Resolution Analysis of Switching Converter Models for Hardware-in-the-Loop

Oscar Goñi, Alberto Sanchez, Elias Todorovich, *Member, IEEE*, Angel de Castro, *Member, IEEE*

***Abstract*—This work proposes two methods to determine the resolution of state variables in models of switching-mode power converters. The target models are intended for Hardware-In-the-Loop, i.e., closed loop emulation using a model of the power converter implemented in digital hardware with the controller in its final implementation. The focus here is on resolution of fixed-point models, although the results can also be applied to the significand resolution in floating-point representation. The first method is based on simulation, provides the designer with the optimum resolution values, and guarantees that using that resolution, the converter will behave as it was specified. The second method is fast but conservative, intended for applications without hard constraints of area and speed. Despite the simplicity of the second method, its results, though slightly overestimated, have been demonstrated to be correct by the results of the first method. A boost converter for power factor correction is used as an application example. As the converter model is intended for FPGA implementation, its area and maximum clock frequency are also analyzed. In this application example, the results show that the area grows linearly with the number of bits of each state variable, and the clock frequency is dominated by the width of one of the variables.**

***Index Terms*—Resolution analysis, Hardware-In-the-Loop, functional verification, digital control, switching converters, field programmable gate arrays.**

I. Introduction

DIGITAL regulators for power converters must be simulated before being implemented, to avoid any malfunction. There are usually several simulation stages. The first one is frequently done when the algorithm of the regulator is obtained, using transfer functions or other high-level models for the power converter. This stage can be accomplished with Matlab or similar control modeling tools. The second simulation stage should be carried out when the regulator is translated into its implementation model. In this way, implementation errors can be detected and the non-idealities and features of the digital implementation — such as conversion delay and pipelining — can be analyzed. Therefore, this simulation stage can check the controller in its final version state because the final regulator is being used along with a model of the power converter. As the final system has digital and analog parts, i.e., regulator and power converter, the simulation can be accomplished using a commercial mixed-simulation tool which simulates both parts. However, the simulation can be a long process, making this testing stage unaffordable. For instance, in PFC (Power Factor Correction), the voltage loop needs hundreds of milliseconds to reach its steady state while the switching and frequency can be high, so the simulation might be too long. Moreover, the controller may implement complex functionalities like heuristic algorithms, or it may use embedded processors [1], so the simulation could take even longer.

The simulation of digital regulators has been thoroughly studied, and multiple solutions have been proposed. Alternative models with different levels of accuracy have been compared using mixed-signal simulators, converter models described in VHDL (VHSIC Hardware Description Language, where VHSIC is Very High Speed Integrated Circuit) or VHDL-AMS (an analog and mixed-signal extension of VHDL) [2], [3], or using two simulators: one for the HDL (Hardware Description Language) controller and another for the power converter [4]. Simulations with HDL models of the power converter obtain faster simulations, although the design is more complex because the model must be designed by hand. To accelerate even further the debugging process, instead of simulating these models, they can be emulated in a FPGA (Field Programmable Gate Array) if the model is synthesizable. This is known as HIL (Hardware in the Loop),and it can use any hardware devices, including computers and FPGAs, for emulating the power converter. As shown in [5], the emulation of HIL systems can increase the speed of the second testing stage up to 29,000 times approximately in comparison with mixed-simulation tools, so the usefulness of the HIL approach is beyond doubt.

Most previous works have used computers for HIL purposes, but the integration step was usually in the order of hundreds or tens of microseconds [6]. This is enough for low switching frequencies, in the order of kilohertz or tens of kilohertz. If the target switching frequency is higher, the integration step should be smaller so that an accurate model may be obtained. With the aim of reaching tens or hundreds of nanoseconds of integration step, FPGAs have been introduced into HIL emulation [7]–[13].

Floating point is probably the easiest arithmetical representation for designing a power converter model. In [14], an HIL system using VHDL2008 *float_pkg* package is shown, while in [15] a power converter is synthesized using a HLST (High Level Synthesis tool), where the accuracy of floating-point operations is evaluated. However, the main problem of floating-point arithmetic is that it was not natively supported by synthesis tools until recent times. Another problem of

This work was partially supported by the Agencia Nacional de Promoción Científica y Tecnológica, Argentina, through Project PICT 2009 - 0041.

Oscar Goñi and Elias Todorovich are with Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina. (e-mail: oegoni@exa.unicen.edu.ar; etodorov@exa.unicen.edu.ar).

Alberto Sanchez and Angel de Castro are with the HCTLab Research Group, Universidad Autonoma de Madrid, Spain. (e-mail: alberto.sanchezgonzalez@uam.es; angel.decastro@uam.es).

synthesizable floating point is the low speed and high occupied area of the HIL system due to the complexity of the floating-point arithmetic.

Finally, a comparison of several simulation techniques, such as mixed-signal simulators, VHDL synthesizable and non-synthesizable models, fixed and floating-point models, and automatically translated VHDL models, is shown in [5]. This comparison proves that fixed point arithmetic obtains the best synthesis results (area and speed), but increases design time.

To the best of the authors' knowledge, no previous work has studied the resolution of the state variables of a model of a converter. When a power converter model is designed, its state variables must be calculated and stored (registered in the case of an HDL implementation). In every time step, the variables must be updated, by adding small incremental values. The width of a register defines the resolution of a variable. The International Vocabulary of Metrology [16] defines resolution as the smallest change in a quantity being measured that causes a perceptible change in the corresponding indication. In addition, it defines accuracy as the closeness of agreement between a measured quantity value and a real quantity value. In [17] the authors explain that the resolution of internal variables is directly related to the resulting accuracy of the computation. Moreover, in an iterative algorithm, the relationship can be far more complex.

Given a constant signal width, the smaller the time step (which aims at higher accuracy), the smaller the resolution of calculations. Therefore, when the system uses high switching and integration frequency, the signal width should also be increased. A tradeoff between resolution in the calculus and speed of simulation must be reached. Besides, in the case of emulation systems, the resources of the HIL system (for example, area occupied in an FPGA) should be analyzed. When fixed-point is used, the width of the arithmetical signals is critical, but resolution issues should also be taken into account when floating-point arithmetic is used. The resolution required for the arithmetical signals depends on the design parameters of the model, but it also varies over time when the input and the output currents and voltages change their values. Hence, the required resolution is not easy to calculate a priori.

An almost imperceptible error produced by insufficient resolution could turn out to be unacceptable many cycles later. The immediate solution to this problem involves using wider registers to store variables. However, this solution may result in high occupied area on the target device and a low clock frequency, if the correct register width is not used. Whatever technology is used, the optimization of the clock frequency and of the resource utilization is an important stage in the design of a hardware system. On the one hand, the resource optimization either makes it possible to include more synthesized modules on a single chip or reduces power consumption. On the other hand, a higher frequency provides faster HIL emulation. This motivates the need to make a resolution analysis to use the right register width for each specific application.

Some papers, such as [8], use fixed-point representation for the model of switching power converters. However, these authors use registers with specific fixed widths to store intermediate results. These widths are set by construction aspects beyond the mathematical model of the converters, such as FPGA hardware multipliers width (e.g., 18 bits on Xilinx Virtex 5) or the resolution of the used ADC (Analog to Digital Converter). In [12] a robust method for minimizing the error in propulsion-drive line-currents reconstructed from a single dc-link current measurement is presented. In [18] a parametrizable ultra low-latency digital processor core is presented and then used on an HIL system. The core is intended to be used in the validation of different industrial designs with a wide range of voltages and currents. However, the resolution used in registers for storing variables is fixed to support the worst scenario. If the correct resolution for each scenario were used, the latency, the frequency, and the occupied area could be improved.

In this work, two methods are proposed to find the optimum width for the state variables in HIL applications. The first method is simulation-based and it provides the optimum resolution needed based on the performance of the actual model. The second one is analytical and results in a conservative value for resolution of voltage and current. Although the first method requires more computing resources, it provides the most precise results about the minimum resolution that allows the desired accuracy. Furthermore, it guarantees that the converter will behave as expected, and that unexpected values of current and voltage will not occur. The second method can be easily applied by the designer when there are no rigid restrictions of area and frequency. The validity of its results is demonstrated using the results of the first method. Finally, the area and frequency are analyzed in order to show the importance of the right resolution selection.

The rest of the paper is organized as follows. Section II presents the importance of HIL systems and the resolution issues for an application example. Sections III and IV present Method 1 and Method 2, respectively. Section V presents the experiment scenarios and shows the analysis of area and frequency and the resolution analysis results of Method 1 and 2. Finally, section VI provides the conclusions.

II. Background

This section describes one of the possible applications for the proposed methods. In II-A the model of a boost converter for PFC (Power Factor Correction) is presented as an application example, while in II-B the resolution issues for this specific application are presented.

A. HIL Models

In this paper, a boost converter (Fig. 1) designed for PFC is used as an application example, but similar conclusions could be drawn for other topologies or applications.

A model of a boost converter needs to calculate the state variables, i.e., the input current (i_{in}) and the output voltage (v_{out}), every time step. In this experiment, the switching frequency (f_{sw}) has been set to $100\ kHz$, and each switching period has been divided into equal 1,000 time steps. Therefore, the time step is $\Delta t = 10\ ns$. It should be noted that $10\ ns$ is also the resolution of the PWM (Pulse Width Modulation).

In this application example, each component of the power converter is initially modeled with its ideal features. However,

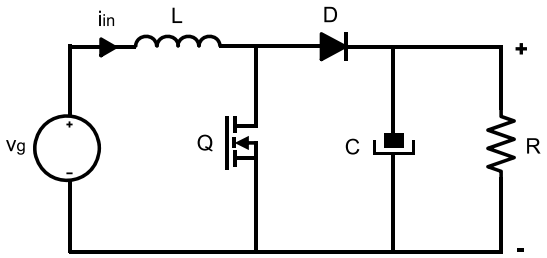

Fig. 1: Schematic of a boost converter

electrical and timing parameters of commercial components could also be added.

The model must take into account whether the switch Q shown in Fig. 1 is open or closed. Besides, the state of the converter depends on the value of the input current because this can be positive so the diode does conduct (CCM or Continuous Current Mode), or can be zero so the diode does not conduct (DCM or Discontinuous Current Mode). Hence, there are three possibilities (closed switch, open switch in CCM, and open switch in DCM), which are defined in (1), (2) and (3), respectively:

$$\begin{aligned} i_{in}(k) &= i_{in}(k-1) + \frac{\Delta t}{L} \cdot v_g \\ v_{out}(k) &= v_{out}(k-1) - \frac{\Delta t}{C} \cdot i_R \end{aligned} \tag{1}$$

$$\begin{aligned} i_{in}(k) &= i_{in}(k-1) + \frac{\Delta t}{L} \cdot (v_g - v_{out}) \\ v_{out}(k) &= v_{out}(k-1) + \frac{\Delta t}{C} \cdot (i_{in} - i_R) \end{aligned} \tag{2}$$

$$\begin{aligned} i_{in}(k) &= 0 \\ v_{out}(k) &= v_{out}(k-1) - \frac{\Delta t}{C} \cdot i_R \end{aligned} \tag{3}$$

The model of the boost converter should implement these three equations in order to simulate its behavior. Other elements could be added, like electrical parasitics, losses, and other nonidealities, but these are not added for the sake of clarity.

B. Resolution issues

Resolution issues about HIL systems have not been thoroughly studied in the literature. Early papers about HIL used low switching frequencies, so resolution problems did not arise because there is a relation between resolution and switching frequency, as will be shown later. However, improvements in digital electronics have made it possible to increase the switching frequency and diminish the integration step of the state variables.

The signal width should be chosen to achieve a trade-off between the speed and accuracy of the simulation. The width required for a signal is determined by both the maximum magnitude that can be stored in it and its resolution. In the case of a state variable, the resolution must be chosen taking into account the incremental values that have to be added every time step. The width increases with the difference between the orders of magnitude of signal value and its increments. That signal should have the correct width to store both values with enough resolution.

The width of a signal x can be determined by (4):

$$width_x = \lceil \log_2 \frac{x}{\Delta x} \rceil + n \tag{4}$$

where x is the value of the variable, Δx is the incremental value and n is the number of bits used to store the incremental value. When $n = 1$, this equation gives the minimum number of bits which are required to store x and Δx simultaneously. In this way, Δx is represented with a '0' or '1', so only multiples of Δx could be added, which results in low resolution in these values. Therefore, n should be greater than 1, so small incremental values around Δx can be stored. Next, two methods to get valid variable widths are presented.

III. Method 1 - Simulation-based approach

This method is based on the simulation of the actual converter model. Indeed, a set of boost converter models with different configurations is simulated in order to find the resolution for the worst-case scenario. As the converter models are targeted to synthesizable hardware, the simulation of the final converter can be accelerated by using emulation on a reconfigurable hardware (i.e., FPGA). In order to achieve flexibility on the test of each converter, a verification environment is built. This environment is useful not only to obtain the optimum resolution with minimum resources usage and maximum clock speed but also to guarantee the correct behavior of the final converter. The evaluation is done by instantiating the converter model with different configurations, collecting signal values, and comparing those values with a reference model. The proof consists in evaluating a converter model by setting many combinations of input current and output voltage widths, (n_i) and (n_v) respectively. The model of the power converter in the verification context becomes the DUV (Design Under Verification). The DUV used in this work accepts n_i and n_v values from 16 to 47 bits. Preliminary tests have shown that a resolution smaller than 16 bits causes unacceptable errors in I_{in} and V_{out}. In this work, an error is considered unacceptable when the difference between the converter model output and that of a reference model is greater than 20%. The reference model is essentially the implementation of the model of the boost converter that uses 64-bit floating-point variables.

The testbench environment (Fig. 2) is based on OVM (Open Verification Methodology) [19] framework since it provides a flexible and reusable structure needed to verify and analyze the converter models in the proposed scenarios and tests. The environment follows a layered approach. The lowest layer (DUV) interacts with the boost converter at bit signal level. At the mid layer, the *Driver* converts floating-point values of voltage to fixed-point registers. Additionally, the *Monitor* decodes the registers at the converter output to floating-point values for analysis purposes. At the top level, the

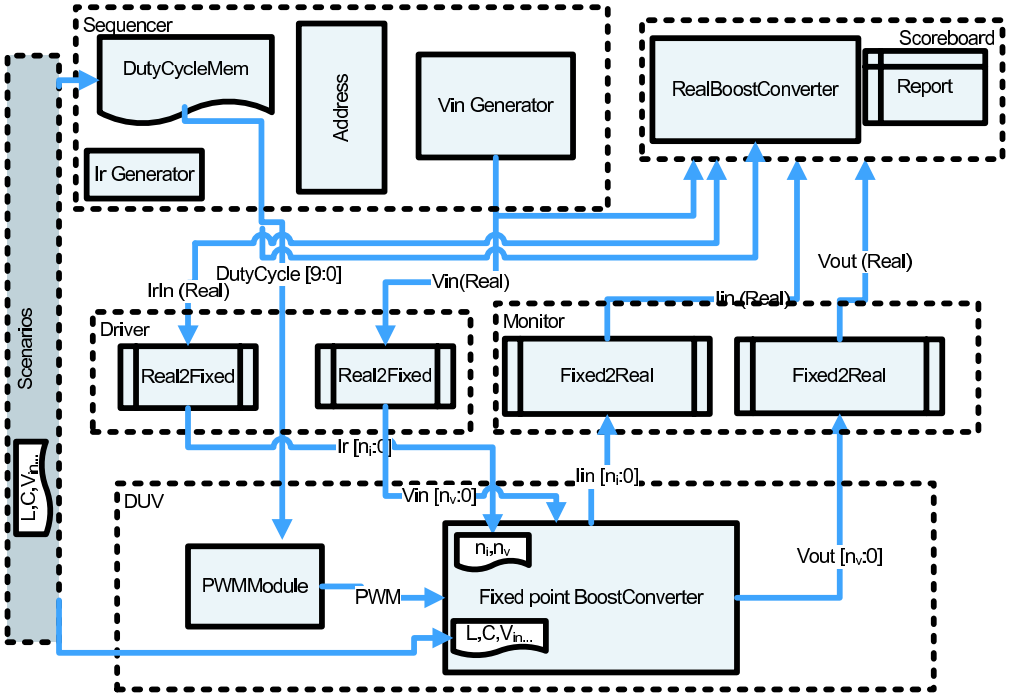

Fig. 2: Simulation environment

Sequencer provides mechanisms to create sequences of voltage values from mains line. The Scoreboard carries out the error calculation of I_{in} and V_{out} and stores them for later analysis. Moreover, assertion-based methods stop the simulation when the error of I_{in} or V_{out} exceeds the specified threshold. This mechanism sets aside invalid combinations of (n_i, n_v), thus speeding up the simulation process.

IV. Method 2 - Analytical approach

The main problem of the previously proposed method (Method 1) is that many simulations are necessary in order to get the minimum resolution that achieves the desired accuracy. This process can take long, so a faster method would be desirable. In this section, an analytical method that uses a simple formula similar to Eq. (4) is proposed. However, Method 2 leads to slightly greater number of bits than Method 1, so it does not guarantee the minimum resolution for the desired accuracy. In order to be conservative, Eq. (4) should not use mean or typical values but the worst case, so x should be the maximum value of the state variable that will be allowed in the simulation, while Δx should be the minimum incremental reasonable value of the state variable. The maximum value, or at least its approximate value, is easy to know because it only depends on the transients that may be simulated. However, the minimum incremental value is not trivial to calculate. Δx can be 0, so no bits would be needed to store that incremental value. Therefore, the minimum Δx excluding 0 should be found. The problem arises when Δx is near 0, because it can be an extremely small value, needing many bits to store the state variable. Nevertheless, very small incremental values have a low impact on the simulation, especially if they only appear during short periods of time. In order to find a trade off between accuracy and simulation time, or between accuracy and hardware resources for emulation, the second method determines state variables that are big enough to store the incremental values for most of the time, but not all the time.

This work proposes choosing a width which is able to calculate the state variable with a high level of accuracy 95% of the time. The accuracy is lower during 5% of the time because, during that time, the incremental values are extremely near 0. The error generated in this period of time is inherently small, because the incremental values are also small.

The boost converter presented in (II-A) is used as an application example. Similar calculations would be carried out for other applications or topologies taking into account the equations of their state variables and range of possible values. The state values for a boost converter with power factor correction are described in (1), (2), and (3).

In (1), the incremental value of the output voltage cannot usually be near $0V$, because the load is normally not near $0\ A$ in this application. However, the incremental value of the input current can be near $0A$ because it depends on the ac input voltage. Fig. 3 shows the rectified input voltage through a semi period. The 5% of the time with smaller values is discarded, so the minimum value of the input voltage that will be taken into account in Eq. (1) is $25.01\ V$. With this value, the incremental value of the input current in Eq. (1) is $\frac{\Delta t}{L} \cdot v_g = 5.002 \cdot 10^{-5}\ A$, considering $L = 5\ mH$. Taking this incremental value as the worst case, the width of the input current variable should be:

$$width_{i_{in}} = \lceil \log_2 \frac{i_{in}}{\Delta i_{in}} \rceil + n$$

$$width_{i_{in}} = \lceil \log_2 \frac{8}{5.002 \cdot 10^{-5}} \rceil + n$$

$$width_{i_{in}} = 18 + n \tag{5}$$

The width of the input current should be $18 + n$, where n is the number of bits used to store the incremental value, as explained above. In the previous equation the maximum value of the input current has been set to $8\ A$, but this parameter, which is set by the designer, is a limitation for the transients that can be simulated by the model.

The same analysis must be done with the other equations of the boost converter. The input current in Eq. (2) does not face any resolution problem, because the difference between the input and output voltages is not near $0V$. However, the difference between currents, needed to calculate the output voltage, is periodically near $0V$. Fig. 4 shows input and output currents. The same methodology is accomplished, discarding the 5% of the time when the difference is nearest $0V$. As can be seen, the minimum difference, once this percentage is discarded, is $0.0656\ A$. With this difference, the incremental value of the output voltage in Eq. (2) is $\frac{\Delta t}{C} \cdot (i_{in} - i_R) = 6.56 \cdot 10^{-6}\ V$, if $C = 100\ \mu F$. With this minimum incremental value, the width of the output voltage variable is:

$$\begin{aligned} width_{v_{out}} &= \lceil \log_2 \frac{v_{out}}{\Delta v_{out}} \rceil + n \\ width_{v_{out}} &= \lceil \log_2 \frac{1,000}{6.56 \cdot 10^{-6}} \rceil + n \\ width_{v_{out}} &= 27 + n \end{aligned} \tag{6}$$

Therefore, the width of the output voltage variable should be $28+n$ bits. This time, the maximum output voltage is $1,000\ V$, and it is again a parameter for this application chosen by the designer.

Finally, the same analysis must be done for Eq. (3). In this case, the input current variable does not face any resolution problem, because the input current is $0\ A$ in DCM, with no incremental value. Besides, the output voltage changes with the output current, but the load is not normally near $0\ A$, so no further considerations should be done in this case.

To sum up, the input current signal width should be $18 + n$ bits, while the output voltage signal width should be $28 + n$ bits. The parameter n describes the number of bits of the incremental value, which is at least 1. As explained above, Eq. (4) represents the number of bits required to store a variable and its incremental values. Δx represents the typical values of the increments, but the real incremental values are not exactly equal to Δx, so more than one bit ($n = 1$) is needed to store these incremental values with accuracy. Considering the results of Method 1, this work proposes that n is around 8 so the incremental value has enough resolution. Therefore the width of the input current should be 26 bits, while the width of the output voltage should be 36 bits for this specific case, where L = 5 mH and C = 100 μF.

In this work, the analysis has been performed for fixed-point signals. When floating-point signals are used, it may be thought that it is not necessary to determine the width of the signals, because a floating-point variable can store big or small values without changing its width. However, as stated above, a variable should store its typical value and also its incremental values. A 32-bit IEEE-754 floating-point variable uses 24 bits

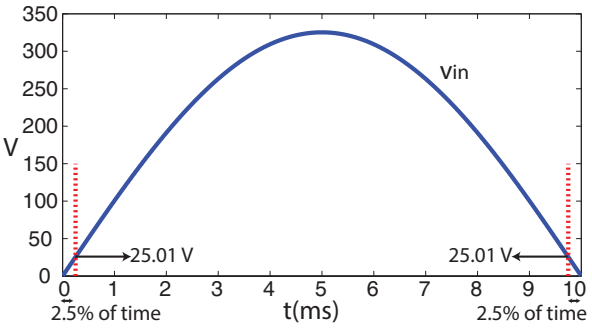

Fig. 3: Minimum values of the input voltage taken for resolution analysis

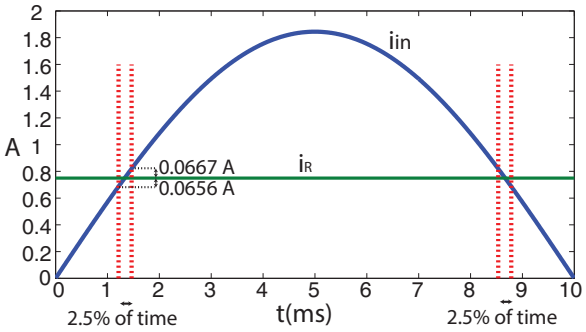

Fig. 4: Minimum values of the difference between currents taken for resolution analysis

for the significand: a fixed 1 and 23 additional bits. If V_{out} is around $1,000\ V$, the MSB in floating point is 2^9 and the LSB is 2^{-14}, that is $6.103 \cdot 10^{-5}\ V$. As explained in Eq. (6), the worst case considered for the incremental value of the output voltage is $6.56 \cdot 10^{-6}$, so a 32-bit IEEE 754 floating-point signal does not have enough resolution to simulate this application.

When the width of a 32-bit IEEE-754 signal is compared with the previous analysis, the floating-point signal has 24 significand bits, while the input current should be 26 bits, and the output voltage should be 36 bits. A possible solution is to use 64-bit IEEE-754 floating-point variables, but the design area in the case of implementation in hardware would be large, and the simulation speed would be seriously affected.

The proposed widths for the input current and output voltage have been chosen as the worst but reasonable case. Method 1 experiments can also check the results of Method 2. On the one hand, the results will check if the decision of discarding the 5% of time can be chosen as the worst case, which means that adding more bits will not significantly improve the accuracy of the simulation. On the other hand, the results will check whether the worst case must be chosen to calculate the widths, or whether a more optimized width can be chosen to speed-up the simulation. The experimental results in Section V have been extracted from fixed-point models, but can also be applied to check if the significand of an IEEE-754 model is accurate enough.

V. RESULTS

A. Test scenarios

The resolution evaluation is done using the boost converter model with several configurations of V_{in}[1] , L, C, V_{out} and P_{out}, one at a time. This spectrum of configurations (Table I) describes some possible real application scenarios. The L and C values are kept fixed for each scenario evaluation.

TABLE I: Converter evaluation scenarios

	L	C	V_{in}	V_{out}	P_{out}
Scenario 1	5 mH	100 μF	230 V	400 V	300 W
Scenario 2	1 mH	100 μF	230 V	400 V	300 W
Scenario 3	1 mH	100 μF	110 V	300 V	150 W
Scenario 4	1 mH	470 μF	230 V	400 V	300 W

To make a broader analysis, each DUV with a specific configuration is simulated using several types of load in every scenario:

a. current test: the load is modeled as a current sink.
b. power test: the load is modeled as a power sink.
c. resistive test: the load is modeled as a resistance.

Method 1 is evaluated by simulating each converter model as a DUV in a verification environment. DUV configuration and the type of load are kept fixed during the simulation. The tests are executed in open loop under the same sequence of duty cycle values. The open loop simulation lets the voltage and current values develop freely. A closed loop configuration can cover up the error produced by insufficient resolution and may lead the designer to make an incorrect design parameter selection.

The performance of each converter model is evaluated considering the mean absolute error values of I_{in} and V_{out} during the simulation. It should be mentioned that the relative error cannot be applied, in particular in current analysis, since there are situations when their values become zero and the relative error is indeterminate. The simulation duration has been set to 140 ms. This duration has been selected to let I_{in} and V_{out} signals stabilize after the transient period, thus ensuring that the model of the converter is evaluated under both a transient and a steady state. The tests execution is automated by a *TCL* (Tool Command Language) script. The script sets the design parameters n_i and n_v, as well as the scenario and test conditions, and finally launches the simulation. Once the simulation is finished, the errors are ordered in tabular form to provide the designer with a set of combinations of (n_i, n_v) that drives a converter with errors up to a certain threshold value.

Next, Method 2 is evaluated by applying Eq. (4) for each scenario and state variable, as explained in IV. The results of applying Method 2 are described in V-D.

B. Frequency and area analysis

The application of Method 1 will lead to a set of solutions (n_i, n_v) to build a fixed-point converter model that fulfills specifications of mean absolute error of I_{in} and V_{out}. In addition, Method 2 can be used to get a conservative solution although it does not consider error specification.

All of these combinations comply with the restrictions, but there is a subset of them - maybe one - that minimizes the occupied area on the target device, maximizes the clock frequency, or both. In order to analyze the impact on area and frequency, all converters are synthesized using each combination of design parameters (n_i, n_v). On the one hand, the area occupied in the target device for the converter is studied in terms of the number of multipliers, LUTs (Look-Up Tables) and slices inferred by the synthesizer. On the other hand, the maximum clock frequency for each converter is evaluated. A *TCL* script is used to automate the process. First the synthesis is carried out with different combinations of design parameters (n_i, n_v), and then their synthesis reports are parsed to obtain the area and maximum frequency information.

In this work, the synthesis is targeted to a *Virtex V xc5vlx20t* FPGA using the XST tool configured with default parameters and automatic constraints. The converter has been synthesized 256 times with different combinations of design parameters: 16 values for n_i x 16 values for n_v.

For all combinations of (n_i, n_v), the synthesis tool has inferred two 18x18-bit multipliers. On the other hand, each labeled curve in Fig. 6 bounds a subset of design parameters (n_i, n_v). The converters that use the combinations above each labeled curve use up to that number of slices. It can be seen that the number of used slices grows linearly when a new bit is added in current or voltage registers.

Next, the information about maximum clock frequency to be used with a specific combination of parameters is considered. The horizontal lines in Fig. 7 show that for any width of current registers, the maximum clock frequency remains for a given width of voltage registers. That means that the maximum circuit frequency depends on the width of V_{out} registers in this application example. This dependence only on V_{out} is caused by a critical path in the voltage calculus. This calculus cannot be pipelined because the computation of each new value, k, needs the immediately previous value, k-1, as stated in Eq. (1), (2) and (3).

C. Method 1 results

The experiments are carried out using the following software/hardware platform:

- *Simulator*: Mentor Graphics Questasim-64bit® 10.0c.
- *Synthesizer*: Xilinx XST®0.4D.
- *Hardware*: Intel i5® - 4Gb RAM running on Microsoft Windows 7 ®.

After executing all tests in all scenarios, current and voltage errors are analyzed. The converters performance is graphically summarized through projections as follows. In Fig. 5, the x axis represents the values assigned to n_i, and the y axis represents the values assigned to n_v. Both axes define a plane of possible design parameters (n_i, n_v). The z axis represents the percentage error, so the V_{out} and I_{in} error information can be seen as 3D surfaces. As an example, Fig. 5 shows a plane for those combinations of parameters with an error in V_{out} of 5% and a plane for those with an error of 2%.

[1]RMS rectified input voltage

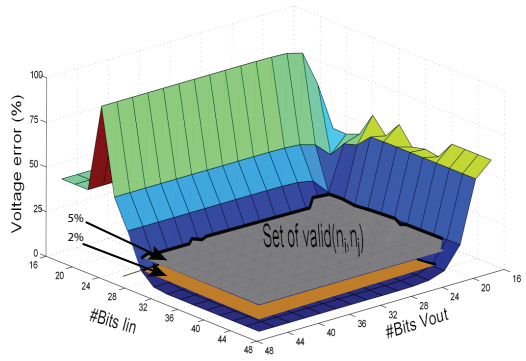

Fig. 5: Projected planes of design parameters for converters with voltage $error \leq 5\%$ and $error \leq 2\%$

Thus, the contour lines in Figs. 8 to 11 are projections of the intersection between the error surface and planes parallel to the (*x*, *y*) plane at different values of percentage error. In this way, each contour line delimits a set of design parameters with an error below a certain threshold. The curves presented with dotted lines stand for projections in current error, while solid lines stand for projections in voltage error.

It can be observed in Figs. 8 to 11 that each labeled voltage curve does not intersect the current curve with the same label. This behavior repeats for all tests in all scenarios and shows that the current error dominates the selection of design parameters within the set of valid combinations. The percentage of voltage error is always smaller than the percentage of current error for any studied combination (n_i, n_v).

Figs. 8 to 11 show the error obtained for converter models. Note that the error in current can vary from 50% to 0% in only five bits of resolution (i.e., 17 to 22 bits in current). This fact strengthens the hypothesis that an incorrect selection of design parameters leads to undesirable current and voltage values. When n_i is greater than 23 and n_v is greater than 30, the mean absolute error in current and voltage is zero or tends to be zero.

TABLE II: Design parameters n_i, n_v for 2% error

	Current	Power	Resistive
Scenario 1	(23, 30)	(24, 30)	(22, 30)
Scenario 2	(24, 31)	(24, 32)	(22, 32)
Scenario 3	(23, 31)	(23, 32)	(23, 32)
Scenario 4	(22, 31)	(23, 31)	(24, 31)

TABLE III: Design parameters n_i, n_v for 5% error

	Current	Power	Resistive
Scenario 1	(21, 26)	(20,25)	(20, 26)
Scenario 2	(21, 29)	(21, 29)	(20, 30)
Scenario 3	(22, 30)	(22, 30)	(22, 31)
Scenario 4	(21, 29)	(21,30)	(22, 29)

Tables II and III summarize the best solutions, according to synthesis results, which fulfill the restrictions to build a converter with a mean average error under 2% and 5% in current and voltage, respectively.

In Figs. 6 and 7, it can be noted that there is a difference of about 5 slices and 2 MHz between the curve with the smallest error in current and voltage and that with the largest error, respectively.

These facts suggest that, for the proposed scenarios, the selection of the correct design parameters can be done by selecting the maximum values in Table II, n_i=24, n_v=32.

D. Method 2 results

The analytical method is applied in order to obtain conservative combinations of design parameters (n_i,n_v). As it is based on the converter model formulae, the method does not consider the load type. However, as every scenario configuration has an impact on the model, different calculi have been done for each configuration. Table IV summarizes the resolution for each scenario computed by Method 2. It can be observed that voltage register resolution ranges from 34 to 38 bits, while current register resolution ranges from 23 to 26 bits. The upper limits of the resolution ranges of Method 1 (considering the different scenarios and the load type) show that Method 2 is conservative, as expected, and that the value n = 8 fits for this specific application. However, the parameter n could be difficult to know a priori if Method 1 was not applied previously.

TABLE IV: Design parameters n_i, n_v using the analytic method when $n = 8$

	(n_i, n_v)
Scenario 1	(26, 36)
Scenario 2	(23, 36)
Scenario 3	(25, 34)
Scenario 4	(23, 38)

As Method 2 does not consider maximum error in Eq. (4), there is a threshold error value for Method 1, where signal widths are larger than those for Method 2. This situation can be observed by comparing Scenario 4 in Tables II (Resistive load) and IV. It can be noted that n_i values for Method 1, are one bit larger. Nevertheless, it can also be observed in Table III that this situation does not occur for errors larger than 5%.

VI. Conclusions

This work focuses on the study of resolution in state variables of fixed-point switching-mode power converter models. Two different methods addressing the correct selection of resolution values have been proposed. The first method is simulation-based. The resolution analysis is done by measuring the mean absolute error of the converter model with several combinations of resolutions in current and voltage registers while its functionality is verified. Although the drawback of Method 1 could be its execution time, it results in the optimum resolution of registers, and it guarantees the correct behavior of the model.

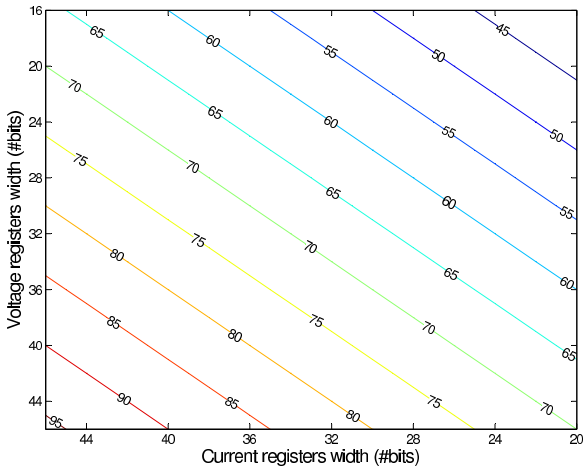

Fig. 6: Slices usage for fixed point converter

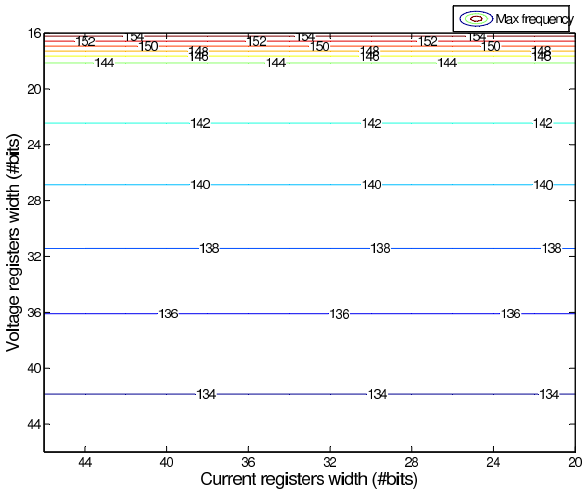

Fig. 7: Maximum clock frequency

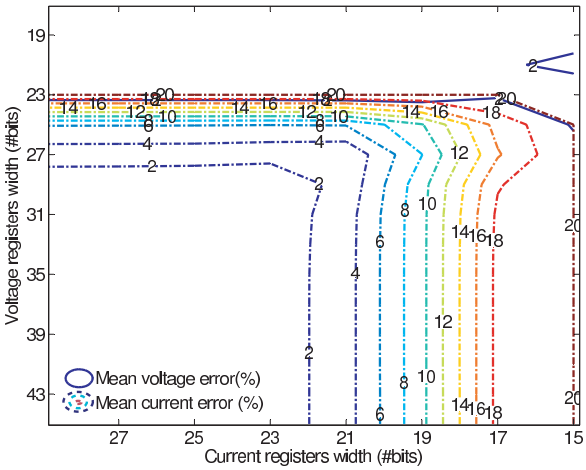

Fig. 8: Voltage and current errors in scenario 1

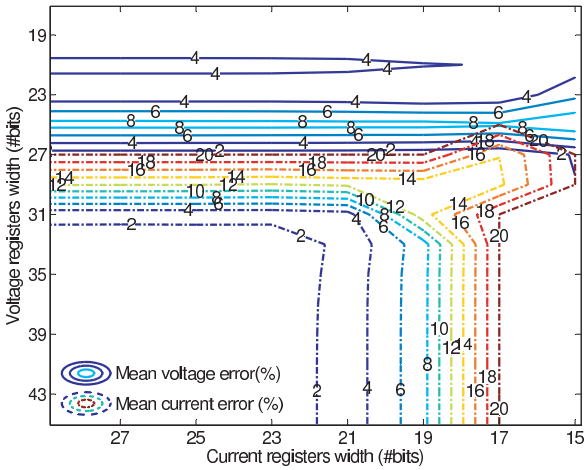

Fig. 9: Voltage and current errors in scenario 2

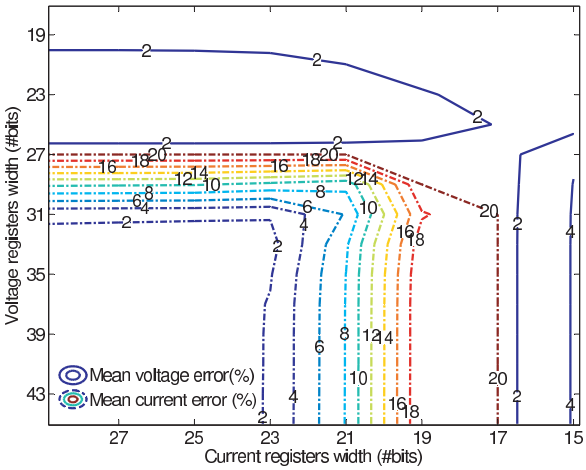

Fig. 10: Voltage and current errors in scenario 3

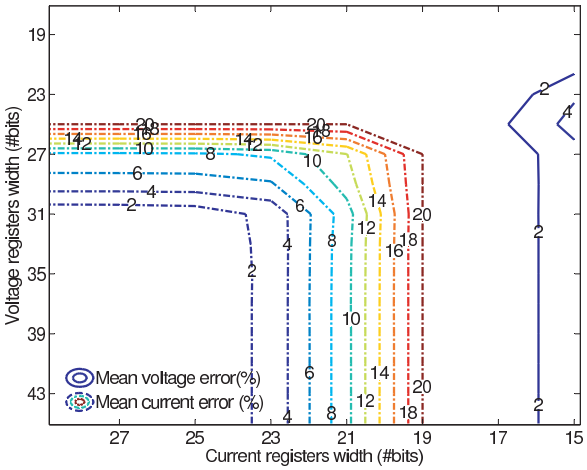

Fig. 11: Voltage and current errors in scenario 4

The second method is analytical and overestimates the resolution of state variables. Its overestimation can increment the resources utilization on the target device and can bound the clock frequency, but more important is that the error of the resulting model is unknown with the second method. However, the application of this method is fast and simple.

After the simulation of a complete set of configurations, the first method demonstrates that the mean absolute error in current is always greater than the mean absolute error in voltage for any resolution in this application example.

Since several configurations fulfill the same current and voltage restrictions, a synthesis analysis is done to select the best resolution in terms of area and speed. The synthesis results indicate that slice usage on the target device increases linearly when a bit is added to the resolution of voltage or current registers. The synthesis has also demonstrated that the maximum clock frequency is dominated by the width of voltage registers.

Method 1 demonstrates that the resolution computed by Method 2 is correct but conservative. Besides, the area and frequency analysis shows the differences in area and maximum clock frequency between using the conservative analytical approach solution and the simulation-based optimum solution. For the proposed example, the resolution computed by Method 1 improves the converter area by about 10%, while the clock frequency optimization is rather small, about 2%.

The experiments have demonstrated that the errors grow sharply from 0% to 50% in only five bits under the optimum resolution for the proposed application example. If the resolution is below that interval, the errors turn out to be unacceptable.

In short, the application of the proposed methods demonstrates that it is possible to implement switching mode power converter models using fixed-point representation without loss of precision, while the area occupied is minimized and the clock frequency is maximized.

REFERENCES

[1] O. Lucía Gil, L. Barragán, J. Burdío, O. Jiménez, 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.

[2] A. Prodic and D. Maksimovic, "Mixed-signal simulation of digitally controlled switching converters," in *Computers in Power Electronics, 2002. Proceedings. 2002 IEEE Workshop on*, June 2002, pp. 100 – 105.

[3] L. Barragan, I. Urriza, D. Navarro, J. Artigas, J. Acero, and J. Burdio, "Comparing simulation alternatives of FPGA-based controllers for switching converters," in *IEEE International Symposium on Industrial Electronics (ISIE)*, June 2007, pp. 419 – 424.

[4] P. Zumel, M. García-Valderas, A. Lázaro, C. López-Ongil, and A. Barrado, "Co-simulation PSIM-ModelSim oriented to digitally controlled switching power converters," in *Control and Modeling for Power Electronics (COMPEL), 2010 IEEE 12th Workshop on*, June 2010, pp. 1 – 7.

[5] 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. 99, pp. 491 – 500, 2012.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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, January 2011.

[11] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 2, pp. 1300 – 1309, February 2012.

[12] E. Adžić, M. Adžić, V. Katić, D. Marčetić, and N. Ćelanovič, "Development of high-reliability ev and hev im propulsion drive with ultra-low latency HIL environment," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 2, pp. 630 – 639, 2013.

[13] A. Hasanzadeh, C. Edrington, N. Stroupe, and T. Bevis, "Real-time emulation of a high speed micro-turbine permanent magnet synchronous generator using multi-platform hardware-in-the-loop realization," pp. 1–1, 2013.

[14] O. Lucía, I. Urriza, L. Barragán, D. Navarro, O. Jiménez, and J. Burdío, "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.

[15] D. Navarro, O. Lucía Gil, L. Barragán, I. Urriza, and O. Jiménez, "High-level synthesis for accelerating the FPGA implementation of computationally-demanding control algorithms for power converters," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 3, pp. 1371 – 1379, 2013.

[16] *International vocabulary of metrology - Basic and general concepts and associated terms (VIM) Vocabulaire international de metrologie - Concepts fondamentaux et generaux et termes associes (VIM)*, JCGM, 2008.

[17] G. Constantinides, N. Nicolici, and A. Kinsman, "Numerical data representations for FPGA-based scientific computing," *Design Test of Computers, IEEE*, vol. 28, no. 4, pp. 8 – 17, July-August. 2011.

[18] D. Majstorovic, I. Ćelanovič, N. Teslic, N. Ćelanovič, and V. Katic, "Ultralow-latency hardware-in-the-loopin-the-loop platform for rapid validation of power electronics designs," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 10, pp. 4708 – 4716, October 2011.

[19] M. Glasser, *Open Verification Methodology Cookbook*. Springer Dordrecht Heidelberg London New York, 2009.

Oscar Goñi was born in Necochea, Argentina, in 1980. He received a computer engineering degree from the State University UNCPBA of Tandil (Buenos Aires), Argentina, in 2009. From 2009 to 2013 he was an assistant professor at the UNCPBA. His research interests include functional verification, programmable logic and embedded systems.

Alberto Sanchez was born in Madrid, Spain, in 1986. He received the M.Sc. and the Ph.D. degrees in computer science and telecommunication engineering from the Universidad Autónoma de Madrid, Spain, in 2010 and 2013 respectively. He is a Ph.D. Assistant Professor in the Technology for Electronics and Communications Department. His research interests include simulation of mixed digital and analog systems, digital control of switching mode power supplies and wireless sensor networks with mobile nodes.

Elías Todorovich (M'10) received a computer engineering degree from the State University UNCPBA of Tandil (Buenos Aires), Argentina, in 1997, and a Ph.D. degree in computer engineering from the Universidad Autónoma de Madrid (UAM), Spain, in 2006. From 2003 to 2009 he was an assistant professor in the Department of Computer Engineering, UAM and is currently a professor at the University FASTA of Mar del Plata, Argentina and associate professor at the UNCPBA. His research interests include programmable logic, digital image processing, and functional verification.

Angel de Castro (M'08) was born in Madrid, Spain, in 1975. He received the M.Sc. and the Ph.D. degrees in electrical engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 1999 and 2004, respectively. He has been an Associate Professor in the Universidad Autońoma de Madrid since 2010, and as Assistant Professor from 2006 to 2010. Previosly, he was an Assistant Professor in the Universidad Politécnica de Madrid since 2003. His research interests include digital control of switching mode power supplies, power factor correction and field programmable gate arrays.