



Universidad Autónoma
de Madrid

Biblos-e Archivo
Repositorio Institucional UAM

Repositorio Institucional de la Universidad Autónoma de Madrid

<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:
This is an **author produced version** of a paper published in:

IEEE Transactions on Power Electronics 38.5 (2023): 6024-6035

DOI: <https://doi.org/10.1109/TPEL.2023.3243702>

Copyright: © IEEE 2023

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

Oversampling techniques to improve the accuracy of Hardware-in-The-Loop switching models

Marina Yushkova, *Student Member, IEEE*, Alberto Sanchez, and Angel de Castro, *Senior Member, IEEE*

Abstract—Real-time Hardware-in-The-Loop implementations for power converters involve digital sampling of the on-off signals that are applied to the switches. The most basic way of considering it, is to process only one switch state during a simulation step. This usually causes an incorrect input reading because it can change during the step. The negative effects include the appearance of undesirable subharmonics and numerical inaccuracy. In this work, input signal oversampling is proposed. Thus, several switch states can be properly processed within the same simulation step. The question is how to use this additional information. This paper proposes two oversampling approaches: a sequential one, which is based on previous works, and a parallel implementation, which is the main contribution of the paper. The results obtained with these techniques are compared between them and with the model without oversampling, showing a significant increase in accuracy for both oversampling approaches around 98%. Although the accuracy is very similar in both methods, synthesis results for FPGA implementation are clearly better for the parallel method, which can be easily adapted to process multiple switching events. The final experimental results show a great subharmonic decrease up to 99%.

Index Terms—real-time simulation, oversampling, power electronics, Hardware-in-the-Loop, FPGA, switching converter.

I. INTRODUCTION

Recently Hardware-in-The-Loop (HIL) has become a popular technique to verify real-time (RT) systems without using physical plants, including the simulation of switching power supplies [1]–[3]. The growing complexity of these systems requires improving the performance of HIL in terms of the ability to easily reconfigure systems and to use higher simulation frequencies, among other challenges [4], [5]. Therefore, the new generation of HIL simulators includes Field Programmable Gate Arrays (FPGAs) that offer superior performance compared to microprocessors and allow simulation with kilohertz and megahertz switching frequencies both for ad-hoc [6]–[8] and for commercial systems [9], [10].

The main goal for FPGA designers in HIL is to make the model as similar as possible to the real system [11]. One of the typical problems is choosing an appropriate numerical method and integration step for the simulation, which was addressed in [12], [13]. Another important challenge to solve comes from the asynchronism between the PWM (Pulse-Width Modulation) controller signal and the HIL system. The information about the digital input can be sampled only with the frequency of the simulation, while in real conditions the commutation usually occurs during a time step. This inaccuracy in the identification and proceeding of the switching event

can cause various negative effects in the output (undesirable output oscillations, numerical errors, etc.), proportional to the simulation step [14], [15]. The most intuitive way to solve this problem is reducing the simulation step, what can be realized successfully in simple power plant models or for offline simulations. However, for the complex RT topologies such a reduction of the time step is not applicable and the time step cannot be reduced more than some minimum value. This value is normally much higher than a reasonable step to decrease the input reading error, especially for high switching frequency converters. For example, commercial HIL systems usually use an integration step of hundreds of nanoseconds [16]. Therefore, when the reduction of the time step is not feasible or not sufficient, another technique should be added.

A. Previous works

Oversampling of the input signal has become a widespread solution, which consists in obtaining the additional information of the PWM signal during the simulation step, and using the extra information for further computation. Acquired data can be processed in various ways, which are present in research and industry. For instance, to smooth the undesirable oscillation in the output, averaging techniques can be applied to averaged converter models. Averaging implies using the oversampled ratio of the digital input to filter it, when a switching event happens during the simulation step. The precision of the filtering will be proportional to the sampling frequency. Variations of this technique were applied to averaged models of power converters in [17]–[20]. For example, in [17] time averaging method is applied to increase the accuracy of voltage source converter models. The results show that the method efficiently removes artificial low-harmonics caused by inaccurate input reading and can handle multiple switching events during one simulation step. The authors state that the possible drawbacks of the method are attenuation of high-frequency switching harmonics (so, a precision decrease) and computational delays in RT (what is unavoidable if any technique is applied). In general, the idea of the averaging methods is similar to approaches presented later in the paper, but it is applied to the averaged models, while the new proposal is designed for switching models.

Another way to use the oversampling information is to correct the output values after a missed switching event is detected. For switched models, several correction algorithms to account a single commutation event are explained in [21]. These algorithms are based on post-interpolation with variable or fixed step-size. Later in [22] the same authors explain in detail that the mentioned techniques violate real-time constraints

The authors are with the HCTLab Research Group, Universidad Autonoma de Madrid, Madrid 28049, Spain (e-mail: marina.yushkova@uam.es; alberto.sanchezgonzalez@uam.es; angel.decastro@uam.es).

because of various reasons including altering the original time grid of the simulation, stepping back in time, varying step size, etc. Therefore, the algorithms can only be applied to offline simulations. Authors then propose a new approach that consists of fixed step-size and clock synchronization (FICS) in order to fit in the online simulations [22]. The proposed approach linearly interpolates the output value during the next time step after the switching event not altering the original time grid, since the calculation is finished until the next step. Also authors introduce the RT simulation and the experimental implementation of their approach and conclude its feasibility for real-time applications. An analysis of computational burden and gained accuracy of the FICS approach is not provided. Nevertheless, it can be assumed that its time overhead is significant due to the necessity of increasing the minimum simulation step to execute the interpolation and resynchronize the new value with the input. In [23] various post-correction algorithms, which apply oversampling of multiple switching events in switched models, are presented. The authors propose using numerous combinations of interpolation and extrapolation techniques with fixed and variable time-step for the post-correction of the output. The algorithms proposed in [23] show significant improvement in accuracy for offline simulations, but the additional computational complexity and if they would be feasible in the presence of real-time constraints remains unclear. In industry, Typhoon HIL uses oversampling techniques for high switching frequency power electronics applications [16], [24]. Authors introduce the Global GDS (Gate Drive Signal) Oversampling, which consists in sampling the input signal, identifying the instant when the change occurs, calculating state variables and using the oversampled information to compensate the next calculated value, while the current value is not visible at the output until the compensation is done. In [24] theoretical description of the Global GDS oversampling feature is shown. The plots with received waveforms are demonstrated in [16] showing a noticeable decrease of the undesirable oscillation due to the input error. However, Typhoon HIL does not provide details on mathematical nor FPGA implementation of the algorithm in public access.

B. Novelty and organization of the paper

This paper aims to develop an oversampling algorithm which would be suitable and efficient for FPGA implementation in switching models. Firstly, a sequential approach similar to the software implementations available in the literature, is synthesized in hardware analysing its performance. The sequential algorithm is concluded to be extensively expensive for hardware implementations, especially in the case of multiple switching events. Therefore, a new parallel approach is proposed after the error analysis, since the simulation results prove its mathematical efficiency. Both techniques consist in receiving the oversampled ratio of on-off events which happened during the previous simulation step. However, the sequential approach calculates the state variables taking into consideration the order of on-off times and their ratio, while the parallel one uses only the ratio. To the best of author's

knowledge, the parallel algorithm has never been proposed before.

The paper shows evidences of superiority of the proposed parallel design above the sequential one for hardware implementations. Two topologies — asynchronous buck converter (allows one switching event per simulation step) and synchronous buck converter (allows two switching events) are used for obtaining the simulation and synthesis results, and making conclusions regarding the most suitable oversampling approach. Although two different configuration of a buck converter have been used in this paper, the proposed approaches can be directly translated to any other topology. Later experimental validation is made with the basic asynchronous converter. Re-simulation of the converter model behaviour with the experimental parameters prove the efficiency of the parallel approach for removing low-frequency harmonics in real HIL platforms.

For adapting the software concepts of oversampling for hardware, the implemented sequential and parallel approaches do not output a wrong value and then post-correct it, like it was done in previous works. Instead, they calculate the state variables with a constant time delay of one simulation step and update the output already with the correct value. However, since the simulation step of the model is usually much smaller than the switching period, this synchronization delay is affordable. For example, in this research the working simulation steps are up to a few hundreds of *ns*. In exchange, it allows to simulate the models with a smaller integration step.

The rest of the paper is organized as follows. The theoretical description of the problem caused by the lack of resolution during input reading is discussed in Section II. In Section III the oversampling approaches for the used buck converter model are presented. The precision results of the MATLAB simulation and the hardware synthesis results from Vivado are presented in Section IV. The experimental implementation of the selected parallel approach into an FPGA is made in Section V. Finally, the conclusion is given in Section VI.

II. BACKGROUND

Hardware-in-the-loop simulators for switched-mode power supplies should detect the moments of the switch transitions as precisely as possible. The lack of resolution on the input reading provokes significant error in the simulation output. The PWM signal is one of the main ways to control a power supply. Therefore, it is used in this study as an example, though the proposed approaches can be applied to any gate signal. In the general case, i.e. when the low and high times of the PWM signal are not perfect multiples of the sampling frequency, inaccurate PWM reading produces higher error than the other common sources of error such as calculation imprecision, limited resolution (number of bits), etc. [25], independently of the converter topology. In other words, if the duty cycle is read ineffectively, choosing a more accurate numerical method or reducing the simulation step may not improve the accuracy, because the PWM sampling error will mask the accuracy of the method.

As an example to demonstrate the influence of the lack of resolution in the PWM sampling, Fig. 1 shows the simulation

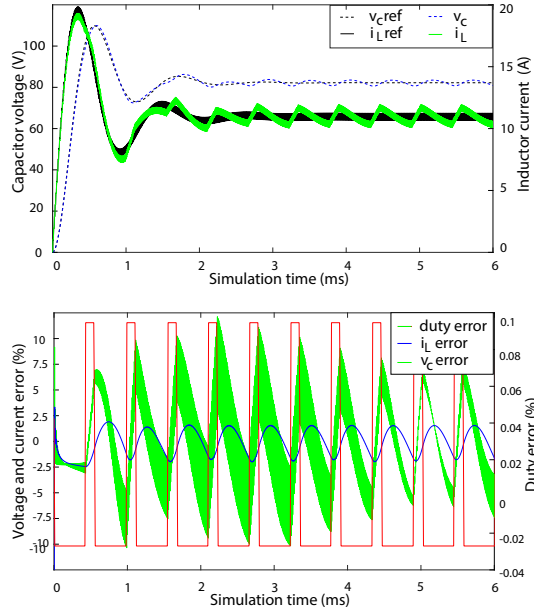


Figure 1. Influence of a wrong PWM duty cycle reading in a buck converter model ($dt = 500$ ns)

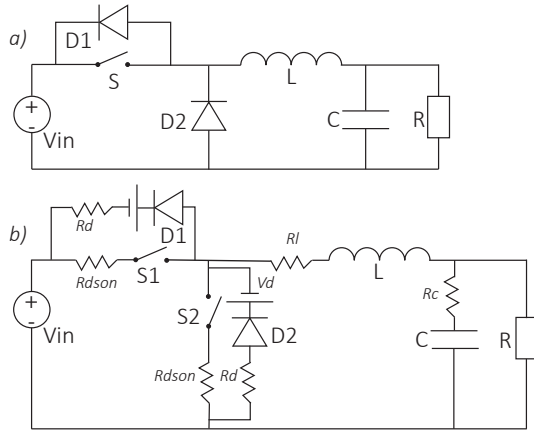


Figure 2. Converter models used in the study: a) Asynchronous buck converter without losses; b) Synchronous buck converter with losses

of a transient of 6 ms of an asynchronous buck converter (Fig. 2.a)). The parameters of the converter are provided in the first row of Table I. The simulation is made with a switching frequency $f_{sw} \approx 99.7118$ kHz, and a fixed duty cycle of 41.11% which are intentionally not divisors of the PWM sampling steps in order to perform a realistic simulation. The capacitor voltage and inductor current and their errors dependent on the duty cycle error are presented with a simulation step $dt = 500$ ns. The simulation step is representative for the application under this research — for instance, the high-end model Typhoon HIL 606 reaches 200 ns. At the bottom of the plot, the red lines are the PWM sampling errors, and the blue and green lines are the state variable errors. The plot shows the errors per unit in relation to their average values for both state variables and sampled duty cycle. It is evident that the error in the state variables follows the same periodic pattern of the duty cycle reading error. For

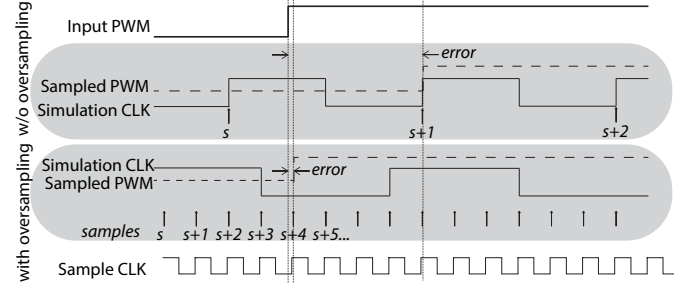


Figure 3. Difference of the sampling error with and without applying the oversampling techniques in a HIL simulation with a single switching event

instance, for the inductor current the quantitative error exceeds 10% with the duty reading error of around 0.09% only. As can be seen, the error in PWM sampling leads to subharmonics in the state variables, which are not present in a real converter. This effect distorts the simulation results and often prevents the simulation to be useful. Therefore, improving the PWM sampling is key in HIL simulations.

Oversampling the PWM is an effective way to decrease the PWM reading error. When the input oversampling is not applied, the switching input signal is sampled at the same frequency as the simulation step of the model. It turns out that this step can be not sufficiently small to detect a PWM input change exactly at the moment when it occurs, and the detection error will be proportional to the step size (half step on average). To avoid this problem, the oversampling approach proposes to sample and store the input PWM signal with a higher frequency than the simulation one, and execute the calculation with respect to the received data. In Fig. 3 the difference in the reading accuracy with and without multiple sampling is graphically shown when a single switching event occurs inside a simulation step. At the top of the figure oversampling is not applied, so the PWM input is read in the rising edge of the simulation clock. As it can be seen, in this case the detection of the PWM change occurs much later than it actually happens and the reading error is high. At the bottom of the figure since oversampling is applied, i.e. more than one sample is taken during a simulation step, the PWM change is detected much faster what significantly decreases the sampling error. The sampling error will be proportional to the chosen sampling step, not to the relatively big simulation step. The practical improvement of the simulation due to the application of oversampling will be shown in Section V.

The possibility of multiple input sampling implies a considerable improvement of the final simulation results, but along with that, it requires additional time and area. While using an oversampling technique, the states of the switching signal have to be collected at several points inside the simulation step. Moreover, the calculation with the accumulated information of the PWM-change cannot take place immediately, but it is used during the next time-step. So, time and area usage is increased due to the overall increment of the model complexity and also additional latency. Therefore, it is important to achieve a reasonable trade-off between overused resources and obtained accuracy improvement. Feasible approaches for oversampling

Table I
BUCK CONVERTERS' PARAMETERS USED IN THE STUDY

	V_{in}	V_{out}	C	L	R	V_d	R_L	R_d	R_{dson}	R_c
<i>async</i>	200 V	100 V	35 μ F	850 μ H	7.5 Ω	0 V	0 Ω	0 Ω	0 Ω	0 Ω
<i>sync</i>	200 V	100 V	35 μ F	850 μ H	7.5 Ω	0.6 V	0.2 Ω	0.05 Ω	0.01 Ω	0.02 Ω

will be proposed in Section III.

III. POSSIBLE OVERSAMPLING APPROACHES

In this section, the significance of the issue and possible oversampling techniques will be discussed. For the sake of simplicity, the mathematical explanation of the oversampling approaches is provided using the formulas of the simple model of asynchronous buck converter without losses in Section III-A. This converter has a single switch, which therefore allows only one switching event per simulation step. However, since the proposed techniques can also be applied to more complex topologies with multiple switching events, the graphical illustration of the oversampling in a synchronous buck with losses with a discussion is given in Section III-B. Furthermore, simulation and synthesis results will be provided for both converter models in Section IV.

A. Oversampling with a single switching event

The functionality of the asynchronous buck converter without losses (see Fig. 2.a) and *async* row in Table I) can be described with the following system of equations:

$$\begin{aligned} \frac{di_L}{dt} &= \frac{v_L}{L} \\ \frac{dv_C}{dt} &= \frac{i_C}{C} \end{aligned} \quad (1)$$

where i_L is the inductor current and v_C is the capacitor voltage — interdependent state variables of the model. The calculation mode is varying depending on possible states of the system, such as closed switch, open switch with positive i_L , open switch with slightly negative or zero i_L :

$$\begin{aligned} \left. \begin{aligned} i_C &= i_L - \frac{v_C}{R} \\ v_L &= v_{in} - v_C \end{aligned} \right\} && \text{closed S} \\ \left. \begin{aligned} i_C &= i_L - \frac{v_C}{R} \\ v_L &= -v_C \end{aligned} \right\} && \text{open S, } i_L > 0 \\ \left. \begin{aligned} i_C &= i_L - \frac{v_C}{R} \\ v_L &= v_{in} - v_C \end{aligned} \right\} && \text{open S, } i_L < 0 \\ \left. \begin{aligned} i_C &= i_L - \frac{v_C}{R} \\ v_L &= 0 \end{aligned} \right\} && \text{open S, } i_L = 0 \end{aligned} \quad (2)$$

where i_C is the current through the capacitor, v_L is the voltage across the inductor, and v_{in} is the input voltage. Since the way of the calculation is changing depending on the state of the switch, the commutation moment is critical.

The system of equations (1) which describes the converter is a first-order differential system that is to be solved with a numerical method. Nowadays, one of the most common numerical methods in both industry and academia is 1st order Forward Euler method (later *FE*). Applying this method to (1), the computation formulas may be written as:

$$\begin{aligned} i_{L_{n+1}} &= i_{L_n} + v_{L_n} \frac{dt}{L} \\ v_{C_{n+1}} &= v_{C_n} + i_{C_n} \frac{dt}{C} \end{aligned} \quad (3)$$

As it can be seen from (3), *FE* method implies approximation of the next step taking into consideration the previous value of the function. It is one of the most common explicit numerical methods for the discretization of continuous systems and that is why it is often used in commercial real-time models. Hence, this method is taken in this work as a basis to expose the effects of applying different oversampling techniques in the sense of the gained accuracy improvement and hardware (HW) overuse. However, the same oversampling techniques can be applied when different numerical methods are used. As long as Forward Euler method is using the previous point to calculate the next one, the computation is made once during a time-step and it is based only on the value of the switch at the event of rising (or falling) clock's edge. Nevertheless, as it was explained in the previous section, in reality a commutation of the switch can happen inside the time-step, what leads to an erroneous calculation. Therefore, it is desirable to obtain intermediate values of the switch state (i.e. oversample) and take them into account during the computation.

To improve the accuracy of the simulation not changing the applied numerical method, two oversampling approaches which use the stored switch information are proposed in this paper. Let us assume that an event of a switch commutation happens during the step dt in the moment t_e , as it is shown in Fig. 4. The application of the first sequential approach *ApSeq* can be described by the following system of equations:

$$\begin{aligned} i'_{L_1} &= i_{L_0} + v_{L_0} \frac{dt_e - t_0}{L} \\ v'_{C_1} &= v_{C_0} + i_{C_0} \frac{dt_e - t_0}{C} \\ i_{L_1} &= i'_{L_1} + v'_{L_1} \frac{dt_1 - t_e}{L} \\ v_{C_1} &= v'_{C_1} + i'_{C_1} \frac{dt_1 - t_e}{C} \end{aligned} \quad (4)$$

In this approach, the information of the on-off times and their occurrence order is accumulated during the simulation step. The computation is made in two steps calculating transitional values of the state variables i'_{L_1} and v'_{C_1} and then using these values to calculate i_{C_1} and v'_{L_1} and the final results i_{L_1} and v_{C_1} in the second step. It is important to notice,

that the execution of the second step is not possible until the first step values are obtained, since the order of the on-off occurrence is taken into consideration. Due to this extra delay in *ApSeq*, the total latency is presumed to increase up to two times approximately in comparison with the plain Forward Euler implementation. However, the same hardware is used twice for the calculation of both steps. Thus, the hardware utilization is not expected to increase significantly, because the main arithmetic blocks are reused.

Another possibility of the hardware implementation for the sequential approach is not reusing hardware and calculating the final result in one step in order to decrease the total calculation time:

$$\begin{aligned} i_{L_1} &= i_{L_0} + \frac{v_{L_0} dt_{e-t_0} + v'_{L_1} dt_{t_1-t_e}}{L} \\ v_{C_1} &= v_{C_0} + \frac{i_{C_0} dt_{e-t_0} + i'_{C_1} dt_{t_1-t_e}}{C} \end{aligned} \quad (5)$$

Mathematically, equations (5) are equal to (4), so their expected accuracy is the same, but their FPGA implementation is different. The values of i'_{C_1} and v'_{L_1} are calculated as in (4), but instead of waiting for them to be executed, they are initially inserted in a single equation and are solved at once. Like that, the synthesis tool can estimate the critical path at the beginning and optimize the calculus. Apart from that, there is no need of a register to store the middle point, what also adds flexibility to the tool. However, the expected improvement in time is smaller than the increase in HW due to the application of longer formulas, which are not reused. The names *ApSeq_r* and *ApSeq_{nr}* will be used accordingly for the implementation reusing memory defined in (4) and not reusing it (5).

The second oversampling approach (*ApPar*) implies calculating the equations in parallel independently of the internal slope order:

$$\begin{aligned} i_{L_1} &= i_{L_0} + \frac{v'_{L_0} dt_{e-t_0} + v''_{L_0} dt_{t_1-t_e}}{L} \\ v_{C_1} &= v_{C_0} + \frac{i'_{C_0} dt_{e-t_0} + i''_{C_0} dt_{t_1-t_e}}{C} \end{aligned} \quad (6)$$

Here only the total proportion of the on-off appearance, but not their order, is stored using some registers as for the previous approach. The next value of the function is calculated during the following step based on the proportion of the switch states applying corresponding formulas (2). Since the on-off order is not considered in *ApPar*, the accuracy result of the approach can be decreased in comparison with *ApSeq*, as it is shown in Fig. 4. Nevertheless, for FPGA applications this accuracy decrease is not supposed to give a significant influence due to the use of a very small integration step. Meanwhile, the latency of *ApPar* is expected to be noticeably lower than of *ApSeq*, because the calculation is made in parallel.

To sum up, intuitively expected calculation time, area and accuracy results, respectively, would be as following:

$$\begin{aligned} dt_{ApPar} &< dt_{ApSeq_{nr}} < dt_{ApSeq_r} \\ HW_{ApSeq_r} &< HW_{ApPar} < HW_{ApSeq_{nr}} \\ error_{ApSeq} &< error_{ApPar} \end{aligned} \quad (7)$$

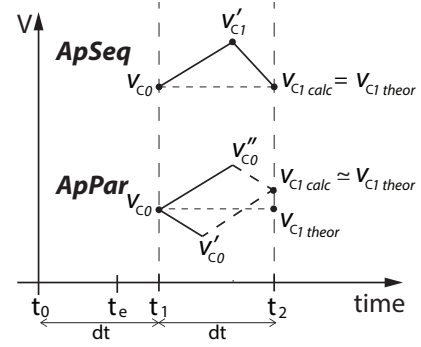


Figure 4. Graphical explanation of oversampling with one switching event

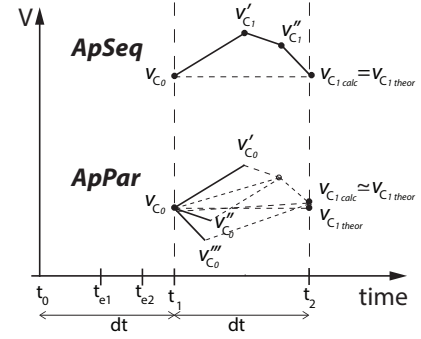


Figure 5. explanation of oversampling with multiple events

The sequential approach reusing HW is supposed to show the maximum timing, but the minimum area usage due to the multicycling. The same approach not reusing HW is supposed to use a bit greater area and an intermediate timing. The accuracy in both sequential cases is the same, because their mathematical formulas are identical as it was said before. The parallel approach is expected to give a bigger error, as it was said before, but to have the smallest area usage and a median timing due to the overall simplicity of its execution. The error and synthesis results from MATLAB and HLS implementations and the best trade-off between the three models (*ApSeq_r*, *ApSeq_{nr}* and *ApPar*) will be given in Section IV.

B. Multiple switching event case

Now let us consider a model which allows various switching events per simulation step using as an example the synchronous buck converter with losses ((Fig. 2.b) and *sync* row in Table I). This converter contains two switches S1 and S2, so functional PWM states include high-side MOSFETs (S1 on, S2 off), low-side MOSFETs (S1 off, S2 on), and deadtime (both S1 and S2 off \rightarrow deadtime). Graphically it can be shown in Fig. 5, where two switching events t_{e1} and t_{e2} happen during the time-step t_1 . In the figure it is seen that the PWM states are in the order *on-deadtime-off*, nevertheless, their line-up can be different.

Same as for the single switching event case from Fig. 4, *ApSeq* approach calculates the provisional values of the state variables in sequence, and *ApPar* simply sums up the vectors multiplied by the proportion of their appearance. Therefore,

analyzing the model with multiple switching events in terms of proposed oversampling techniques, the sequential approach for the single event can be adapted to the multiple events with significant extra time delay. This delay will be proportional to the number of extra switching events in the model. However, the parallel approach will have much smaller time penalties, as all the events are applied simultaneously and just a weighted average is applied afterward. Therefore, in the case of more complex topologies, the implementation of the parallel approach is remarkably more promising. In the next section, it will be proven by the analysis of accuracy, time and area of the proposed models.

IV. SELECTION OF THE OVERSAMPLING APPROACH

The sequential and parallel approaches described in the previous section are analyzed concerning their accuracy and utilization results for both converter models. The accuracy of both sequential options $ApSeq_r$ and $ApSeq_{nr}$ was calculated as one since they are mathematically equivalent as it was mentioned before. In this research, the experiments are carried out using the asynchronous buck converter model (Fig. 2.a), Table I), and the synchronous buck converter with losses (Fig. 2.b), Table I). However, the oversampling approaches can be applied to any switched circuit topology. Instead of using an exact switching frequency of 100 kHz (as Table I shows), arbitrary frequencies around that value have been chosen to show realistic cases, when the synchronization between the controller and the model does not match. All the simulation and synthesis results are obtained using 32-bit floating-point arithmetic defined by the IEEE 754 standard.

Before obtaining the accuracy and synthesis results, a few examples of simulations in open and closed loop are demonstrated. In Fig. 6 an example of the MATLAB simulation in open loop without and with oversampling is shown in comparison with the reference model. For this simulation, the open-loop control has been applied with a constant duty cycle. Therefore, the improvement in oversampling can be easily noticed. For the reference (golden model) the simulation is made with no error in the input sampling. That is, the same equations of the converters are used for the golden model as for the models under estimation, but all its times, such as *on*, *off* and *deadtime*, are intentionally multiples of the simulation step of the reference model which is 1 ns . This situation is unachievable since in real conditions it is impossible to reach a perfect synchronism between the converter and HIL clocks. As it can be noticed, undesirable subharmonics appear when oversampling is not applied, what has a great influence on the final simulation reliability. In the case when oversampling is applied, the unwanted oscillation is basically removed and the result wave is much more similar to the reference one. Therefore, oversampling can be included into the model to smooth the subharmonics and significantly increase the accuracy of the simulation.

Closed-loop simulations were also carried out to check the importance of applying oversampling in this situation. Fig. 7 shows the inductor current and capacitor voltage when the asynchronous buck converter is controlled by the following

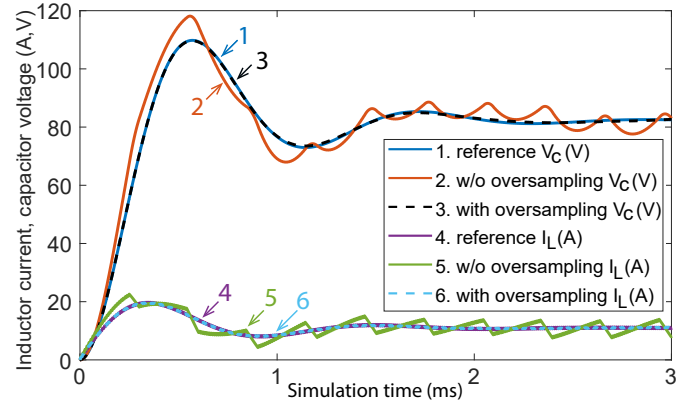


Figure 6. Open-loop example of the capacitor voltage and inductor current simulation in asynchronous buck converter with and without oversampling

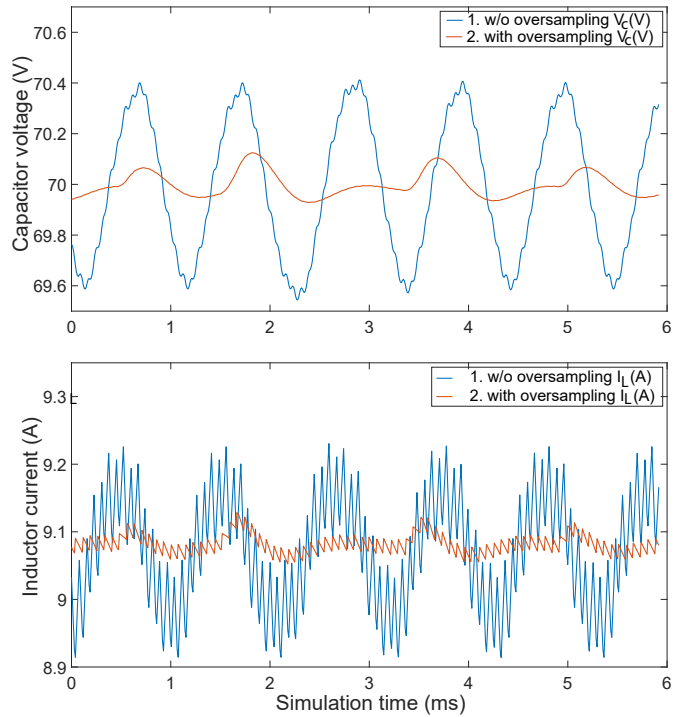


Figure 7. Closed-loop example of the capacitor voltage and inductor current simulation in asynchronous buck converter with and without oversampling

PI (Proportional Integrator) controller ($f_{reg} = 101.419\text{ kHz}$) that regulates the capacitor voltage:

$$R(z) = \frac{0.0001}{z - 1} \quad (8)$$

The figure demonstrates the steady state after the converter has reached the reference capacitor voltage of 70 V . The switching ripple was removed from the figure to show only the sampling subharmonics and the dynamics of the converter. It can be seen that even in steady state, there are subharmonic oscillations caused by the sampling of the input signals, which are not compensated properly by the regulator. When no oversampling is applied, the capacitor voltage has an artificial ripple of 0.9 V , which is 1.3% of the average value. This oscillation prevents the HIL system to be applicable, since the user cannot test their

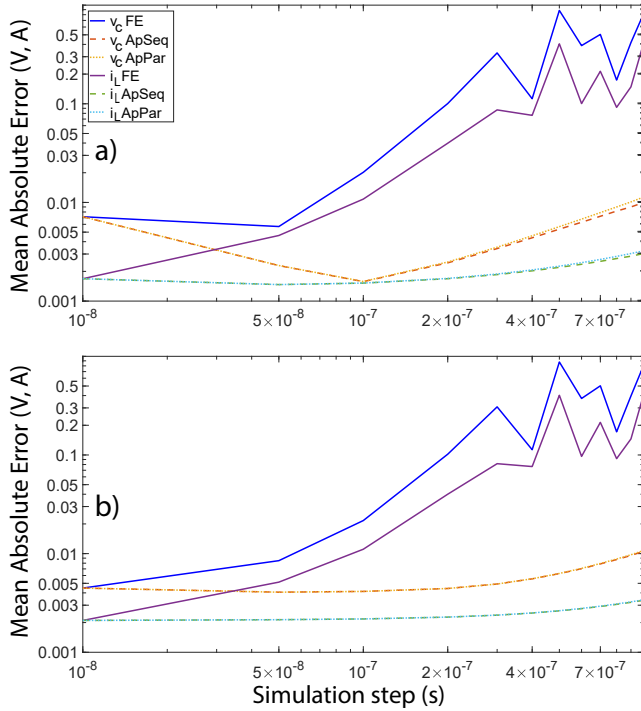


Figure 8. Absolute Errors of the capacitor voltage and inductor current during the simulation with $f_{sw} = 99.71$ kHz, $t_{on} = 41.11\%$: a) asynchronous buck without losses; b) synchronous buck with losses

controller in a reliable way. When any oversampling approach is applied, the oscillations are removed drastically up to 0.19 V (0.27%) in the capacitor voltage, proving the effectiveness of the proposed method.

A. MATLAB accuracy results

In this work, the accuracy evaluation is done for open-loop models and is defined using MAE (Mean Absolute Error) formula, which demonstrates a relative sum of absolute error from the reference golden model described above:

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - x_j| \quad (9)$$

where y_j is the value calculated inside a model and x_j is the reference value. The model errors received during the simulation are shown in Fig. 8, for the asynchronous buck — on top, for the synchronous one — on bottom (logarithmic scale). Notice that on the left part of the plots with the decrease of the integration step the error value increases with oversampling. It means that the machine epsilon of the simulation was reached, therefore, smaller dt values were not used. This issue is addressed in the previous works [13], [26].

Since previously reported ad-hoc HIL systems reach tens of nanoseconds as a simulation step, and commercial systems usually use around hundreds of nanoseconds and often around one microsecond [16], [27], the models under experiment are simulated applying various typical real-time integration steps dt , such as from 10 to 1000 ns. The PWM sampling time $dt_{samp} = 10$ ns is used for the oversampling approaches

for all models and approaches in this research. However, any sampling period could be used for the oversampling as long as it is a submultiple of the simulation period in order to obtain an integer number of cycles in the counters. The switching frequency $f_{sw} = 99.71$ kHz and the duty cycle of 41.11% are arbitrary numbers. As it was said before, these numbers were chosen for the accuracy comparison so that the *on*, *off* and *deadtime* are not perfect multiples of either the simulation step or the sampling step. The total simulation length is 27 ms.

In both Fig. 8.a) and Fig. 8.b), two solid curves at the top show the simulation error of the Forward Euler implementation without oversampling of the state variables. The errors of the proposed *ApSeq* and *ApPar* are the dashed lines at the bottom of the plots. They are basically overlapped in this resolution for the voltage and current of two converter models. As can be noticed comparing synchronous and asynchronous converters in Fig. 8, both of them show very similar error curves. Although the topologies are distinct, most of the error in the converters is caused by the same source — the insufficiently fast input sampling. That is why both converter models show the same accuracy patterns without oversampling. In the case when the simulation step is equal to the sampling step (10 ns), the error values match in all the cases (with and without oversampling). Then, using greater simulation steps of 50 – 1000 ns (from 5 to 100 samples per switching period are taken accordingly) the error of the simulation is decreased when any of the oversampling approaches is applied. The accuracy of *ApPar* in some cases is totally matching the accuracy of *ApSeq* and in the others it is very similar to it (slightly worse). Namely, for particular simulation steps in *ApPar*, the error may become greater by values of the order of maximum four decimal places for voltage and five decimal places for current. These are approximately 0.00001% of the total value in volts/amperes, which implies very limited importance to this precision difference between *ApSeq* and *ApPar*. Thus, in this case the *on-off-deadtime* order does not influence significantly the global error of the simulations with multiple sampling. This means that choosing either *ApSeq* or *ApPar* has a similar effect on the final accuracy increase of the simulations. On the other hand, the accuracy growth using some oversampling approach is important in comparison with not using any for both asynchronous and synchronous models.

In Tables II and III the received accuracy improvement for the simulation with *ApSeq* and *ApPar* is shown for the asynchronous and synchronous buck models respectively. The accuracy improvement was defined using formula (10):

$$improvement (\%) = \frac{errorFE - errorApproach}{errorFE} \cdot 100 \quad (10)$$

Error calculations of *errorFE* and *errorApproach* in (10) were carried out with MAE formula (9). From the tables can be noted that only with 5 middle points taken into account, the accuracy is already improved by 58-68% for i_L and 52-59% for v_C . Later, using more middle points, the accuracy starting from 30 points per simulation step is raised up to 99% for both current and voltage. This advancement makes the proposed approaches a powerful tool to increase the accuracy

Table II
ASYNCHRONOUS BUCK ACCURACY IMPROVEMENT WITH OVERSAMPLING

	<i>Sim. step (ns)</i>	50	100	200	300	500	700	1000
i_L	ApSeq	68.21%	85.93%	95.77%	97.86%	99.46%	98.82%	99.28%
	ApPar	68.20%	85.91%	95.74%	97.83%	99.44%	98.76%	99.23%
v_C	ApSeq	59.82%	92.20%	97.57%	98.96%	99.39%	98.57%	98.79%
	ApPar	59.75%	92.17%	97.52%	98.93%	99.36%	98.44%	98.63%

Table III
SYNCHRONOUS BUCK WITH LOSSES ACCURACY IMPROVEMENT WITH OVERSAMPLING

	<i>Sim. step (ns)</i>	50	100	200	300	500	700	1000
i_L	ApSeq	58.39%	80.35%	94.32%	97.09%	99.35%	98.64%	99.19%
	ApPar	58.38%	80.35%	94.32%	97.08%	99.35%	98.63%	99.18%
v_C	ApSeq	52.09%	80.88%	95.65%	98.40%	99.28%	98.43%	98.71%
	ApPar	52.10%	80.88%	95.65%	98.40%	99.28%	98.42%	98.68%

of the models not changing neither the integration step nor the applied numerical method.

B. HLS synthesis and utilization

Since the paper is focused on FPGA-based HIL systems, proposed oversampling approaches were synthesized in Xilinx Vivado using HLS (High Level Synthesis) to analyze the area and time overhead, and then implemented into an xc7a35ticsg324-1L FPGA. This low-end FPGA was used because any modern FPGA can sample an input and operate a binary counter with a $5\text{-}10\text{ ns}$ period or even less (in this case the sampling time was 10 ns). There is no need to store a big amount of information, because only several counters and the order of the input sequence are managed. Therefore, a high-end FPGA or processor are not necessary to perform the oversampling.

The utilization results of the models are presented in Table IV for the asynchronous converter, and in Table V for the synchronous converter with losses. The models which include the sequential technique are implemented in two different ways — reusing and not reusing hardware — as it was explained in Section III. The used resources and time growth are analysed in comparison with the simplest *FE* method. As it can be noticed from Tables IV and V, the resource utilization increase for the oversampling approaches is quite similar. The LUTs and FFs increment is around 2 times for all of them. The DSPs utilization growth is less significant for *ApSeq_r*, than for other methods, since its logic overhead above *FE* is small. For the rest of cases the DSPs increase is also around 2 times.

Still the most important characteristic to compare here is the minimum achievable period of the approach. As it can be seen from both HLS utilization tables, *ApPar* is reaching the best timing of three oversampling models by a great margin for both converter models. For the model with a single switching event (Table IV) the minimum period grows up to 2 times for *ApSeq_r* and *ApSeq_{nr}*, while for *ApPar* it does only 1.5 times. For the case of two possible switching events (Table V), the time overhead is predictably even bigger for the sequential techniques, since it is proportional to the number of switching events. While for *ApSeq_r* and *ApSeq_{nr}* the minimum period grows 2.3 times, for *ApPar* it does only 1.4 times. This significant time improvement is the key factor for

Table IV
ASYNCHRONOUS BUCK HLS UTILIZATION RESULTS

	FE	ApSeq_r	ApSeq_{nr}	ApPar
<i>LUTs</i>	531	1154 (x2.17)	1385 (x2.6)	1140 (x2.15)
<i>FFs</i>	595	1252 (x2.1)	1504 (x2.53)	1327 (x2.23)
<i>DSPs</i>	5	5 (x1)	10 (x2)	8 (x1.6)
<i>T_{min} (ns)</i>	124.94	250.72 (x2)	241.82 (x1.94)	188.73 (x1.51)

power electronics applications not only because of the speed of the execution itself, but also because of the possibility of implementing converters with higher switching frequencies.

C. Selection summary

Apart from the novel contributions already discussed in the Introduction (such as application of oversampling to switching instead of average models, real time feasibility, fixed synchronization delay, straightaway outputting of the correct value), another important assertion after analyzing the simulation and synthesis results is the rationality of using the parallel approach. In HIL models with sufficiently small simulation step, like the ones used in power converters modeling, the gained accuracy difference after sequential calculation does not play crucial role in contrast to the important minimum period decrease with parallel calculation. However, the parallel calculation was not proposed yet in previous research (see Section I).

To conclude, all the oversampling approaches receive a similar error decrease while the difference between their used area does not make a big impact. This gives an advantage to the parallel approach for HW implementations in FPGA, due to its great win in minimum timing, along with the possibility of easily extending *ApPar* to multiple switching events. Thus, it is recommended to use *ApPar* for oversampling of switching converters input signals, discarding the sequential techniques, since they are less efficient for FPGA implementations. The experimental implementation of the chosen oversampling approach *ApPar* will be provided in Section V using the topology of the asynchronous buck converter for the sake of clarity.

Table V
SYNCHRONOUS BUCK WITH LOSSES HLS UTILIZATION RESULTS

	FE	ApSeq _r	ApSeq _{nr}	ApPar
<i>LUTs</i>	1174	2246 (x1.91)	3204 (x2.73)	2234 (x1.9)
<i>FFs</i>	1173	2022 (x1.72)	3414 (x2.91)	2150 (x1.83)
<i>DSPs</i>	7	9 (x1.29)	18 (x2.57)	18 (x2.57)
<i>T_{min} (ns)</i>	236.95	552.11 (x2.33)	540.33 (x2.28)	322.91 (x1.36)

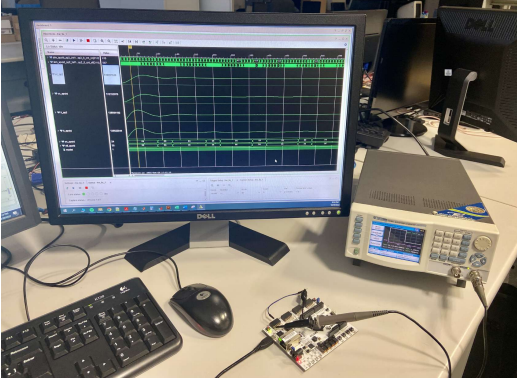


Figure 9. Experimental setup used in the study

V. EXPERIMENT. CASE STUDY

The models of the asynchronous buck converter — the chosen model with oversampling (*ApPar*) and the regular model without oversampling (*FE*) — were built in Vitis HLS tool using C++ language for hardware implementation. The models were exported to and implemented inside Xilinx Vivado environment. To test the HIL models, instead of using a closed-loop controller that would mask the effectiveness of the oversampling approaches, an external PWM signal generator was used, with a constant PWM signal. All the data were collected through an Integrated Logic Analyzer (ILA), which is an IP (Intellectual Property) block from Xilinx. The picture of the experimental setup is shown in Fig. 9. The setup includes an FPGA xc7a35ticsg324-1L, an external wave generator and a computer. The state variable values were only sampled in every switching transition because the limited memory resources of the FPGA do not allow storing the data at the FPGA clock speed.

In Fig. 10 some of the received waveforms during the experiment are presented. The switching transitions (sampling triggers) correspond to the high and low peaks of inductor current. The state variables are taken during the steady state with different combinations of arbitrary switching frequencies and duty cycles. Notice that both approaches were implemented with their minimum corresponding simulation step — *FE* with 130 ns and *ApPar* with 190 ns. The red waves show results not applying oversampling, the dashed orange ones — results of the model with oversampling, and the blue ones — the reference model, which was made by offline simulation with 1 ns step, while the rest are experimental results. As it can be seen, the oversampling approach and the reference waveforms overlap, showing the high accuracy of the proposed method.

Fig. 11 shows a zoom of the inductor current from Fig. 10.a) where the switching ripple can be seen. It should be remarked that if oversampling is not used, large undesirable subharmonic oscillations appear for all the presented frequencies due to the lack of the PWM-reading precision. It produces a significant error increase since this oscillation will not be found in real systems. In contrast, applying the oversampling technique improves the simulation so much so that a real steady-state is reached as it is expected when the PWM generation is stable.

To calculate decrease of the undesirable subharmonic oscillations using the oversampling technique, the switching ripple was filtered capturing only the lower peaks of the current in every cycle. The results for some switching frequencies are given in Tables VI and VII. The results for the current and voltage give similar improvement since the state variables are interconnected, i.e. a change of one influences the other. The absolute values of the oscillation amplitude in amperes and volts are presented without and with oversampling applied. Then, the total oscillation improvement, i.e. its decrease, is calculated as:

$$Oscill.(%) = \frac{Oscill.w/o - Oscill.with}{Oscill.w/o} \cdot 100 \quad (11)$$

The results show up to 99% oscillation decrease with oversampling technique. The undesirable oscillation is significantly removed with oversampling, as it was asserted by theory.

For all the experimental cases, the simulation validation was carried out with the experimental sets of parameters. The parameters include switching frequency, on-time of duty cycle and minimum achievable time-steps, corresponding to the methods. The comparison of received accuracy for current and voltage is presented in Tables VI and VII, columns 3-4. The errors were calculated using MAE approach, described in Section IV-A. As in can be observed from tables, the accuracy during the re-simulation meets the experimental accuracy for all the cases with a possible small margin of error not greater than +1%. This means that proposed oversampling approach works as expected in experimental environment and can be applied in real-time converter models to remove the undesirable oscillation.

Comparing the waves received in the experiment (Fig. 10) with the wave from MATLAB (Fig. 6), it can be stated that the experimental waveform behavior meets the simulation expectation showing same stabilization character. The output signal obtained with the oversampling technique practically coincides with the reference model in both theory and experiment. Taking into consideration this excellent stabilization and the accuracy increase due to the use of more exact timing each time-step, oversampling becomes a great technique to improve the quality of digital simulations.

VI. CONCLUSION

In this study, the use of the oversampling technique has been proposed for the simulation of switched power electronic models in real time. The technique implies taking additional points during the PWM input reading at each simulation step. Oversampling provides a significant improvement to the

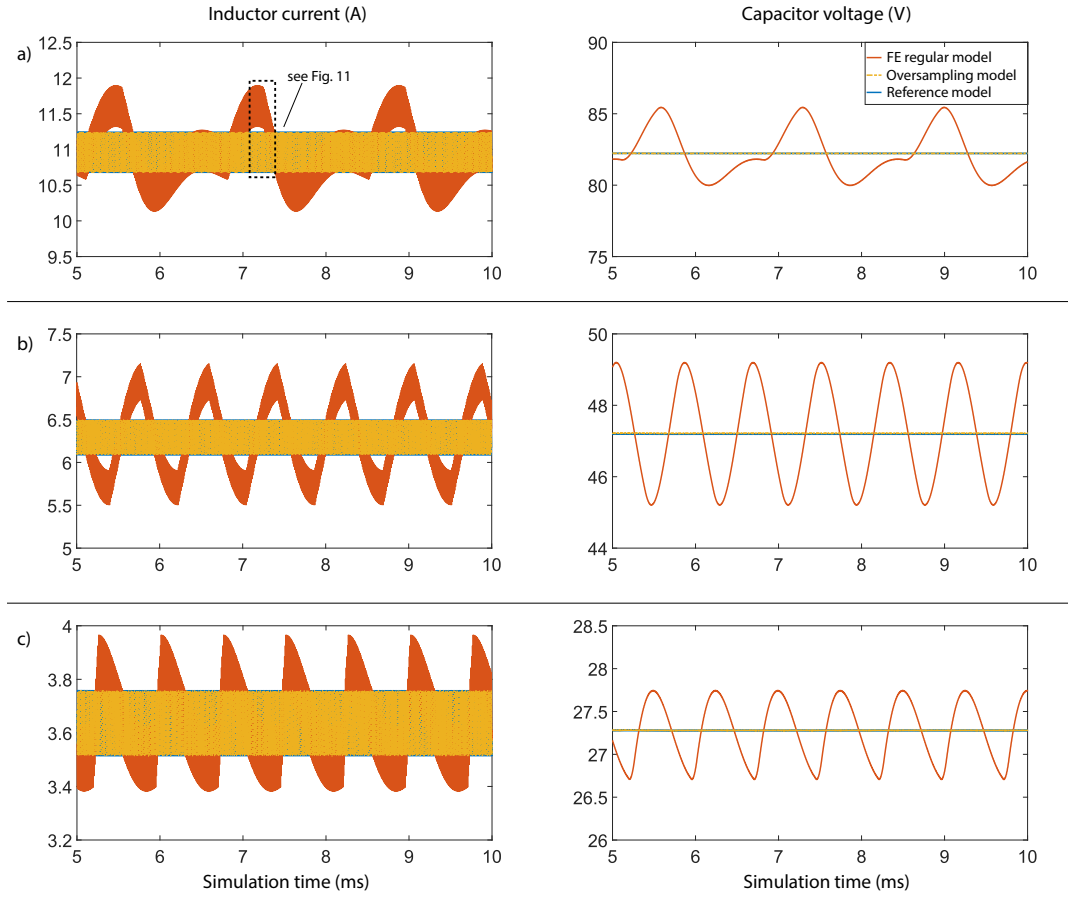


Figure 10. Experimental state variable waveforms in asynchronous buck converter during steady state: a) $f_{sw} = 99.71$ kHz, duty = 41.11%; b) $f_{sw} = 103.20$ kHz, duty = 23.6%, c) $f_{sw} = 113.09$ kHz, duty = 13.64%

Table VI
INDUCTOR CURRENT OSCILLATION W/O AND WITH OVERSAMPLING

fsw (kHz)	on-time (%)	error w/o overs. (A)		error with overs. (A)		oscill. w/o overs. (A)	oscill. with overs. (A)	oscill. decrease with overs. (%)
		simul.	exper.	simul.	exper.			
99.71	41.11	$1.756 \cdot 10^{-2}$	$1.756 \cdot 10^{-2}$	$7.129 \cdot 10^{-3}$	$7.129 \cdot 10^{-3}$	1.288	0.18	98.48
101.42	31.17	$1.412 \cdot 10^{-2}$	$1.414 \cdot 10^{-2}$	$3.501 \cdot 10^{-3}$	$3.502 \cdot 10^{-3}$	1.169	1.08	92.48
103.20	23.60	$1.622 \cdot 10^{-2}$	$1.623 \cdot 10^{-2}$	$1.612 \cdot 10^{-3}$	$1.614 \cdot 10^{-3}$	1.183	1.11	94.18
104.46	16.74	$1.573 \cdot 10^{-2}$	$1.573 \cdot 10^{-2}$	$2.327 \cdot 10^{-3}$	$2.328 \cdot 10^{-3}$	1.022	0.24	98.98
113.09	13.64	$1.036 \cdot 10^{-2}$	$1.035 \cdot 10^{-2}$	$2.642 \cdot 10^{-3}$	$2.642 \cdot 10^{-3}$	0.689	0.27	98.59

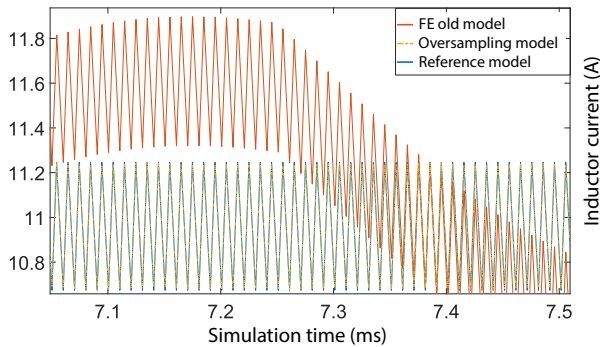


Figure 11. Zoom of the inductor current simulation from Fig. 7 a)

simulation by reducing the negative effects of inaccurate PWM

reading. The proposed techniques are based on the application of an intermediate simulation step, in order to apply the state equations correctly in relation to the read inputs. Through this, undesirable subharmonics are eliminated and the numerical error is reduced up to 98% starting from the acquisition of only 30 oversampling points. In addition, the application of this technique opens the possibility of simulating complex systems that require larger simulation steps to be executed in real time, since oversampling drastically reduces the error regardless of the simulation step.

Two possible oversampling approaches have been analyzed in the paper: the sequential, which takes into account the order of on-off appearance, and the parallel which does not. The concept of the parallel approach was developed by the authors, while the sequential approach is based on previous state of the art. The analysis of the gained error and used resources

Table VII
CAPACITOR VOLTAGE OSCILLATION W/O AND WITH OVERSAMPLING

fsw (kHz)	on-time (%)	error w/o overs. (V)		error with overs. (V)		oscill. w/o overs. (V)	oscill. with overs. (V)	oscill. decrease with overs. (%)
		simul.	exper.	simul.	exper.			
99.71	41.11	$5.951 \cdot 10^{-2}$	$5.951 \cdot 10^{-2}$	$1.332 \cdot 10^{-3}$	$1.331 \cdot 10^{-3}$	7.38	0.03	99.64
101.42	31.17	$8.520 \cdot 10^{-2}$	$8.521 \cdot 10^{-2}$	$1.922 \cdot 10^{-3}$	$1.923 \cdot 10^{-3}$	9.02	0.49	94.62
103.20	23.60	$1.007 \cdot 10^{-1}$	$1.007 \cdot 10^{-1}$	$1.893 \cdot 10^{-3}$	$1.892 \cdot 10^{-3}$	12.34	0.54	95.62
104.46	16.74	$8.982 \cdot 10^{-2}$	$8.982 \cdot 10^{-2}$	$1.221 \cdot 10^{-3}$	$1.222 \cdot 10^{-3}$	9.84	0.06	99.36
113.09	13.64	$4.828 \cdot 10^{-2}$	$4.828 \cdot 10^{-2}$	$1.635 \cdot 10^{-3}$	$1.635 \cdot 10^{-3}$	5.99	0.11	98.20

has proven that the parallel approach is more appropriate for FPGA implementation while obtaining almost identical accuracy improvement and a possibility to apply it to multiple switching events. Thus, the parallel approach is recommended to use in real-time hardware implementations of switching models. It has been implemented in FPGA and analyzed through experimental tests. The experimental results met the expectations of improving the quality of the simulation by reducing the undesirable subharmonics of state variables up to 99% for various frequencies.

ACKNOWLEDGMENT

This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Autónoma de Madrid in the line for the Excellence of the University Teaching Staff, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

REFERENCES

- [1] P. S. Niklaus, J. W. Kolar, and D. Bortis, "100 kHz large-signal bandwidth gan-based 10 kva class-d power amplifier with 4.8 mhz switching frequency," *IEEE Transactions on Power Electronics*, pp. 1–20, 2022.
- [2] G. Lauss and K. Strunz, "Accurate and stable Hardware-in-the-Loop (HIL) real-time simulation of integrated power electronics and power systems," *IEEE Transactions on Power Electronics*, vol. 36, no. 9, pp. 10 920–10 932, 2021.
- [3] J. Badar, F. Akhter, H. M. Munir, S. S. H. Bukhari, and J.-S. Ro, "Efficient real-time controller design test bench for power converter applications," *IEEE Access*, vol. 9, pp. 118 880–118 892, 2021.
- [4] M. D. Omar Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, K. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, C. Strunz, M. Saeedifard, X. Wang, D. Shearer, and M. Paolone, "Real-time simulation technologies for power systems design, testing, and analysis," *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 63–73, 2015.
- [5] C. Washington and J. Dolman, "Creating next generation HIL simulators with FPGA technology," in *2010 IEEE AUTOTESTCON*, 2010, pp. 1–6.
- [6] J. Zheng, Z. Zhao, Y. Zeng, S. Ji, and L. Yuan, "An event-driven parallel acceleration real-time simulation for power electronic systems without simulation distortion in circuit partitioning," *IEEE Transactions on Power Electronics*, vol. 37, no. 12, pp. 15 626–15 640, 2022.
- [7] J. Xu, K. Wang, P. Wu, Z. Li, Y. Liu, G. Li, and W. Zheng, "FPGA-based submicrosecond-level real-time simulation of solid-state transformer with a switching frequency of 50 kHz," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, no. 4, pp. 4212–4224, 2021.
- [8] S. Mazumdar and K. Basu, "Hardware emulation of energization of a long transmission line by high-frequency power electronic converter," *IEEE Transactions on Power Electronics*, vol. 35, no. 9, pp. 9267–9280, 2020.
- [9] Typhoon HIL. [Online]. Available: <https://www.typhoon-hil.com>
- [10] Opal-RT. [Online]. Available: <https://www.opal-rt.com/>
- [11] M. Zhang, Z. Zhang, Z. Li, H. Chen, and D. Zhou, "A unified open-circuit-fault diagnosis method for three-level neutral-point-clamped power converters," *IEEE Transactions on Power Electronics*, pp. 1–12, 2022.
- [12] M. Yushkova, A. Sanchez, and A. de Castro, "The necessity of resetting memory in adams–bashforth method for real-time simulation of switching converters," *IEEE Transactions on Power Electronics*, vol. 36, no. 6, pp. 6175–6178, 2021.
- [13] M. Yushkova, A. Sanchez, and A. de Castro, "Strategies for choosing an appropriate numerical method for FPGA-based HIL," *International Journal of Electrical Power Energy Systems*, vol. 132, p. 107186, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0142061521004257>
- [14] K. Strunz, L. Linares, J. Marti, O. Huet, and X. Lombard, "Efficient and accurate representation of asynchronous network structure changing phenomena in digital real time simulators," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 586–592, 2000.
- [15] X. Li, X. Ruan, Q. Jin, M. Sha, and C. K. Tse, "Small-signal models with extended frequency range for dc–dc converters with large modulation ripple amplitude," *IEEE Transactions on Power Electronics*, vol. 33, no. 9, pp. 8151–8163, 2018.
- [16] C. R. D. Osório, M. Miletic, J. Zelic, D. Majstorovic, and O. Gagrira, "Advancements on real-time simulation for high switching frequency power electronics applications (invited paper)," in *2021 21st International Symposium on Power Electronics (Ee)*, 2021, pp. 1–6.
- [17] K. Lian and P. Lehn, "Real-time simulation of voltage source converters based on time average method," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 110–118, 2005.
- [18] D. Maksimovic, A. Stankovic, V. Thottuvelil, and G. Verghese, "Modeling and simulation of power electronic converters," *Proceedings of the IEEE*, vol. 89, no. 6, pp. 898–912, 2001.
- [19] S. Zhao, N. Felderer, and J. Allmeling, "Real-time simulation of three-phase current source inverter using sub-cycle averaging method," in *2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2020, pp. 1–6.
- [20] J. Allmeling and N. Felderer, "Sub-cycle average models with integrated diodes for real-time simulation of power converters," in *2017 IEEE Southern Power Electronics Conference (SPEC)*, 2017, pp. 1–6.
- [21] V. Dinavahi, M. Reza Iravani, and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," *IEEE Transactions on Power Delivery*, vol. 16, no. 4, pp. 775–781, 2001.
- [22] V. Dinavahi and M. Iravani, "Coupling digital controllers with real-time digital simulators for switched power systems," in *Proc. Int. Conf. Power Systems Transients*, 2001.
- [23] M. Faruque, V. Dinavahi, and W. Xu, "Algorithms for the accounting of multiple switching events in digital simulation of power-electronic systems," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1157–1167, 2005.
- [24] Typhoon HIL GDS oversampling. [Online]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/concepts/gds_oversampling.html
- [25] E. Zamiri, A. Sanchez, M. S. Martínez-García, and A. de Castro, "Analysis of the aliasing effect caused in Hardware-in-the-Loop when reading PWM inputs of power converters," *International Journal of Electrical Power Energy Systems*, vol. 136, p. 107678, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S014206152100908X>
- [26] R. Saralegui, A. Sanchez, M. S. Martínez-García, J. Novo, and A. de Castro, "Comparison of numerical methods for hardware-in-the-loop simulation of switched-mode power supplies," in *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2018, pp. 1–6.
- [27] N. Lin and V. Dinavahi, "Detailed device-level electrothermal modeling of the proactive hybrid hvdc breaker for real-time hardware-in-the-loop simulation of dc grids," *IEEE Transactions on Power Electronics*, vol. 33, no. 2, pp. 1118–1134, 2018.