



Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:
This is an **author produced version** of a paper published in:

2016 IEEE 17th Workshop on Control and Modeling for Power Electronics
(COMPEL). IEEE, 2016. 7556670

DOI: <http://dx.doi.org/10.1109/COMPEL.2016.7556670>

Copyright: © 2016 IEEE

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

Hardware-in-the-loop using parametrizable fixed point notation

Alberto Sanchez, Irene Villar, Angel de Castro, Fernando López Colino and Javier Garrido

HCTLab research group

Universidad Autonoma de Madrid

Madrid, Spain

Email: {alberto.sanchezgonzalez, angel.decastro, fj.lopez, javier.garrido}@uam.es, irene.villar@hctlab.com

Abstract—The verification of digital regulators designed to control power converters is not trivial because the plant is analog while the regulator is digital. There are several methodologies to accomplish this task, but there is no standard method and, usually, the verification is a slow process. An alternative is to use an HIL (Hardware-in-the-loop) system which emulates in hardware a digital model of the plant, achieving significantly faster simulations. This paper explains how to implement a simple but fast mathematical model for a full-bridge converter and how to implement it using parametrizable fixed point arithmetic. Fixed point arithmetic is able to achieve faster simulations compared to floating point while using less hardware resources. This paper shows that this model can emulate the converter in real-time using a time step of 23 ns.

Index Terms—Hardware-in-the-loop, real-time simulation, FPGA.

I. INTRODUCTION

In recent years, the simulation of power converters using HIL (Hardware-in-the-Loop) systems is increasing, because of its speed-up compared to traditional simulation techniques. The idea is to implement the model of the plant in digital hardware, which can be a computer, a microprocessor, an FPGA, etc. The first debugging step is usually accomplished while designing the transfer function of the regulator in a control tool, such as Matlab. This is not complex, because all the parts (controller and power converter) are modeled in the same tool. The use of digital regulators is very common in recent decades, using DSP (Digital Signal Processor), microcontrollers, ASICs (Application-Specific Integrated Circuit) or FPGAs (Field Programmable Gate Array) [1], [2]. However, once the digital controller is implemented in the final processing unit, it is necessary to debug it again.

One alternative to the HIL systems is using mixed-signal simulators or VHDL-AMS simulators [3], [4], or using two simulators: one for the regulator as it is designed in VHDL and other simulator for the analog plant [5].

To accelerate the debugging process, HIL systems can be used. The first proposals in the literature used computers, but their simulation time step was high and they were only useful for low frequency applications [6]. More recently, FPGAs have been introduced in HIL systems for acceleration. For example, in [7], [8], a Matlab model is used which is automatically translated into VHDL.

In [7]–[12], several HIL systems based on FPGAs model low switching frequency converters. In these proposals, fixed point numerical notation were used. Fixed point representation achieves optimal results in area and speed, but the design effort is increased compared to using floating point representation.

In [13], `float_pkg`, which is a library inside VHDL2008, is proposed for HIL. This library helps to implement synthesizable floating points operations. Floating point helps the designer to model the plant, but its running frequency is around ten times slower, and the used area is more than ten times greater [14].

In [15] a power converter is synthesized using a HLST (High Level Synthesis tool) where the accuracy of floating-point operations is evaluated. HLST models are easy to implement but the results in terms of area and speed are not optimized.

Another remarkable difference between fixed and floating point notations is the representation range of a number. In fixed point, the range is constrained at design time, while in floating point the location of the decimal point can be shifted when necessary, although the resolution of the mantissa is fixed (23+1 bits in IEEE-754 single precision). Therefore, the main drawback of fixed point is that the simulation values cannot exceed the range imposed in the design stage of the model.

In [16], a parameterizable digital processor core is presented to implement 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 the state variables is fixed to support the worst scenario. Therefore, if the correct resolution for each scenario was used, the accuracy would be improved.

This paper proposes the implementation of an HIL model based on parametrizable fixed point representation which can adapt its resolution to any situation. This approach allows configuring the location of the decimal point at simulation time instead of design time, so it is not necessary to reimplement the model if the simulation conditions change. In this way, the advantages of both fixed point and floating point are obtained.

The rest of the paper is organized as follows. Section II explains how to model a power converter plant. Section III describes its implementation. Section IV presents the results and, finally, conclusions are given in Section V.

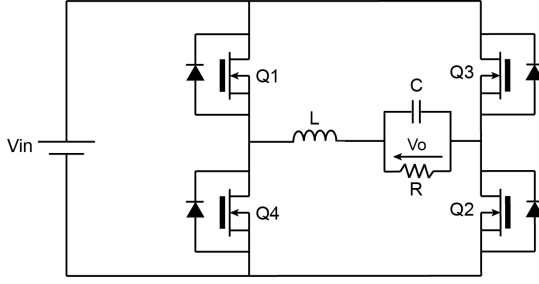


Fig. 1: Topology of a full-bridge converter

II. MODEL OF THE PLANT

This paper presents an HIL system based on an FPGA. The application example is a full-bridge converter (see Fig. 1). Regarding the arithmetics which can be used, there are several possibilities to model the plant in an FPGA:

- 1) Floating point representation: It allows shorter design time because the designer can transcribe the mathematical equations almost directly to VHDL, without taking into account the representation ranges of a number, because the point is shifted automatically when necessary. For example, in [13] it can be found an HIL model using this 32-bit floating point representation (single precision standard). However, resolution problems have arisen in models used with high frequency switching [14], [17] (hundreds of kHz). Resolution problems are directly related to the resulting accuracy of the computation, as it was seen in [18], so 32-bit floating point may not be appropriate to implement models to verify high frequency regulations. Double precision can be used but the results of area and speed dramatically deteriorate even more. It should be noticed that 32-bit floating point is still much slower than fixed point arithmetic.
- 2) Fixed point representation: It allows smaller simulation steps and uses less area than floating point, but it requires more design effort. This is because every signal should be implemented taking into account the value ranges inside the simulation. The point location is fixed, so the number of bits for the integer and fractional parts are constrained at design time. A conservative decision would be to allocate many bits to the integer part to avoid numerical saturation/overflow, but the resolution would decrease.
- 3) Parametrizable fixed point representation: It uses the basis of fixed point representation, taking advantage of the high speed and low area. However, the number of bits for the integer and fractional parts are not known a priori. This model uses integer signals and the model is not aware about the point location. Thus, all inputs and outputs of the model are externally interpreted. Depending on the interpretation, the number of integer and fractional bits are changed so, given the simulation limits, the system adjusts the point location of every signal without resynthesizing.

Independently of the chosen arithmetic, the converter should be modeled. This paper shows a simple mathematical model using difference equations. The full-bridge converter can be modeled analyzing its state variables, which are the output voltage (v_{out}), and the inductor current (i_L). The inductor voltage and the current through the capacitor are defined by:

$$\begin{aligned} v_L &= L \cdot \frac{\delta i}{\delta t} \\ i_C &= C \cdot \frac{\delta v}{\delta t} \end{aligned} \quad (1)$$

These equations can be translated into difference equations in order to define the state variables which depend on the values of v_L and i_C and, then, also on the state of the switches.

$$\begin{aligned} i_L(k) &= i_L(k-1) + \frac{\Delta t}{L} \cdot v_L \\ v_{out}(k) &= v_{out}(k-1) + \frac{\Delta t}{C} \cdot i_C \end{aligned} \quad (2)$$

Where k represents a simulation step, Δt is the simulation time step, L is the inductance, C the capacitance, v_L the voltage of the inductor and i_C the current through the capacitor. Taking into account the states of the switches, Eq. (3) and (4) define the plant behavior when the two control branches are active: 1 (switches Q1 and Q2 closed) and 2 (switches Q3 and Q4 closed) respectively.

$$\begin{aligned} i_L(k) &= i_L(k-1) + \frac{\Delta t}{L} \cdot (v_g - v_{out}) \\ v_{out}(k) &= v_{out}(k-1) + \frac{\Delta t}{C} \cdot (i_L - i_R) \end{aligned} \quad (3)$$

$$\begin{aligned} i_L(k) &= i_L(k-1) + \frac{\Delta t}{L} \cdot (-v_g - v_{out}) \\ v_{out}(k) &= v_{out}(k-1) + \frac{\Delta t}{C} \cdot (i_L - i_R) \end{aligned} \quad (4)$$

Both branches cannot be enabled simultaneously so the regulators usually add deadtimes, when no branch is activated for a short time. When all switches are open, the current crosses the antiparallel diodes. In this way, when i_L is positive, the equations (4) of branch 2 must be applied, and if i_L is negative, the equations of branch 1 (3) are applied.

Electrical losses can be also modeled but they have not been added in this manuscript for the shake of clarity. Once the equations are defined, in section III several arithmetics will be used in order to make a comparison.

III. IMPLEMENTATION

In this section *real* and parametrizable fixed point arithmetics are explained. The *real* arithmetic is quite easy to use because the equations (3) and (4) can be automatically translated into VHDL without any other consideration. However, the main drawback is that *real* arithmetic cannot be synthesized so it cannot be used in an HIL system.

Resolution problems should not be arisen as *real* arithmetic uses IEEE-754 double precision standard with a mantissa width of 53 bits (1+52). For example, if v_{out} variable is in the order of 200 V, the mantissa starts with 1.1001000, where the first one is not written in the 64-bit number, so 45 bits are free in order to store decimals in the integration process. In this case, the resolution of the *real* signal is around $2.84 \cdot 10^{-14}$ V. If IEEE-754 single precision (32-bits) were used, only 16 bits would be free and the resolution would be around $1.53 \cdot 10^{-5}$ V. If the integration step (Δt) is small, resolution problems can appear because the incremental values of the state variables can be under that resolution.

On the other hand, the proposed fixed point notation model uses QX.Y representation. This format contains 1 bit which allows sign, and X and Y bits for the integer and fractional parts respectively, so a Q4.11 signal has 1+4+11=16 bits. To obtain the decimal value of a QX.Y number, it should be multiplied by 2^{-Y} . For example, given an output voltage of 0101001011010000₂ V in binary (21200 in decimal) with a scale of 11, the output voltage is $21200 \cdot 2^{-11} = 10.3515625$ V.

However, the proposed model is not aware of the value of X and Y, so it only operates the numbers as if they were integers. The proposed system is parametrizable, so the scale (value of Y) of every signal should be adapted to the needed value range. The scale of a signal can be obtained with Eq. (5):

$$scale = Total_width - \lceil \log_2 Max_value \rceil \quad (5)$$

where $Total_width$ is the number of bits of the variable, including the integer and the fractional parts. For example, if the simulation must handle output voltages up to 200 V, and the state variable has 57 bits (plus 1 of sign), the scale is defined by $57 - \lceil \log_2 200 \rceil = 49$, so 49 bits are used to store fractional values (Q4.49 variable), with a resolution of $2^{-49} = 1.78 \cdot 10^{-15}$ V.

It can be seen in Eq. (5) that the resolution is adjusted depending on the value range, improving the versatility of the classic fixed point notation. The scale of the signals should be calculated before simulating in order to send to the model the input values and interpret its output values. However, this can be done without resynthesizing but only configuring the *scale change* blocks shown in Fig. 2.

Fig. 2 shows the schematic of the full-bridge model using this approach. Two main parts can be seen: the calculus of the inductor current and the output voltage. The total number of bits of every signal and their scales are shown in the figure. As there is a feedback in the model (there is dependency between current and voltage), the internal variables (in their own scale) are shifted when passed to the other part of the model, so both operands in the adders/subtractors are in the same scale. However, multipliers do not need the operands to be in the same scale, but the operands scales are added in order to get the new scale.

The advantage of this proposal is that fixed point arithmetic is simpler than floating point so wider variables can be used using less resources. Moreover, this model is optimized

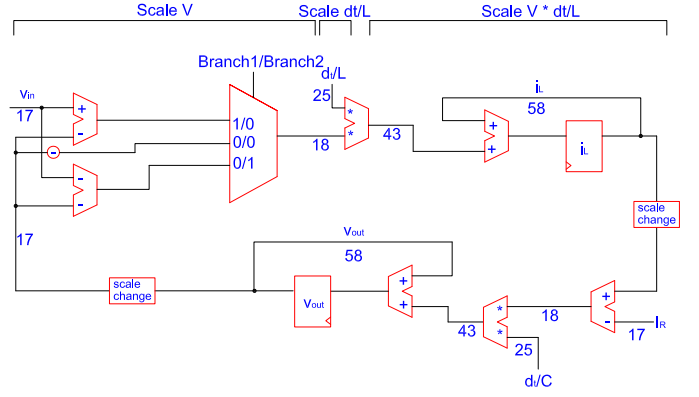


Fig. 2: Implementation of the parametrizable fixed point model

using more bits for the integrators which calculates the state variables, while less bits are used for the feedback signals as these signals do not require high resolution. Therefore, the combinational critical path in the FPGA can be reduced.

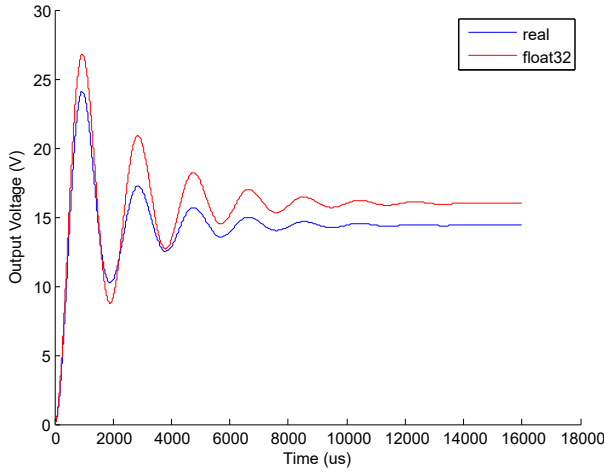
Pipelining techniques cannot be applied in this case because both state variables depend on the previous value of each others, so adding pipeline registers will modify the model equations seen in (3) and (4) because of the pipeline registers.

IV. EXPERIMENTAL RESULTS

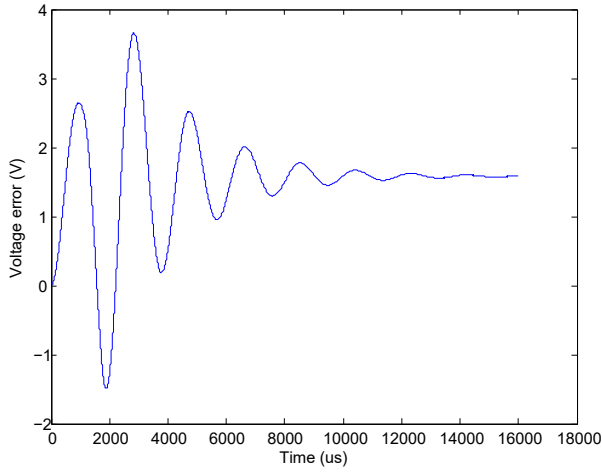
The synthesizable float32 and also the parametrizable fixed point approaches have been implemented using an FPGA Xilinx xc7z020-1. Both models have been compared with *real* model (non-synthesizable 64-bit floating point model), whose resolution is sufficiently high, so it can be considered as a reference system. The physical parameters of the model are $V_{in} = 20$ V, $L = 900 \mu H$, $C = 100 \mu F$, $R = 12 \Omega$, and a simulation time step of 23 ns. In order to achieve an useful HIL system, this time step must be met using real-time emulation so these 23 ns of simulation must be calculated just in 23 ns of the real-time. As it will be seen later, experimental results show that this time step can be reached using the aforementioned FPGA. The model has been simulated in open loop using a fixed duty cycle, as a regulator would minimize the error of the converter model.

Fig. 3 shows the results of the output voltage for the *real* model which uses 64-bit floating point arithmetic compared to the float32 model. As it can be seen, the resolution problems explained in Section III arises. A higher integration step can be chosen in order to reduce the resolution problems but high switching frequency regulations require small integration steps in order to keep the model accurate.

Fig. 4a shows the results of the proposed parametrizable fixed point compared to the *real* model. As the output voltage is similar in both cases, Fig. 4c shows a detail of the output voltage so both systems can be compared. Fig. 4b shows a detail of the output voltage in order to see the resolution of this model. As it was explained before, the resolution is adapted depending on the maximum representable value.



(a) Output voltage results in open loop



(b) Voltage error

Fig. 3: Comparison of *real*(64-bit float) versus float32

TABLE I: Absolute voltage error comparison

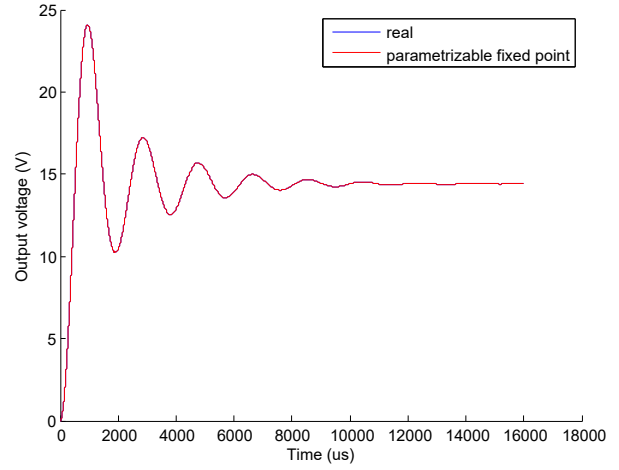
| System | Absolute error | Standard deviation |
|--------------------|--------------------------|--------------------------|
| Float32 | 1.5906 V | $5.901 \cdot 10^{-1}$ V |
| Param. fixed point | $1.2911 \cdot 10^{-4}$ V | $9.0655 \cdot 10^{-5}$ V |

TABLE II: Resources and max. speed of the proposed models

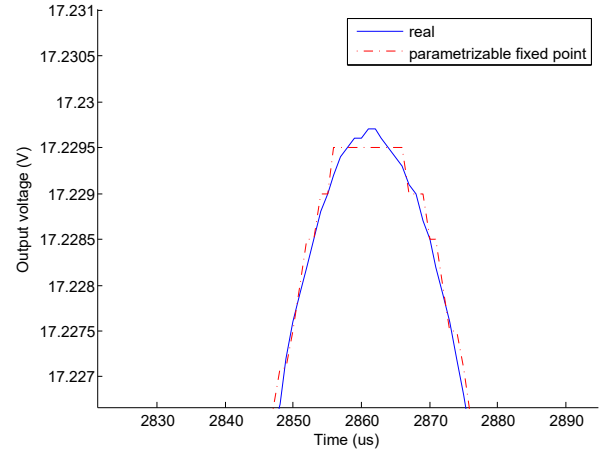
| System | Max. Frequency | LUTS | | Flip flops | | Multipliers | |
|--------------------|----------------|------|-------|------------|-----|-------------|------|
| Float32 | 12.2 MHz | 8987 | 16.9% | 69 | <1% | 4 | 1.8% |
| Param. fixed point | 45.22 MHz | 509 | 1.0% | 106 | <1% | 3 | 1.4% |

Table I shows the absolute error and its standard deviation of the output voltage for both models. Float32 model has an average error of 1.59 V (with voltages around 20 V), so it is not appropriate, whilst the fixed model has an average error of $1.29 \cdot 10^{-4}$ V.

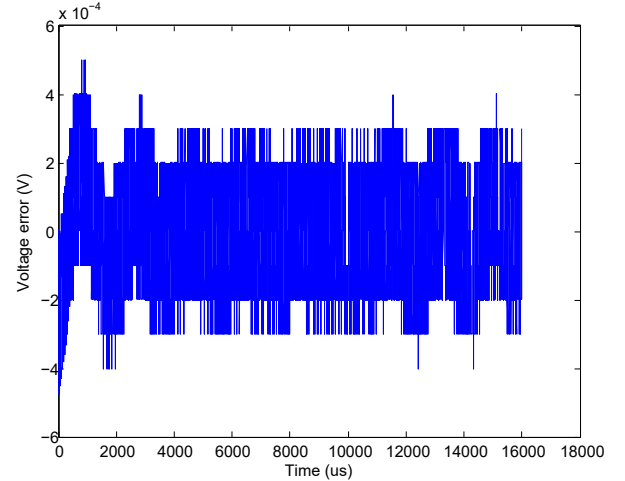
Table II presents the resources used to implement the float32 and the parametrizable fixed point models. *Real* model is not synthesizable so no results can be shown. As it can be seen, the fixed point model uses significantly less resources while



(a) Output voltage results in open loop



(b) Detail zoom in



(c) Voltage error

Fig. 4: Comparison of *real*(float64) versus fixed point models

the area is much higher. The maximum frequency of the model is 45.22 MHz, so the minimum clock period is 22 ns and a time step of 23 ns can be reached. Before it was shown that float32 is not accurate enough because of resolution problems

TABLE III: Simulation/Emulation time to simulate 100 ms of the converter

| System | Simulation/Emulation | Time |
|---------------------------|----------------------|---------------------------------|
| Real | Simulation | 54" 120 ms |
| Float32 | Emulation | 1' 27" 70 ms (non real-time) |
| Param. fixed point | Emulation | 100 ms |

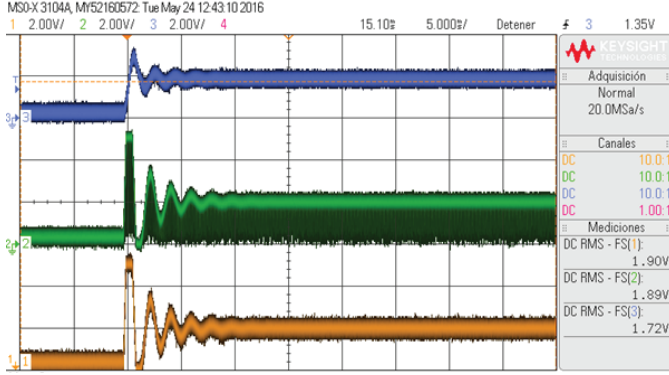


Fig. 5: Integration as an HIL system. Blue (top): output voltage, green (middle): input current, orange (bottom): inductor current

but in this table it can be shown that, moreover, the float32 model cannot achieve real-time with a time step of 23 ns. This is because the frequency of the float32 model is 12.2 MHz, so the minimum achievable time step is 81.9 ns.

Table III shows the time needed to simulate or emulate 100 ms of the full-bridge converter. The *real* model is not synthesizable so it cannot be emulated in actual hardware. As it was explained before, the float32 model simulation cannot be run in real-time, because its FPGA synthesis frequency is lower than 43 MHz (the chosen integration time is 23 ns). However, the parametrizable fixed point proposal can be run in real-time.

The parametrizable fixed point has been integrated as an HIL system using 14-bit high speed DACs (Digital-to-Analog converters), which output the values of the state variables of the model. Fig 5 shows the output voltage, the inductor current and the input current during an emulation. It can be seen that the HIL system is able to represent the dynamic of the plant. The input current presents a noticeable ripple because this current switches between positive and negative signs continuously (between $\pm i_L$), depending on the states of the switches. Negative values are not shown because the DAC is not fed with negative voltages, but a voltage offset can be applied in the output of the DAC in order to see the negative values.

V. CONCLUSIONS

This paper has proposed a hardware-in-the-loop system to emulate a power converter using parametrizable fixed point notation. The proposals found in the literature use floating point and fixed point arithmetics. 32-bit floating point is versatile and allows rapid design, but the resolution may not be

enough for high frequency switching converters. Using more bits, the resolution problem can be avoided, but the results in area and speed may not be affordable. Fixed point notation is more complex but every signal has the optimal number of bits. The main drawback of fixed point is that the simulation cannot exceed the design value limits. The proposed parametrizable fixed point notation achieves the versatility of floating point and the speed and area of traditional fixed point arithmetic. Results show that this proposal achieves almost the same numerical results, whilst the area is kept low and it achieves real-time emulation with a simulation step of 23 ns.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministerio de Economía y Competitividad under project TEC2013-43017-R

REFERENCES

- [1] E. Monmasson, L. Idkhajine, M. Cirstea, I. Bahri, A. Tisan, and M. Naouar, "FPGAs in industrial control applications," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 2, pp. 224–243, May 2011.
- [2] Z. Lukic, N. Rahman, and A. Prodic, "Multibit sigma-delta PWM digital controller IC for DC-DC converters operating at switching frequencies beyond 10 MHz," *Power Electronics, IEEE Transactions on*, vol. 22, no. 5, pp. 1693–1707, Sept. 2007.
- [3] 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.
- [4] 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.
- [5] P. Zumel, M. Garcia-Valderas, A. Lazaro, C. Lopez-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.
- [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] 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.
- [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] 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] 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.
- [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, Feb. 2012.
- [12] E. Adžić, M. Adžić, V. Katić, D. Marčetić, and N. Čelanović, "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] 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.

- [14] A. Sanchez, A. de Castro, and J. Garrido, "A comparison of simulation and hardware-in-the-loop alternatives for digital control of power converters," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 491–500, Aug 2012.
- [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. PP, no. 99, pp. 1371 – 1379, 2013.
- [16] D. Majstorovic, I. Čelanović, N. Teslic, N. Čelanović, and V. Katic, "Ultralow-latency hardware-in-the-loop platform for rapid validation of power electronics designs," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 10, pp. 4708 – 4716, October 2011.
- [17] O. Goni, A. Sanchez, E. Todorovich, and A. de Castro, "Resolution analysis of switching converter models for hardware-in-the-loop," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1162–1170, May 2014.
- [18] 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.