electronics

Article

Comparison of Power Converter Models with Losses for Hardware-in-the-Loop Using Different Numerical Formats

Elyas Zamiri, Alberto Sanchez, Angel de Castro * and Maria Sofia Martínez-García

HCTLab Research Group, Universidad Autónoma de Madrid, 28049 Madrid, Spain; elyas.zamiri@uam.es (E.Z.); alberto.sanchezgonzalez@uam.es (A.S.); sofia.martinez@uam.es (M.S.M.-G.)

* Correspondence: angel.decastro@uam.es; Tel.: +34-914-972-802

Received: 8 October 2019; Accepted: 25 October 2019; Published: 1 November 2019

Abstract: Nowadays, the Hardware-In-the-Loop (HIL) technique is widely used to test different power electronic converters. These real-time simulations require processing large data at high speed, which makes this application very suitable for FPGAs (Field Programmable Gate Array) as they are capable of parallel processing. This paper provides an analytical discussion on three HIL models for a full-bridge converter. The three models use different possible numerical formats, namely float and fixed-point, the latter with and without optimizing the width of signals to the embedded DSP (Digital Signal Processors) blocks of the FPGA. The optimized fixed-point model (OFPM) uses three and two times fewer DSP blocks or LUTs (Look Up Tables), and the maximum achievable clock frequency is also up to 35 % and 25 % higher than the float model and non-optimized fixed-point model (nOFPM), respectively. Furthermore, the models' accuracy is proportional to the clock frequency, thus the OFPM is also the most accurate model. Finally, the paper shows the differences in the simulation when the models include or not losses, proving that not including losses leads to high errors, especially during transients.

Keywords: hardware in the loop; numerical format; field programmable gate array

1. Introduction

Nowadays, it is necessary to find some alternatives to test power electronic converters to reach more advantages over the classical test flow which only includes off-line simulations followed by tests in a real prototype. Off-line simulation is the cheapest and safest possibility to test a power electronic system especially for the initial phase of testing. However, off-line simulation is not enough for guaranteeing a proper behavior of the final hardware implementation so tests with both the real controller and power converter are necessary. However, before testing both the final controller and power converter together, other intermediate steps are possible, which can accelerate the process, and also decrease the risk of managing real power [1] when testing the final controller. In the past, to test a more realistic model of the controller, which is digital in many cases, than just a transfer function or high-level model, some simulation alternatives appeared. For example, mixed-signal simulators [2], a mixture of VHDL (Very high-speed integrated circuit Hardware Description Language) and analog signal extension (VHDL-AMS simulator) [3], or using two different simulators, one for the controller part which usually designs in VHDL and the other one for the analog power converter part [4], were employed to tackle this issue. However, these simulation alternatives were not trivial in many cases, they were usually very slow and, above all, did not meet the requirement of testing the real final controller in hardware.

Recently, it has been possible to emulate controllers and power converters in real-time (RT), which is known as HIL (Hardware-In-the-Loop), regardless of whether the controller is analog or digital.

A comprehensive study on simulation versus HIL alternatives for power converters is accomplished in [5]. In HIL testing, the power converter is replaced by an emulation of it to mimic the real converter and it is made to interact with the real controller to test the controller. HIL model is closer to the actual real system; it saves money, especially in the case of testing expensive systems; allows tests without damaging the real system; and saves a lot of efforts during the implementation of a design [6–20].

Because of the rapid progress of semiconductors, the switching frequency of the power converter is increasing. Precise modeling of switched-power converter needs an integration step at least 100 times less than its switching period. The minimum integration step of the microprocessor-based HIL implementations, which were used traditionally was about hundreds or tens of µs that is not small enough for high-frequency applications [21]. FPGAs (Field Programmable Gate Array) have resulted in a revolution of HIL systems because they make it possible to test a digital model of a mid-high switching frequency power converter in RT (it was nearly impossible by using microprocessors) and they also have excellent parallel processing capabilities and small bus latencies that make them ideal for fast RT simulation [4,22–30]. HDL (Hardware Description Language) models of power converters can be emulated in an FPGA if the model is synthesizable to make it faster in comparison with microprocessor-based HIL implementations [31,32]. Recently, HIL systems which use FPGAs can emulate complex power converters with an integration step about 1 µs or lower without requiring optimization [33]. However, simple and optimized models implemented in FPGA can reach integration steps under 100 ns [31,32,34].

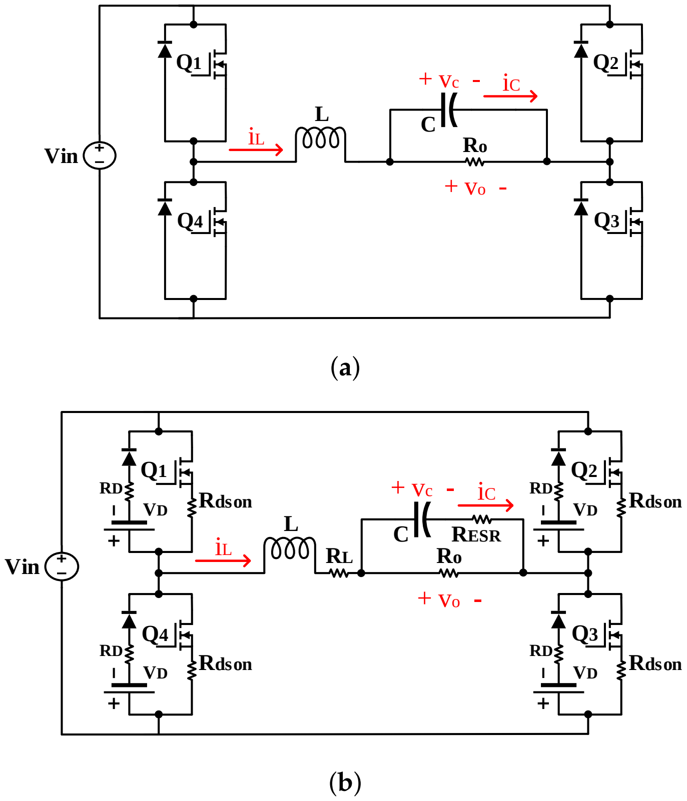
Different numerical formats used in FPGA-based HIL systems play a crucial role in needed hardware resources, the minimum achievable clock frequency, the design time, and the accuracy of the model. A comparison between fixed-point and floating-point representation is done in [5,31], in which FPGA-based HIL systems were proposed. The results confirmed that floating-point representation needs more hardware resources and it is not as fast as fixed-point (up to 10 times more resources and slower) but the effort design is less and the resolution is optimized in different calculations. This is why most HIL applications use floating-point representation [35]. In fixed-point representation, the designer has to define the widths of the signals to provide an optimized model, which is much faster and needs fewer hardware resources [5,31]. It is important to highlight that, even using FPGAs, HIL models remain simple compared with electrical simulators. In many cases, they do not consider any losses because that would make the model slower and may not reach RT. However, it is clear that including some losses would make the model much more accurate.

This paper proposes three different models of the full-bridge converter with and without considering losses based on different possible numerical formats. It will prove that considering losses results in more accurate results even if it makes the models more complex which results in increasing the RT simulation step. The HIL model of the full-bridge is implemented in three different versions: using floating-point representation, using fixed-point representation without taking into account the characteristics of FPGA embedded DSP (Digital Signal Processors) blocks, and optimized fixed-point. The main purpose of this paper is to quantify the differences between these three proposed models and to compare different possibilities of implementation. However, hardware implementation of different numerical formats to solve differential equations presents some issues when they are applied to power electronic converters, which are explained in detail. Furthermore, this paper explores using the idea that, by limiting the width of signals to those of the embedded DSP blocks in FPGAs, the model can achieve smaller simulation step, which results in reaching a more accurate model.

In this paper, initially, a full-bridge converter is presented as an application example with and without considering losses. In Section 2, the equations of the model are calculated by using an explicit Euler approach, and three different possibilities to model a power converter are discussed. In Section 3, the reference model, float model, non-optimized fixed-point model (nOFPM), and OFPM are proposed by using different numerical formats. The implementation of the full-bridge model is also presented and the schematic of the model is discussed in detail. The benefits of the optimized model are confirmed by several experimental and simulation results in Section 4. Finally, conclusions are given in Section 5.

2. Application Example

In this paper, an HIL system based on an FPGA is presented. The application example is a full-bridge converter, which is shown in Figure 1, although the idea can be adapted to other topologies. A full-bridge converter plays a crucial role in industry and it can act as a dc-dc converter to regulate the output voltage or as a multilevel inverter to create an ac output voltage. To have a more accurate model, considering the losses of elements such as MOSFETs, inductors, and capacitors is very important. In Figure 1, two different models are included with and without losses to clarify the differences between both models.



(**a**)

(**b**)

Figure 1. Full-bridge topology with and without losses. (**a**) Ideal full-bridge converter; (**b**) Non-ideal full-bridge converter.

2.1. Model of the Plant

The model of the ideal full-bridge converter is shown in Figure 1a, which is the simplest possible model of a full-bridge. It consists of four power electronic unidirectional switches along with their anti-parallel diodes. The switches of each leg have complementary states and cannot be on simultaneously. The input dc source has been denoted by V_{in} and the output voltage of the full-bridge can get values between $-V_{in}$ and $+V_{in}$ depending on the switching pattern. As mentioned above, several elements of the full-bridge converter are not ideal. Parasitic resistances and electrical losses of the electrical components are considered in Figure 1b to propose a more accurate model.

In Figure 1b, R_D, R_{dson}, R_L, and R_{ESR} are the series resistance of the diode, MOSFET, inductor, and capacitor, respectively. Besides, V_D is the forward voltage of the diode, which is used in the voltage loss due to diode D. The inductor (L) and the capacitor (C) are the LC filter of the output side of the converter and their values are related to the switching frequency of MOSFETs and switching pattern, which are discussed below.

2.2. Equations

To obtain the model of a full-bridge converter, it must be described in equations. The main idea of this section is to extract the related equations that are appropriate in FPGA with a fixed discrete time step. The control signals of switches Q_1–Q_4, which are inputs of the model, play a vital role in the operation of the full-bridge. For instance, the equations of the model when Q_1 and Q_3 are closed are different from those when Q_1 and Q_2 are closed. To simplify the equations, three different situations are introduced, which are shown in Table 1. The current path losses (v_{L-loss}), which affect the calculus of v_L in each of the situations, are calculated based on different switching possibilities. These situations are categorized based on the number of MOSFETs and diodes in the current path. In Situation I, v_{L-loss} is obtained by calculating $(2R_{dson} + R_L)i_L$ and it shows that in this situation two MOSFETs (Q_1 and Q_3 or Q_2 and Q_4) and the inductor are in the current path. In Situation II, all switches are open, and two diodes are conducting. Finally, components in the current path in Situation III are one diode and one MOSFET. Table 1 describes all converter configurations considering different switching possibilities and it is obvious that, in the ideal model (Figure 1a), v_{L-loss} is considered zero. Apart from these losses, which are included in Equation (4), there are conduction losses in the output capacitor resistance (v_{O-loss}), proportional to the capacitor current (i_C). The value of v_{O-loss} is independent of the different situations, which is calculated in Table 1, and it affects the output voltage.

Table 1. Different situations and conducting losses.

	Situation I	**Situation II**	**Situation III**
ON Switches	Q_1andQ_3 or Q_2andQ_4	All OFF	1 MOSFET and 1 diode
$\boldsymbol{v_{L-loss}}$	$(2R_{dson} + R_L)i_L$	$2V_D sign(i_L) + (2R_D + R_L)i_L$	$V_D sign(i_L) + (R_{dson} + R_D + R_L)i_L$
$\boldsymbol{v_{o-loss}}$	$R_{ESR}\ i_C$	$R_{ESR}\ i_C$	$R_{ESR}\ i_C$

Different models for the full-bridge converter are proposed in this paper which use a fixed time step and can be implemented in HDL. The purpose of this paper is to compare different models of the full-bridge converter with a small-time step for testing the control system of a user. It is important to note that the model must calculate the inductor current (i_L) and output voltage (v_O) in every time step. Thus, the model should calculate the exact incremental values of state variables (capacitor voltage and inductor current). The incremental values of capacitor voltage (Δv_C) and inductor current (Δi_L) in discrete time can be formulated as follows:

$$\Delta v_C = \frac{\Delta t}{L} \cdot i_C \tag{1}$$

$$\Delta i_L = \frac{\Delta t}{C} \cdot v_L \tag{2}$$

where Δt is the time step and i_C and v_L are the capacitor current and the inductor voltage, respectively. The capacitor current can be formulated as shown in Equation (3) where $G_L = \frac{1}{R_O}$ is the conductance of the output load. Besides, the inductor voltage depends on the different switching states, thus is calculated as shown in Equation (4).

$$i_C = i_L - i_R = i_L - \frac{v_O}{R_O} = i_L - G_L v_O \tag{3}$$

$$v_L = \begin{cases} V_{in} - v_O - v_{L-loss}I & Q_1 : ON\ and\ Q_3 : ON \\ -V_{in} - v_O - v_{L-loss}I & Q_2 : ON\ and\ Q_4 : ON \\ -V_{in} - v_O - v_{L-loss}II & All\ switches : OFF\ and\ i_L > 0 \\ V_{in} - v_O - v_{L-loss}II & All\ switches : OFF\ and\ i_L < 0 \\ -v_O - v_{L-loss}III & Only\ Q_1\ or\ Q_3 : ON\ and\ i_L > 0 \\ V_{in} - v_O - v_{L-loss}III & Only\ Q_1\ or\ Q_3 : ON\ and\ i_L < 0 \\ -V_{in} - v_O - v_{L-loss}III & Only\ Q_2\ or\ Q_4 : ON\ and\ i_L > 0 \\ -v_O - v_{L-loss}III & Only\ Q_2\ or\ Q_4 : ON\ and\ i_L < 0 \end{cases} \tag{4}$$

For the sake of accuracy, previous equations consider all losses. To obtain the equations related to the ideal model, V_{L-loss} can be removed from the equation. The equations of the capacitor voltage and inductor current can be defined as in Equations (5) and (6). The output voltage of the model depends on the capacitor voltage and the capacitor current, which is calculated in Equation (7).

$$v_C(k) = v_C(k-1) + \frac{\Delta t}{C} \cdot i_C(k-1) \tag{5}$$

$$i_L(k) = i_L(k-1) + \frac{\Delta t}{L} \cdot v_L(k-1) \tag{6}$$

$$v_O(k) = v_C(k) + R_{ESR} \cdot i_C(k-1) \tag{7}$$

After defining the equations, the model can be implemented using different numeric representations. Several possibilities to model the converter in VHDL are discussed in the next subsection.

2.3. Plant Modeling Possibilities

FPGAs have been chosen for the implementation of the models due to their parallel processing capabilities and the need for emulating in RT. There are three main possibilities to model the power electronic converters in an FPGA regarding the used numerical format as shown below. The selected data type determines the minimum time step achievable by the model, hardware resources, the design effort, the area of the design, etc.

- *Real* type: The converter can be modeled with the signal type which is called double precision real. The real data type is defined in the library called MATH_REAL. It is a 64-bits floating-point numeric type, which is not synthesizable but can reduce the time and complexity of the design. Therefore, real type can be used only for simulation purposes but cannot be implemented in an FPGA. In this paper, the real type model of the full-bridge is considered as the reference model and all other models are compared with this model. It is notable that the numerical error of the real model, which uses 64 bits for all signals, is negligible because of the high resolution of the variables.
- *Floating-point* type: A floating-point type is a numeric type consisting of real numbers represented in *IEEE-754* standard. It takes shorter design time in comparison with fixed-point models because the equations of the plant can be translated directly into VHDL without considering range or precision issues. Unlike the real type, the model in float type can be implemented in hardware but it takes more hardware resources to store than the fixed-point model, as confirmed in Section 4. It is also slower than the fixed-point model which has an impact on the accuracy of the model. In this paper, when referring to floating-point, single-precision (32-bits) notation is always used because the double-precision (64-bits) version would lead to much more hardware resources and decreased speed.
- *Fixed-point* type: The logic circuits using floating-point hardware are more complicated than fixed-point hardware, which means that the fixed-point representation uses smaller size and achieves smaller simulation step compared with floating-point representation. The drawback is

that it needs more design effort to determine the optimum signal width considering the fractional and integer part of every variable of the model. It is notable that the converter is modeled only once, thus, the longer design time is not a big problem. In fixed-point notation, the designer determines the number of bits of every variable and they are fixed when the model is implemented. When the cost is an important consideration, especially in complex systems, fixed-point hardware can result in significant savings.

The next step is selecting a method to implement the full-bridge converter in an FPGA. In the following section, an optimized fixed-point model is described that can be useful in implementing the converter in an FPGA because of the lower simulation step and lower hardware resources.

3. Implementation

The implementation of the full-bridge is explained in this section. Three different implementations have been developed using real type, float type, and fixed point numerical format. The simplest approach is using the real type, which allows translating the equations directly into VHDL. It can be implemented by using Equations (5)–(7). However, as mentioned above, the real numerical format is not synthesizable, and it cannot be used in a real-time FPGA implementation, thus it is used only as a reference model for simulation. Real numerical format uses *IEEE-754* double precision standard (64-bits) with a mantisa field of 53 bits. Using this numerical format, numeric resolution problems are avoided because of the high number of signal bits, thus it is the best choice as the reference model to have a comparison between different approaches. In this paper, the real numerical format with a time step of 1 ns is used as the reference model. The schematic of the Full-bridge model without considering the variable width is shown in Figure 2. This schematic is the result of translating Equations (5)–(7) into VHDL and using real type.

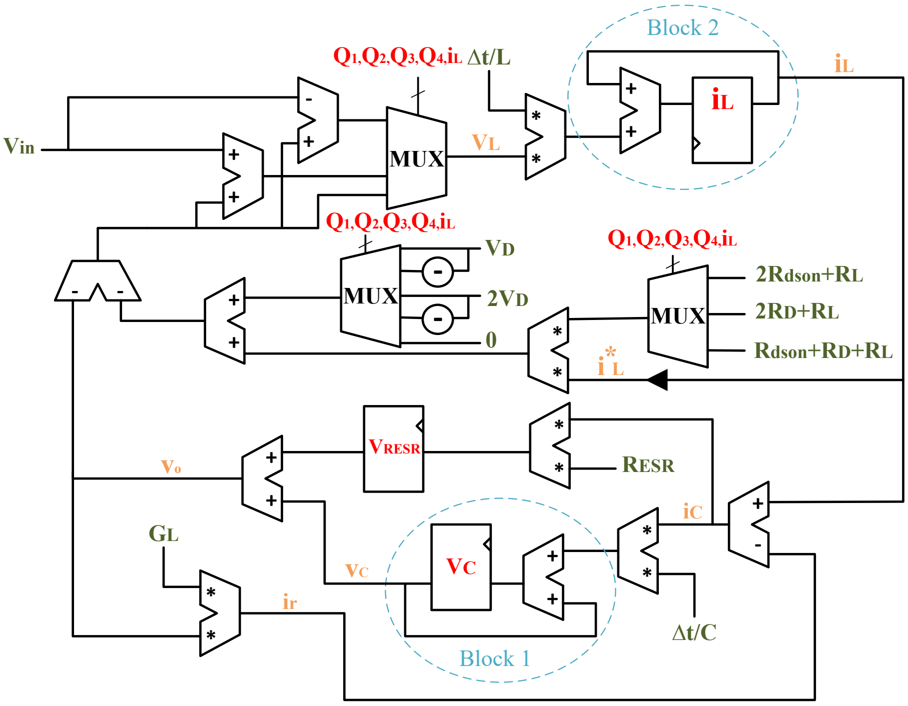


Figure 2. Model schematic.

The model has three outputs, v_C, i_L, and v_o, and some inputs: V_{in}, G_L, switches states, the forward voltage of the diode, the resistance of components, and the values of the output filter (L and C). Thus, different modulation strategies with different components and any load condition can be modeled. Blocks 1 and 2 are the accumulators of the state variables which calculate v_C and i_L, respectively.

The model uses multiplexers with five select lines, which are the switches control signals (Q_1, Q_2, Q_3, and Q_4) and the sign of the inductor current. The final output of these multiplexers is V_L, which is calculated using Equation (4). It is notable that two more multiplications are used to calculate the incremental values of the state variables in the design. In addition, the outputs of Blocks 1 and 2 are the feedback for the next integration step and they will be added to the incremental values in order to obtain the next values of state variables. Finally, to calculate the output voltage, register VRESR is used to consider the conduction losses of the output capacitor, as mentioned in Equation (7).

Resolution problems cannot be ignored if single-precision IEEE-754 (float 32) is used because of the smaller number of fractional bits compared with the reference model, *real*, which uses 64 bits. For example, if v_{out} signal were in the range of 200 V, the resolution in float 32 would be around 1.53×10^{-5} V (mantissa of 24 bits) while it would be around 2.84×10^{-14} V (mantissa of 53 bits) in 64 bits real type. It is notable that Δv_C and Δi_L can be around μV and μA when dt (time step) is very small (around tens of ns). Therefore, float 32 may not have enough resolution for some converters using a small dt. However, float 32 models are easily designed, basically the same as real type models, but can be synthesized in an FPGA. Those are the main advantages of float 32 models.

The last possibility for the implementation of the model in an FPGA is fixed-point representation. In this paper, two different fixed-point models are presented, with and without optimizing the model to the hardware resources of the FPGA such as the embedded multipliers in Family-7 FPGAs. The non-optimized fixed-point model, which is called nOFPM in this paper, can be implemented directly by Equations (5)–(7), but the design time is higher than in the floating-point model, which is the main disadvantage of fixed-point. The number of bits in this model is high enough and there are no resolution problems, as discussed below. All signals in this model except constants and the only independent input, v_{in}, use 40 bits in total and the number of integer bits is calculated based on the maximum expected value of those signals. For example, the signals of i_L and v_C state variables have six and nine integer bits while the number of the fractional bits are 33 and 30 bits, respectively. To improve the result of area, speed, and accuracy of the fixed-point model, an optimized fixed-point model based on the number of bits of the embedded DSP blocks of the FPGA is proposed.

QX.Y representation is used for the proposed fixed-point models. In this format, X and Y are the numbers of the integer and fractional bits, respectively. The number of bits in this format is $X + Y + 1$, including the sign bit (most significant bit), thus a Q8.2 signal has 11 bits. The decimal value of the QX.Y signal can be calculated by multiplying by 2^{-Y}. The X and Y values of all signals should be calculated by the designer. The important signals' widths, format, and the resolution of the Full-bridge fixed-point model are shown in Table 2. In this paper, a comparison between different representations including an optimized fixed-point model is done to find a trade-off among the resolution, area, and maximum clock frequency.

Table 2. Signed QX.Y signal formats of the fixed-point model.

	Number of bits		**Format**		**Resolution**		
Signal	**nOFPM**	**OFPM**	**nOFPM**	**OFPM**	**nOFPM**	**OFPM**	**Unit**
v_{in}	21	11	Q8.12	Q8.2	2^{-12}	2^{-2}	V
i_r	40	30	Q6.33	Q6.23	2^{-33}	2^{-23}	A
v_C/v_o	40	40	Q9.30	Q9.30	2^{-30}	2^{-30}	V
i_L	40	40	Q6.33	Q6.33	2^{-33}	2^{-33}	A
i_L^*	40	25	Q6.33	Q6.18	2^{-33}	2^{-18}	A
v_L	40	25	Q6.33	Q9.15	2^{-33}	2^{-15}	V
i_C	40	25	Q6.33	Q6.18	2^{-33}	2^{-18}	A
$\frac{\Delta t}{L}$	27	18	Q-14.41	Q-15.32	2^{-41}	2^{-32}	$\frac{s}{H}$
$\frac{\Delta t}{C}$	27	18	Q-11.38	Q-12.29	2^{-38}	2^{-29}	$\frac{s}{F}$

DSP blocks are integrated into most modern FPGA devices in order to improve the speed and efficiency of computations. The hardware multipliers in the Family-7 Xilinx series FPGA, DSP48E1 slice, are improved from 18 × 18 in the Family-6 series to 18 × 25 in the Family-7 series [36,37]. Thus, the input signals width of the multiplier in OFPM is truncated to 18 and 25 bits to minimize the number of DSP blocks in the optimized model and maximize speed. The minimization of the number of the multipliers can affect the maximum clock frequency because of shortening the critical path in the model. In fact, the main idea of this paper is choosing the optimized signal width including the fractional and integer bits to increase the clock frequency as much as possible. The increase in maximum achievable clock frequency can improve the accuracy, however, the reduction of the signal width is not negligible. The optimized model uses more bits for the integrators to calculate the state variables, while the feedback signals' widths are changed to meet the multiplier limitations. In OFPM, fewer bits are used for feedback signals because they do not need high resolution. To minimize the number of DSPs, it is necessary to change the signal width of i_L by defining the signal i_L^*, which has 25 bits. Furthermore, in the schematic of the model, there are some constants, such as G_L, $\frac{\Delta t}{L}$, $\frac{\Delta t}{C}$, and V_D, which are the inputs of the multipliers, as can be seen in Figure 2. In OFPM, these constants are represented with 18 bits to use the minimum number of multipliers. However, in nOFPM, there is no limitation and more bits are considered for the mentioned constants. It is notable that the pipelining technique is not used in the proposed models because it would modify Equations (5)–(7). The output of both state variables depend on the previous values, thus inserting pipeline registers is not allowed. It would be equivalent to using (k-2) or even previous values instead of (k-1).

4. Results

As explained in Section 2, this paper presents two different models of the full-bridge converter with and without losses intended to be implemented in FPGAs. As can be seen in Figure 3, the model without losses produces noticeably different results, especially during the transient. The error of the model without losses is calculated in Table 3, which is categorized into two different parts (transient and steady-state error). The steady-state zone is the interval in which the state variable of the model without losses is in the ±2% band of the final value. It is obvious that the error of the model without losses cannot be neglected especially in the case of transient error. Thus, it is necessary to include different losses to the model as in Figure 1b to have a more accurate model.

Table 3. Percentage error of the model without losses.

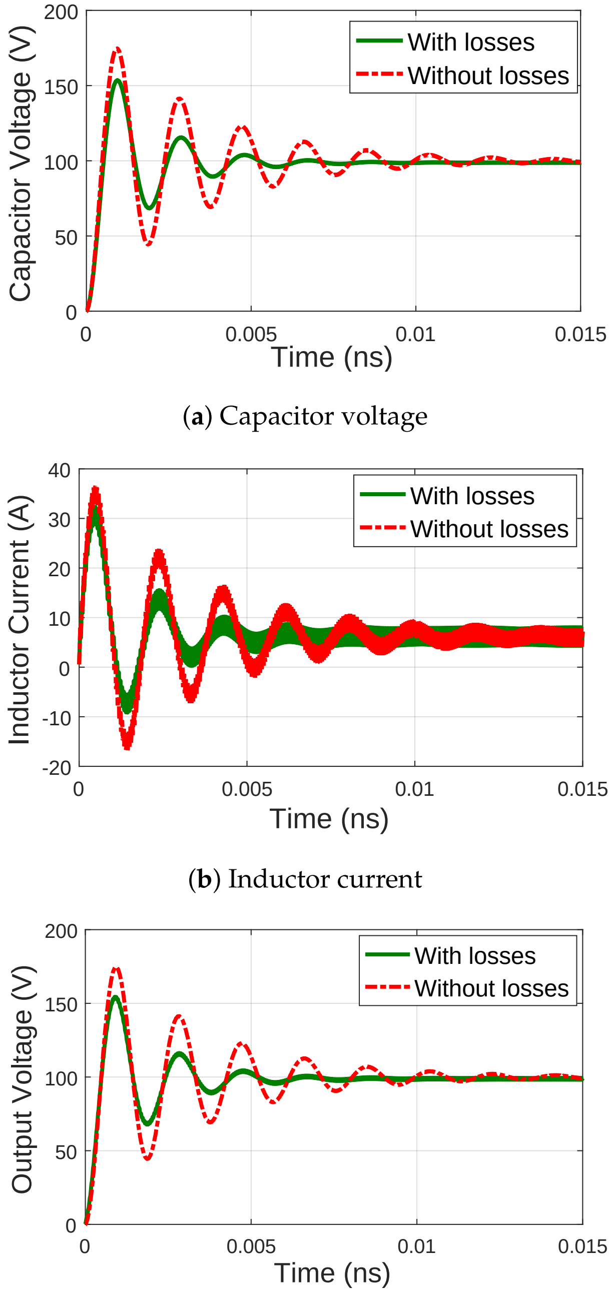
	Transient	Steady-state
Capacitor Voltage Error (%)	8.4933	1.2862
Inductor Current Error (%)	38.2581	1.9710
Output Voltage Error (%)	8.2440	1.2943

Once the importance of losses in the model is clear, the next question is which is the most appropriate numeric representation system. A thorough comparison is done between the reference model (real model) with losses and three other models with and without losses: floating-point 32-bits, non-optimized fixed-point, and optimized fixed-point. The accuracy of the reference model is previously confirmed by comparing its outputs with the same model in MATLAB/Simulink and the theoretical equations. The differences between the reference model and the MATLAB/Simulink model are shown in Table 4. All the errors shown in this paper are MAE (Mean Absolute Error). The values in this table show that the results of the VHDL reference model match the simulation results in MATLAB.

$$v_O = (2D-1)\cdot V_{in} \tag{8}$$

$$v_{O,loss} = \frac{(2D-1)}{1+(2R_{dson}+R_L)G_L}\cdot V_{in} \tag{9}$$

The theoretical value of the output voltage without considering losses and with losses is calculated as Equations (8) and (9), respectively. The ripple and the mean value of the state variables based on the real model (reference model) are shown in Table 5, which are compatible with the theory results in Table 6 that shows the physical parameters of the implemented full-bridge converter. In all tests, the input voltage is a fixed 200 V dc voltage source and the switching frequency (f_{sw}) is set at 20 kHz, while the duty cycle is 0.75. As can be seen, a resistive load has been chosen for the output load and the switching period (T_{sw}) of the full-bridge model is 50 µs.



(**a**) Capacitor voltage

(**b**) Inductor current

(**c**) Output voltage

Figure 3. Comparison of the models simulations with and without losses.

In the following, all comparisons are done based on the reference model, which uses floating-point of double precision. This is to ensure that the only error sources are the numerical representation or the simulation time step, as all the other aspects are equal in the reference model and the rest of the compared models.

Table 4. Percentage difference between the reference model with losses and the MATLAB/Simulink model.

	V_C	i_L	v_{out}
Transient difference (%)	0.0258	0.1289	0.0259
Steady-State difference (%)	0.0011	0.0363	0.0011

Table 5. Ripple and mean values of the signals.

	Ripple		**Mean value**	
	Without Losses	**With Losses**	**Without Losses**	**With Losses**
Capacitor voltage (V)	0.251	0.243	99.9965	98.7356
Inductor current (A)	3.715	3.510	6.2381	6.1707
Output voltage (V)	0.251	1.277	99.9965	98.7355

Table 6. Physical parameters of the full-bridge model.

	$R_{dson}(\Omega)$	$R_L(\Omega)$	$G_L(\Omega^{-1})$	$R_{ESR}(\Omega)$	$R_D(\Omega)$	$V_D(V)$	$V_O(V)$
Without losses	0	0	0.0625	0	0	0	100.000
With losses	0.1	0.005	0.0625	0.36	0.8	0.7	98.735

The proposed models were tested in open loop, without using any closed-loop regulators. This is important to compare the accuracy of the different models since a closed-loop regulation would lead to very similar results, masking small model inaccuracies [5]. The control signals of the model were implemented with a simple DPWM (Digital Pulse Width Modulator). These inputs to the model, which define the switches states, are used for choosing the appropriate equations, as shown in Equation (4). It is notable that, although in this example PWM signals are used for the control, the model actually reads the instantaneous values of the switches control signals, which are the inputs of the model, so any modulation can be used, without requiring constant frequency or any other limitation. The evaluation of the proposed systems is done by instantiating the different models, monitoring capacitor voltage, inductor current, and output voltage, and comparing those values with the ones of the reference model. As mentioned above, the reference model in VHDL is based on variables of real type and a simulation step of 1 ns.

Four different tests were done to show the numerical errors related to the different numerical formats. The first test focused on the error of different numerical formats with a simulation step of 1 ns. It is obvious that it cannot be reached in RT but it can be very useful to show the resolution problem in different approaches. The second, third, and fourth tests were carried out with simulation steps of 16, 20, and 24 ns, respectively. These simulation steps were chosen because they correspond to the limits of RT emulation when using optimized fixed-point, non-optimized fixed-point, and floating-point, respectively, as shown below.

Figure 4 shows the relation between the transient and the steady-state error of the capacitor voltage versus the clock period. As can be seen, the numerical error is nearly linear if the clock period is equal or greater than 16 ns. This situation can be seen in Figure 4, where the accuracy of the models should be proportional to the simulation step, which means that the error of the model with a lower clock period should be smaller. This is the expected result because, as the simulation step is reduced, Equations (5) and (6) are more accurate. It is obvious that there is an anomaly in some of the models for a simulation step of 1 ns, but it is due to resolution issues in the numerical format.

The error of different numerical formats is very small but it can be analyzed in Tables 7–9 regarding different integration steps. As can be seen in Table 7, which has the same information as Figure 4, the error is very similar between the different models when they use the same simulation step (T_{clk}) if it is 16 ns or higher. This is because for those simulation steps the numerical resolution of all the models is high enough. However, when using $T_{clk} = 1ns$, the different models have quite different errors. That is caused by the insufficient resolution of some of the models, especially 32-bit floating-point and optimized fixed-point. The reason is that the increments in Equations (5) and (6), which are proportional to $\Delta t = T_{clk}$, become so small that numerical issues appear. However, a simulation step of 1 ns is not achievable for RT emulation. This is just to show that numerical resolution issues may appear for high switching frequency converters (small simulation steps) depending on the application, and that the simulation step cannot be decreased indefinitely without also increasing the number of bits.

The other main conclusion of Tables 7–9 is that, when numerical issues are not present, the error is mainly proportional to the simulation step. Thus, the design rule is to decrease the simulation step as much as the model allows. The minimum simulation step that can be reached in RT greatly depends on the complexity of the model, which determines the minimum achievable clock period for RT execution of each model.

Table 7. Percentage error of the capacitor voltage (the values of error in RT are highlighted).

	Float		**nOFPM**		**OFPM**	
$T_{clk}\,(ns)$	**Transient**	**Steady-state**	**Transient**	**Steady-state**	**Transient**	**Steady-state**
24	3.8538×10^{-3}	1.8121×10^{-4}	3.8461×10^{-3}	9.8411×10^{-5}	3.9712×10^{-3}	1.2883×10^{-4}
20	3.2838×10^{-3}	1.1166×10^{-4}	3.1765×10^{-3}	8.2517×10^{-5}	3.2263×10^{-3}	1.1342×10^{-4}
16	2.4927×10^{-3}	1.6808×10^{-4}	2.5065×10^{-3}	6.6892×10^{-5}	2.6252×10^{-3}	9.6293×10^{-5}
1	3.5829×10^{-2}	1.5891×10^{-2}	9.0971×10^{-5}	8.3662×10^{-5}	1.9425×10^{-4}	2.9091×10^{-5}

Table 8. Percentage error of the inductor current (the values of error in RT are highlighted).

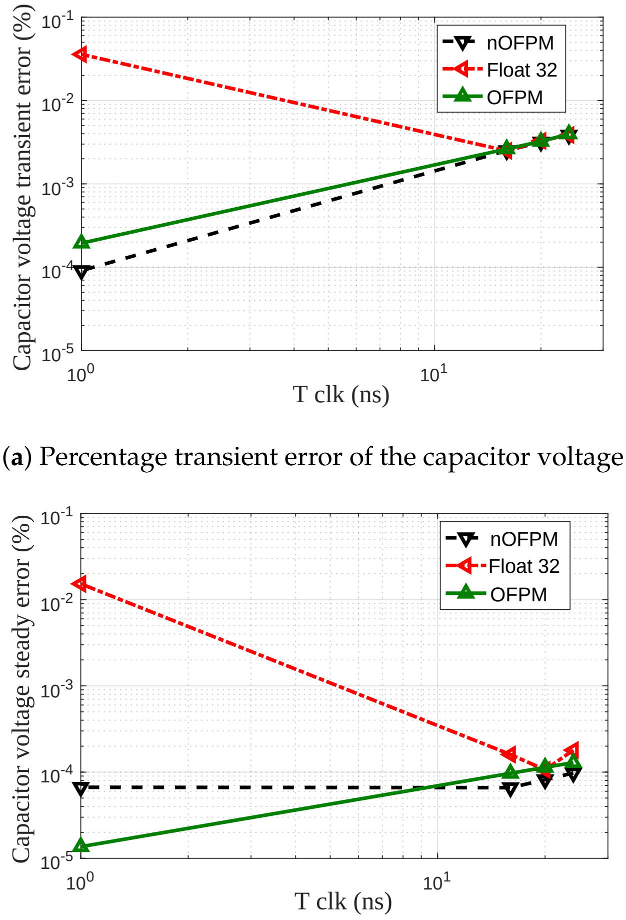
	Float		**nOFPM**		**OFPM**	
$T_{clk}\,(ns)$	**Transient**	**Steady-state**	**Transient**	**Steady-state**	**Transient**	**Steady-state**
24	1.6279×10^{-2}	1.0107×10^{-3}	1.6278×10^{-2}	5.6759×10^{-4}	1.6725×10^{-2}	6.2125×10^{-4}
20	1.3860×10^{-2}	7.9861×10^{-4}	1.3443×10^{-2}	4.7458×10^{-4}	1.3798×10^{-2}	5.3714×10^{-4}
16	1.0475×10^{-2}	8.5086×10^{-4}	1.0606×10^{-2}	3.8393×10^{-4}	1.1206×10^{-2}	4.4490×10^{-4}
1	1.5092×10^{-1}	4.4609×10^{-2}	5.3126×10^{-4}	6.8676×10^{-4}	1.0264×10^{-3}	7.6533×10^{-4}

Table 9. Percentage error of the output voltage (the values of error in RT are highlighted).

	Float		**nOFPM**		**OFPM**	
$T_{clk}\,(ns)$	**Transient**	**Steady-state**	**Transient**	**Steady-state**	**Transient**	**Steady-state**
24	3.9179×10^{-3}	1.7985×10^{-4}	3.9128×10^{-3}	9.7579×10^{-5}	4.0350×10^{-3}	1.2881×10^{-4}
20	3.3360×10^{-3}	1.0625×10^{-4}	3.2317×10^{-3}	8.1659×10^{-5}	3.2777×10^{-3}	1.1367×10^{-4}
16	2.5371×10^{-3}	1.6001×10^{-4}	2.5502×10^{-3}	6.5965×10^{-5}	2.6614×10^{-3}	9.6861×10^{-5}
1	3.5101×10^{-2}	1.5261×10^{-2}	7.4757×10^{-5}	6.6688×10^{-5}	1.8305×10^{-4}	1.3657×10^{-5}

Table 10 shows the synthesis results of the emulation systems after implementation in an xc7a35ticsg324-1L FPGA, which is a low-cost FPGA. The table presents the results in area and speed with and without losses. Three different synthesis results are provided including floating-point model, nOFPM, and OFPM. Furthermore, the three models were synthesized enabling and disabling the use of DSP blocks, to show the impact of these blocks on the rest of the necessary area (especially Look Up Tables (LUTs)) and necessary clock period. All previous models were hand-coded for optimum synthesis results. Besides, an automatic-translated model from MATLAB code to HDL using *Fixed-Point Designer/HDL Coder* by MATLAB is shown in this table and is discussed below.

It can be seen that in all models with and without losses, the fixed point models require much fewer hardware resources than the float model, even 3 and 2.5 times fewer DSP blocks or LUTs and the minimum possible clock period is also up to 35% and 28% smaller in the models with and without losses, respectively. The main reason is that floating-point adders and multipliers are much more complex than fixed-point ones. It is notable that in these models, which are a direct translation of Equations (5) to (7), the FPGA clock period is equivalent to the simulation step. Therefore, fixed-point models can work in real-time using smaller simulation steps, which is the best way of minimizing model errors as seen before. This is also crucial for RT emulation of middle-high switching frequency converters. Tables 7–9 highlight the error of each model when using their best achievable simulation step in each case: 16 ns for OFPM, 20 ns for nOFPM, and 24 ns for float 32.



(**a**) Percentage transient error of the capacitor voltage

(**b**) Percentage steady-state error of the capacitor voltage

Figure 4. Percentage error of the capacitor voltage, depending on the simulation step logarithmic scale.

Regarding both fixed-point models (nOFPM and OFPM), it can be seen that the OFPM implementation area is quite smaller than nOFPM (it needs fewer FPGA resources), and its maximum clock frequency is about 25% higher. It was expected, as OFPM uses fewer bits in general, and, furthermore, its widths are chosen for fitting exactly in one DSP block each one (multipliers 18 × 25 bits).

In Table 10, the synthesis results include versions without using the DSP blocks to clarify the impact of these blocks both in area and speed. In fact, most of the logic resources are dedicated to the multipliers, which are implemented in the DSP blocks. To have a fair comparison, it can be seen that the models with losses and without DSP blocks use 752 LUTs for OFPM, 1546 for nOFPM, and 2332 for floating-point. The same results are obtained for the models without losses as they use 472, 754, and 1182 LUTs for OFPM, nOFPM, and the float model, respectively. Removing DSP blocks is not a good approach because it can increase the minimum achievable clock frequency, as can be seen in Table 10 for the OFPM. The minimum clock period ($T_{clk,min}$), which is equal to the execution time needed by the FPGA to calculate the integration equations, is the most important parameter for comparing different notations because not only a small simulation step is necessary to emulate high-frequency converters but it also affects the error, as discussed above.

Table 10 also includes results of the automatic translation from MATLAB to HDL code using fixed-point. To have a fair comparison, this translation uses the same data widths of OFPM. Its synthesis results are clearly worse than the hand-coded OFPM with area sometimes approaching floating-point results and with time results even worse than hand-coded floating-point. Therefore, for optimum synthesis results, hand-coding is highly recommended.

The bar chart in Figure 5 illustrates the numbers in Table 10 to highlight the area and clock period differences between all models. As a conclusion, area results of the three models are quite different, which has a direct impact in the final price of the HIL systems. The number of LUTs when not using DSP blocks is about three times more in floating-point than in OFPM, and a similar or even higher proportion in the number of DSP blocks when they are enabled. It can also be seen that the minimum achievable clock period reduces if the OFPM is used, but in less proportion than area. Taking into account that fixed-point requires more design effort than floating-point, time results may not compensate the extra design effort depending on the application, but, when area is the main concern, fixed-point is highly recommended.

Table 10. FPGA resources used by the design and the synthesis results with and without losses (WL/WoL).

	LUTs		**Flip Flops**		**DSPs**		$T_{clk,min}(ns)$	
	WL	**WoL**	**WL**	**WoL**	**WL**	**WoL**	**WL**	**WoL**
Floating-Point	907	351	82	64	17	9	21.283	17.807
Non-Optimized Fixed-Point	671	391	101	80	11	9	19.754	17.412
Optimized Fixed-Point	628	412	96	77	5	1	15.784	13.932
Floating-Point (no DSPs)	2332	1182	93	63	0	0	21.050	16.489
Non-Optimized Fixed-Point (no DSPs)	1546	754	100	62	0	0	17.810	15.646
Optimized Fixed-Point (no DSPs)	752	472	94	77	0	0	17.644	13.023
MATLAB HDL translation *	782	305	121	80	9	4	22.342	17.205
MATLAB HDL translation (No DSPs) *	2303	1182	120	80	0	0	22.807	20.981

* Using fixed-point with the same data width as OFPM.

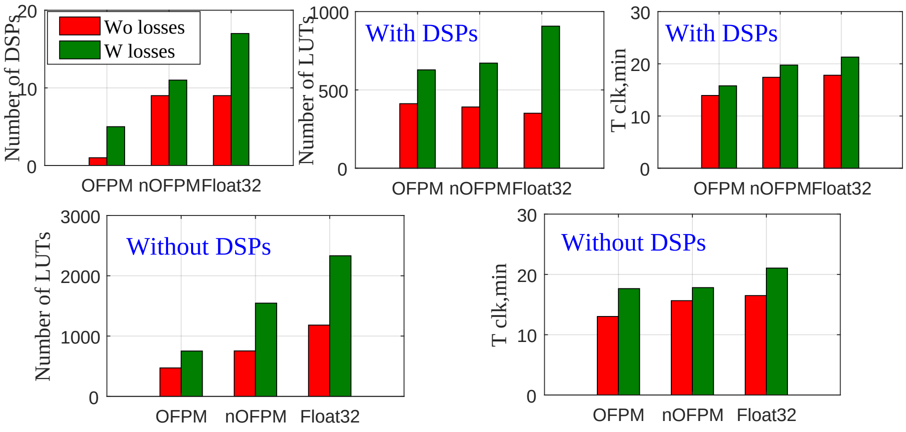


Figure 5. Synthesis results of different models.

5. Conclusions

In this paper, three different HIL models of the full-bridge converter based on different possible numerical formats are proposed. The float model, nOFPM, and OFPM achieve simulation steps of 21.283, 19.754, and 15.784 ns in RT, respectively. The main purpose of this paper is to demonstrate the differences between the different proposed HIL models and compare the used area based on different numerical formats. OFPM is based on the idea of limiting the width of signals to that of the embedded DSP blocks in FPGAs in order to save hardware resources. The comparison has shown that the hardware resources of this model are one-half and one-third of the nOFPM and the float model, respectively. A comparison of all proposed models based on the number of LUTs without using the DSP blocks was accomplished, and the results clarified the differences of hardware resources as they used 2332, 1546, and 752 LUTs for the float model, nOFPM, and OFPM, respectively. OFPM was tested using a simulation step of 16 ns (62.5 MHz) with negligible numeric errors (2.6614×10^{-3} and 9.6861×10^{-5} for transient and steady-state, respectively). This noticeably small simulation step also allows modeling high switching-frequency converters, while the simulation step limits the applications that can be modeled precisely. The calculated mean absolute errors in RT have proved that the OFPM is the most accurate model among the three different proposed models in both transient and steady-states while it needs more design effort than the float model. As can be seen, the decision about which numerical format should be used is not trivial and it should be taken considering the application. As a starting point, the model based on float 32 type could be the first implementation choice in terms of design effort. If the necessary resources must be decreased or the simulation step should be lower, the fixed-point notation is the solution. However, in this case, an optimized version taking into account the FPGA resources, such as the width of the DSP blocks, should be used because it has an important impact on both necessary resources and minimum simulation step.

Author Contributions: Conceptualization, E.Z., A.S. and A.d.C.; methodology, E.Z. and A.d.C.; software, E.Z. and A.S.; validation, E.Z., A.S., A.d.C. and M.S.M.-G.; formal analysis, E.Z. and A.d.C.; investigation, E.Z. and A.S.; resources, E.Z. and A.d.C.; writing–original draft preparation, E.Z.; writing–review and editing, E.Z., A.S., A.d.C. and M.S.M.-G.; visualization, E.Z. and M.S.M.-G.; supervision, A.d.C.

Funding: This research was funded by Spanish Ministerio de Economía y Competitividad grant number TEC2013-43017-R.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vijay, A.S.; Doolla, S.; Chandorkar, M.C. Real-Time Testing Approaches for Microgrids. *IEEE J. Emerg. Sel. Top. Power Electron.* **2017**, *5*, 1356–1376. [CrossRef]
2. Ghosh, S.; Giambiasi, N. Modeling and simulation of mixed-signal electronic designs—Enabling analog and discrete subsystems to be represented uniformly within a single framework. *IEEE Circuits Devices Mag.* **2006**, *22*, 47–52. [CrossRef]
3. Pecheux, F.; Lallement, C.; Vachoux, A. VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2005**, *24*, 204–225. [CrossRef]
4. Lucia, O.; Urriza, I.; Barragan, L.A.; Navarro, D.; Jimenez, O.; Burdio, J.M. Real-Time FPGA-Based Hardware-in-the-Loop Simulation Test Bench Applied to Multiple-Output Power Converters. *IEEE Trans. Ind. Appl.* **2011**, *47*, 853–860. [CrossRef]
5. Sanchez, A.; de Castro, A.; Garrido, J. A Comparison of Simulation and Hardware-in-the- Loop Alternatives for Digital Control of Power Converters. *IEEE Trans. Ind. Inform.* **2012**, *8*, 491–500. [CrossRef]
6. Jia, J.; Yang, G.; Nielsen, A.H.; Roenne-Hansen, P. Hardware-in-the-loop tests on distance protection considering VSC fault-ride-through control strategies. *J. Eng.* **2018**, *2018*, 824–829. [CrossRef]
7. Prieto-Araujo, E.; Olivella-Rosell, P.; Cheah-Mañe, M.; Villafafila-Robles, R.; Gomis-Bellmunt, O. Renewable energy emulation concepts for microgrids. *Renew. Sustain. Energy Rev.* **2015**, *50*, 325–345. [CrossRef]
8. Jandaghi, B.; Dinavahi, V. Real-Time FEM Computation of Nonlinear Magnetodynamics of Moving Structures on FPGA for HIL Emulation. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7709–7718. [CrossRef]
9. Jandaghi, B.; Dinavahi, V. Real-Time HIL Emulation of Faulted Electric Machines Based on Nonlinear MEC Model. *IEEE Trans. Energy Convers.* **2019**. [CrossRef]
10. Liang, T.; Dinavahi, V. Real-Time System-on-Chip Emulation of Electrothermal Models for Power Electronic Devices via Hammerstein Configuration. *IEEE J. Emerg. Sel. Top. Power Electron.* **2018**, *6*, 203–218. [CrossRef]
11. Karimi, S.; Poure, P.; Saadate, S. An HIL-Based Reconfigurable Platform for Design, Implementation, and Verification of Electrical System Digital Controllers. *IEEE Trans. Ind. Electron.* **2010**, *57*, 1226–1236. [CrossRef]

12. Liu, C.; Ma, R.; Bai, H.; Li, Z.; Gechter, F.; Gao, F. FPGA-Based Real-Time Simulation of High-Power Electronic System With Nonlinear IGBT Characteristics. *IEEE J. Emerg. Sel. Top. Power Electron.* **2019**, *7*, 41–51. [CrossRef]
13. Montano, F.; Ould-Bachir, T.; David, J.P. An Evaluation of a High-Level Synthesis Approach to the FPGA-Based Submicrosecond Real-Time Simulation of Power Converters. *IEEE Trans. Ind. Electron.* **2018**, *65*, 636–644. [CrossRef]
14. Sanchez, A.; Todorovich, E.; de Castro, A. Exploring the Limits of Floating-Point Resolution for Hardware-In-the-Loop Implemented with FPGAs. *Electronics* **2018**, *7*, 219. [CrossRef]
15. Yushkova, M.; Sanchez, A.; de Castro, A.; Martínez-García, M.S. A Comparison of Filtering Approaches Using Low-Speed DACs for Hardware-in-the-Loop Implemented in FPGAs. *Electronics* **2019**, *8*, 1116. [CrossRef]
16. Lee, J.; Kang, D.; Lee, J. A Study on the Improved Capacitor Voltage Balancing Method for Modular Multilevel Converter Based on Hardware-In-the-Loop Simulation. *Electronics* **2019**, *8*, 1070. [CrossRef]
17. Saito, K.; Akagi, H. A Power Hardware-in-the-Loop (P-HIL) Test Bench Using Two Modular Multilevel DSCC Converters for a Synchronous Motor Drive. *IEEE Trans. Ind. Appl.* **2018**, *54*, 4563–4573. [CrossRef]
18. Fernández-Álvarez, A.; Portela-García, M.; García-Valderas, M.; López, J.; Sanz, M. HW/SW Co-Simulation System for Enhancing Hardware-in-the-Loop of Power Converter Digital Controllers. *IEEE J. Emerg. Sel. Top. Power Electron.* **2017**, *5*, 1779–1786. [CrossRef]
19. Amin, M.; Aziz, G.A.A.; Durkin, J.; Mohammed, O.A. A Hardware-in-the-Loop Realization of Speed Sensorless Control of PMa-SynRM with Steady-State and Transient Performances Enhancement. *IEEE Trans. Ind. Appl.* **2019**, *55*, 5331–5342. [CrossRef]
20. Tian, J.; Liu, J.; Shu, J.; Tang, J.; Yang, J. Engineering modelling of wind turbine applied in real-time simulation with hardware-in-loop and optimising control. *IET Power Electron.* **2018**, *11*, 2490–2498. [CrossRef]
21. Faruque, M.O.; Strasser, T.; Lauss, G.; Jalili-Marandi, V.; Forsyth, P.; Dufour, C.; Dinavahi, V.; Monti, A.; Kotsampopoulos, P.; Martinez, J.A.; et al. Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis. *IEEE Power Energy Technol. Syst. J.* **2015**, *2*, 63–73. [CrossRef]
22. Matar, M.; Paradis, D.; Iravani, R. Real-time simulation of modular multilevel converters for controller hardware-in-the-loop testing. *IET Power Electron.* **2016**, *9*, 42–50. [CrossRef]
23. Deter, M.; Ha, Q.; Plöger, M.; Puschmann, F. FPGA-Based Real-Time Simulation of a DC/DC Converter. *Atzelektronik Worldw.* **2014**, *9*, 32–35. [CrossRef]
24. Rosa, A.; Silva, M.; Campos, M.; Santana, R.; Rodrigues, W.; Morais, L. SHIL and DHIL Simulations of Nonlinear Control Methods Applied for Power Converters Using Embedded Systems. *Electronics* **2018**, *7*, 241. [CrossRef]
25. Lin, N.; Shi, B.; Dinavahi, V. Non-linear behavioural modelling of device-level transients for complex power electronic converter circuit hardware realisation on FPGA. *IET Power Electron.* **2018**, *11*, 1566–1574. [CrossRef]
26. Bai, H.; Liu, C.; Zhuo, S.; Ma, R.; Paire, D.; Gao, F. FPGA-Based Device-Level Electro-Thermal Modeling of Floating Interleaved Boost Converter for Fuel Cell Hardware-in-the-Loop Applications. *IEEE Trans. Ind. Appl.* **2019**. [CrossRef]
27. Herrera, L.; Li, C.; Yao, X.; Wang, J. FPGA-Based Detailed Real-Time Simulation of Power Converters and Electric Machines for EV HIL Applications. *IEEE Trans. Ind. Appl.* **2015**, *51*, 1702–1712. [CrossRef]
28. Barragán, L.A.; Urriza, I.; Navarro, D.; Artigas, J.I.; Acero, J.; Burdio, J.M. Comparing simulation alternatives of FPGA-based controllers for switching converters. In Proceedings of the 2007 IEEE International Symposium on Industrial Electronics, Vigo, Spain, 4–7 June 2007; pp. 419–424.
29. Ruelland, R.; Gateau, G.; Meynard, T.A.; Hapiot, J.C. Design of FPGA-based emulator for series multicell converters using co-simulation tools. *IEEE Trans. Power Electron.* **2003**, *18*, 455–463. [CrossRef]
30. Ibarra, L.; Rosales, A.; Ponce, P.; Molina, A.; Ayyanar, R. Overview of Real-Time Simulation as a Supporting Effort to Smart-Grid Attainment. *Energies* **2017**, *10*, 817. [CrossRef]
31. Sanchez, A.; de Castro, A.; Garrido, J. Parametrizable fixed-point arithmetic for HIL with small simulation steps. *IEEE J. Emerg. Sel. Top. Power Electron.* **2018**. [CrossRef]
32. Goni, O.; Sanchez, A.; Todorovich, E.; de Castro, A. Resolution Analysis of Switching Converter Models for Hardware-in-the-Loop. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1162–1170. [CrossRef]
33. Vekić, M.S.; Grabić, S.U.; Majstorović, D.P.; Čelanović, I.L.; Čelanović, N.L.; Katić, V.A. Ultralow Latency HIL Platform for Rapid Development of Complex Power Electronics Systems. *IEEE Trans. Power Electron.* **2012**, *27*, 4436–4444. [CrossRef]

34. Sanchez, A.; Todorovich, E.; de Castro, A. Impact of the hardened floating-point cores on HIL technology. *Electr. Power Syst. Res.* **2018**, *165*, 53–59. [CrossRef]
35. Majstorovic, D.; Celanovic, I.; Teslic, N.D.; Celanovic, N.; Katic, V.A. Ultralow-Latency Hardware-in-the-Loop Platform for Rapid Validation of Power Electronics Designs. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4708–4716. [CrossRef]
36. Ago, Y.; Nakano, K.; Ito, Y. A Classification Processor for a Support Vector Machine with Embedded DSP Slices and Block RAMs in the FPGA. In Proceedings of the 2013 IEEE 7th International Symposium on Embedded Multicore Socs, Tokyo, Japan, 26–28 September 2013; pp. 91–96.
37. 7 Series DSP48E1 Slice User Guide. Available online: https://www.xilinx.com/support/documentation/user_guides/ug479_7Series_DSP48E1.pdf (accessed on 27 March 2018).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).