A Real Time Simulator of a Phase Shifted Converter for High Frequency Applications

Tudor GHERMAN[1], Dorin PETREUS[1], Marcian Nicolae CIRSTEA[2]
[1]*Applied Electronics Department, Technical University of Cluj-Napoca, Cluj-Napoca, Romania*
[2]*School of Computing and Information Science, Anglia Ruskin University, Cambridge, United Kingdom*
Tudor.Gherman@ael.utcluj.ro

***Abstract*—This paper presents a switched function FPGA-based Real Time Simulator (RTS) of a synchronous Phase Shifted (PS) converter. The design methods developed contribute to improving the accuracy, the portability, to lowering the cost and the resource demand of RTS models, enabling them to be easily deployed both in hardware in the loop (HIL) simulations, but also in error detection or health monitoring systems where these properties are essential. The research work carried out demonstrates the importance of reducing the simulation time step for avoiding false limit cycling behavior and obtaining an accurate closed loop response of the RTS. The very small time step (20 ns), not achievable with commercial real time simulation tools, helped in accurately modeling the time and frequency response of the converter for switching frequencies of 200 kHz (tested) and above. Although applied to a particular type of DC-DC converter, the methods presented can be used to successfully model a wide range of Switched Mode Power Supply (SMPS) topologies. An innovative hardware platform that enables running the real time simulation model in parallel with the reference converter and facilitates a comparative analysis that proves the fidelity of the RTS of the PS converter was also developed.**

***Keywords*—real-time systems, closed loop systems, field programmable gate arrays, high level synthesis, DC-DC power converters.**

I. INTRODUCTION

Real time simulation techniques have been developed for more than three decades [1] but the recent progress in digital systems' performance facilitates the development of higher bandwidth and more accurate models. The most common use of RTSs is as part of HIL test platforms, aiming to facilitate control systems test and development in applications such as wind energy conversion systems [2], maximum power point tracking systems [3], fuel cell hybrid vehicles [4], power systems analysis [5], machine testing or rapid prototyping [1], with HIL simulations taking the form of signal level, power level or mechanical level simulations [6]. Benefits such as increased flexibility, reduced software development time and cost, possibility of testing control systems in situations that may result in hardware damage have established HIL simulation as part of the development process in automotive, avionics, defense or industrial equipment applications [7].

To date, HIL simulations have been used to model mostly high power devices or systems. However, SMPSs with power levels in the range of kW or below have been increasingly used in automotive industry, particularly as part of Level 1 and Level 2 on board battery chargers [8], or in renewable energy systems. Development of RTSs for power converters integrated in such complex systems is justified, allowing software to be designed and tested early in the development process. Moreover, as high performance parallel resources are now available in small low cost FPGAs and because recent development of high level synthesis (HLS) tools including MATLAB's HDL Coder, System Generator or Vivado HLS have contributed to making FPGA programming more accessible for users not familiarized with conventional hardware description languages, HIL simulation platforms can be migrated from expensive platforms to low cost and more flexible embedded systems. These factors have contributed to extending the use of RTSs from their conventional functionality as part of HIL simulations to functionalities such as estimation, observation, diagnostics or health monitoring [9] - case in which the RTS runs on the same embedded device as the controller and available resources constitute a major constraint.

Several approaches of implementing real time simulation models of DC-DC power converters can be found in literature. Examples of using HIL simulation systems for accelerating control loop development for a Buck converter are presented in [6] and [10]. However, the RT simulation models presented in these papers are averaged models running on conventional CPUs limiting the accuracy and utility of the models as part of the HIL simulation. In [11], an electrical vehicle battery charger real time simulator is presented but no details regarding the discretization technique or the performance obtained are offered.

A high fidelity real time switched model was also presented in [12], but authors reported a much larger time step of 1 μs for a simpler DC-DC converter topology. The use of ideal components also limits the utility of the model since inductor and capacitor equivalent series resistors (ESR) influence the frequency response of the converter close to or below the crossover frequency. Another drawback of the approach presented in [12] is the fact that it gives no details about the discretization method used and relies on a particular software and hardware to generate the model, limiting its portability and utility. More detailed analysis with very low time steps obtained and convincing results were presented in [13-15], but the methods presented were applied to IGBT based voltage source converters (VSC) and SMPS control particularities were not studied.

The concepts and implementation methods proposed in this paper are bringing to a higher level the techniques and results reported previously in literature ([16-23]) with significant improvements on the PS converter RTS model, the reduction in time step, an extended HIL platform that

Digital Object Identifier 10.4316/AECE.2020.03002

allows a comparative analysis between the reference converter and the RTS and the addition of limit cycling analysis.

The main goals of the present paper are: i) to develop a highly portable, low resource demand RTS of the PS converter that can be deployed on low cost embedded platforms and is capable of accurately modeling the behavior of the converter for switching frequencies in the range of hundreds of kHz ii) to demonstrate that obtaining a simulation time step lower than what is achievable with RTS dedicated commercial tools is critical for avoiding false limit cycling conditions and accurately modeling the closed loop behavior of a SMPS; iii) to develop a configurable hardware platform in order to demonstrate the use of the RTS in both HIL simulations and health monitoring applications.

The rest of the paper is organized as follows: Section II presents methodology aspects; Section III discusses the modelling of the PS converter in the continuous time domain and in the discrete time domain; Section IV presents in detail the implementation of the RTS model and the hardware platform used for validation, while in Section V the simulation and experimental results are presented. Conclusions are finally drawn in Section VI.

II. METHODOLOGY

A. Methods, Tools and Techniques

Electronic Design Automation (EDA) tools and techniques have evolved over the years, ultimately enabling the design and development of complex electronic systems of high performance. In order to evaluate the functionality of electronic systems, mathematical models were traditionally used. A new generation of design methods has then brought closer the design's behavioral definition and its hardware implementation in a unique environment, through the use of hardware description languages (HDL). Due to the availability of FPGAs as reconfigurable multi-million-gate hardware chips that also contain processor cores, design methods had to change and adapt to enable a faster time-to-market for complex products. The outcome is a modeling, simulation, design environment based on modern EDA tools that enables an early stage functional simulation ahead of hardware implementation, hence maximizing performance for design processes and products. In order to achieve a design environment that allows simultaneous consideration of all system parameters at an early stage in the design process, holistic modelling and simulation of complex electronic systems are the key.

The most advanced generation of design methodologies for electronic systems marks a step forward in terms of achieving a high level of abstraction. Nowadays, engineers design hardware using either HLS tools based on high-level languages or schematic-based Electronic System Level (ESL) design.

The first type of methods, based on HLS tools, enable the behavioral system description written in a high-level language to be transformed into a Register Transfer Level (RTL) implementation / description ([24-26]). In this approach, the underlying details of implementation are not visible to the user and their automatic generation enables a shorter time to market and reduces costs.

A second type of methods is that based on schematic blocks in ESL design. The tools facilitating the top-down approach enable the design to be developed at high level of abstraction, in a graphical format. There are two main families of tools in this category, namely the Mathworks family (including MATLAB, Simulink, etc. [27]) and the National Instruments (NI) family (including tools such as Modelsim, Labview). From such tools, it is possible to deploy models to other simulation environments, including HIL systems. C-code or HDL generation is possible, too, thus facilitating the direct hardware implementation of complex electronic control systems. Rapid prototyping of electronic system designs is thus streamlined. However, the high level of abstraction may decrease the design flexibility and the performance, especially in terms of RTS simulation time step with achievable values in the range of hundreds of ns or even μs [28-29]. For this reason a tool based on a high level description language (Vivado HLS) was chosen in the detriment of schematic-based ESL tools.

B. Models and Simulators

According to [15], offline simulators can be classified as system level and device level simulators, classification that can be applied to RTSs as well. System level simulators are mostly oriented towards modeling the network behavior of the power converter and use numerical integration methods to implement discrete solvers for the linear ordinary differential equations (ODE) that describe the circuit. Three behavioral models are commonly used for power electronic devices within this class of simulators: i) ideal models, ii) switching function models and iii) averaged models.

The use of averaged models is the simplest approach since it considers only the low frequency components of the switching function. Most controller design guidelines rely on the averaged model, thus using an averaged model may be sufficient strictly for assessing the SMPS stability. The main drawback of this approach is the fact that in a HIL simulation the PWM generator will need to be replaced by its averaged model as well, thus the PWM generator will not be tested as part of the simulation. Besides this, in the case of digital control, voltages and currents that need to be controlled are sampled on each switching period [30, pp 89-93]. Determining the correct sampling time is essential, particularly for waveforms that exhibit high ripple. An averaged model will not be able to predict errors caused by improper sampling. A second effect associated with digital control of SMPS that averaged models are not able to model is limit cycling.

Real time simulators based on switching function models replace the switch network with controlled voltage or current sources obtaining, at the end of the process, a time - invariant topology. In comparison with ideal model based RTSs, which model each switching device individually, switched function model RTSs allow smaller time steps and are still able to model high frequency effects [31].

Device level simulators aim to model switching transients, power losses and thermal behavior of devices and conventionally rely on time consuming simulation methods not suitable for real time simulations. While successful real time device level simulators of voltage source converters

implemented on FPGAs have been reported in literature [14-15], the methods proposed lead to a significant increase of required FPGA resources. Furthermore, switching devices used in converters below 1 kW have switching characteristics that can't be accurately modeled with time steps in the range of tens of ns that can be achieved with mid-range or low-end FPGAs. Thus, the most suitable approach for simulating SMPS in real time is the use of a system level simulator based on switched function models.

C. RTS use case and validation

The intended use cases of the developed RTS are signal level HIL simulations and error detection or health monitoring systems. The scope of signal level HIL simulations is to facilitate and accelerate the control loop design, all the elements of the loop except the controller and the PWM generator being simulated. Thus, the RTS is part of the loop. For error detection and health monitoring applications it is the real hardware that runs in the loop and the RTS runs in parallel on the same digital device as the controller. For these applications it is essential that the real (reference) converter and the RTS are driven by the same input signals and the outputs of the two blocks are available at all time to the digital system for comparison.

A highly configurable digital platform that accommodates both use cases was developed. Taking advantage of the parallelism and flexibility of the FPGA, the system can be configured to include either the RTS or the reference PS converter in the loop. This architecture also enables a comprehensive analysis of the RTS performance. A simplified block diagram of the developed system is presented in Fig. 1.

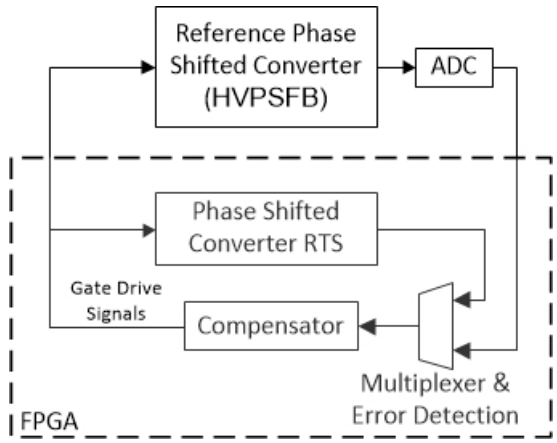

Figure 1. Hardware system simplified block diagram

For convenience, the digital system was implemented on a Digilent Eclypse Zynq based board with a Zmod ADC 1410 and a Zmod DAC 1411 (modules also manufactured by Digilent) implementing the analog to digital and digital to analog interfaces. However, the described project is by no means constrained to using these development boards. A HVPSFB evaluation board from Texas Instruments [33] provides the reference PS converter.

III. Converter Modeling

The selected reference converter to be modeled is a PS converter. This choice is based on the broad range of applications in which the PS converter is used including telecom rectifiers, renewable energy systems or battery charging systems ([8], [32]). Such applications may require various elaborate control schemes. As robustness and ease of maintenance are also critical for these systems, the ability of performing HIL simulations or health monitoring with the aid of the proposed RTS is a considerable benefit.

A. Phase Shifted Converter Continuous Time Domain Modeling

Depending on the application demands, the topology of the PS converter has been developed with several variations. The topology chosen to validate the real time simulation model presented in this paper is illustrated in Fig. 2 a).

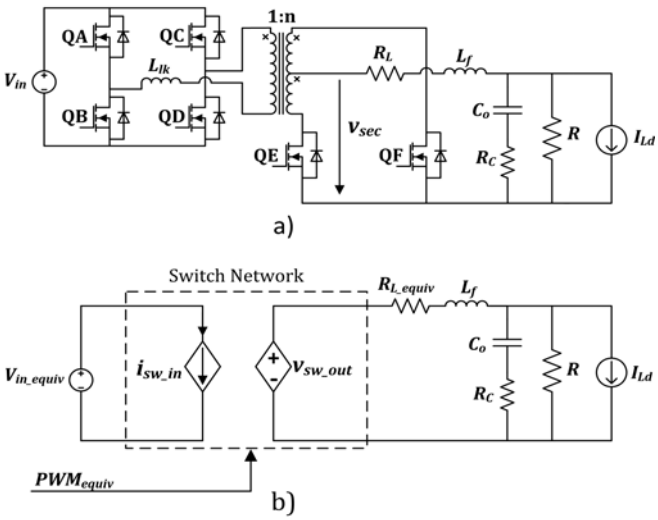

Figure 2. PS converter a) simplified schematic b) equivalent circuit

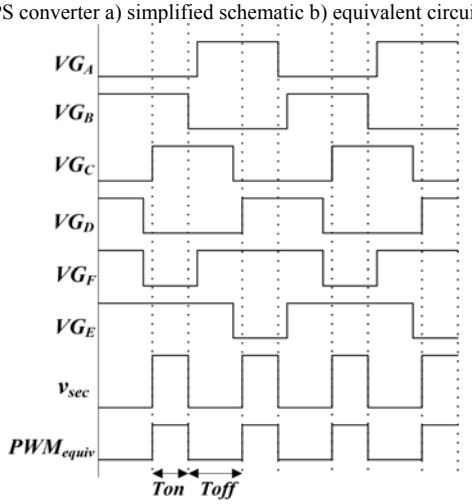

Figure 3. PS converter gate drive signals, secondary mid winding voltage and equivalent PWM

The gate drive signals of the switching elements (VG_x, where x is the index letter of the transistor) and the voltage on the mid winding of the transformer (v_{sec}) are described in Fig. 3 [33]. The phase shift between the two legs of the primary side of the converter will determine the duty cycle of the PWM signal on the mid winding of the transformer in the secondary side. Therefore, the secondary of the PS converter can be equivaled with a synchronous Buck converter (Fig. 2 b)). However, the effect of the leakage inductance of the transformer (L_{lk}) needs to be accounted for in order to obtain an accurate model. The leakage inductor is necessary to enable zero voltage switching, but also limits the slope of the primary current when voltage is applied to

the transformer. As a result, the duty cycle will depend not only on the phase shift between the legs of the full bridge of the primary, but also on its value, the filter inductor current, the input voltage, the ratio of the transformer and the switching frequency. This effect can be modeled by adding a resistor (R_d) in series with the filter inductor. The value of R_d is computed as follows [34]:

$$R_d = 4n^2 L_{lk} F_{sw} \tag{1}$$

where F_{sw} represents the switching frequency, n the ratio of the transformer and L_{lk} the leakage inductance. The winding resistance of the transformer and R_d can be lumped together with the series resistance of the filter inductor, the equivalent resistance being labeled R_{L_equiv} (as shown in Fig. 2 b)). While this approach implies knowing the switching frequency of the converter in advance, it is not limitative for most applications since the PS converter usually operates at a well-known constant frequency. Besides limiting the slope of the primary side current, the leakage inductance also influences the amplitude of the voltage across the primary and secondary of the transformer. During time interval *Toff* (Fig. 3), both QE and QF are in the on state and the voltage across the windings of the transformer is zero. During *Ton*, the filter inductor current is reflected in the primary causing a voltage drop on L_{lk}. The input voltage of the equivalent Buck converter (v_{in_equiv}) is equal to the mid winding voltage of the transformer during *Ton* and can be computed as:

$$v_{in\_equiv}(t) = n\left(V_{in} - nL_{lk}\frac{di_{Lf}(t)}{dt}\right) \tag{2}$$

where i_{Lf} is the current through the filter inductor. As described below, the switching function makes the value of v_{in_equiv} irrelevant during *Toff*. The PWM drive signal of the equivalent synchronous Buck converter is described by:

$$PWM_{equiv} = \left(VG_A \text{ and } VG_D\right) or \left(VG_B \text{ and } VG_C\right) \tag{3}$$

The PS converter switch network behavior is also not identical to that of a synchronous Buck. The output voltage of the network (v_{sw_out}) is a PWM signal but the input current of the network (i_{sw_in}) will not be zero during time interval *Toff* (when QF and QE are in the on state). In practice, it is difficult to determine the exact value of the input current for the time interval *Toff* and such an attempt would have little value since typical implementations of the input current sensors use circuits based on current transformers that also alter the input valley current waveform [33]. Control methods that rely on the input current (such as peak current mode control or average current mode control) only use the peak current or average current which can be easily and precisely modeled based on the filter inductor current waveform during *Ton*. Thus, the value of the input current during *Toff* can be considered to be equal with the reflected filter inductor current without any impact on the usage of the model. The input and output of the equivalent switch network of the PS converter can therefore be modeled by a controlled current source and a controlled voltage source being described by:

$$i_{sw\_in}(t) = ni_L(t) \tag{4}$$

$$v_{sw\_out}(t) = sw(t)v_{in\_equativ} \tag{5}$$

In the above equations the switching function *sw(t)* of the switched function model is 1 when the PWM_{equiv} signal is in the logic high state and 0 when the PWM_{equiv} signal is in the logic low state. The RTS is designed to emulate either a passive or an active load. Therefore, the output stage of the converter is modeled with a passive resistor in parallel with a current source (Fig. 2 b)). The characteristic state space representation of the equivalent synchronous Buck converter is described by:

$$\frac{dx(t)}{dt} = Ax(t) + Bv(t) \tag{6}$$

$$y(t) = Cx(t) + Ev(t) \tag{7}$$

The state vector x, the input vector v and the output vector y are defined as:

$$x(t) = \begin{bmatrix} i_{Lf}(t) \\ v_c(t) \end{bmatrix} \tag{8}$$

$$v(t) = \begin{bmatrix} v_{sw\_out}(t) \\ i_{Ld}(t) \end{bmatrix} \tag{9}$$

$$y(t) = \begin{bmatrix} i_{Lf}(t) \\ v_o(t) \end{bmatrix} \tag{10}$$

In the above definitions v_c represents the voltage across the output capacitor, while v_o represents the output voltage of the converter. The matrices A, B, C and E can be computed as:

$$A = \begin{pmatrix} -\frac{1}{L_f}\left(R_{L\_equiv} + \frac{RR_C}{R+R_C}\right) & -\frac{1}{L_f}\frac{R}{R+R_C} \\ \frac{1}{C_o}\frac{R}{R+R_C} & -\frac{1}{C_o}\frac{1}{R+R_C} \end{pmatrix} \tag{11}$$

$$B = \begin{pmatrix} \frac{1}{L_f} & -\frac{1}{L_f}\frac{RR_C}{R+R_C} \\ 0 & -\frac{1}{C_o}\frac{R}{R+R_C} \end{pmatrix} \tag{12}$$

$$C = \begin{pmatrix} 1 & 0 \\ \frac{RR_C}{R+R_C} & \frac{R}{R+R_C} \end{pmatrix} \tag{13}$$

$$E = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{RR_C}{R+R_C} \end{pmatrix} \tag{14}$$

where L_f represents the output filter inductor, C_o the output capacitance, R_C the ESR of the output capacitor and R the load resistance (Fig. 2).

B. Model Discretization

The operation of the RTS consists in sampling the input variables at the beginning of each time step and estimating, with some latency, the value of the state and output variables for the next time step based on their current value and the value of the sampled inputs. For SMPS it is very common to use discrete PWM (DPWM) generators that rely on high frequency clock sources to generate high resolution PWM outputs. In such a case it is most likely that the RTS time step has a greater value than the clock period of the DPWM generator. The sampling action performed by the RTS reduces therefore the effective resolution of the PWM signal "seen" by the RTS. This further reduces the precision of the model but may also lead to the presence of limit

cycling when the RTS is part of a closed loop (HIL simulation). While for time step values approximately 100 times smaller than the switching period the resulted precision is acceptable [9], any larger values of the time step in comparison with the DPWM clock period may lead to false limit cycling conditions. The effort of the discretization process is therefore concentrated on minimizing the time step of the RTS. As shown in Fig. 4 it is also essential that the switching period is a multiple of the time step value. Steady state perturbations may occur otherwise.

Besides cost reduction, portability and low resource demand, one of the reasons this paper presents an alternate method from commercial real time simulators available in industry is the reduced time step obtained, essential in modeling the closed loop behavior of the RTS. As demonstrated by the experimental results, a time step in the range of hundreds of ns achievable with tools provided by OPAL-RT [29] or MATLAB [28] may lead to inexact behavior of the RTS when used in HIL simulations.

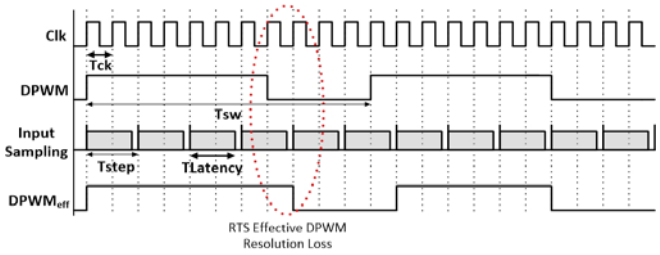

Figure 4. RTS timing related errors

The solution of the ODE system described by (6) and (7) obtained with the 1st order Adams-Moulton (Backward Euler) discretization method is represented by:

$$x_{(n+1)} = A_{BE}\,x_{(n)} + B_{BE}\,v_{(n)} \tag{15}$$

where

$$A_{BE} = \begin{pmatrix} 1-T_{step}A_{11} & -T_{step}A_{12} \\ -T_{step}A_{21} & 1-T_{step}A_{22} \end{pmatrix}^{-1} \tag{16}$$

$$B_{BE} = T_{step}B\begin{pmatrix} 1-T_{step}A_{11} & -T_{step}A_{12} \\ -T_{step}A_{21} & 1-T_{step}A_{22} \end{pmatrix}^{-1} \tag{17}$$

and T_{step} represents the value of the time step. The filter inductor current, the output capacitor voltage and the output voltage can be finally expressed as:

$$\begin{aligned} i_{Lf(n+1)} &= A_{BE_{11}}i_{Lf(n)} + A_{BE_{12}}v_{c(n)} + \\ &+ B_{BE_{11}}v_{sw\_out(n)} + B_{BE_{12}}i_{Ld(n)} \end{aligned} \tag{18}$$

$$\begin{aligned} v_{c(n+1)} &= A_{BE_{21}}i_{Lf(n)} + A_{BE_{22}}v_{c(n)} + \\ &+ B_{21}v_{sw\_out(n)} + B_{22}i_{Ld(n)} \end{aligned} \tag{19}$$

$$v_{o(n)} = C_{21}i_{Lf(n)} + C_{22}v_{c(n)} + E_{22}i_{Ld(n)} \tag{20}$$

The value of each state variable at time step n+1 will depend on the values of the state variables at time step n, on the circuit inputs (input voltage and load current), on the time step value and on the coefficient matrices (A, B, C, E). By applying the Backward Euler method to (2), the discrete input of the equivalent model can be obtained:

$$v_{in\_equiv(n+1)} = nV_{IN} - A_{VIN}\left(i_{Lf(n+1)} - i_{Lf(n)}\right) \tag{21}$$

where:

$$A_{VIN} = \frac{L_{1k}}{T_{step}}n^2 \tag{22}$$

The stability properties of the Backward Euler Method will make the resulting RTS A-stable [35 p 5.54]

C. Control Loop Design and Limit Cycling Analysis

A voltage mode control scheme will be used to close the loop for the HIL simulation. As explained in [30, p 116], when sampling the output voltage of a Buck derived topology, the small aliasing approximation is satisfied and the averaged converter model can be used to compute the parameters of the compensator. A detailed analysis of the small signal model of the PS converter is presented in [34]. A parallel PID structure with the discrete transfer function described by (23) will be used to implement the compensator.

$$G_{PID}(z) = Kp + \frac{K_i}{1-z^{-1}} + Kd\left(1-z^{-1}\right) \tag{23}$$

The compensator design follows the procedure described in [30, pp 165-217]. When designing the compensator for a digital controlled power supply, besides obtaining the desired phase margin, another concern is eliminating limit cycling. Limit cycling is a steady state disturbance that may affect the accuracy and the performance of the converter [36]. This effect is caused by the time and amplitude quantization introduced by the ADC and the DPWM generator. All subsequent references to the DPWM signal properties will refer to the equivalent DPWM of the PS converter (*PWMequiv*). The ADC divides the output voltage in quantization bins ($q_{vo(AD)}$) which can be expressed as:

$$q_{vo(AD)} = \frac{V_{FS\_ADC}}{G_{Vout}\,2^{N_{ADC}}} \tag{24}$$

where V_{FS_ADC} is the full scale voltage of the ADC, N_{ADC} is the number of resolution bits of the ADC and G_{Vout} is the gain of the voltage sensor. The variation of the output voltage produced by the minimum variation of the duty cycle is labeled $q_{vo(DPWM)}$. For the PS converter, assuming that the DPWM generator implements a digital ramp with a maximum value equal to a power of 2 and with a unitary increment, $q_{vo(DPWM)}$ is expressed as:

$$q_{vo(DPWM)} = \frac{V_{in\_equiv}}{2^{N_{DPWM}}} \tag{25}$$

where N_{DPWM} represents the number of resolution bits of the DPWM generator and V_{in_equiv} is the steady state value of v_{in_equiv} during *Ton*. The no limit cycling condition is synthetized by [30, pp 227-235]:

$$q_{vo(AD)} > q_{vo(DPWM)} \tag{26}$$

For the PS converter (Buck) topology, the minimum number of DPWM resolution bits can be determined as:

$$N_{DPWM\,\min} > \frac{V_{in\_equiv}G_{Vout}\,2^{N_{ADC}}}{V_{FS\_ADC}} \tag{27}$$

A distinction between the DPWM resolution and the DPWM resolution "seen" by the RTS ($DPWM_{eff}$) needs to be made. The RTS asses the state of the switches for each

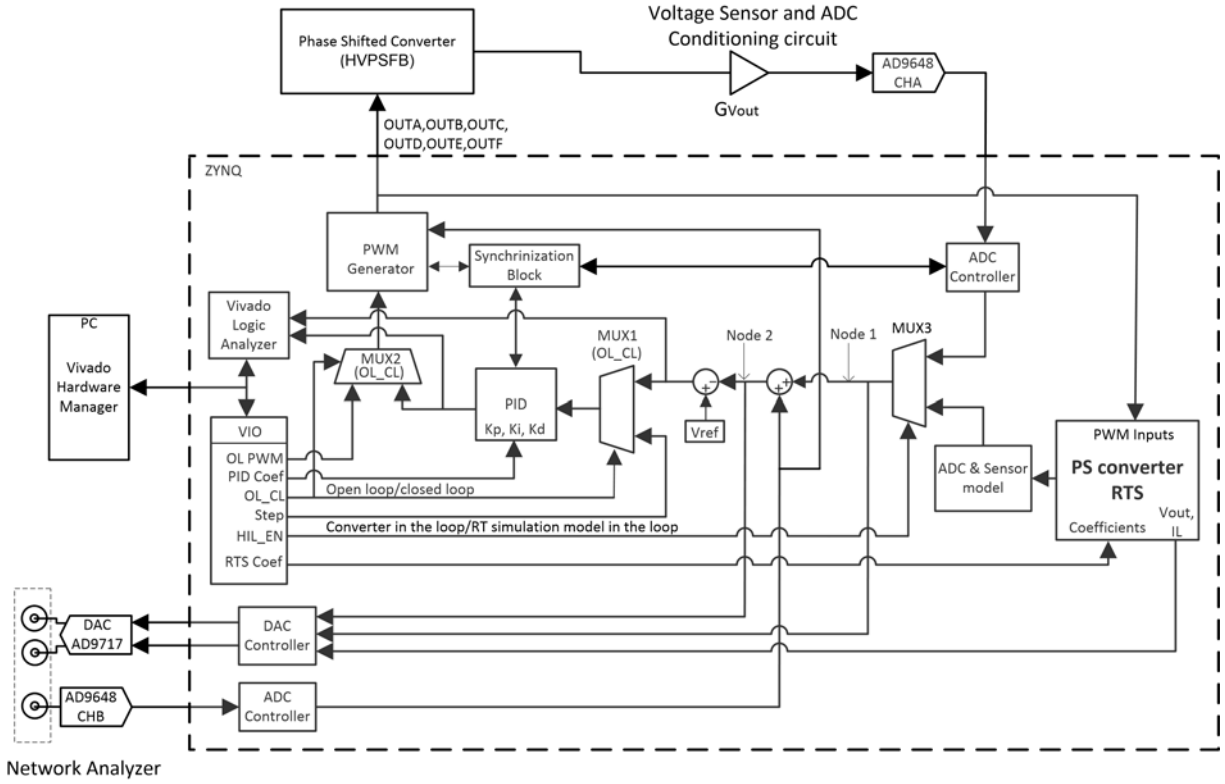

Figure 5. RTS validation hardware platform

time step, based on the gate drive inputs, altering the DPWM real resolution. The effective DPWM resolution becomes:

$$N_{DPWMeff} = \log_2\left(\frac{T_{sw}}{T_{step}}\right) \tag{28}$$

where T_{sw} represents the period of the DPWM. Therefore, the time step has a critical value beyond which the RTS can no longer model the behavior of the loop accurately. If $N_{DPWMeff}$ will be smaller than $N_{DPWMmin}$, a false limit cycling condition can occur. Limit cycling can also be caused by high values of the integrative coefficient of the PID controller, but this constraint is assumed to be satisfied.

IV. Validation Platform and RTS Implementation

A. Hardware Platform Design

The main purpose of the platform (illustrated as block diagram in Fig. 1) is to allow the RTS to run in parallel with the reference converter, both being driven by the same gate drive control signals. This approach allows easily measuring and comparing the outputs of the two blocks. A more detailed diagram of the hardware platform used to validate the RTS of the PS converter is presented in Fig. 5.

The features of the hardware include: a flexible architecture that enables running the converter and the RTS in open loop or closed loop, the capability of running the reference converter in parallel with the RTS for a comparative time domain analysis and the capability of injecting a perturbation in order to perform a comparative frequency domain analysis with the loop open, case in which the frequency response of the PS converter/RTS is measured or with the loop closed, case in which the loop gain is measured.

The main elements that compose the hardware platform are the PS converter, a 2 channel ADC, a 2 channel DAC and a XC7Z020 Zynq hybrid device. The Zynq is divided in the ARM Cortex A-9 based processing system part and the programmable logic part. Tasks that do not have demanding timing requirements can run on the processing system, taking advantage of the rich set of peripherals that facilitate system integration and the ease of use associated with conventional programming languages, while functionalities that require low latency and high throughput will run on the FPGA.

The FPGA configuration consists of a DPWM Generator block that generates the specific gate drive signals of the PS converter, the RTS of the PS converter, the ADC and voltage sensor model, a PID controller, a Synchronization block that manages event timing, a Vivado Logic Analyzer (ILA) IP and a Virtual Input Output Port (VIO) IP. The ILA IP is a logic analyzer provided by Xilinx that connects to the Vivado development environment over a JTAG interface and enables monitoring all critical signals in the design [37]. The VIO, also an IP provided by Xilinx, enables modifying defined signals from the Vivado environment at run time [38].

The VIO eliminates the need of reprogramming the FPGA with a new bitstream for each new set of coefficients or for each control scheme modification, significantly accelerating the validation process. Through the multiplexer control signals that it drives, the VIO can configure the circuit to work in closed loop or in open loop. Furthermore, the control loop can be closed either through the PS converter or through the RTS. The drawback of this approach is the large number of coefficients that need to be manually introduced in the VIO interface.

Instead of using a VIO, the hybrid nature of the Zynq can be exploited. The RTS and the PID coefficients can be computed by the software running on the processing system based on the parameters of the circuit and passed to the

FPGA through an AXI Lite interface. The whole system configuration can be managed by software.

The implemented system parameters are listed in Table I.

TABLE I. HARDWARE PLATFORM PARAMETERS

Parameter	Symbol	Value
Filter inductor	L_f	2 µH
Filter Capacitor	C_o	1500 µF
Capacitor ESR	R_C	6 mΩ
Equivalent inductor ESR	R_{L_equiv}	100 mΩ
Resonant Inductor	L_{lk}	26 µH
Transformer ratio	n	0.048
Closed loop nominal output voltage	V_o	5 V
Input voltage	V_{in}	200 V
Equivalent Switching frequency	F_{sw}	200 kHz
System clock frequency	T_{sys_clk}	100 MHz
Inductor current sensor gain	G_{cs}	0.559
Output voltage sensor gain	G_{Vout}	0.062
ADC and DAC number of bits	$N_{ADC/DAC}$	14
ADC full scale voltage	V_{FS_ADC}	1 V
ADC delay	$T_{ADCdelay}$	170 ns
ADC and DAC sampling rate	ADC_{SR}	100 MSPS
DAC and output filter gain	G_{DAC}	1.327e-3 V
Equivalent DPWM resolution bits	N_{DPWM}	10
Voltage loop PID proportional coefficient	K_P	2.659
Voltage loop PID integrative coefficient	K_i	0.248
Voltage loop PID derivative coefficient	K_d	0
PS converter RTS Latency	$T_{RTS\_Latency}$	20 ns
PID Latency	$T_{PID\_Latency}$	60 ns
ADC model multiplier latency	$T_{Mult\_Latency}$	30 ns

B. RTS Model Implementation

An implementation of the ODE system discrete solver of the PS converter using generic digital structures that can be easily implemented in MATLAB, Plecs or described in the C programming language is presented in Fig. 6.

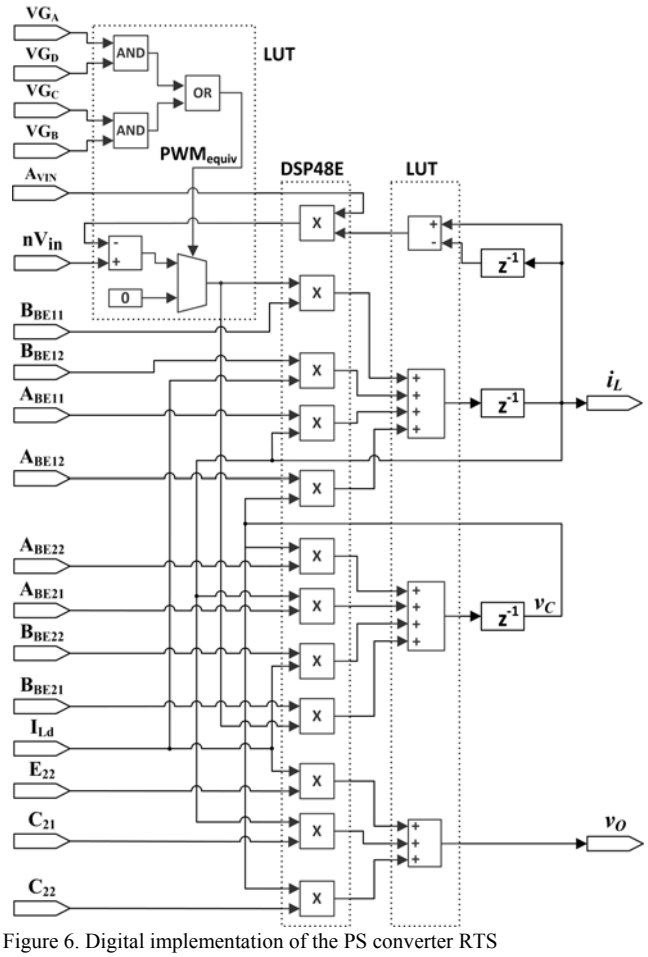

Figure 6. Digital implementation of the PS converter RTS

The digital structures used to compute the filter inductor current and the output voltage are designed based on (18), (19) and (20), while the input voltage of the model, as expressed in (21), is computed statically as $V_{in}n$. Since an FPGA implementation is targeted, the hardware primitives used for implementation are also indicated in Fig. 6.

As mentioned in Section II, the tool of choice for implementing the RTS of the PS converter was Vivado HLS. The C language that Vivado HLS accepts for the input files represents a natural way of describing arithmetical operations, logical operations, registers or the conditional elements used to model the switch network. The multiplexers used in Fig. 6 as conditional elements are replaced in the C description by if statements. Multiplications and part of the additions/subtractions are implemented in the FPGA's DSP48E dedicated primitives obtaining an excellent performance.

The FPGA utilization report for the XC7Z020 used for the experimental setup is summarized in Table II, while the timing performance of different Xilinx FPGAs is summarized in Table III.

TABLE II. XC7Z020 FPGA RESOURCE UTILIZATION

BRAM_18K	DSP48E	Flip Flop	LUT
0 (0%)	12 (5%)	597 (≈0%)	675 (1%)

TABLE III. FPGA PERFORMANCE COMPARISON

FPGA	Latency	Interval	$N_{DPWMeff}$
XC7Z020clg484-1 (Zynq)	30 ns	20 ns	8
XC7K325tfbg900-2(Kintex)	25 ns	10 ns	9
Xcvu3p-ffvc1517-2-i (Virtex Ultrascale +)	7.5 ns	5 ns	10

The latency is defined as the number of clock cycles the IP core requires between registering data on its inputs and producing valid data on its outputs. The interval represents the number of clock cycles required before the IP core can accept new data on its input (and produce valid data on its outputs). If operations can be pipelined it is possible to obtain a smaller value for the interval than for the latency. The latency needs to be considered in the control loop since it adds to the delay introduced by the PID controller, the ADC model, the sensor model and the DPWM Generator. The interval parameter will dictate the rate at which the inputs, including the gate drive signals, will be sampled by the RTS. Therefore, it is the interval parameter that defines the time step value. The number of resolution bits of the DPWM "seen" by the model is expressed as follows:

$$N_{DPWMeff} = \log_2\left(\frac{T_{sw}}{Interval}\right) \tag{29}$$

Effects such as limit cycling for DPWM generators with resolutions beyond $N_{DPWMeff}$ will not be accurately modeled. The values of $N_{DPWMeff}$ in Table III are computed for a switching frequency of 200 kHz.

C. Signal Level HIL Simulation

In order to demonstrate one of the use cases of the RTS, a signal level HIL simulation was developed to test the control circuit of the PS converter. MUX3 (Fig. 5) is the element that switches the system between HIL operation and real hardware control. As shown in Fig. 5, all the loop elements are developed on the same platform. The external elements that need to be modeled are the PS converter, the ADC and

the voltage sensor. The main parameters of the ADC that influence the control loop behavior are the ADC acquisition and conversion delays (T_{ADC_delay}), the quantization error, and the ADC gain. Therefore, the ADC is modeled as a gain block followed by a delay block. Since the output voltage sensor is ideally characterized only by its gain, the ADC gain and the sensor gain will be modeled together using only one multiplier block in order to minimize the FPGA resources used for implementation. The output of the multiplier will be saturated at $2^{N_{ADC}}$-1. The ADC quantization error is modeled by truncating the output of the RTS, only the most significant N_{ADC} bits being used by the multiplier as input. The ADC parameters, the voltage sensor parameters and the latency of the HIL loop components are listed in Table I.

An external controller can also be tested with the RTS developed. In this case, the outputs of the RTS will be multiplied by the sensor gains of the emulated hardware and converted to an analog format by the DAC.

V. Simulation and Experimental Results

A. Simulation Validation

The first method proposed for the RTS validation relies on simulation means. Because the components of the reference PS converter and their parasitic elements have tolerances that may affect the high frequency characteristic significantly, the best way of assessing the accuracy of the RTS is to compare its behavior against the idealized PS converter, comparison that can be best performed using a simulation tool. The RTS schematic (Fig. 6) and a synchronous PS converter (Fig. 2 a)) with the circuit parameters specified in Table I are both implemented in Plecs. Both simulation blocks share common inputs. The time domain (steady state) and frequency domain comparative simulation results are presented in Fig. 7 and Fig. 8. The transient response corresponding to a phase shift increase from 0 to 90 degrees is also presented in Fig. 9.

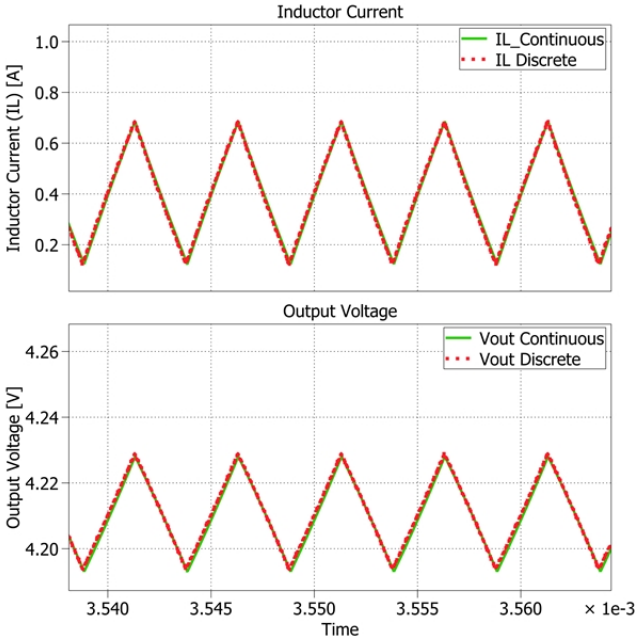

Figure 7. Time domain comparison between the continuous and the discretized models for the synchronous PS converter.

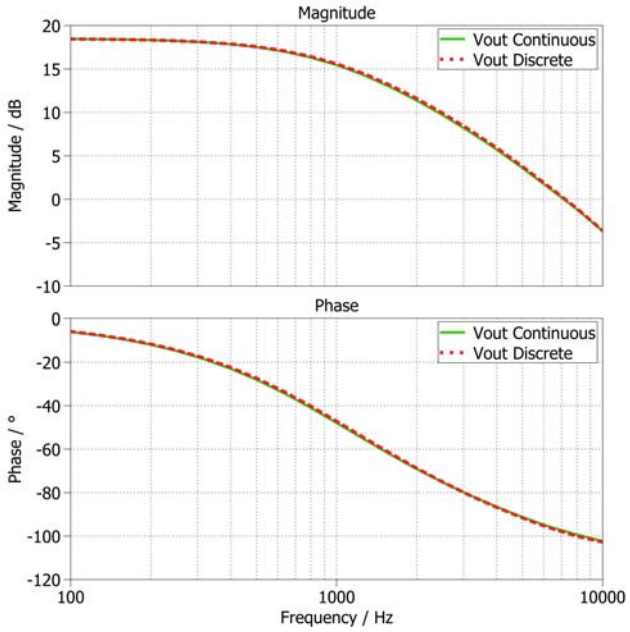

Figure 8. Frequency domain comparison between the continuous and the discretized models for the synchronous PS converter.

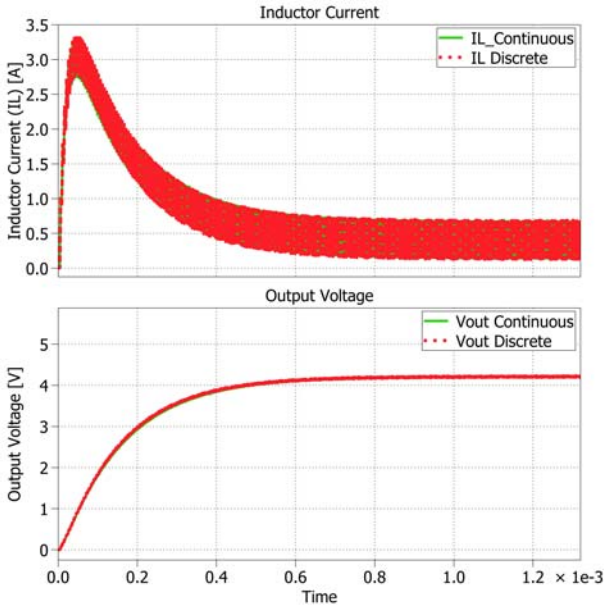

Figure 9. Transient response comparison between the continuous and the discretized models for the synchronous PS converter.

B. Experimental Validation

The following section describes in detail the configuration options applied to the schematic presented in Fig. 5 and the measurement procedures. For the first validation step proposed, the VIO IP core configures MUX2 to pass a constant value, also generated by the VIO (OL_PWM), as input to the DPWM generator. For the results presented in Fig. 10, a constant phase shift of 90 degrees between the two legs of the full bridge of the primary was generated. The DAC controller will output the emulated sensed inductor current and the emulated output voltage on the two available channels. The emulated output voltage is captured with an Analog Discovery oscilloscope on channel 1 (red, labeled as C1) while the sensed inductor current on channel2 (blue, labeled as C2). The purpose of this measurement is to perform a comparison between the simulation and the experimental results. The numeric values are presented in Table IV and include (in the case of the experimental

results) both the errors associated with the model discretization and the errors added by the digital to analog conversion.

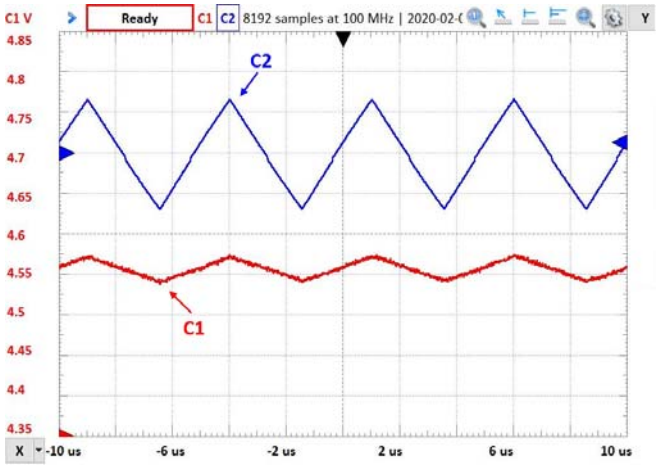

Figure 10. RTS output voltage and inductor current sensor output (horizontal scale: 5 µs/div, Channel1 vertical scale: 0.5 V/div, Channel2 vertical scale: 0.5 V/div, Channel1 offset: -0.5 V, Channel2 offset: 1.5 V).

TABLE IV. COMPARISON BETWEEN THE IDEAL REFERENCE CONVERTER SIMULATION AND RTS EXPERIMENTAL RESULTS

Measurement	Ideal Reference Converter Simulation Results [V]	Experimental Results [V]
Peak to peak output voltage	0.0352	0.0358
Peak to peak sensed inductor current	0.553	0.546
Output DC voltage	4.580	4.55
Sensed inductor current DC value	0	0.0031

In order to obtain a comparative view of the RTS and the PS converter (Fig. 11), the output voltage of the PS converter will first be captured and saved as reference (green, labeled as R1). The RTS will then be captured on channel 1 with the input supply of the PS converter powered down.

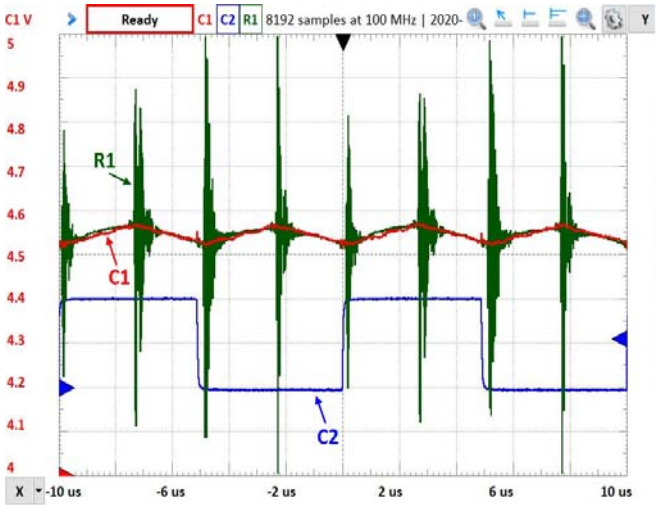

Figure 11. RTS and reference PS converter output voltage (horizontal scale: 2 µs/div, Channel 1 vertical scale: 0.1 V/div, Channel 2 vertical scale: 2 V/div, Reference 1 vertical scale: 0.1 V/div, Channel 1 offset: -4.5 V, Channel 2 offset: -3 V, Reference 1 offset: -4.5 V)

The output of the RTS can also be captured with the PS converter powered up, but the measurement would be affected by noise injected in the measurement system. The VG_B gate drive signal is used as trigger in order to assure that the reference channel and channel1 waveforms are synchronized.

For the second validation step the configuration of the multiplexers will remain unchanged. The Analog Discovery's network analyzer will be used to inject a perturbation on channel B of the ADC. The perturbation will be added to the constant value at the input of the DPWM generator. The output voltage frequency response of the reference PS converter will first be measured with the network analyzer and saved as reference after which the output voltage frequency response of the RTS will be added to the same plot. In order to limit the amplitude of the perturbed output voltage, the perturbation signal input will be divided by 2^5. This will result in a -30.103 dB offset added to the measured bode plots (Fig. 12).

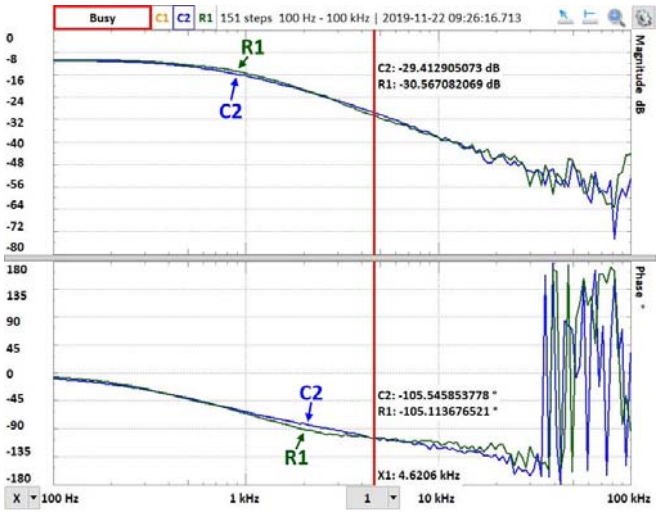

Figure 12. Synchronous PS converter (Reference 1) and RTS (Channel 2) output voltage frequency response

The hardware platform setup used for this validation step allows, optionally, the evaluation of the response of the PID controller at a Heaviside step excitation. The input of the PID controller can be driven by the VIO core through MUX1 while its outputs can be captured by the ILA IP core.

The third proposed validation step consists in the loop gain measurement. For this purpose, the circuit will be configured to run in closed loop. MUX3 will decide if the loop either closes through the PS converter (reference measurement) or through the RTS block. MUX 1 will be configured to pass the output of MUX3 to the PID controller while MUX2 will select the output of the PID controller as an input for the DPWM generator. ADC channel B will add a perturbation to the output of MUX3 while the node before the injection point (Node1 in Fig. 5) and after the injection point (Node2 in Fig. 5) will be converted into an analog format by the DAC. The open loop transfer function gain and phase margin of the system can be obtained by plotting the ratio between Node 2 and Node 1. This method of obtaining the loop gain is a digital implementation of the conventional method of performing this measurement which requires adding an injection transformer, an extra resistor in the feedback network and costly equipment [39]. The first loop gain measurement is performed with the loop closed through the PS converter and saved as reference by the network analyzer. A second measurement is then performed

with the loop closed through the RTS and plotted on channel 2 of the oscilloscope. The comparative result is presented in Fig. 13.

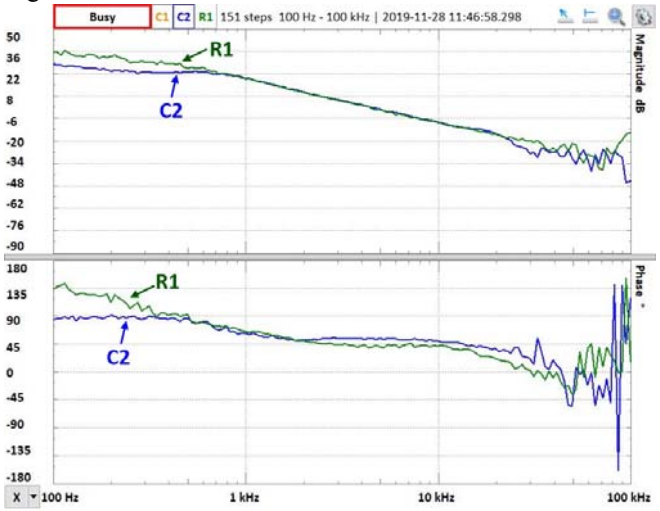

Figure 13. Loop gain measured with the loop closed through the synchronous PS converter (Reference 1) and through the RTS (Channel 2)

For demonstrating the ability of the PS converter RTS to model limit cycling and to point out the limitations imposed by the time step, three setups are proposed. The ADC and DPWM generator parameters for each setup are listed in Table V.

TABLE V. LIMIT CYCLING ANALYSIS SETUP

Setup number	ADC resolution bits	DPWM resolution bits	Effective DPWM resolution bits	No limit cycling condition
1	8	6	6	Not Satisfied
2	8	8	8	Satisfied
3	9	10	8	Satisfied

For the first experiment, the DPWM resolution is reduced so that limit cycling is expected to occur for the selected ADC resolution. For the second experiment, the DPWM and ADC resolution are selected so that the no limit cycling condition is met. In the third experiment the DPWM and ADC resolution are also selected so that the no limit cycling condition is met. However, in this case, the DPWM effective resolution is smaller than the real DPWM resolution resulting in a false limit cycling condition.

In order to synchronize the error signal (which is computed and exported as an output by the PID controller block) with the output voltage, the error signal is converted into an analog format. By multiplying the error, which is represented as an integer, with the inverse of the DAC gain, channel B of the DAC will output the value of the error in volts.

For the waveforms presented form Fig. 14 to Fig. 18 channel 1 represents the output voltage of the RTS/PS converter, while channel 2 represents the error signal in analog format. The results obtained for setup 1 (Fig. 14 and Fig. 15) prove that for DPWM resolutions below the maximum effective DPWM resolution the RTS correctly models the limit cycling effect when the no limit cycling condition is not met. The results for setup 2 (Fig. 16) prove that for DPWM resolutions below the maximum effective DPWM resolution the RTS behaves as expected when the no limit cycling condition is met, the error being stabilized to 0. The results of setup 3 (Fig. 17 and Fig. 18) shows that for DPWM resolutions beyond the maximum effective DPWM resolution, false limit cycling may occur. This highlights the importance of the time step value for obtaining accurate results with the HIL simulation.

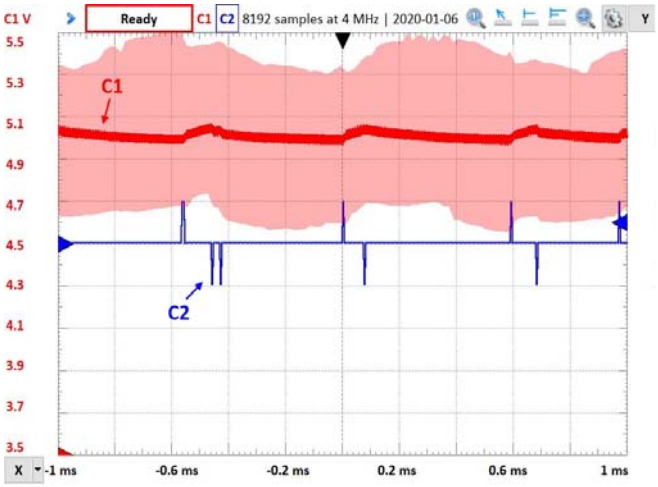

Figure 14. Setup1 running on the reference PS converter (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.2 V/div, Channel2 vertical scale: 1 V/div, Channel1 offset: -4.5 V, Channel2 offset: 0 V)

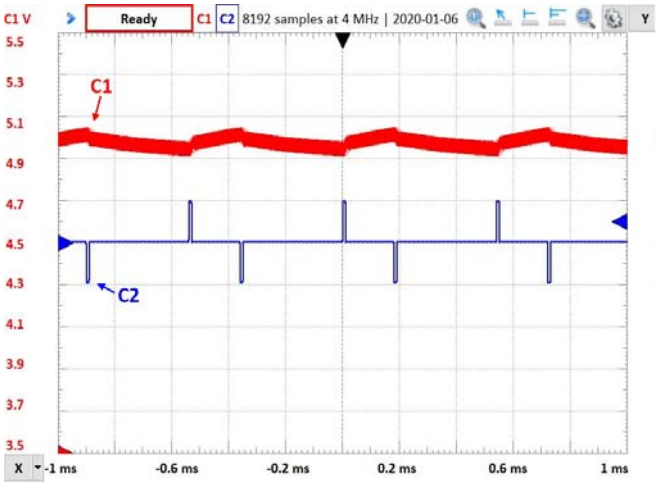

Figure 15. Setup1 running on the RTS (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.2 V/div, Channel2 vertical scale: 1 V/div, Channel1 offset: -4.5 V, Channel2 offset: 0 V)

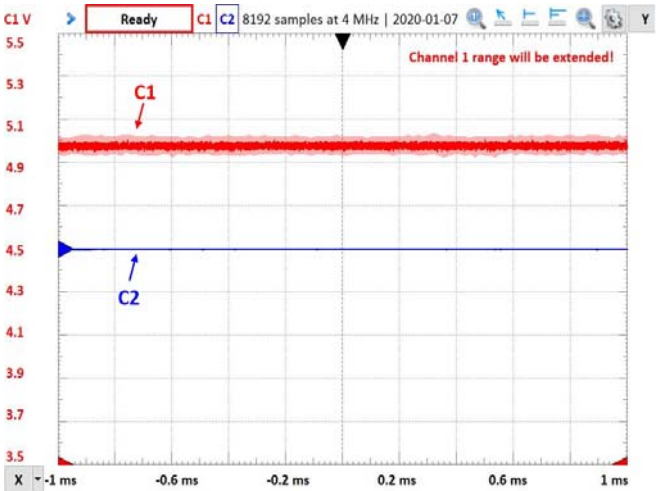

Figure 16. Setup2 running on the RTS (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.2 V/div, Channel2 vertical scale: 1 V/div, Channel1 offset: -4.5 V, Channel2 offset: 0 V)

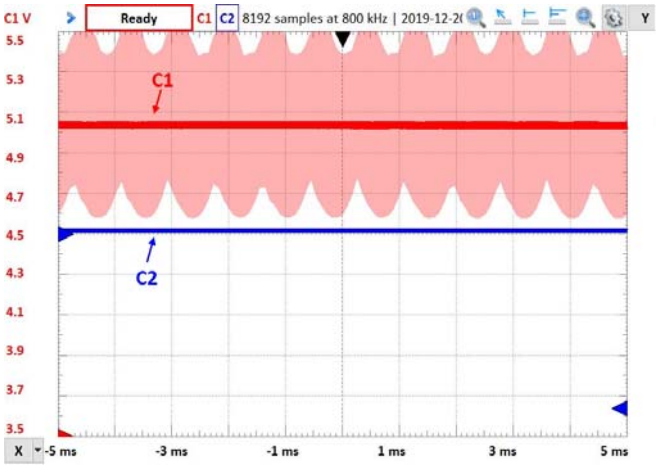

Figure 17. Setup3 running on the reference PS converter (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.2 V/div, Channel2 vertical scale: 1 V/div, Channel1 offset: -4.5 V, Channel2 offset: 0 V)

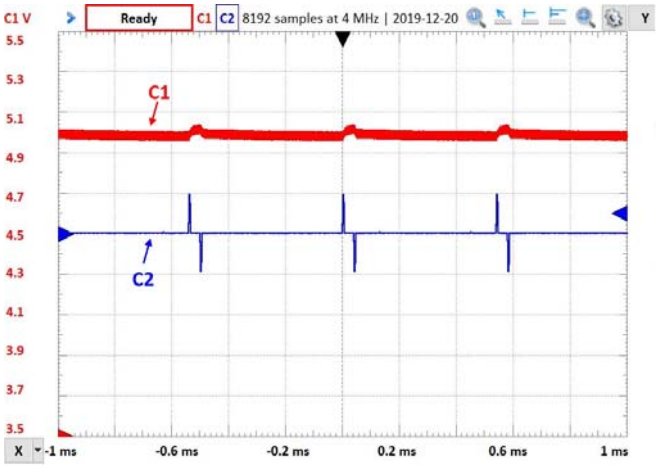

Figure 18. Setup3 running on the RTS (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.2 V/div, Channel2 vertical scale: 1 V/div, Channel1 offset: -4.5 V, Channel2 offset: 0 V)

The final validation step proposed consists in generating a step variation on the reference signal. For this setup, the loop is closed through the reference converter with the outputs of the reference converter and the RTS measured simultaneously.

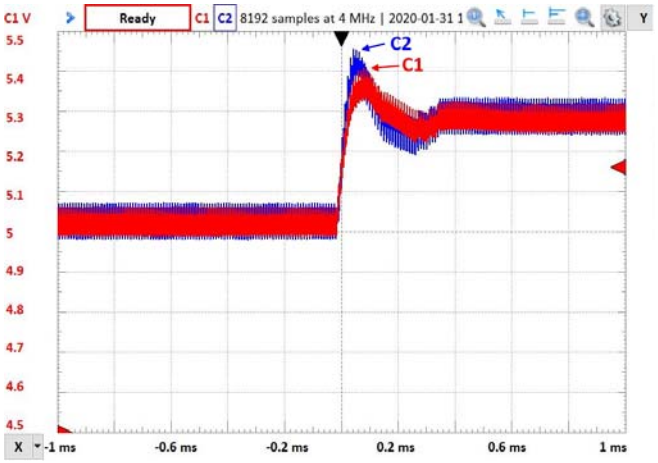

Figure 19. Reference step variation response (horizontal scale: 200 µs/div, Channel1 vertical scale: 0.1 V/div, Channel2 vertical scale: 0.1 V/div, Channel1 offset: -5 V, Channel2 offset: -5 V)

The reference variation can be generated by the VIO IP or, alternatively, by the software running on the processing system. The responses of the reference converter (channel 1) and of the RTS (channel 2) are illustrated in Fig. 19. Component aging effects or hardware faults can be evaluated based on analyzing the difference between the outputs of the two blocks.

VI. Conclusion

In this paper, a method for designing and implementing a RTS model of a synchronous PS converter was presented. Simulation and experimental results prove that the switched function model developed is capable of accurately modeling the output ripple of the converter. A very low time step and minimized resource utilization were obtained with a small FPGA, the RTS proving to be well adapted to low cost embedded platform implementations either for HIL simulations or for error detection or monitoring applications. The 20ns time step obtained with the XC7Z020 Zynq device can be improved to 5ns by using a higher end FPGA. Although the RTS is able to model the output ripple of the reference converter with time steps much larger than the ones obtained, in order to precisely model the control loop behavior, the time step has to be minimized as much as possible, otherwise false limit cycling can occur.

To sum up, the novelty of this paper is represented by the following: i) the development of an RTS for the PS topology; ii) the very low time step (20ns) obtained considering that the RTS also accounts for inductor and capacitor ESRs; iii) the discretized model is generic and can be easily described using conventional hardware description languages, C, or graphical description languages; iv) the RTS is highly portable with compatible target platforms including CPUs, GPUs but, for maximized performance, FPGA implementation is required; v) the flexible architecture and the manner in which the analog to digital interface in conjunction with the FPGA configuration enable system characterization methods necessary to assess the accuracy of the RTS model; vi) the study of the impact that the time step has on the ability to model limit cycling; vii) the low level FPGA resources required that make the RTS ideal for both HIL simulation or for health monitoring applications.

The high level synthesis tool (Vivado HLS) chosen for the FPGA implementation has contributed to accelerating development and helped minimizing the time step obtained, proving to be an optimal compromise between cost, design effort, flexibility and performance. The innovative architecture of the hardware platform used for validation takes advantage of the FPGA capabilities and allows modifying the architecture of the system and parameters at run time, enabling a comprehensive comparative analysis between the RTS and the PS converter. As a final major advantage – this solution is fully scalable: the digital platform designed, alongside the methods presented, can be used to successfully develop RTS models for a wide range of SMPS topologies.

References

[1] E. Perez and J. de la Ree, “Development of a real time simulator based on ATP-EMTP and sampled values of IEC61850-9-2,” Int. J. Electr. Power Energy Syst., vol. 83, pp. 594–600, 2016. doi:10.1016/j.ijepes.2016.06.001

[2] B. Li, H. Zhao, S. Gao, and S. Hu, “Digital real-time co-simulation platform of refined wind energy conversion system,” Electr. Power

Energy Syst., no. July, p. 105676, 2019. doi:10.1016/j.ijepes.2019.105676

[3] H. Bounechba, A. Bouzid, H. Snani, and A. Lashab, "Real time simulation of MPPT algorithms for PV energy system," Int. J. Electr. Power Energy Syst., vol. 83, pp. 67–78, 2016. doi:10.1016/j.ijepes.2016.03.041

[4] X. Luo, Z. Zhong, and Y. Xiong, "A HIL Test Bench for FCHV Control Units," 2009 IEEE Veh. Power Propuls. Conf., pp. 1783–1787, 2009. doi: 10.1109/VPPC.2009.5289665.

[5] O. Craciun, A. Florescu, I. Munteanu, I. A. Bratcu, S. Bacha, and D. Radu, "Hardware-in-the-loop simulation applied to protection devices testing," vol. 54, pp. 55–64, 2014. doi: 10.1016/j.ijepes.2013.06.031

[6] A. Ben Hadid and K. Ben Saad, "HIL simulation of a DC-DC converter controller on a Zynq," in 2015 7th International Conference on Modelling, Identification and Control (ICMIC), 2015, pp. 1–5. doi: 10.1109/ICMIC.2015.7409495.

[7] C. Washington and S. Delgado, "Improve Design Efficiiency and Test Capabilities with HIL Simulation," in 2008 IEEE AUTOTESTCON, 2008, pp. 539–549. doi:10.1109/AUTEST.2008.4662686.

[8] M. Yilmaz and P. T. Krein, "Review of charging power levels and infrastructure for plug-in electric and hybrid vehicles," IEEE Trans. Power Electron., vol. 28, no. 5, pp. 2151–2169, 2013. doi: 10.1109/IEVC.2012.6183208.

[9] M. Dagbagi, A. Hemdani, L. Idkhajine, M. W. Naouar, and I. Slama-Belkhodja, "ADC-Based Embedded Real-Time Simulator of a Power Converter Implemented in a Low-Cost FPGA : Application to a Fault-Tolerant Control of a Grid-Connected," IEEE Trans. Ind. Electron., vol. 63, no. 2, pp. 1179–1190, 2016. doi: 10.1109/TIE.2015.2491883.

[10] Y. Yonezawa, H. Nakao, and Y. Nakashima, "Novel Hardware-in-the-Loop Simulation (HILS) Technology for Virtual Testing of a Power Supply," 2018 IEEE Appl. Power Electron. Conf. Expo., pp. 2947–2951, 2018. doi: 10.1109/APEC.2018.8341438.

[11] L. Bao, L. Fan, and Z. Miao, "Real-Time Simulation of Electric Vehicle Battery Charging Systems," 2018 North Am. Power Symp., pp. 1–6, 2018. doi: 10.1109/NAPS.2018.8600543.

[12] E. Chai, I. Celanovic, and J. Poon, "Validation of Frequency- and Time-domain Fidelity of an Ultra-low Latency Hardware-in-the-Loop (HIL) Emulator," 2013 IEEE 14th Work. Control Model. Power Electron., pp. 1–5, 2013. doi: 10.1109/COMPEL.2013.6626402.

[13] C. Liu, R. Ma, H. Bai, F. Gechter, and F. Gao, "A new approach for FPGA-based real-time simulation of power electronic system with no simulation latency in subsystem partitioning," Electr. Power Energy Syst., vol. 99, no. August 2017, pp. 650–658, 2018. doi:10.1016/j.ijepes.2018.01.053

[14] M. Matar and R. Iravani, "FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients," vol. 25, no. 2, pp. 852–860, 2010. doi:10.1109/TPWRD.2009.2033603.

[15] A. Myaing and V. Dinavahi, "FPGA-Based Real-Time Emulation of Power Electronic Systems with Detailed Representation of Device Characteristics," IEEE Trans. Ind. Electron., vol. 58, no. 1, pp. 358–368, 2011. doi:10.1109/TIE.2010.2044738.

[16] Ó. Jiménez et al., "Implementation of an FPGA-Based Online Hardware-in-the-Loop Emulator Using High-Level Synthesis Tools for Resonant Power Converters Applied to Induction Heating Appliances," vol. 62, no. 4, pp. 2206–2214, 2015. doi:10.1109/TIE.2014.2360138.

[17] F. Montano, T. Ould-bachir, and J. P. David, "An Evaluation of a High-Level Synthesis Approach to the FPGA-Based Submicrosecond Real-Time Simulation of Power Converters," IEEE Trans. Ind. Electron., vol. 65, no. 1, pp. 636–644, 2018. doi:10.1109/TIE.2017.2716880.

[18] S. Lucia, D. Navarro, O. Lucia, P. Zometa, and R. Findeisen, "Optimized FPGA Implementation of Model Predictive Control for Embedded Systems Using High-Level Synthesis Tool," IEEE Trans. Ind. Informatics, vol. 14, no. 1, pp. 137–145, 2018. doi:10.1109/TII.2017.2719940.

[19] D. Tormo, L. Idkhajine, E. Monmasson and R. Blasco-Gimenez, "Evaluation of SoC-based embedded Real-Time simulators for electromechanical systems," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 4772-4777, doi:10.1109/IECON.2016.7793185.

[20] D. Tormo, Vidal-Albalate, L. Idkhajine, E. Monmasson, and R. Blasco-Gimenez, "Modular Multi-level Converter Hardware-in-the-Loop Simulation on low-cost System-on-Chip devices," IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc., vol. 1, pp. 2827–2832, 2018. doi:10.1109/IECON.2018.8591256.

[21] D. Tormo, R. Vidal-Albalate, L. Idkhajine, E. Monmasson, and R. Blasco-Gimenez, "Study of System-on-Chip Devices to Implement Embedded Real-Time Simulators of Modular Multi-Level Converters Using High-Level Synthesis Tools," 2018 IEEE Int. Conf. Ind. Technol., pp. 1447–1452. doi:10.1109/ICIT.2018.8352393.

[22] T. Gherman, D. Petreus and R. Teodorescu, "A Real Time Simulator of a PEV's On Board Battery Charger," 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 2019, pp. 329-335, doi: 10.1109/ACEMP-OPTIM44294.2019.9007152.

[23] T. Gherman, D. Petreus and R. Teodorescu, "A Method for Accelerating FPGA Based Digital Control of Switched Mode Power Supplies," 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 2019, pp. 322-328, doi:10.1109/ACEMP-OPTIM44294.2019.9007156

[24] J. S. Shen and P. A. Hsiung, "Dynamic Reconfigurable Network-on-chip Design: Innovations for Computational Processing and Communications". New York: IGI Global publisher, 2010, p136. doi: 10.4018/978-1-61520-807-4

[25] M. Lattuada and F. Ferrandi, "A Design Flow Engine for the Support of Customised Dynamic High Level Synthesis Flows," ACM Trans. Reconfigurable Technol. Syst., no. 12, pp. 1–26, 2019. doi:10.1145/3356475

[26] M. F. Dossis, "Formal ESL Synthesis for Control-Intensive Applications," Adv. Softw. Eng., vol. 2012, pp. 1–30, 2012. doi:10.1155/2012/156907

[27] D. K. Chaturvedi, "Modeling and Simulation of Systems using MATLAB® and Simulink". Taylor & Francis, CRC Press, 2010. doi:10.1201/9781315218335

[28] Power Electronics Real-time HIL Testing with FPGA Acceleration, Mathworks, Natick, MA, USA, 2019, p 31.

[29] K. Palaniappan, B. Seibel, M. Cook and C. Dufour, "Real Time Hardware-in-the-Loop Validation of Common Bus Inverter Low Voltage Drives," 2019 IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, CA, USA, 2019, pp. 2554-2558. doi: 10.1109/APEC.2019.8722148.

[30] L. Corradini, D. Maksimovic, P. Mattavelli, and R. Zane, "Digital Control of High-Frequency Switched Mode Power Converters", 1st ed. Wiley-IEEE Press, 2015, pp 89-93, p 116, pp 165-217, pp 227-235. doi: 10.1002/9781119025498

[31] H. Jin, "Behavior-mode simulation of power electronic circuits," IEEE Transactions on Power Electronics, vol. 12, no. 3, pp. 443-452, May 1997, doi: 10.1109/63.575672.

[32] T. H. Kim, S. J. Lee, and W. Choi, "Design and control of the phase shift full bridge converter for the on-board battery charger of electric forklifts," J. Power Electron., vol. 12, no. 1, pp. 113–119, 2012. doi:10.1109/ICPE.2011.5944761.

[33] H. Nene, "Phase Shifted Full Bridge CCS User Guide," Texas Instruments, March 2012.

[34] V. Vlatkovic, J. A. Sabatc, R. B. Ridley, F. C. Lee, and B. O. H. Cho, "Small-Signal Analysis of the Phase-Shifted PWM Converter," IEEE Trans. Power Electron., vol. 7, no. 1, pp. 128–135, 1992. doi:10.1109/63.124585.

[35] M. B. Patil, V. Ramanarayanan, V. T. Ranganathan, "Simulation of power electronic circuits". Oxford: Alpha Science International, 2009, p 5.54.

[36] A. V Peterchev and S. R. Sanders, "Quantization Resolution and Limit Cycling in Digitally Controlled PWM Converters," IEEE Trans. Power Electron., vol. 18, no. 2, pp. 301–308, 2003. doi: 10.1109/TPEL.2002.807092.

[37] PG172: Integrated Logic Analyzer LogiCORE, V6.2, Xilinx, San Jose, CA, USA, 2016.

[38] PG159: Virtual Input / Output LogiCORE, V3.0, Xilinx, San Jose, CA, USA, 2018.

[39] DC/DC Converter Stability, V3.3, OMICRON Lab, Houston, TX, USA, 2018.