Modeling of power converters for debugging digital controllers through FPGA emulation

Fernando López-Colino, Alberto Sanchez, Angel de Castro and Javier Garrido
Human Computer Technology Laboratory
Universidad Autónoma de Madrid
Francisco Tomas y Valiente 11
Madrid, Spain
Phone: +34 (91) 497 36 13
Fax: +34 (91) 497 2107
Email: [fj.lopez, alberto.sanchezgonzalez, angel.decastro, javier.garrido]@uam.es
URL: http://www.hctlab.com

Acknowledgments

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovacion under project TEC2009-09871.

Keywords

<<Modeling>>, <<Device simulation>>, <<Field Programmable Gate Arrays (FPGA)>>, <<Switched-mode power supply>>.

Abstract

Debugging a digital controller for power converters can be a lengthy process due to the long time required in mixed-signal simulations. This paper focuses on the design of a power converter model for debugging digital controllers in closed loop. The testing may be performed by means of simulation or emulation. This paper shows the results of simulating and emulating the power converter using different data representations. Experiments will show that through a good selection of data and emulation, testing can be speeded up over $28,000$ times.

Introduction

Every engineering system requires an exhaustive testing process prior to its approval. This testing aims for avoiding damages to real systems or even injuries to people. The design and development of digital controllers for power converters are not an exception.

This paper focuses on debugging digital controllers for power converters implemented in FPGAs [1, 2, 3]. These controllers are usually described in VHDL or Verilog, but this is not the only possible solution. Literature shows examples of embedded μProcessors used for running C code [4]. The verification of the final regulator implies not only debugging its implementation, but also its interaction with the plant to be controlled. This way, it is possible to find out if any problem would arise when applied to the real plant.

One of the multiple possibilities is using a computer for running the full simulation, as shown in [5]. Another solution is to design a model of the plant to be controlled using the same HDL as used for the controller. Hence, using the correct implementation, both the controller and the plant can be tested in an FPGA [6, 7, 8, 9]. This solution is referred as the HIL (Hardware-in-the-Loop) approach. The plant's model requires solving equations including real numbers. In [10], the use of the VHDL2008 *float_pkg* package is proposed for modeling the plant.

This paper will compare existing HIL solutions and propose new alternatives for modeling the plant. We will also present the influence of including electrical losses in the plant model, checking the accuracy of the results.

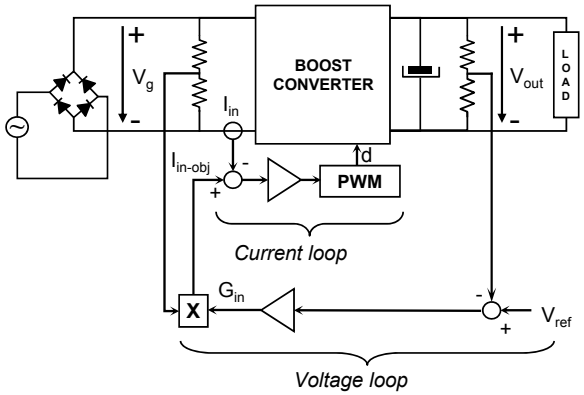


Figure 1: PFC technique with a boost converter

This paper is structured as follows: the second section presents the different alternatives for data representation in the plant's model, the power converter implementation and some required optimizations. The third section presents the results of three different evaluations for each of the implemented models. Finally, the fourth section presents a brief discussion about the results presented in this work.

Verification of digital controllers

Application example

This paper aims to provide a solution for testing digital controllers for boost converters using power factor correction (PFC), although it could be easily adapted to other topologies or applications. The regulator follows the classic solution using two loops (Fig. 1). The transfer functions of the plants related to both loops are described in the literature [11].

Plant modeling possibilities

Considering the complexity of the whole system, the main problem is the simulation of both the controller and the model of the power converter. There are several simulation and emulation approaches to verify the correct implementation of the controller. There are commercial programs, such as Questa and SystemVision of Mentor Graphics, that allow the simulation of both analog systems and digital HDL defined systems. These applications also allow including electrical parasitic and losses. However, the simulation time is so long that makes them very impractical for many applications, such as PFC. Another possibility is the design of a digital version of the plant including both the plant and the ADC. This design can be done using the same language used for designing the controller, in our case: VHDL. VHDL provides different data types which allow the design of the plant. The main different options for designing the plant are presented below:

1. *Floating point, not synthesizable*: VHDL allows modeling any system using a signal type called *real*. The implementation of the plant is very simple as it allows a direct representation of the electrical formulas. However, the main drawback of this signal type is that it can be only simulated in the computer and it is not synthesizable.
2. *Floating point, synthesizable*: The *float* type, which is implemented in the VHDL2008 *float_pkg* package [12]. The implementation of the plant is as simple as using the *real* type, but it can be also implemented in hardware. This allows both simulation and emulation. However, the floating point consumes many hardware resources.
3. *Fixed point, synthesizable*: The last proposed approach for modeling the plant relies on fixed point arithmetic. This approach focuses on reducing hardware resources and decreasing emulation time. In the past, designers had to handle all the considerations related to the precision of the operators, the arithmetic operations and the rounding management. However, the VHDL2008 *fixed_pkg* package [12] automatically handles many of these considerations, leaving to the designer the decision about the operators precision.

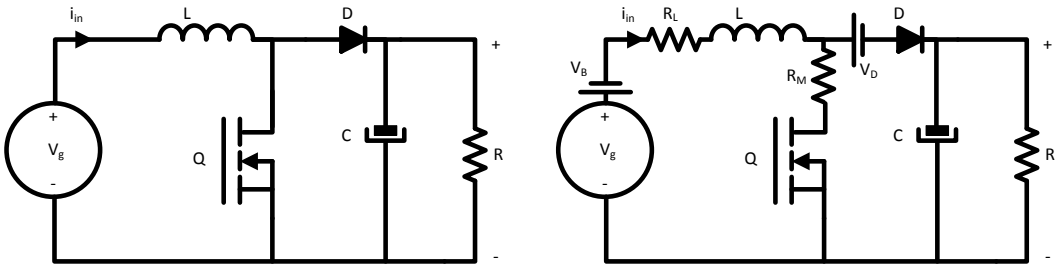
In the next section, we will discuss the differences between modeling the boost using VHDL with these different numeric notations: *fixed_pkg*'s fixed point, *float_pkg*'s floating point, and *real* types.

Table I: Boost Converter Parameters

Parameter	f_{sw}	L	C	P	V_{out}	R_L	R_M	v_D	v_B
Value	$100\,kHz$	$5\,mH$	$100\,\mu F$	$300\,W$	$400\,V$	$0.6965\,\Omega$	$0.4\,\Omega$	$1.03\,V$	$1.14\,V$

Model of the plant

An ideal boost converter has been depicted in Fig. 2(a). However, several elements of a boost converter present parasitic resistances and electrical losses. The model of the plant includes the following modifications compared to an ideal boost converter (Fig. 2(b)):



(a) Ideal boost converter

(b) Losses within the boost converter

Figure 2: Boost converter topology

- v_B is the voltage loss due to the diode bridge that converts AC into DC.
- v_D is the voltage loss due to the diode D.
- R_L is the serial resistance of the inductor L.
- R_M is the on resistance of the MOSFET Q.

The final values of v_B, v_D and R_M were obtained from their datasheets, while the value of R_L had to be measured because it is a hand-made inductance. These values of the modeled boost converter are presented in Table I.

The discrete model of the plant updates every time step k the values of the output voltage (v_{out}) and input current (i_{in}). This update must take into account the status of the switch. Hence, the update period must be equal to or a fraction of the signal's period controlling the status of this switch. Considering both the status of the switch and the current through the diode D, we define the following three states: (a) closed switch, (b) open switch and the diode D is forward-biased and (c) open switch and the diode D is reversed-biased.

The ideal model of a boost converter defines the input voltage as $v_L = L\frac{di_{in}}{dt}$ and the output capacitor current is defined by $i_C = C\frac{dv_{out}}{dt}$. Converting the previous differential equations into difference equations, considering the duration of each step k as Δt, the input current and the output voltage for each time step are defined by (1):

$$\begin{aligned} i_{in}(k) &= i_{in}(k-1) + \tfrac{\Delta t}{L} v_L(k) \\ v_{out}(k) &= v_{out}(k-1) + \tfrac{\Delta t}{C} i_C(k) \end{aligned} \tag{1}$$

Each step k has a constant duration of Δt. Hence, the values of $\frac{\Delta t}{L}$ and $\frac{\Delta t}{C}$ are also constant. i_C is the current through the capacitor, which is determined by the output load: when the switch is closed (state (a)), $i_C = -i_R$; and $i_C = i_{in} - i_R$ when the switch is open, depending on the value of i_{in} the diode would be forward-biased (state (b)) or reverse-biased (state (c)). A possible value for i_R is considering a resistive load. Hence, its value would be $i_R = \frac{v_{out}}{R}$, but the proposed model lets i_R as an independent variable, so any load can be modeled. This approach provides a more flexible implementation.

Each step k, the equation (1) must consider the three previously defined states (a, b and c), for updating the values of i_{in} and v_{out}. The following equations show how to update i_{in} (2) and v_{out} (3) for each of the three states:

$$i_{in}(k) = \begin{cases} i_{in}(k-1) + \frac{\Delta t}{L} v_g & \text{(a)} \\ i_{in}(k-1) + \frac{\Delta t}{L} (v_g - v_{out}(k)) & \text{(b)} \\ 0 & \text{(c)} \end{cases} \quad (2)$$

$$v_{out}(k) = \begin{cases} v_{out}(k-1) - \frac{\Delta t}{C} i_R(k) & \text{(a)} \\ v_{out}(k-1) + \frac{\Delta t}{C} (i_{in}(k) - i_R(k)) & \text{(b)} \\ v_{out}(k-1) - \frac{\Delta t}{C} i_R(k) & \text{(c)} \end{cases} \quad (3)$$

It was previously stated that each step k has a constant duration of Δt. Hence, the values of $\frac{\Delta t}{L}$ and $\frac{\Delta t}{C}$ are also constant in (2) and (3).

To include the electrical losses (shown in Fig. 2(b)) in the equations (2) and (3) the following modifications must be considered:

- v'_g is the output voltage of the diode bridge that converts AC into DC. Its value is obtained from the following formula: $v'_g = v_g - v_B$ when $v_g > v_B$, or 0 otherwise. In this case, v_B is the voltage lost in the diode bridge.
- When the system is in the (b) state, the MOSFET Q is open and the diode is forward-biased, we must consider the voltage lost in the diode v_D.
- When the system is in the (a) or (b) states, i_{in} flows through the inductor L, which has an inherent resistance R_L. Hence, the voltage loose in the inductor is $i_{in} \cdot R_L$.
- Finally, in the (a) state, the MOSFET Q also shows a resistance R_M. The voltage loss in this case is $i_{in} \cdot R_L$.

These losses applied the equations (2) and (3) result in the following formulas:

$$i_{in}(k) = \begin{cases} i_{in}(k-1) + \frac{\Delta t}{L} \left[v'_g - i_{in}(k) \cdot (R_L + R_M) \right] & \text{(a)} \\ i_{in}(k-1) + \frac{\Delta t}{L} \left[v'_g - (v_{out}(k) + v_D + i_{in}(k) \cdot R_L) \right] & \text{(b)} \\ 0 & \text{(c)} \end{cases} \quad (4)$$

$$v_{out}(k) = \begin{cases} v_{out}(k-1) - \frac{\Delta t}{C} i_R(k) & \text{(a)} \\ v_{out}(k-1) + \frac{\Delta t}{C} (i_{in}(k) - i_R(k)) & \text{(b)} \\ v_{out}(k-1) - \frac{\Delta t}{C} i_R(k) & \text{(c)} \end{cases} \quad (5)$$

Optimizations to the model

It can be observed in equations (4) and (5) that several multiplications must be performed. Particularly, in equations (4.a) and (4.b), two sequential multiplications are required. The first one is for obtaining the electrical losses in the inductor and the MOSFET, and the second one is for the constant value $\frac{\Delta t}{L}$. These two multiplications create a critical path in the hardware implementation of the model. If the model can be optimized someway, the emulation speed would be incremented.

To optimize the model, we propose the following variable substitutions in the discrete model of the plant:

- $i^*_{in} = \frac{L}{\Delta t} i_{in}$
- $v^*_{out} = \frac{C}{\Delta t} v_{out}$
- $R^* = \frac{\Delta t}{L} R$

Applying these substitutions to the equation (1), the following is obtained:

$$\begin{aligned} i^*_{in}(k) &= i^*_{in}(k-1) + v_L(k) \\ v^*_{out}(k) &= v^*_{out}(k-1) + i_C(k) \end{aligned} \quad (6)$$

If we apply the same reasoning that led from equation (1) to equations (4) and (5); starting from equation (6) we will obtain the following two equations for describing the values of the input current and the output voltage:

$$i^*_{in}(k) = \begin{cases} i^*_{in}(k-1) + v'_g - i^*_{in}(k) \cdot (R^*_L + R^*_M) & \text{(a)} \\ i^*_{in}(k-1) + v'_g - (v_{out} + v_D + i^*_{in}(k) \cdot R^*_L) & \text{(b)} \\ 0 & \text{(c)} \end{cases} \quad (7)$$

$$v^*_{out}(k) = \begin{cases} v^*_{out}(k-1) - i_R(k) & \text{(a)} \\ v^*_{out}(k-1) + (i_{in}(k) - i_R(k)) & \text{(b)} \\ v^*_{out}(k-1) - i_R(k) & \text{(c)} \end{cases} \quad (8)$$

Using this transformation, the multiplication from equation (5) is removed in (8). In equations (7.a) and (7.b) the sequence of multiplications is removed. It must be noted that a multiplication is required for obtaining v_{out} using $v^*_{out} = \frac{C}{\Delta t} v_{out}$. Apart from that, a multiplication is still necessary for $i^*_{in}(k) \cdot R^*$. However, these multiplications can be done in parallel as there is not any dependency between them. This modification increases the working frequency, thus reducing the time required for the emulation. This optimization is explained in more detail in [13].

Results

The experiments have been done using four different approaches for implementing the model: using *mixed simulation*, using the *real* type in a VHDL implementation, using the *float* type in the *float_pkg* library of VHDL and using the QX.Y fixed point notation described in the *fixed_pkg* library of VHDL. For these implementations using simulation and emulation (when available) the following elements were measured: steady state accuracy, accuracy during a load step and the required time for solving the same simulation of the system.

Steady state accuracy

The first evaluation focused on the accuracy of the different implementations. For the first evaluation we will measure the steady state accuracy of the implementations within the design shown in Fig. 1 for power factor correction. To avoid comparing the different elements of the system, we have chosen a single significant element of the system. The output of the voltage loop (G_{in}) is modified by both the voltage loop and the current loop. Therefore, if the model does not calculate accurately one of the two variables, the parameter G_{in} will change. The first evaluation consisted on requesting a fixed output voltage $v_{out} = 400\,V$ and waiting for the stabilization of the G_{in} value. In Table III we present the results of the theoretical values and the obtained results using the ideal model and the loose model.

Considering the ideal models, the result obtained when using the *real* type shows the lowest difference compared to the theoretical value. It must be noticed that the *real* type uses a 64-bit floating point representation. The results obtained using the QX.Y fixed point representation are also quite similar to the theoretical ones. However, the difference between QX.Y and the theoretical results is slightly greater than the difference between the *real* type and the theoretical ones. The worst result is obtained with the *float* type as it shows an error of 10 % when compared to the theoretical one. This error is caused by the 32-bit precision of the *float* type, compared to the 64-bit precision used by the *real* type. The *float* type does not allow the required resolution for storing the increments of the v_{out} and the i_{in} signals. Consider the increment of the v_{out} signal is around $7.5 \cdot 10^{-5}\,V$ for a $v_{out} = 400\,V$. The *float-32* type uses 24 bits for the mantisa: a fixed "1" and 23 additional bits. The v_{out} value is around $400\,V$ so the MSB is 2^8 and the smallest value that can be stored is $3.05 \cdot 10^{-5}$. Therefore, the increment of $7.5 \cdot 10^{-5}\,V$ is rounded to $6.1 \cdot 10^{-5}\,V$. A possible solution would be incrementing the value of Δt, so the variation of v_{out} and i_{in} would be larger and the error due to the resolution would be smaller. However, the system uses a $100\,kHz$ clock for the duty cycle management and a $\Delta t = 10\,ns$, which allows a resolution of the duty cycle of 0.1 %. If we were to increase the value of Δt to $\Delta t = 100\,ns$, the resolution of the duty cycle would be reduced to 1 %, which is far from optimal for simulating power converters. The discussion about the problems due to the resolution of the different data types is extended in [13].

This evaluation has been also applied to the models including losses. However, considering the error obtained with the *float* data type, we have not performed the evaluations including losses with this data type. The *real* model is also the most similar to the *Mixed simulation*, which is considered as base line for the losses model. The difference between the QX.Y fixed point notation and the base line is lower than 1 %. In this comparison the values obtained from the experimental results are omitted. The reason of this omission is that G_{in} is not only modified by electrical losses but also by the measurement accuracy (e.g., resistor dividers).

Table II: Accuracy of the model - PFC converter

System	Simulation Emulation	G_{in}	G_{in} error related to ideal G_{in}
Ideal G_{in} without losses		0.005 671 08	
"Real" type without losses	Simulation	0.005 653 38	−0.31%
32-bit "Float" type without losses	Sim/Emulation	0.005 123 14	−9.66%
QX.Y without losses	Sim/Emulation	0.005 649 57	−0.38%
Mixed simulation (includes losses)	Simulation	0.005 767 82	1.71%
"Real" type with losses	Simulation	0.005 733 49	1.1 %
QX.Y with losses	Sim/Emulation	0.005 714 42	0.76%
Results taken in steady state with V_{out} reference set to 400 V			

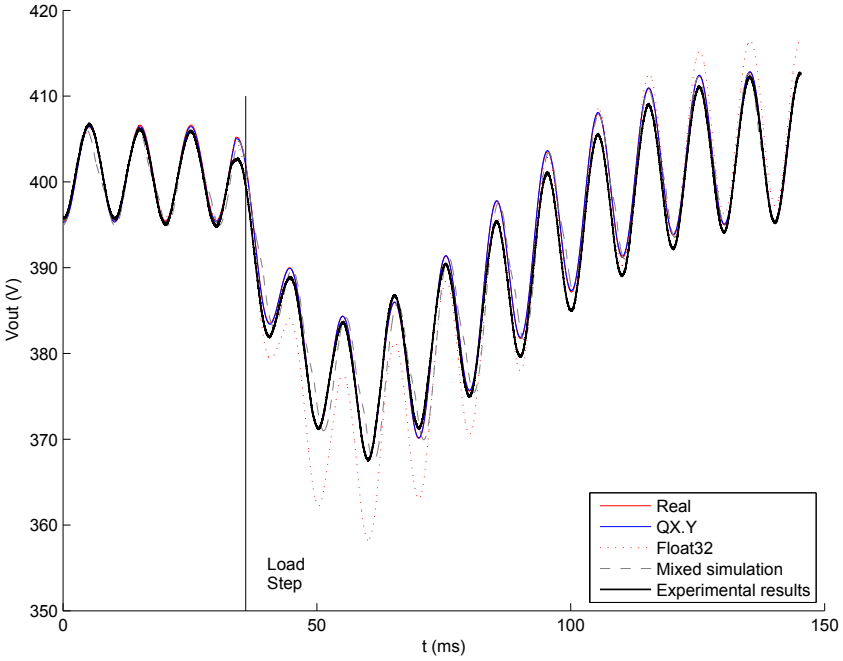


Figure 3: Comparison of the proposed systems after load step from 1176 Ω to 741 Ω ($v_{out}Ref = 400\ V$)

Accuracy during a load step

To perform a significant comparison with the real prototype a different evaluation has been chosen. For this comparison, a load step has been simulated with the different models and measured in the real prototype. All the systems start in a steady state for an output voltage of $v_{out} = 400\ V$ and a load of 136 W. From this initial conditions, the load is modified so the new value is 216 W. The real data has been captured and compared with the results of the emulations showing their behavior in this dynamic transition. Fig. 3 shows the evolution of v_{out} for this experiment.

The evolution of this transition is highly dependent of the values of the inductor, the capacitor and the controller's gain. Fig. 3 only includes the results of the ideal models as the models which included losses provided the same results. As it can be observed in Fig. 3, the results of the QX.Y, *real* and *Mixed simulation* are very similar as they use the same values for the inductor, the capacitor and the controller's gain. The tolerance of the real components in the prototype make the measured values a bit different of the simulated ones. However, this figure shows a relevant difference between the previous values an the results obtained using the *float 32-bit* data type. The lack of resolution of this data type makes the difference in the results very relevant.

Required time for obtaining the results

The third evaluation measured the time required by each model to provide the results. Considering the two loops of the controller, the slowest one requires around 109 *ms* to reach a steady state. Considering that $\Delta t = 10\ ns$, several millions of iterations must be calculated to obtain the final results.

To compare the different implementations of the model, we have measured the time required for simulating or emulating the same task. The emulation was performed in an FPGA Xilinx XC3S1000. Emulation times are extracted from the maximum clock frequency of each model.

Table III: Time results of a simulation of 200 ms

System	Simulation/Emulation	Time	Speedup
Mixed simulation	Simulation	2h 13' 21" 751 *ms*	Reference
"Real" type	Simulation	2' 14" 646 *ms*	59.4×
"Float" type	Simulation	2h 5' 14" 438 *ms*	1.1×
"Float" type	Emulation	3" 228 *ms*	2478.9×
QX.Y (fixed_pkg)	Simulation	29' 30" 780 *ms*	4.5×
QX.Y (fixed_pkg)	Emulation	276 *ms*	28991.2×

The results of this evaluation are presented in Table III. After, this experiment we can conclude that the *float* data type is not suitable for this task. Both the lack of accuracy in the results and the second longest simulation time discards its use. The simulation speed of the *real* type and the provided accuracy, shown in the previous experiments, stand out this data type as the best solution. However, if a large number of simulations or a longer simulation were required, the emulation approach using the QX.Y data type would be highly recommended.

Conclusions

This paper has presented different approaches for simulating digital controllers for boost converters. These controllers must be simulated during the design step using specific tools. However, their implementation using HDL may introduce severe errors leading to hardware failures or even human injuries. The final implementation also shows non-idealities such as delays in the ADC, limited word width, limits to the maximum and minimum duty cycle, etc. Therefore, simulating the final implementation of the controller is really important.

Mixed simulation allows an easy design of the testbench, merging the use of a GUI for designing the circuitry and a simple instantiation of the HDL-designed controller. Its main drawback is the lengthy simulations, which can last several hours for complex designs.

The other main approach for simulating the final implementation of the controller is designing the boost converter in HDL. This implementation is highly dependent on the chosen data type. This election will condition the simulation speed and the implementation effort. The *real* type is a floating point representation, which allows an easy implementation of the converter and takes minutes for simulation. The *float* type is also a floating point representation which can be synthesized, allowing the emulation within an FPGA. This allows an incredible speedup of the simulation time. However, the lack of resolution of the 32-bit *float* leads to significant accuracy errors when used for high frequency switching power converters. Another drawback of this data type is the considerable amount of resources when synthesized. Finally, we have also proposed a fixed point notation for implementing the power converter. The QX.Y notation requires more effort during the implementation, but provides a similar accuracy in the results when compared to the *real* type and allows finishing the simulations within miliseconds. The effort of this implementation can be reduced using the *sfixed* library.

This paper has shown how including first order electrical losses modifies the results. The results obtained using the HDL models are similar to the ones obtained using *Mixed simulation* for a steady state simulation. It must be noted that *Mixed simulation* includes these electrical losses.

The dynamic response depends on the controller's gain (independent of the converter models), the capacitance of the output capacitor (C) and the inductance of the inductor (L). Therefore, all converter models, with and without losses, show similar dynamic simulations. The measured data from the real prototype is a bit different from these simulations. The reason of this difference is the small tolerance in the values of L and C, which are different for every real converter. Hence, the simulations will show the average results for every possible converter built using those specifications.

References

[1] J. Alvarez, O. Lopez, F. Freijedo, and J. Doval-Gandoy, "Digital parameterizable VHDL module for multilevel multiphase space vector PWM," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 9, pp. 3946–3957, Sept. 2011.

[2] E. Monmasson, L. Idkhajine, and M. Naouar, "FPGA-based controllers," *Industrial Electronics Magazine, IEEE*, vol. 5, no. 1, pp. 14–26, March 2011.

[3] F. Taeed, Z. Salam, and S. Ayob, "FPGA implementation of a single-input fuzzy logic controller for boost converter with the absence of an external analog-to-digital converter," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 2, pp. 1208–1217, Feb. 2012.

[4] O. Lucia, L. Barragan, J. Burdio, O. Jimenez, D. Navarro, and I. Urriza, "A versatile power electronics test-bench architecture applied to domestic induction heating," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 3, pp. 998–1007, March 2011.

[5] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 2, pp. 919–931, April 2007.

[6] S. Karimi, P. Poure, and S. Saadate, "An hil-based reconfigurable platform for design, implementation, and verification of electrical system digital controllers," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 4, pp. 1226–1236, April 2010.

[7] M. Matar and R. Iravani, "FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *Power Delivery, IEEE Transactions on*, vol. 25, no. 2, pp. 852–860, April 2010.

[8] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 358–368, Jan. 2011.

[9] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *Power Delivery, IEEE Transactions on*, vol. 22, no. 2, pp. 1235–1246, April 2007.

[10] O. Lucia, I. Urriza, L. Barragan, D. Navarro, O. Jimenez, and J. Burdio, "Real-time FPGA-based hardware-in-the-loop simulation test bench applied to multiple-output power converters," *Industry Applications, IEEE Transactions on*, vol. 47, no. 2, pp. 853–860, March-April 2011.

[11] R. Erickson and D. Maksimovic, *Fundamentals of power electronics*. Kluwer Academic, 2001.

[12] [Online]. Available: http://www.eda.org/fphdl/

[13] A. Sanchez, A. de Castro, and J. Garrido, "A comparison of simulation and hardware-in-the- loop alternatives for digital control of power converters," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 3, pp. 491 –500, aug. 2012.