DSP IMPLEMENTATION OF A REAL-TIME BOOST CONVERTER SIMULATOR HANDLING INTER-STEP SWITCHING DELAY

Felipe Dicler*, Matheus Soares*, Gustavo Gontijo*, Thiago Tricarico *, Marcello Neves*, Luis G. B. Rolim†, Mauricio Aredes*

*Laboratório de Eletrônica de Potência e Média Tensão - UFRJ
Av. Athos da Silveira Ramos, sala I-156
Cidade Universitária, Rio de Janeiro - RJ, 21941-594

†Laboratório de Fontes Alternativas de Energia - UFRJ
Av. Athos da Silveira Ramos, salas I-152 e I-154
Cidade Universitária, Rio de Janeiro - RJ, 21941-594

Emails: dicler@lemt.ufrj.br, matheussoares@lemt.ufrj.br, gustavo@lemt.ufrj.br, thiago@lemt.ufrj.br, marcello@lemt.ufrj.br, rolim@poli.ufrj.br, aredes@lemt.ufrj.br

Abstract— This paper introduces the development and implementation of a boost converter real time simulator dealing with inter-step switching delays, which occur due to the different time steps and resulting asynchronism between the digital controller that outputs switch firing signals and the microprocessor that simulates the boost converter. It is shown that, if left unaccounted for, these delays can cause inaccuracy levels such as to make both closed and open loop simulations unreliable. In order to avoid this, a time step changing algorithm is implemented. This algorithm is based on the detection of switching orders and a subsequent correction routine to account for the delay by calculating the system state at the detection moment. The boost circuit is modeled through the differential equations of each possible converter state-configuration (that is, each mode of the circuit). The equations are then numerically solved by Backward Euler method. The real-time platform is composed by two Texas Instruments DSPs, the first one simulating the converter and the other implementing the Proportional-Integral control system. The implemented algorithm works as intended and allows for an increase in the converter simulation time step to values which would previously give incorrect results.

Keywords— Discrete event simulation, real time simulation, inter-step time switching, hardware-in-the-loop

1 Introduction

As the total amount of worldwide electricity consumption is continuously growing, it is essential to ensure that such an increase comes with an improvement in overall system efficiency and a better usage of different energy sources. These demands imply in the development of new technologies to handle increasing power demand and assure system stability and reliability.

In this context, power electronics devices and their control systems are employed in a wide range of applications, such as FACTS (Flexible AC Transmission Systems), HVDC (High Voltage Direct Current) and also in low power driving systems. Before commissioning a controller on a real power system, one should test it in digital platforms in order to achieve three main objectives: (Dinavahi and Iravani, 2001) (Bélanger et al., 2010).

- Evaluate controller performance under open-loop and closed-loop control conditions;
- Evaluate controller performance under steady-state and dynamic system conditions;
- Verify built-in diagnostic and protection strategies.

Off-line transient simulation is not sufficient to achieve these objectives and, therefore, real-time simulation becomes necessary (Dinavahi, 2000).

Before the digital simulation advent, Transient Network Analyzers (TNAs), which were based on scaled down models of analogical devices, were used aiming for real time simulation testing of new control strategies. As the costs were often prohibitive, and due to the emergence of low-cost computers, analog simulators gave way to fully digital real-time simulators. This fact brought new technical challenges unseen in analog simulations.

Interfacing the simulator with its controller is one of the main issues concerning digital real-time simulations because the controller output discrete firing signals may not be (and in fact almost never are) in synchronism with the time step of the simulator. In a physical continuous system, the switches responses would be almost instantaneous, but a digital simulator can only respond at a fixed time grid. As the incoming control signals vary and do not occur at the same instant relative to the time grid, the delay introduced is not constant. The larger the time step of the simulator, the larger is the delay.

Switching delay was first observed in off-line simulations of thyristor based power electronic converters (Dinavahi, 2000), (Manitoba, 2010).

Thyristors models should switch as their currents goes to zero, but if the moment of the current zero-crossing happens between two time steps, the switching is delayed by a fraction of time step.

In order to correct the error introduced by the delay, several approaches have been used for off-line simulations. The most common is to carry out the entire simulation with a small step size, so the error is also reduced, but at the cost of a larger total simulation time (Dinavahi, 2000). Another common approach is to backtrack in time when some event is detected (e.g. the zero-current crossing in the thyristor case), reduce the step size, and reevaluate only this time step with a smaller step size. Both approaches are not suitable for real time applications, due to their high computational efforts.

An approach with smaller computational cost is to use linear interpolation in order to obtain the state at the switching time (Gole et al., 1997),(Faruque et al., 2005). The main advantage is that the admittance matrix does not have to be recalculated, since the time step is fixed. This method requires two solution steps and two interpolation steps for every inter-time step event, this may not be feasible in real-time (Dinavahi, 2000).

The present work implements a time step changing algorithm that does not backtrack in time, making it suitable for real-time applications with physical systems that are not large enough so as to make the recalculation of the admittance matrix unpractical in nowadays commercially available microprocessors. A boost converter circuit is simulated to verify the feasibility of the algorithm. This circuit is a good example of a system in which the delay, if not considered, causes spurious behavior. (Vasca and Iannelli, 2012). Real-time simulation results were compared with commercial software results and good agreement was found.

This paper is structured as follows: this first section introduces the subject, Section II describes the switching delay correction algorithm, Section III presents the converter model and its numerical solution, Section IV shows the real time platform composed of two TI TMS320F28377S microprocessors, Section V gives results and Section VI contains the authors conclusions.

2 Delay Correction Algorithm

Any electrical circuit's dynamics describable by differential equations can be written in the state space representation, as shown in Equation 1 (Franklin et al., 1994).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{A}$ is the state matrix and $\mathbf{B}$ is the input matrix which multiplies the input vector $\mathbf{u}$.

In each iteration k, the last state is used for calculating the current one through some numerical integration method. Equation 2 shows Backward Euler (implicit) Method, which can be applied to Equation 1 (Butcher, 2016). Hence, a time domain solution can be achieved starting from some initial conditions.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \times \dot{\mathbf{x}}_{k+1} \tag{2}$$

If the circuit of interest is a switched one, it is necessary to check the switching status before solving Equation 1, because matrices $\mathbf{A}$ and $\mathbf{B}$ change with the switching status, as the circuit topology changes.

Besides changing matrices $\mathbf{A}$ and $\mathbf{B}$, a digital real-time simulator must take into account the delay observed between the exact instant of the switching order and the current time step.

Figure 1 illustrates a sequence of states between times t_0 and t_3, with time step $\Delta t = h$. A switching event labelled "A", which would ideally occur at instant t_A is only processed by the fixed-step simulator at instant t_2, introducing a delay $\gamma = t_2 - t_A$. By measuring this delay, it is possible to numerically solve state x_A changing the time step to $h_1 = h - \gamma$. In order to return to the original time grid, state x_3 is calculated with time step $h_2 = h + \gamma$.

As every real time simulation, once state x_2 is calculated, it is impossible to get back in time and recalculate it, but the present algorithm ensures that the next state x_3 is correct. On the other hand, if a further switching event occurs between t_2 and t_3, it is necessary to obtain the correct state x_2 in order to proceed with the algorithm. In this case, it is possible to linearly interpolate the state x_2, using its next states x_A and x_3 and their respective instants t_A and t_3.

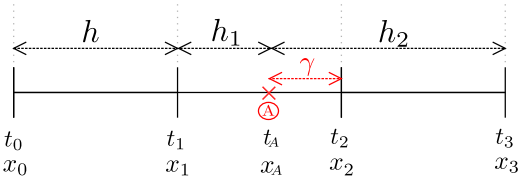

Figure 1: Illustration of a switching event occurring at t_A between time grid points t_1 and t_2.

The following steps summarize the algorithm routine, which are also depicted in the flowchart of Figure 2:

- In t_0, $\mathbf{x}_1$ is calculated;
- In t_1, $\mathbf{x}_2$ is calculated;

- In t_2:
  - An event at t_A is detected;
  - Time step is changed into $h_1 = h - \gamma$;
  - From $\mathbf{x}_1$, $\mathbf{x}_A$ is calculated;
  - Circuit topology is refreshed, recalculating matrices $\mathbf{A}$ and $\mathbf{B}$, according to switching state;
  - Time step is changed into $h_2 = h + \gamma$;
  - From $\mathbf{x}_A$, $\mathbf{x}_3$ is calculated;
  - Time step returns to h;
- In t_3, $\mathbf{x}_4$ is calculated.

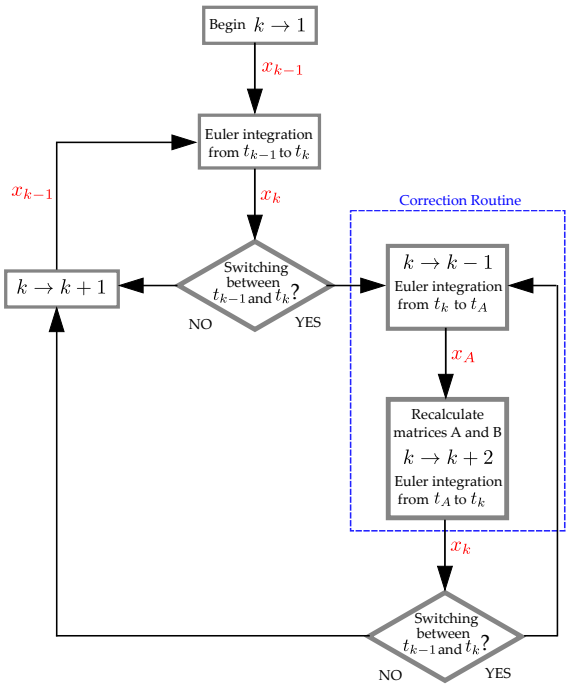

Figure 2: Simulation flowchart with correction algorithm routine.

3 Boost converter model

The modeled dc-dc boost converter circuit is shown in Figure 3. The semiconductor devices acting as switches are considered to be ideal, that is, they present perfect conduction and perfect interruption of current, and the commutation between these states take place instantly.

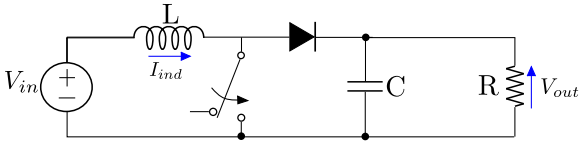

Figure 3: Boost converter circuit.

Boost converters, due do their possibles switching positions and due do the unidirectional current nature of the devices, present the three modes illustrated in Figures 4, 5 and 6 (Marouchos, 2006).

As the switch shown in Figure 3 is on, the inductor is charged by the source and the capacitor discharges into the resistor. When the switch opens, the inductor current forces diode direct polarization, discharging the inductor and charging the capacitor. The state variables adopted were the capacitor voltage and the inductor current.

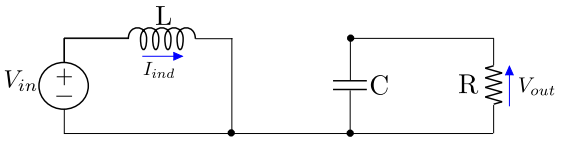

Figure 4: Mode I: Switch is conducting, the diode is reversely polarized.

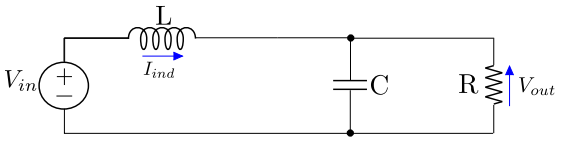

Figure 5: Mode II: Switch is off, the diode is conducting.

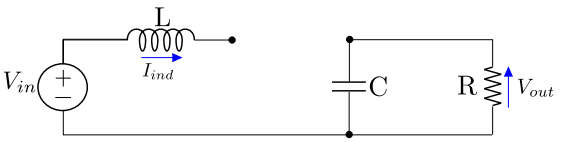

Figure 6: Mode III: Both the switch and the diode are off, this mode is characteristic of converters operating with discontinuous current conduction.

Equation 3 describes Mode I circuit.

$$\begin{bmatrix} \frac{dV_{out}}{dt} \\ \frac{dI_{ind}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} & 0 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} V_{out} \\ I_{ind} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \times V_{in} \quad (3)$$

Equation 4 describes the dynamics of Mode II.

$$\begin{bmatrix} \frac{dV_{out}}{dt} \\ \frac{dI_{ind}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix} \times \begin{bmatrix} V_{out} \\ I_{ind} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \times V_{in} \quad (4)$$

Equation 5 describes Mode III.

$$\frac{dV_{out}}{dt} = -\frac{1}{RC} \times V_{out} \quad (5)$$

Applying Euler Backward method (Equation 2) to equations 4 and 3 yields Equation 6.

$$\mathbf{x}_{k+1} = [\mathbf{I} - h\mathbf{A}]^{-1} \times [\mathbf{x}_k + hV_{in}\mathbf{B}] \qquad (6)$$

where $\mathbf{x}$ is the state vector, k is the current iteration, h is the time step, $\mathbf{A}$ is the state matrix, $\mathbf{B}$ the input matrix and $\mathbf{I}$ is the identity matrix.

Circuit solving is preceded at every time step by a reading of the firing signal's level, which directs the program to the proper circuit mode. Interleaved solution of modes equations yield the dynamics of the switching circuit.

The converter discrete model was implemented in MATLAB code and the obtained result was compared with a simulation of the same system in PSIM, in order to validate the model. Figure 7 show voltage and current results with the following parameters: R = 4 Ω, L = 1 mH, C = 1 mF, V_{in} = 1 V, h = 1 μs. The results agreed well, validating the proposed model.

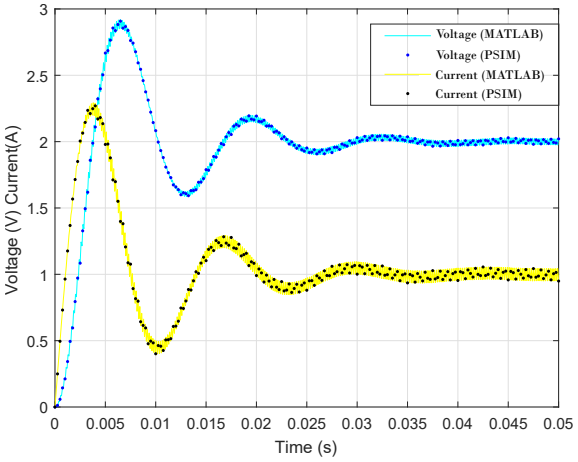

Figure 7: Voltage and current results from MATLAB and PSIM simulations.

4 Real-time Simulator Setup

The real-time simulator is composed of two 32 bits, 200 MHz Texas Instruments LAUNCHXL-F28377S, one simulating the boost converter and the other performing the control system. The controller receives the capacitor voltage signal from the boost converter microprocessor DAC, into its ADC, both of them working on 12 bits. An error value is generated comparing the ADC reading with an internal reference value. This error serves as an input to a PI controller that outputs a modulation signal (CMPA), which is passed to a PWM register and is compared with the PWM carrier wave, generating a PWM signal at an output pin. The PWM resolution is defined by the TBPRD variable. The converter DSP receives the PWM signal through its GPIO input. This scheme is illustrated in the block diagram of Figure 8.

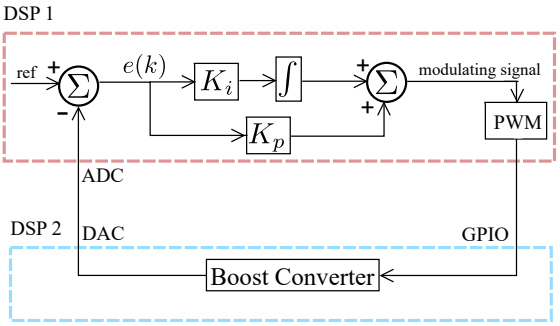

Figure 8: Real-time simulator block diagram.

The ADC sampling frequency is defined by the PWM frequency. The acquisition is configured to occur when the rising portion of the triangular signal passes through zero.

The converter GPIO sampling frequency is defined by the internal CPU interruption routine in which the simulator is implemented, and is chosen to be such that gives a simulation time step of 20.0 μs.

Another GPIO is configured to receive the PWM signal in order to measure the switching delay. This GPIO triggers an externally triggered interruption at every change in PWM level. This externallly triggered interruption starts a timer, which is stopped at every beginning of CPU interruption, if it is on. Figure 9 depicts the system's illustration and the interruptions pseudocodes.

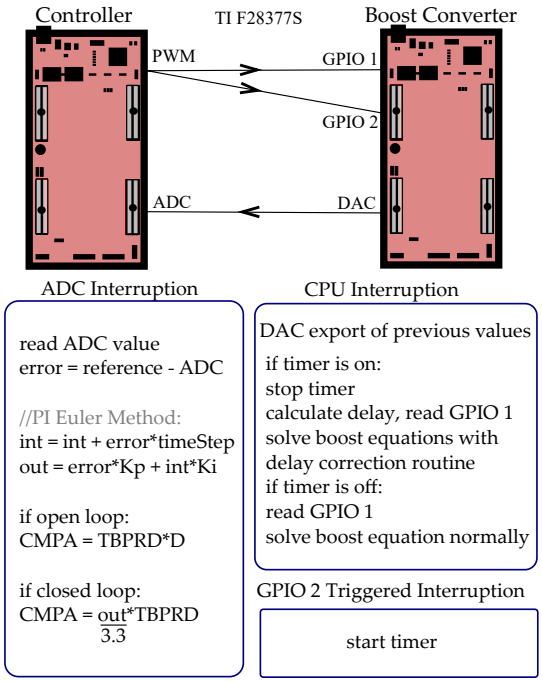

Figure 9: Real-time simulator setup illustration and interruptions pseudocodes.

In short, at every CPU interruption, the timer register is checked. If it is on, it is stopped, the

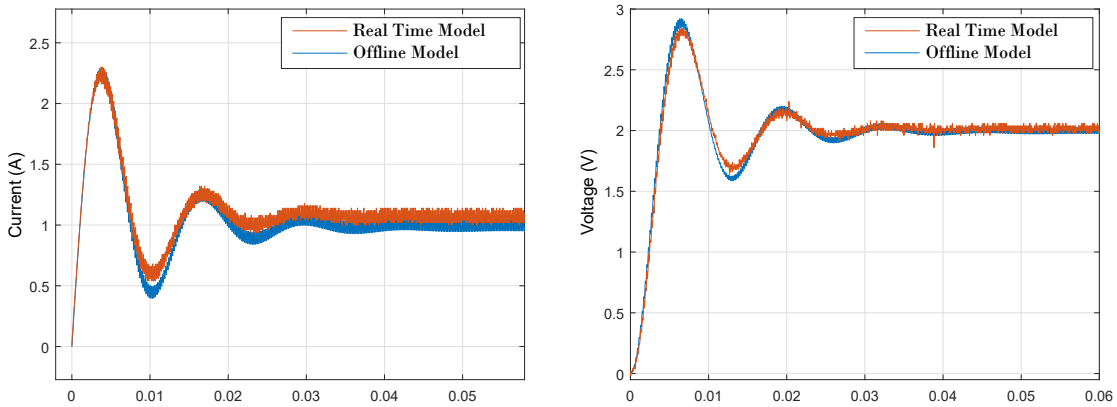

Figure 10: Real time model validation. (a) Current simulation; (b) Voltage simulation.

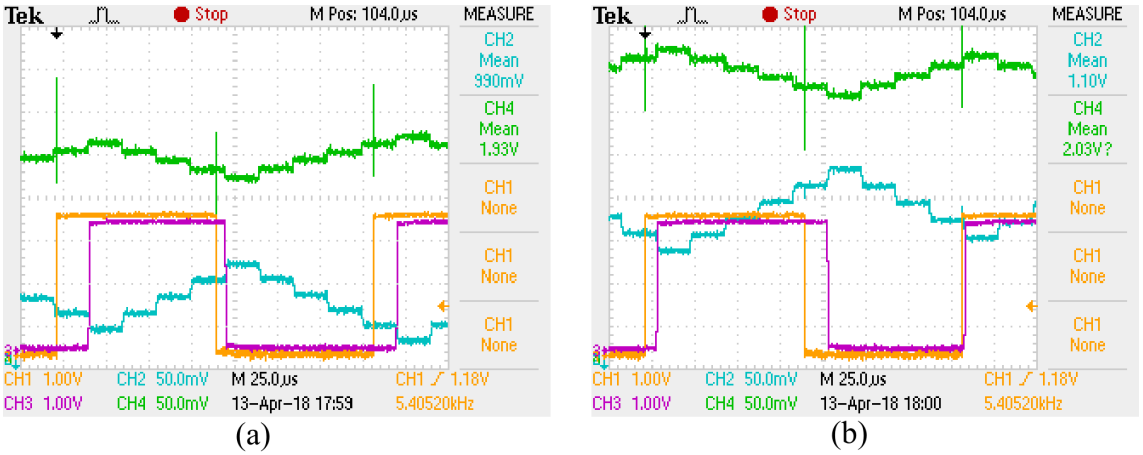

Figure 11: Open loop without corrector algorithm results. Voltage (green), Current (cyan), PWM signal (orange), PWM as seen by converter (purple). (a) Duty cycle seen by converter is less than real; (b) Duty cycle seen by converter is greater than real.

delay is measured, and the simulator goes to the correction algorithm routine. If it is off, no event happened between time steps, so the simulator proceeds with normal operation.

All embedded code was written in C. Since there is no support for linear algebra libraries in the used devices, all matrix operations have to be translated into arithmetical operations inside the DSPs.

5 Results

In order to validate the corrector algorithm, several tests have been performed, with different time steps for the converter simulation, and different PWM frequencies, in both open and closed loops.

It was observed that the corrector algorithm is necessary for time steps greater than 5 μs, with a switching frequency of 5400 Hz. Without the algorithm the voltage and current simulated values do not reach steady state and periodic oscillations appears.

Table 1 shows circuit and simulation parameters.

Table 1: Parameters.

R	4.0 Ω
L	1.0 mH
C	1.0 mF
V_{in}	1.0 V
Reference for V_{out} (closed loop)	2.0 V
Duty Cycle (open loop)	50%
Time Step	20.0 μs
PWM Frequency	5400.0 Hz
K_p	1.0×10^{-4}
K_i	10.0

The circuit parameters were chosen in order to make the voltage and current ripples large enough so the gap between discrete simulated values could be clearly distinguished from noise.

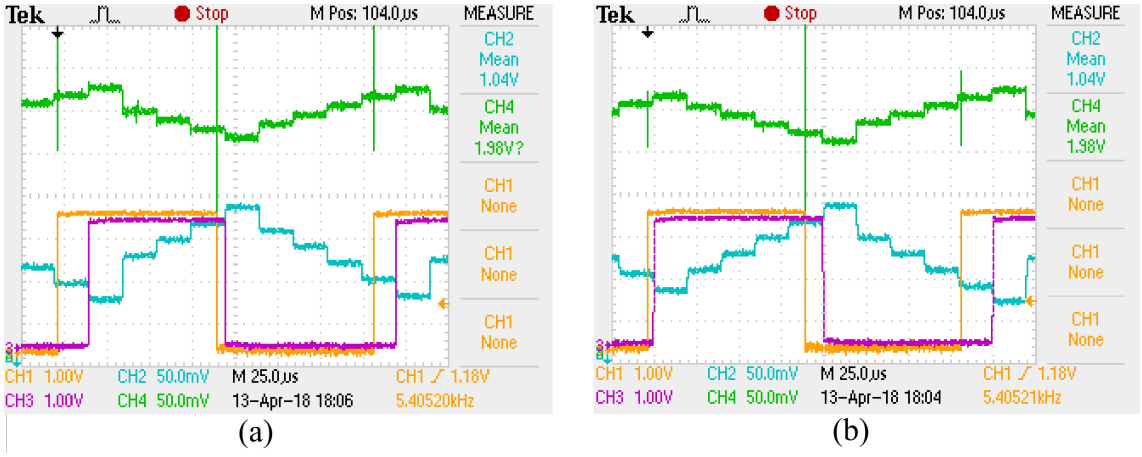

Figure 12: Open loop with corrector algorithm results. Voltage (green), Current (cyan), PWM signal (orange), PWM as seen by converter (purple). (a) Duty cycle seen by converter is less than real; (b) Duty cycle seen by converter is greater than real.

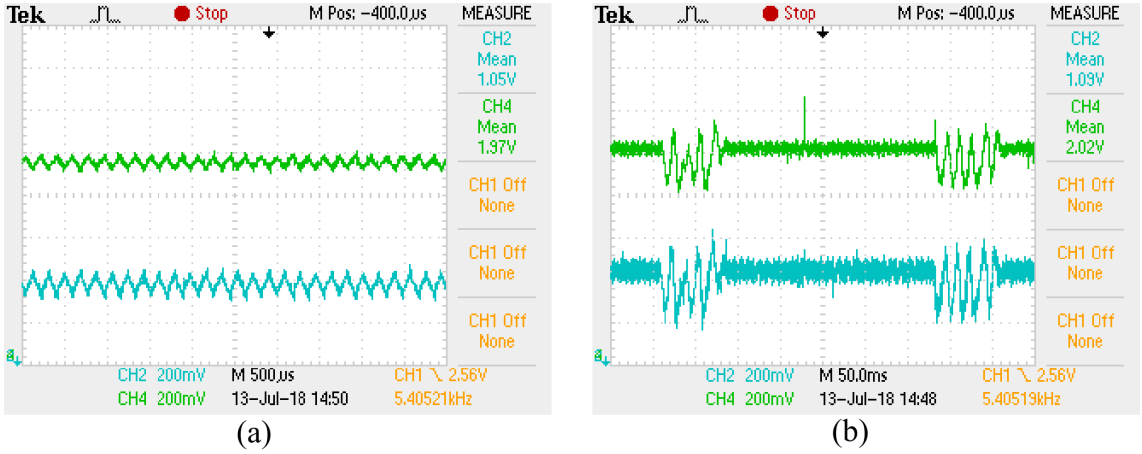

Figure 13: Closed loop results. Voltage (green), Current (cyan). (a) With corrector algorithm; (b) Without corrector algorithm.

5.1 Real-time model validation

Figures 10 (a) and 10 (b) show real-time and offline open loop transient voltage and current comparison, with a duty cycle of 50%. Offline result was obtained solving the boost model in MATLAB, as shown in section III.

Good agreement has been observed between models, with a steady state error of about 5% in current, and less than 1% in voltage.

5.2 Open loop real-time results

Figures 11(a) and 11(b) show two distinct instants with a duty cycle of 50% without the corrector algorithm. Channel 2 (blue) is the current, channel 4 (green) is the output voltage. Channel 1 (orange) is the PWM signal, and channel 3 (purple) is the PWM as seen by the converter. This last signal is obtained exporting the PWM signal measured in the boost converter DSP, through its DAC output with the same frequency of the simulator.

As a discrete system, the boost converter DSP can only recognize a pulse width multiple of its own time step. As the real pulse width sent by the control is not multiple of the converter time step, it will see different duty cycles according with the instant of switching. In Figure 11(a), the switching instant is such that the converter sees a duty cycle less than the real one, while in Figure 11(b) the duty cycle is greater than the real one.

This constant changing in duty cycle makes the voltage and current oscillate between higher and lower values than would be expected with the real duty cycle. In the present case, a duty cycle of 50% should deliver a voltage of 2.0 V as depicted in Equation 7, but the simulated voltage oscillates between 1.93 V and 2.03 V.

$$V_{out} = \frac{1}{1-D} V_{in} = 2 \text{ V} \tag{7}$$

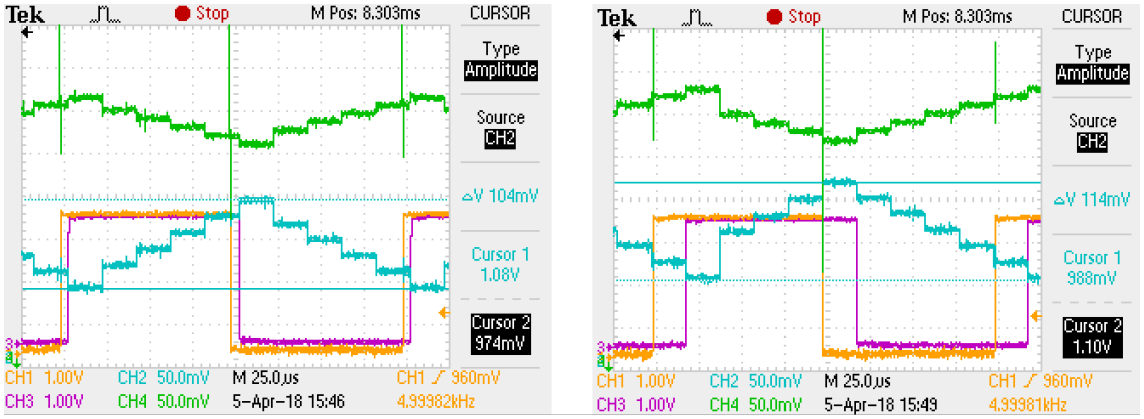

Figure 14: Current ripple measurement. Voltage (green), Current (cyan), PWM signal (orange), PWM as seen by converter (purple). (a) Minimum delay instant; (b) Maximum delay instant.

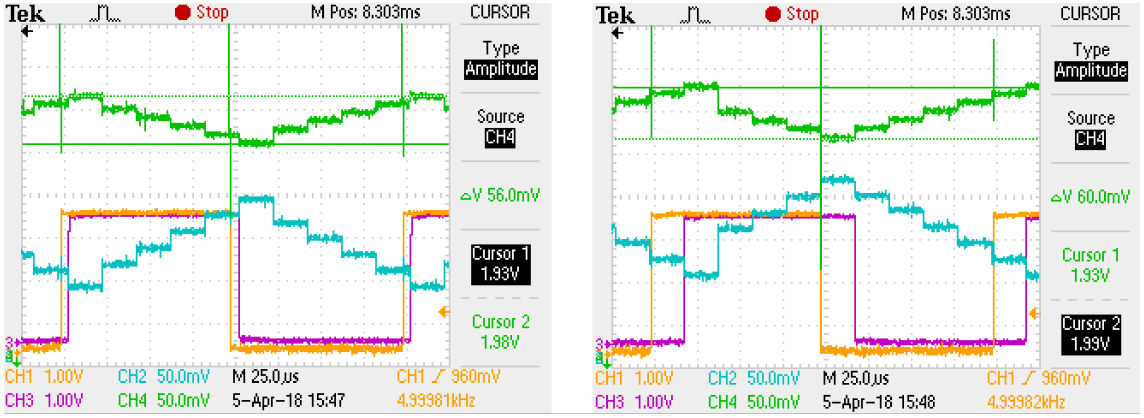

Figure 15: Voltage ripple measurement. Voltage (green), Current (cyan), PWM signal (orange), PWM as seen by converter (purple). (a) Minimum delay instant; (b) Maximum delay instant.

In order to account for the delay, the second state vector (voltage and current values) after switching should be corrected, but as in this case there is no correction algorithm, the gap between discrete values keeps constant.

Figures 12(a) and 12(b) show two distinct instants for a duty cycle of 50% with the corrector algorithm. Here, the voltage and the current values do not change according with the duty cycle seen by the converter. In this case, the second iteration after PWM switching is proportionally handling the delay. The gap between discrete values changes in the second step after switching.

5.3 Closed loop real-time results

Figures 13(a) and 13(b) show closed loop results with a voltage reference of 2.0 V, with and without corrector algorithm, respectively. It is seen that the algorithm is necessary to ensure a good result, giving a steady error of 1.0% in voltage. Without accounting for the delay, the voltage and current simulated values do not reach steady state. Instead, periodic oscillations associated with the limit cycle concept are observed. The reference (Buso and Mattavelli, 2006) gives more information about limit cycle on digital control of power electronics switched circuits.

5.3.1 Ripple measurement

In order to show how the ripple error depends on the delay, the PWM frequency was changed to 5 kHz. By doing so, a duty cycle of 50% gives a pulse width of 100.0 μs, a value multiple of the time step of 20.0 μs, what makes the duty cycle seen by the converter equals the real one.

Figures 14(a), 14(b) show current ripple measurements. Equation 8 gives the expected ripple value (Hauke, 2014).

$$\Delta I_{ind} = \frac{V_{in} D}{f_s L} = \frac{0.5}{5} = 0.1 \Rightarrow \Delta I_{ind} = 100.0 \text{ mA} \quad (8)$$

where ΔI_{ind} is the peak to peak current ripple, V_{in} is the input voltage, D is the duty cycle, f_s is the switching frequency and L is the inductance value.

An error of about 4.0% can be seen in current ripple on the minimum delay instant and of about 14.0% on the maximum delay. As the post-switching iteration can not be corrected in real-time, it is expected that this uncorrected value causes an error in ripple, both in rise and fall switchings.

Figures 15(a), 15(b) show voltage ripple measurements. Equation 9 gives the expected ripple value (Mohan and Undeland, 2007).

$$\Delta V_{out} = \frac{V_{out} D}{f_s RC} = \frac{1}{4 \times 5} = 50.0 \text{ mV} \quad (9)$$

It can be seen a voltage ripple error of 12.0% on minimum delay instant and 20% on maximum delay.

It was observed that for voltage reference values which lead to PWM cycles less than one simulation period, the algorithm fails. In this case, there is more than one switching event inside one simulation period (a rise and a fall in PWM level), so the algorithm should be able to account more than one event inside one time step.

6 Conclusion

This work presents the development and implementation of a boost converter real time simulator. Account for the switching delays is one of the main challenges in real time simulation of switching devices. For doing so, this work implements an algorithm which corrects the error introduced by the delay between the real PWM signal and the PWM as seen by the simulator, due to its discrete nature. The boost circuit was modeled through the differential equations of each possible mode, and then solved by Backward Euler method. The real-time platform is composed by two Texas Instruments DSPs implementing both the converter and its control system.

The implemented algorithm worked well as intended and allowed for an increase in the converter simulation time step to values which would previously yield wrong results. It was observed that in both open and closed loops the delay have to be handled for time steps greater than 5.0 μs. It was also seen that the present algorithm does not work for duty cycles which leads to pulse widths less than one simulation time step. For doing so, it would be necessary to implement a multiple event accounting algorithm, what is recommended as future work.

References

Buso, S. and Mattavelli, P. (2006). Digital control in power electronics, *Lectures on power electronics* **1**(1): 1–158.

Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*, John Wiley & Sons.

Bélanger, J., Venne, P. and Paquin, J. (2010). The What, Where and Why of Real-Time Simulation, *IEEE PES General Meeting*, pp. 25–29.

Dinavahi, V. (2000). *Real-time Digital Simulation of Switching Power Circuits*, PhD thesis.

Dinavahi, V. and Iravani, M. (2001). Coupling Digital Controllers with Real-Time Digital Simulators for Switched Power Systems, *2001's IPST International Conference on Power Systems Transients*, pp. 178–183.

Faruque, M., Dinavahi, V. and Xu, W. (2005). Algorithms for the accounting of multiple switching events in digital simulation of power-electronic systems, *IEEE Transactions on Power Delivery* **20**(2): 1157–1167.

Franklin, G. F., Powell, J. D., Emami-Naeini, A. and Powell, J. D. (1994). *Feedback control of dynamic systems*, Vol. 3, Addison-Wesley Reading, MA.

Gole, A., Fernando, I., Irwin, G. and Nayak, O. (1997). Modeling of power electronic apparatus: Additional interpolation issues, *Proc. Int. Conf. Power System Transients*, pp. 23–28.

Hauke, B. (2014). Basic calculation of a boost converter's power stage, *Texas Instruments, Application Report January* pp. 1–9.

Manitoba, H. (2010). Research centre, pscad/emtdc, user manual 4.7, *Tutorial manual* .

Marouchos, C. (2006). *The Switching Function: Analysis of Power Electronic Circuits*, The Institute of Engineering and Technology.

Mohan, N. and Undeland, T. M. (2007). *Power electronics: converters, applications, and design*, John Wiley & Sons.

Vasca, F. and Iannelli, L. (2012). *Dynamics and control of switched electronic systems: advanced perspectives for modeling, simulation and control of power converters*, Springer.