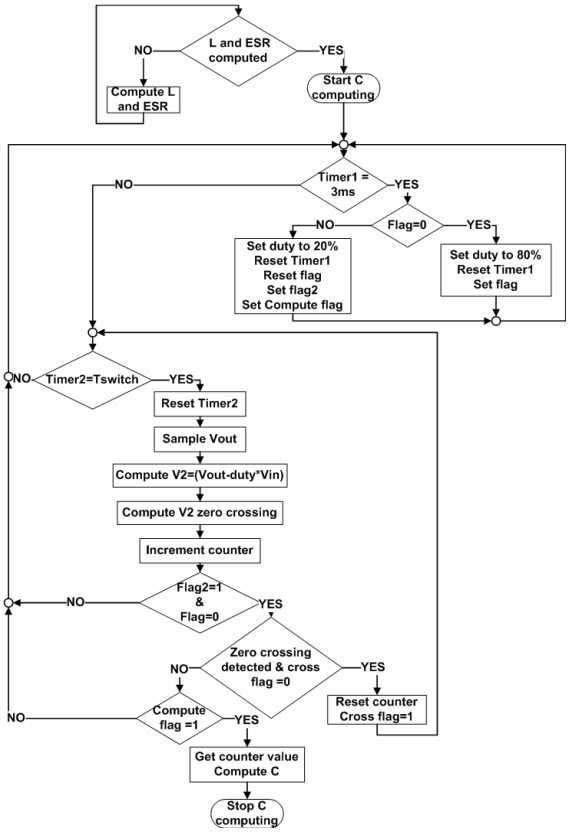

Figure 8. Software flowchart for capacitance computation.

As figure 8 shows, the value of the capacitance is computed when the duty is changed from 20% to 80%. The variable "flag" is used to determine the current value of the duty (20% or 80%). The variable "flag2" is used to detect

the zero crossing only when the duty is set to 80% and the "flag compute" flag is used to start the computation of the capacitance.

By using all the three software flowcharts together, an algorithm for detecting the passive components of the step down converter is implemented in the DLL block. This software has six sequential tasks. The first task is to set a constant duty cycle. The second task of computing the inductance starts after the transient response has passed. It is followed by the task of computing the ESR. For the first three tasks the duty was kept constant but for the fourth task the duty is changed between two values in order to compute the capacitance value. After the values of the inductance, the ESR and the capacitance are computed and saved. The angular frequencies of the power supplies poles and zeros can be determined and the coefficients of the digital filter can be computed in task five. In task six the controller is set and the compensator now sets the duty cycle value and updates it on every switching cycle [14].

The waveform of the output voltage with the tasks highlighted is presented in figure 9.

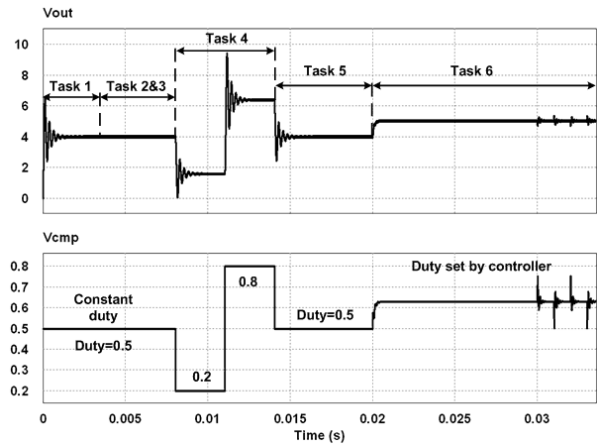

Figure 9. Output voltage and duty cycle value during tasks.

In the last part of task six the output load changed with a fixed period of time (1 ms) drawing a current of 500mA and 1A.

The digital controller implemented with the computed coefficients based on the values obtained during tasks 1-5 computes the duty cycle every switching cycle and keeps the output voltage to the value set by the reference.

In figure 10 are presented the values of the inductance, capacitance and ESR computed and used by the algorithm to determine the values of the angular frequencies of the poles and zeros and the coefficients of the digital compensator.

The values of the passive components used in the circuit are shown in the first row in table 1 and the computed ones in the second row.

Table 1 The values of the components

Passive component	Inductance	Capacitance	ESR
Circuit values	47 uH	36 uF	220 mΩ
Computed values	47.37 uH	36.822 uF	217 mΩ

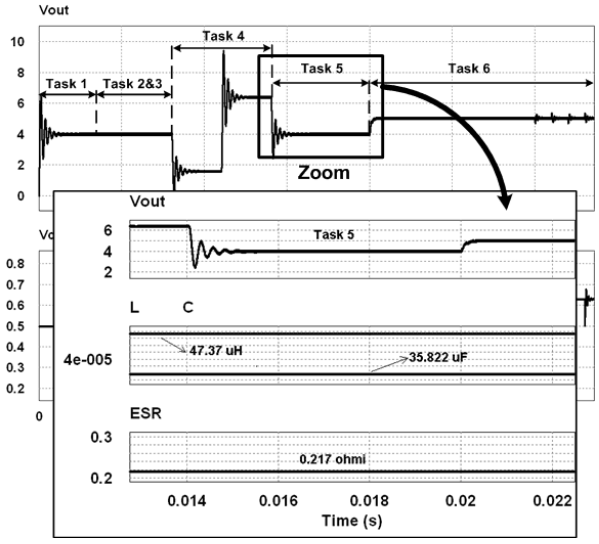

Figure 10.Computed values of the components after task 5.

.

From table 1 the error for the capacitance is 2.2%, for the inductance is 0.7% and for the ESR is 1.38%.

In figure 11, task 6 is magnified and the response of the converter at output load change is observed.

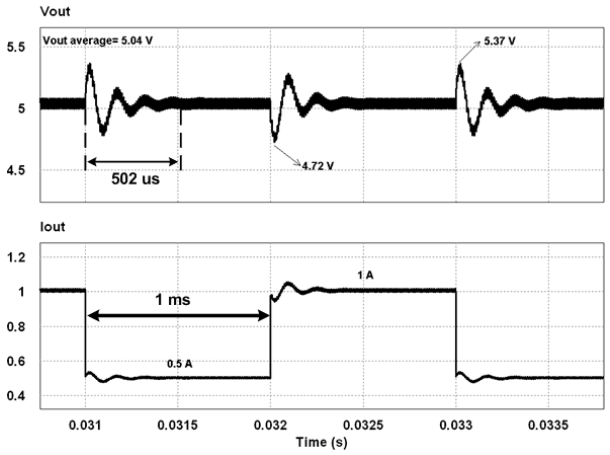

Figure 11. Output voltage at load change.

The time needed by the controller to settle the output voltage of the converter at the value set by the reference is 502 microseconds for the zeros and poles frequencies [14] obtained during task 5, presented in table 2.

Table 2 Angular frequencies of poles and zeros

ω_{p0}	ω_{p2}	ω_{p3}	ω_{z1}	ω_{z2}
3140	147580	314000	24240	24240

One can observe that the frequency of the two zeros is the same, determined by the position of the pole introduced by the output LC filter, distinct from [14] where the frequency of the first pole was set at half of the frequency of the output LC filter.

The algorithm implemented in Psim after the method was

validated, is written to be used on the microcontroller from Texas Instruments.

III. EXPERIMENTAL RESULTS

For the experimental setup, figure 12, a step down converter was used as the power topology and a Piccolo 320F28027 microcontroller for the control. The experimental waveforms were acquired using a Tektronix TDS2000 series oscilloscope and the variable load used was a digital load from Chroma.

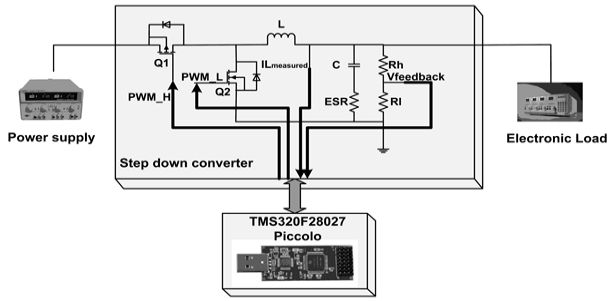

Figure 12. Experimental setup.

The software flowcharts presented in figures 8,5,3 were used when the software for the microcontroller firmware was written. In figure 13 a block diagram of the firmware is implemented. In the main function the software flow is determined by flags. The flags are used to sequentially execute the tasks and to use the information provided by the ADC in the active task. The timer interrupt routine set the active time for each task. The ADC interrupt routine is triggered by the PWM module every cycle and provides the voltage and the current value to the tasks in order to compute the algorithm.

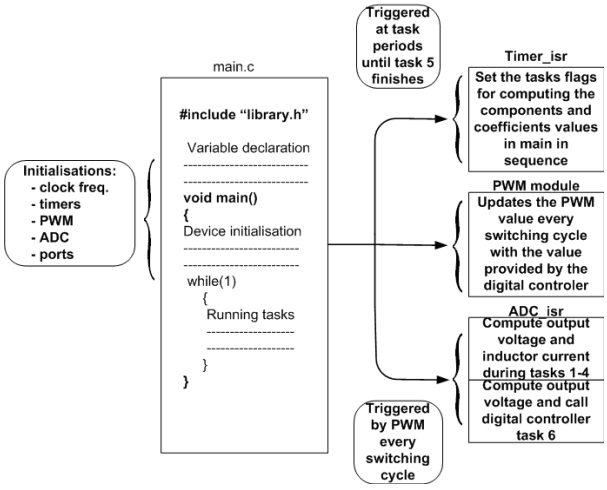

Figure 13. Microcontroller modules block diagram.

The firmware was loaded in the microcontroller and the debug option was used to run the algorithm on the experimental setup. The same waveforms obtained in the simulation were acquired. In figure 14 the first waveform represents the inductor current ripple. The second waveform is a signal that indicates the moments when the current is sampled. When this waveform changes its value from low to high state the current minimum value is saved inside the microcontroller's memory. The next PWM cycle when waveform 2 changes its value from high to low, the current maximum value is saved. With this two values saved in the memory, the microcontroller can compute the inductor current ripple.

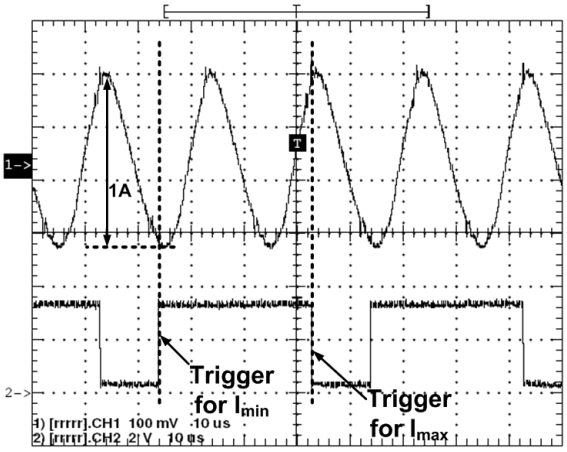

Figure 14. Inductor current ripple.

The output voltage value is read at the same time as the inductor current. This means that the inductor current ripple and the output voltage ripple are computed at the same time and the two values correspond to the same duty cycle value.

The output voltage and the PWM waveform are presented in figure 15. The voltage ripple is 110 mV, the switching frequency 100 kHz and the duty cycle kept constant to 50%.

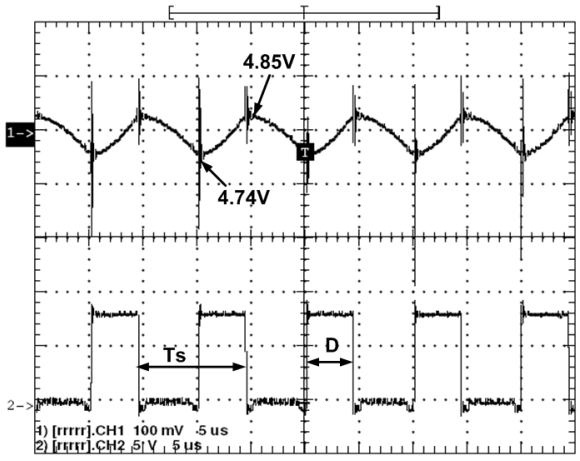

Figure 15. Output voltage ripple.

After these measurements are made, and the two ripple values are computed, the value of the inductance and the ESR can be determined. The flags for the tasks are set and the switching frequency is doubled to have better accuracy for the measurements of the values used in the capacitance computation.

Figure 16 shows the output voltage waveform during the capacitance computation task. Using the algorithm presented in figure 8, the duration of the first overshoot when the duty is changed from 20% to 80% is computed. By counting the number of PWM's periods passed in this period of time the oscillations period can be determined. For the setup used in this case a number of 13 PWM cycles was obtained.

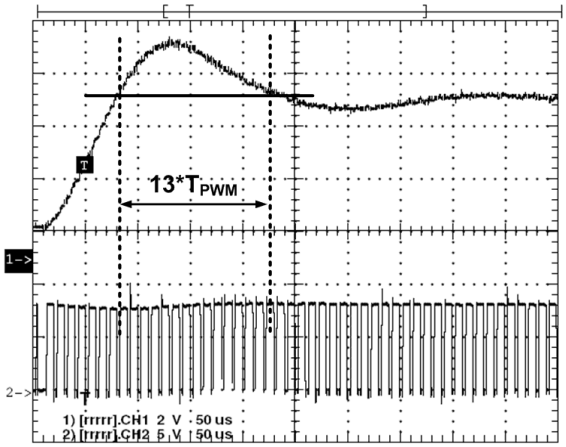

Figure 16. Oscillation period measurement.

The value "T_osc" from figure 17 is obtained after multiplying the PWM period TPWM with the variable retaining the number of PWM periods counted, "count_osc".

Name	Value	Type	Radix
T_osc	0.00013	double	float
count_osc	13	unsigned int	unsigned
TPWM	1e-05.0	double	float

Watch Locals | Watch 1

Figure 17. Watch window with T_osc monitored.

This variable "T_osc" represents the value of "per" in equation 8.

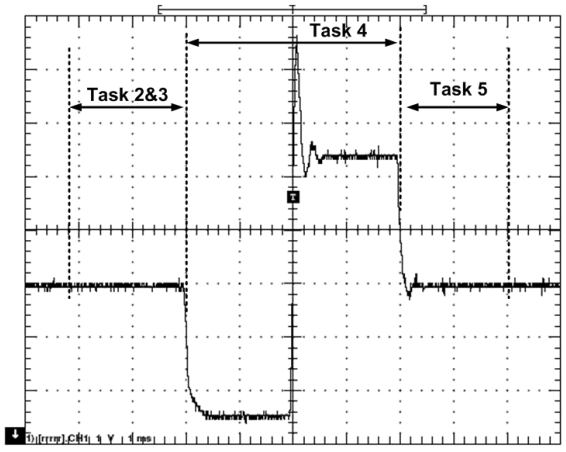

Figure 18. Output voltage during computing tasks.

The counterpart of figure 9 is figure 18. Comparing the two results one can conclude that the firmware loaded in the controller respects the flowcharts of the programs used during simulation. In figure 18 the output voltage of the DC-DC converter during tasks 2-5 can be observed. The change in duty cycle used to compute the value of the capacitance is identified during task 4. The overshoot from task 4 is magnified and presented in figure 16.

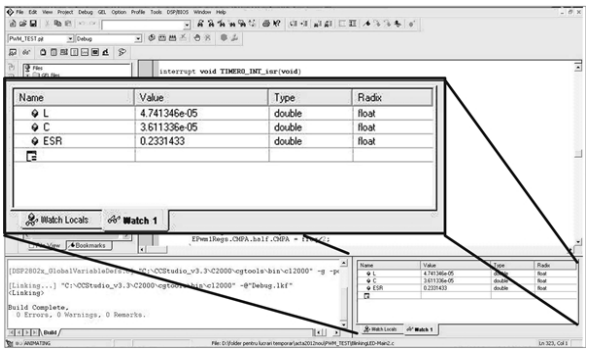

Figure 19. Components values monitored in watch window.

At the end of task 4, the values of the inductance, ESR and capacitance are already computed and saved in the memory of the microcontroller. A screenshot of the watch window during debug mode is presented in figure 19, with the three values. The value of the determined inductance differs by less than 1% from the value of the inductor used in the circuit and the capacitance value with 11%. The value of the ESR is 233 mΩ and in simulation a value of 220 mΩ was used.

By substituting these values in the equations used to compute the angular frequencies of the poles and zeros of the compensator, the values from figure 20 are obtained. These values are very close to the ones obtained through simulation table 2. At the end of task 5 the values of the digital filter coefficients are computed and the values obtained are passed in IQ_26 format to the implemented control loop function. In task 6, the duty cycle is updated every switching period with the value provided by the 3rd order IIR filter. The reference voltage for the output is set to 5V and the ADC now monitors only the output voltage of the converter in order to feed it to the control function. The control method used is voltage mode control and the switching frequency is 100 kHz.

Name	Value	Type	Radix
wz1	24166.58	double	float
wz2	24166.58	double	float
wp0	3379.523	double	float
wp2	144775.6	double	float
wp3	314000.0	double	float
B0	0.3729598	double	float
B1	-0.21213	double	float
B2	-0.3556213	double	float
B3	0.2294685	double	float
A1	0.9383852	double	float
A2	0.09714007	double	float
A3	-0.0355252	double	float

Watch Locals | Watch 1

Figure 20. Angular frequencies and coefficients.

The dynamic response of the digital controlled DC-DC converter is presented in figure 21 at an output current change from 0.5A to 1A at a rate of change of 1kHz. The settling time of the output voltage is 500 μs.

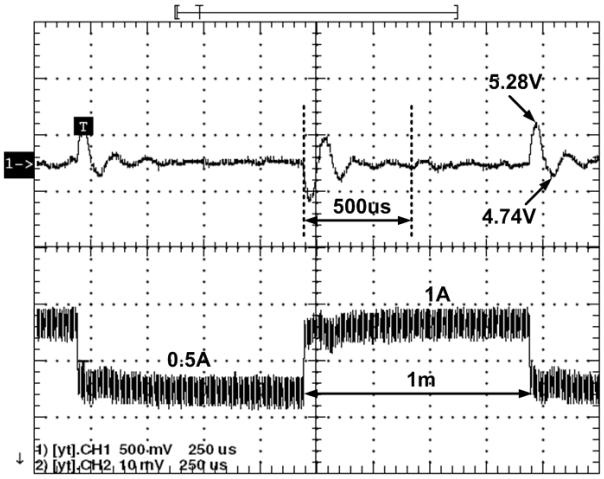

Figure 21. Output voltage at load change.

As one can observe the system is stable and the method proposed to achieve an automated digital controller was reached.

IV. Conclusions

An automated compensation method was proposed for a DC-DC converter that has a second order transfer function. This method is based on computing the values of the components in the circuit that influence the dynamic response of the converter. The method is validated through simulations and experimental results. The control algorithm is implemented in a DLL in C language and transposed in Code Composer Studio to generate the firmware for the C2000 microcontroller from Texas Instruments used in the experimental setup.

As one can observe by comparing the experimental results with the simulations, the differences are under 5%, excepting the capacitance measurement. This is due to the fact that the oscillation period needed to compute its value depends on the number of PWM periods counted. If the period is not an integer multiple of the PWM period, than an error occurs. The error is acceptable because the capacitor used has a variation of 15% from its nominal value provided by the manufacturer due to aging, temperature and stress.

The values of the digital filter coefficients obtained with the experimental setup are similar the one obtained from the Psim program simulations. This can be observed by analyzing figure 11 and figure 21. In this case the response obtained with the experimental setup has a better settling time than the one obtained in Psim. This difference is caused by the errors introduced in the computation of the components taking into account that the temperature and the non idealities of the physical components. The method proposed has the advantage that it doesn't need a very powerful and expensive computation unit to determine the values of the components and also the algorithm once implemented doesn't need recalibration of the compensator coefficients made by an operator in case of a hardware change in the power circuit.

In conclusion, the method proposed is a low cost auto compensation digital controller for a step down converter.

ACKNOWLEDGEMENT

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-20013.

REFERENCES

[1] Z. Zhao and A. Prodic, "Limit-cycle oscillations based auto-tuning system for digitally controlled DC–DC power supplies," *IEEE Trans. Power Electron*., vol. 22, no. 6, pp. 2211–2222, Nov. 2007.

[2] W. Stefanutti, S. Saggini, E. Tedeschi, P. Mattavelli, and P. Tenti, "Simplified model reference tuning of PID regulators of digitally controlled DC–DC converters based on crossover frequency analysis," in *Proc. IEEE Power Electron. Spec. Conf. (PESC)*, 2007, pp. 785–791.

[3] M. Shirazi, R. Zane, D. Maksimovic, L. Corradini, and P. Mattavelli, "Autotuning techniques for digitally-controlled point-of-load converters with wide range of capacitive loads," in *Proc. IEEE Appl. Power Electron. Conf. (APEC)*, 2007, pp. 14–20.

[4] W. Stefanutti, P. Mattavelli, S. Saggini, and M. Ghioni, "Autotuning of digitally controlled DC–DC converters based on relay feedback," *IEEE Trans. Power Electron*., vol. 22, no. 1, pp. 199–207, Jan. 2007.

[5] J. Morroni, R. Zane, and D. Maskimovic, "Design and implementation of an adaptive tuning system based on desired phase margin for digitally controlled DC–DC converters," *IEEE Trans. Power Electron*., vol. 24, no. 2, pp. 559–564, Feb. 2009.

[6] M. Shirazi, R. Zane, D. Maksimovic, L. Corradini, and P. Mattavelli, "Autotuning techniques for digitally-controlled point-of-load converters with wide range of capacitive loads," in *Proc. IEEE Appl. Power Electron. Conf. (APEC)*, 2007, pp. 14–20.

[7] W. Stefanutti, P. Mattavelli, S. Saggini, and M. Ghioni, "Autotuning of digitally controlled DC–DC converters based on relay feedback," *IEEE Trans. Power Electron.*, vol. 22, no. 1, pp. 199–207, Jan. 2007.

[8] V. Yousefzadeh, A. Babazadeh, B. Ramachandran, E. Alarcon, L. Pao, and D. Maksimovic, "Proximate time-optimal digital control for synchronous buck DC–DC converters," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 2018–2026, Jul. 2008.

[9] A. Babazadeh and D. Maksimovic, "Hybrid digital adaptive control for synchronous buck DC–DC converters," in *Proc. IEEE Power Electron. Conf. (PESC)*, 2008, pp. 1263–1269.

[10] E. Meyer, Z. Zhang, and Y.-F. Liu, "An optimal control method for buck converters using a practical capacitor charge balance technique," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 1802–1812, Jul. 2008.

[11] A. Soto, A. de Castro, P. Alou, J. A. Cobos, J. Uceda, and A. Lotfi, "Analysis of the buck converter for scaling voltage of digital circuits," *IEEE Trans. Power Electron.*, vol. 22, no. 6, pp. 2432–2443, Nov. 2007.

[12] J. Morroni, L. Corradini, R. Zane and D. Maksimovic, "Adaptive Tuning of Switched-Mode Power Supplies Operating in Discontinuous and Continuous Conduction Modes", *IEEE Transactions on Power Electronics*, vol. 24, no. 11, Nov. 2009.

[13] M. Shirazi, R. Zane, D. Maksimovic, "An Autotuning Digital Controller for DC-DC Power Converters Based on Online Frequency-Response Measurement", *IEEE Transactions on Power Electronics*, vol. 24, no. 11, Nov. 2009.

[14] R. Etz, D. Petreus, "An Adaptive Digital Compensation Design For Buck Converter Topology", *Acta Technica Napocensis*, vol. 52, no. 2, 2011.