Real-time emulation of boost inverter using the Systems Modeling Language and Petri nets

A. Gutierrez, Michael Bressan, Fernando Jimenez, Corinne Alonso

Real-time Emulation of Boost Inverter Using the Systems Modeling Language and Petri Nets

A. Gutierrez[1–2], M. Bressan[1], J.F. Jimenez[1–3], and C. Alonso[2–3]

[1] *Universidad de los Andes, Department of Electrical and Electronic Engineering, Bogotá, Colombia*

[2] *Université de Toulouse III, UPS; Toulouse F-31400, France*

[3] *LAAS-CNRS, 7 Avenue du Colonel Roche, Toulouse F-31077, France*

Email: a.gutierrez75@uniandes.edu.co (A. Gutierrez), alonsoc@laas.fr (C. Alonso)

Abstract

A boost inverter is a versatile architecture able to supply DC or AC outputs from diverse alternative energy sources. The most relevant advantage of this multipurpose converter is to provide boosting and inversion in a single stage. The boost inverter has been studied from design and control perspectives with renewable energy sources. However, the real-time emulation of boost inverters is less widespread in literature. In this context, the main contribution of this paper is proposing an innovative methodology based on the Systems Modeling Language and Petri nets to emulate in real-time power converters using the boost inverter as a case of study. This approach develops real-time Hardware-in-the-Loop models using a graphical language and Petri nets. These graphical and Petri net features allow a formal validation of computational and time constraints before implementation in FPGA. The proposed methodology is also able to transform the developed models to the High Level Specification of Embedded Systems for automatic code generation. Comparison of real-time emulations and experimental results shows a suitable trade-off between the model accuracy and the computational time.

Keywords: Boost inverter, Hardware-in-the-Loop, SysML, Petri net, real-time emulation, FPGA.

1. Introduction

Nowadays, alternative energy sources have increased their integration in power systems. As a result, innovative power converters are required to accomplish the power system conditions [1]. Modern power converters should be robust architectures able to interact with intermittent energy sources and electrical grids [2]. Currently, the boost inverter is a power converter able to achieve these requirements which can be tested using real-time emulations [3].

The boost inverter shows interesting characteristics because provide both boosting and inversion in a single stage [4]. These characteristics improve the power system efficiency [5], decrease the harmonic distortion [6], and increase the system reliability [7]. Authors in [8] introduce the boost inverter and propose a nonlinear control to take advantages of this converter. Also, further studies present advanced control strategies to improve the boost inverter performance as presented by [9][10][11]. Indeed, advanced controllers allow the integration between renewable energy sources and the boost inverter [12][13].

According to [14], the integration of boost inverters and renewable energy sources requires innovative modeling approaches. For instance, authors in [5] and [6] study the boost inverter interconnection with electrical grids and PV modules. According to [15], the boost inverter can be modeled and integrated in wind power systems. Other works also study the use of boost inverters in hybrid architectures based on batteries, supercapacitors, and fuel cells [16][17]. However, the boost inverter shows a highly nonlinear behavior and control complexity which leads to unusual industrial applications despite their remarkable advantages. As a consequence, the real-time emulation of this power converter provides an interesting alternative for the control system design to encourage their industrial application such as the current integration of renewable energies in power systems [18][19].

Real-time emulations provide several advantages for the study of power converters [20]. Analyses based on real-time emulations allow evaluating changes under different scenarios [21][22]. A relevant advantage of a real-time approach is to test the safety operation of power converters closer to working conditions using a Hardware-in-the-Loop method [23]. However, the real-time emulation of boost inverters is less widespread in the literature despite its powerful features.

In order to contribute in the fields of power electronics modeling and real-time emulation, this study proposes an innovative modeling approach using the graphical Systems Modeling Language (SysML)[24]. In this specific case, the boost inverter is modeled given its special characteristics and potential range of applications in current renewable energy systems. Nevertheless, the mathematical framework and the modeling methodology can also be extended to other power converters.

The Systems Modeling Language (SysML) is a general-purpose modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, and procedures [25][26]. SysML provides a systems engineering methodology able to model complex systems and useful to describe systems based on power converters and renewable energies [27]. This study takes advantages of powerful modeling possibilities of SysML to describe power electronics systems for further real-time emulation in Field Programmable Gate Arrays (FPGAs) [28][29]. The FPGA is selected as a processing device because of the characteristics of parallel processing, versatility, and adaptability [30][20].

The proposed methodology also employs the High Level Specification of Embedded Systems (HiLeS) to transform automatically the developed SysML models in Petri nets and VHDL code for model implementation in FPGA [31][32]. This study contributes with the use of Petri nets in two directions. First, the Petri nets are employed to verify the structural consistency of the developed SysML models. Second, the analysis of generated Petri nets allows identifying delays and defining the sequence of operations to achieve the time constraints for real-time emulation of power converters [33]. Fig.1 outlines the proposed modeling methodology through SysML and Petri nets.

Section II introduces the boost inverter operation and their modeling using a component-based approach. Section III presents the concepts, current state, and scope of real-time emulation methods for power converters. Section IV presents the SysML and the HiLeS - Petri nets framework to emulate in real-time power converters. Section V describes the experimental setup to validate the proposed approach. Finally, emulated and experimental results are discussed.

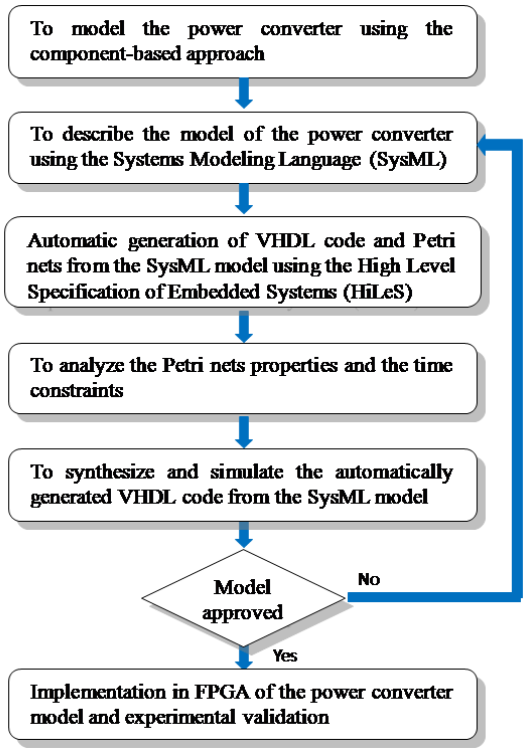


Figure 1: Modeling methodology based on SysML and Petri nets

2. Operation principle of boost inverter and modeling approach

This section introduces the operation principle and theoretical analysis of the boost inverter as a study case for further real-time emulation.

2.1. Operation principle

A boost inverter is a power converter build by two complementary bidirectional boost converters [4]. Fig.2 shows the operation principle of the boost converter. The output voltage of a boost inverter is given by eq.(1). When v_1 and v_2 are sinusoidal 180° phase-shifted and dc-biased signals, as shown in eq.(2) and eq.(3), the differential output of the boost inverter is given by eq.(4).

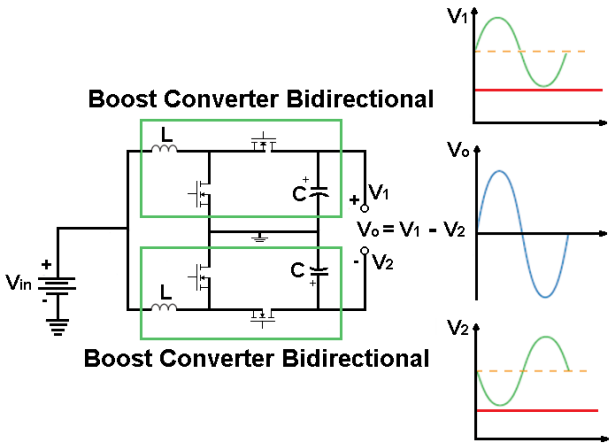


Figure 2: Operation principle of Boost inverter

$$v_o = v_1 - v_2 \tag{1}$$

$$v_1 = V_{dc} + \frac{1}{2} A_o Sin\left(\omega t\right) \tag{2}$$

$$v_2 = V_{dc} + \frac{1}{2} A_o Sin\left(\omega t - \pi\right) \tag{3}$$

$$v_o = A_o Sin\left(\omega t\right) \tag{4}$$

2.2. Gain analysis

The gain is the relation between the output and the input voltages of a power converter. The gain is one of the most important parameter for power converters. The ideal boost inverter gain is given by [4], where D is the duty cycle.

$$\frac{V_o}{V_{in}} = \frac{2D - 1}{D(1 - D)} \tag{5}$$

Taking into account the conductive losses, the boost inverter circuit is drawn in Fig.3. Where S_1 to S_4 are ideal switches, R_{on} represents the conductive losses of switching devices, and R_L are the conductive losses associated to the inductors [34].

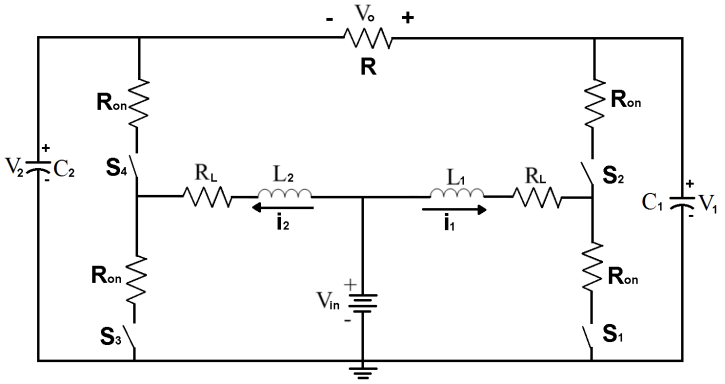


Figure 3: Boost inverter circuit with conductive losses

The boost inverter has two fundamental operation states. In the first operation state, $S_1 - S_4$ are active. In the second operation state, $S2 - S3$ are active. For the first operation state,

$$v_{L1} = v_{in} - i_1 \left(R_L + R_{on}\right) \tag{6}$$

$$v_{L2} = v_{in} - i_2 \left(R_L + R_{on}\right) - v_2 \tag{7}$$

$$i_{c1} = \frac{v_2 - v_1}{R} \ and \ i_{c2} = i_2 + \frac{v_1 - v_2}{R} \tag{8}$$

In the second operation state,

$$v_{L1} = v_{in} - i_1 \left(R_L + R_{on}\right) - v_2 \tag{9}$$

$$v_{L2} = v_{in} - i_2 \left(R_L + R_{on}\right) \tag{10}$$

$$i_{c1} = i_1 + \frac{v_2 - v_1}{R} \ and \ i_{c2} = \frac{v_1 - v_2}{R} \tag{11}$$

Applying the principle of inductor volt-second balance and the capacitor charge balance to equations eq.(6) to eq.(11) [35]. Where $R_s = R_L + R_{on}$ and $D_1 = 1 - D$,

$$\langle V_{L1} \rangle = D \left(V_{in} - I_1 \left(R_s\right)\right) + D' \left(V_{in} - I_1 \left(R_s\right) - V_1\right) = 0 \tag{12}$$

$$\langle V_{L2} \rangle = D\left(V_{in} - I_2\left(R_s\right) - V_2\right) + D'\left(V_{in} - I_2\left(R_s\right)\right) = 0 \tag{13}$$

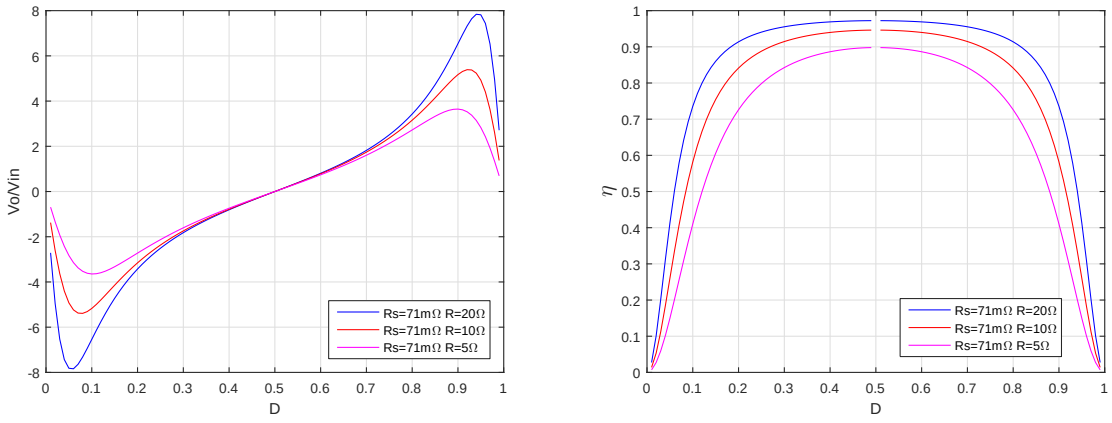
$$\langle I_{c2} \rangle = D\left(\frac{V_2 - V_1}{R}\right) + D'\left(I_1 + \frac{V_2 - V_1}{R}\right) = 0 \tag{14}$$

$$\langle I_{c2} \rangle = D\left(I_2 + \frac{V_1 - V_2}{R}\right) + D'\left(I_1 + \frac{V_1 - V_2}{R}\right) = 0 \tag{15}$$

Solving the system of equations from eq.(12) to eq.(15), the boost inverter gain is given by eq.(16) [34].

$$\frac{V_o}{V_{in}} = \frac{2D - 1}{\frac{R_s}{R}\left(\frac{1}{D'} + \frac{D'}{D} - 1\right) + DD'} \tag{16}$$

Fig. 4a shows the relationship between the output voltage and the input voltage given a series resistance R_s associated to conductive and inductance losses and a set of resistive loads R. As shown in Fig. 4a, the gain of a boost inverter can be affected by the load changes. Fig. 4a depicts that an AC voltage can be produced when the duty cycle oscillates around $D = 0.5$. Finally, eq.(16) determines that the factor R_s/R limits the boost inverter gain.



(a) Boost inverter gain

(b) Boost inverter efficiency

Figure 4: Boost inverter gain and efficiency

2.3. Efficiency analysis

The efficiency η is another important parameter of power converters. The efficiency is defined as a relation between the output power and the input power. Thus,

$$\eta = \frac{P_{out}}{P_{in}} = \frac{V_{out} I_{out}}{V_{in} I_{in}} = \left(\frac{V_{out}}{V_{in}}\right)\left(\frac{I_{out}}{I_{in}}\right) \tag{17}$$

Given the following relation for the boost inverter currents,

$$I_{in} = I_1 + I_2 = I_o\left(\frac{2D-1}{DD'}\right) \tag{18}$$

Calculating the efficiency η from eq.(16) and eq.(18),

$$\eta = \left(\frac{2D-1}{\frac{R_s}{R}\left(\frac{1}{D'} + \frac{D'}{D} - 1\right) + DD'}\right)\left[\frac{1}{\frac{2D-1}{DD'}}\right] \tag{19}$$

Fig.4b depicts the boost inverter efficiency η according to eq.(19). Fig.4b shows the influence of the duty cycle on the efficiency. As shown in Fig.4b, variations of $D \geq 0.8$ and $D \leq 0.2$ impact drastically in the converter efficiency [18]. Fig. 4a and Fig.4b also show the better operation range between $0.2 \leq D \leq 0.8$ where variation is more linear and efficiency is higher. Taking into account previous results, it is clear that a well developed model is necessary to describe the nonlinear behavior of the boost inverter. Next section describes the component-based approach for boost inverter modeling.

2.4. Component-based approach to model the boost inverter

The previous section showed the non-linearity and behavior of boost inverter which requires innovative modeling approaches. This section introduces a component-based approach suitable to describe the complex behavior of power converters which is applied to the boost inverter a case study. This component-based approach is intended to model power converters as building blocks interacting with their environment. This methodology allows to model the power converter with a high degree of detail disregarding subsequent changes in the power source and load models.

The state of the art about modeling of boost inverter is focused on oriented-control approaches [18]. Caceres *et .al* introduce the model presented in [8] and consider for analysis half boost inverter coupled with a fixed source. Approach in [8] simplifies the analysis; however, the dynamic of half boost inverter is disregarded. In [4] is proposed a model to design a slide-mode control on each half boost inverter. According to authors in [4], the output transfer function is evaluated to analyze the boost inverter stability. Flores-Bahamonde *et .al* in [36] assume the boost inverter operation with complementary signals; therefore, two configurations are obtained. In [36], the state equations are given and the second method of Lyapunov is used for stability analysis. According to [37], the proposed model can express the converter behavior using a simplify average model. AC operation is represented by a simplified model of the electrical grid [37]. The model is intended to study the harmonic input current impact on the boost inverter [37]. Finally, the boost inverter model in [15] introduces a converter bilinear model from a state-space model considering the grid connection.

As aforementioned, most of proposed models in literature are intended to control design. Additionally, proposed models simplify the boost inverter behavior, use linear approaches, and disregard the influence of conductive losses. However, these characteristics have an important effect on the efficiency and gain of actual power converters [35]. Another drawback of current models is the lack of scalability and adaptability; therefore, most of these models should be updated when the operation condition changes. For instant, changes in either the type of load or the power source require significant changes in the model. These drawbacks are a disadvantage for real-time emulations because the study of versatile converters such as the boost inverter requires the test under several scenarios.

In this context, an innovative component-based approach is presented. This methodology explores characteristics of adaptability, scalability, and model integration suitable for the study of power systems built by several types of power sources, converters, and loads. In addition, this method is intended to digital real-time emulations.

The component-based approach employs the concept of instantaneous power conservation to analyze the power converter systems as structures formed by functional components [18]. Fig.5 shows the general concept

about the component-based approach. In this case, the power converter is modeled independently of changes in the power source or the load. This modeling approach uses the flow of information between the power converter and its electrical neighborhood. To achieve this goal, the power converter is modeled with virtual sources able to maintain the equivalent instantaneous power at the input and output ports. The modeling procedure by using the component-based approach is described as follows:

Step 1: Substitution of the source power model by a virtual voltage source and the load model by a virtual current source. The aim of these changes is to formulate the power converter equations in terms of input and output voltages and currents independently of its electrical neighborhood.

Step 2: Formulation of the power converter equations in terms of inputs, outputs, and internal parameters. These systems of equations are converted into the discrete domain because this method is intended to digital real-time emulations.

Step 3: Formulation of equations for describing the interaction between inputs and outputs with the electrical boundaries.

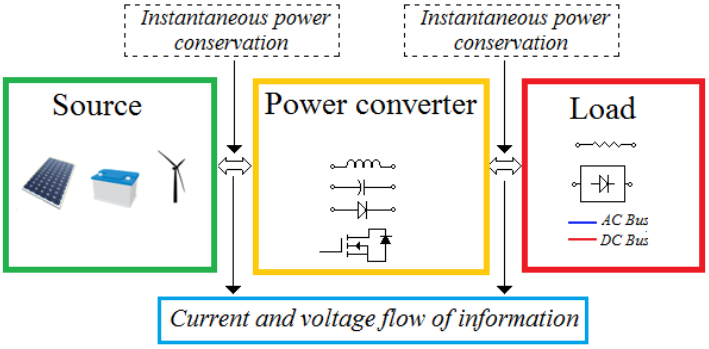


Figure 5: Component-based modeling approach

The boost inverter modeling applies the component-based approach. Four functional components are identified using this approach. The identified functional components are two bidirectional boost converters, the power source, and the load. Fig.6 shows the functional components of the boost inverter [18].

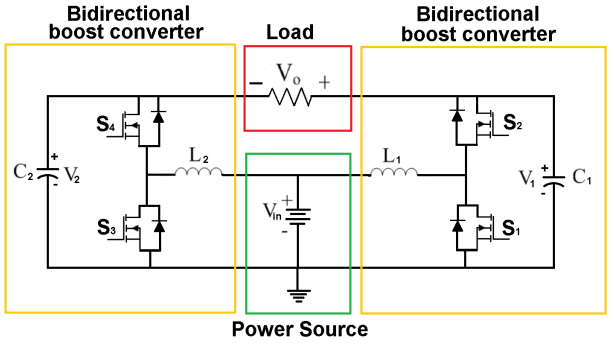


Figure 6: Boost inverter circuit and functional components

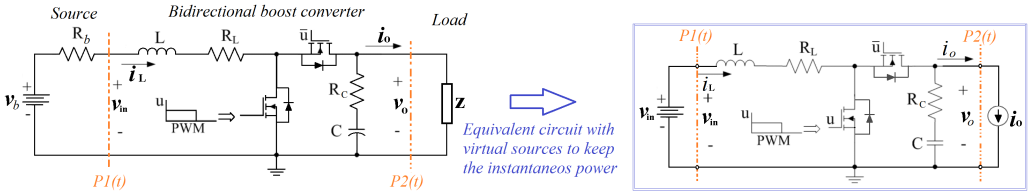


Figure 7: Operation states of bidirectional boost converter modeled with virtual sources

The boost inverter analysis takes as a start point the study of the bidirectional boost converters. In this case, the bidirectional boost converter is studied using the component-based approach and the instantaneous power conservation. Fig.7 represents the bidirectional boost converter but interconnected to input and output virtual sources. These virtual sources are intended to keep the instantaneous power $p_n(t) = v_n(t)i_n(t)$ at the input and output ports. For this reason, the input power source is replaced by a voltage source keeping the input current, and the load is replace by a current source keeping the output voltage. The power conservation property allows to model the converter without considering changes in the power source or the load, but only taking into account the information flow with the electrical boundaries (see Fig.5) [18].

As shown in Fig.5, the bidirectional boost converter block should interact with the power source and load blocks through information flow. Then, in order to achieve this goal the following mathematical framework is proposed [34].

From circuit in Fig.7, the bidirectional boost converter behavior is given by two operation states represented by the control signal $u = \{0, 1\}$. Then, the converter behavior is given by,

$$\frac{di_L}{dt} = \frac{1}{L}\left[v_{in} - i_L(R_L + R_{on}) - (i_L R_c - i_o R_c + v_c)(1 - u)\right] \tag{20}$$

$$\frac{dv_c}{dt} = \frac{1}{C}\left[-i_o + i_L(1 - u)\right] \tag{21}$$

Given the discrete form with a sample time h, and let us consider $R_s = R_L + R_{on}$,

$$i_{L_{k+1}} = \frac{h}{L}[v_{in_k} - i_{L_k}(R_L + R_{on}) - (i_{L_k} R_c - i_{o_k} R_c + v_{c_k})(1 - u_k)] + i_{L_k} \tag{22}$$

$$V_{c_{k+1}} = \frac{h}{C}\left[-i_{o_k} + i_{L_k}(1 - u_k)\right] + v_{c_k} \tag{23}$$

Finally, the input and output interactions are defined. The output voltage interaction is then,

$$v_{o_k} = v_{c_k} + R_c\left[i_{L_k}(1 - u_k) - i_{o_k}\right] \tag{24}$$

Eq.(22) and eq.(24) are the output variables from the bidirectional boost converter block. The calculated $i_{L_{k+1}}$ can be sent to the power source block and v_{o_k} is able to be sent to the load block. Furthermore, eq.(22) to eq.(24) are expressed to variables v_i and i_{o_k} from associated power source and load blocks. Additional terms in eq.(22) to eq.(24) depend only of internal states and parameters. Then, any change in the power source and load models does not affect the developed converter model. Indeed, this is an advantage for real-time emulation because model operations and execution time can be well defined before model implementation.

Section IV will integrate the boost bidirectional model with the power source and the load models using the graphical language SysML. This model integration of section IV is proposed to develop real-time emulations. For this reason, before to model the boost inverter in real-time, the next section will introduce the currents state and the scope of real-time emulations in power electronics.

3. Real-time emulation of power converters

This section describes briefly the basic concepts about real-time emulation, and its current state and scope in power electronics.

A real-time emulation of power electronics systems search to reproduce accuracy voltage and current signals in a safety environment. These signals represent the modeled systems and can be used to study control systems or power interactions. This goal requires a fast digital real-time emulator able to solve the equations that model the system [38]. Therefore, in discrete time intervals are produced the output signals equivalent to the emulated system [39]. Two situations can arise from the time constraints to evaluate the model equations. First, the model execution time is shorter than the required time-step. Then, the emulation is considered real-time. In contrast, when emulation solves the equations after the time-step the emulation is considered non real-time or offline [39].

The real-time emulations are classified in two categories [38]. The first category is a full digital real-time emulation. In this case, the emulation occurs inside of the processing device without information interchange with the environment. The second category is the Hardware-in-the-Loop. For this category, an interchange of electrical signals occurs and digital-analog and analog-digital conversion are required [23]. When the Hardware-in-the-Loop system involves an actual controller, it is called a Controller Hardware-in-the-Loop (CHIL). Our study is oriented toward Controller Hardware-in-the-Loop. In contrast, the emulation that interacts with power transfer is known as a Power Hardware-in-the-Loop (PHIL) [38].

According to Faruque et al. [39] a real-time emulator needs to solve an electrical grid-scale model by roughly $50\mu s$ or a smaller time-step to reproduce electrical transients. Additionally, smaller time-steps are required in power electronics systems because of the high switching frequency of power converters [40]. For Controller Hardware-in-the-Loop, the internal clock of microprocessors for the control system has speed periods of $1\mu s$ to 10ns. Then, an emulation of these systems requires more computing capability and the ability to emulate in the nanosecond scale [41].

Given these features of calculation speed and time constraints, the Field

Programmable Gate Arrays (FPGAs) arise as a suitable devices to achieve the requirements of real-time emulation in power electronics [21]. Fig.8 depicts the frequency range versus simulation complexity [42]. In this figure, the frequency emulation above 100 KHz is feasible for FPGA and represents a research challenge [42]. However, Myaing *et .al* argue that the main drawback of FPGAs is the complexity for developing code [43]. Indeed, Andina *et .al* highlight the potential of FPGAs for real-time emulations; however, tools to take more advantages of these devices are still required [44]. For these reasons, the next section describes the SysML and the HiLeS approaches to develop FPGA-based models to contribute with a structural methodology to emulate in real-time power converters.

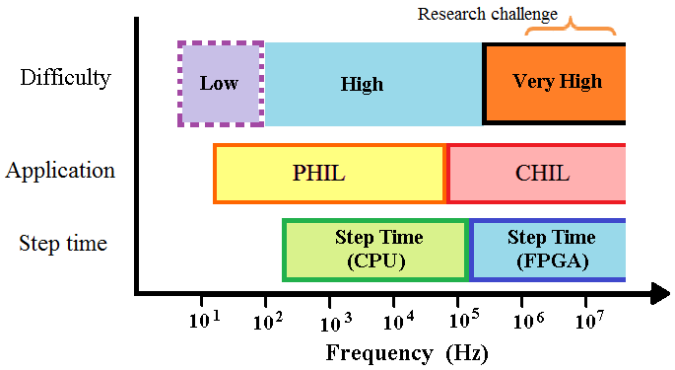


Figure 8: Range of applications and processing devices for real-time emulations

4. The SysML and Petri nets to emulate power converters

This section presents the Systems Modeling Language (SysML) [25], the High Level Specification of Embedded Systems (HiLeS) [31], and the Petri nets concepts [45] to develop FPGA-based real-time models of power converters. These models are developed using the component-based approach described in section II with the boost inverter as a study case.

4.1. SysML models

The Systems Modeling Language (SysML) is a general-purpose graphical modeling language for specifying, designing, and verifying of engineering systems [25]. The SysML language uses international standardized diagrams to specify the system concepts and architectures. The main diagrams in SysML

are the Block Definition Diagrams, the Internal Block Diagrams, and the Activity Diagrams [25]. The Block Definition Diagrams define the structure and hierarchy of components in a system. The Internal Block Diagrams describe the interaction between components. Finally, the Activity Diagrams are intended to model both computational and flow processes.

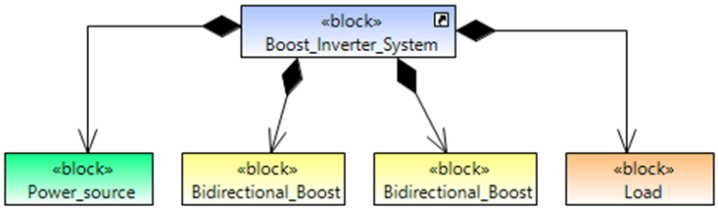


Figure 9: Block Definition Diagram for the boost inverter model

As described in section 2.4, the boost inverter was analyzed as a system built by four functional components. The SysML representation is shown in the Block Definition Diagram of Fig.9. This figure depicts the structural composition of the boost inverter. Fig.10 shows the interaction between the functional components of the boost inverter. This figure represents the Internal Block Diagram with the specification of the information flow between the bidirectional converters, the power source, and the load [34]. In this case, the main components are the bidirectional boost converters which interchange current and voltage information with the load and the power source. Finally, the Activity Diagram of Fig.11 represents a simplified sequence of process to evaluate the boost inverter model. In this case, the Activity Diagram represents the operation flow and the concurrency of parallel process.

The proposed approach allows describing the behavior of each component in contrast with conventional modeling methodologies [18]. As a result, each functional component is modeled with independence of external models. This approach is feasible for modeling current power converters because the interactions should be analyzed for the different types of power sources, converter topologies, and loads [34].

4.2. Model transformation from SysML to Petri nets and VHDL

The High Level Specification of Embedded Systems (HiLeS) framework is a free license tool able to transform automatically the SysML diagrams into

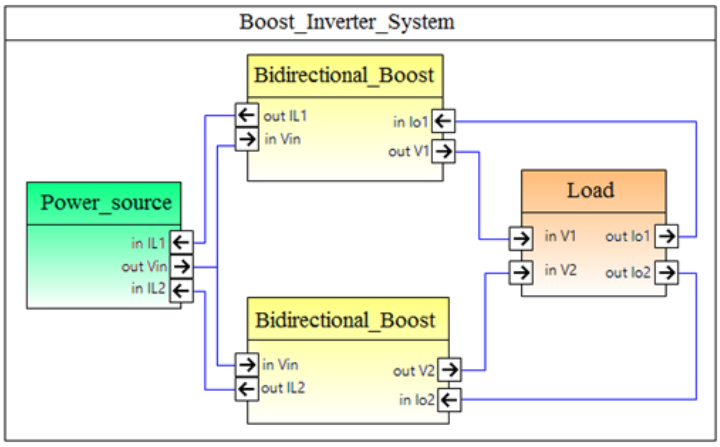


Figure 10: Internal Block Diagram for the boost inverter model

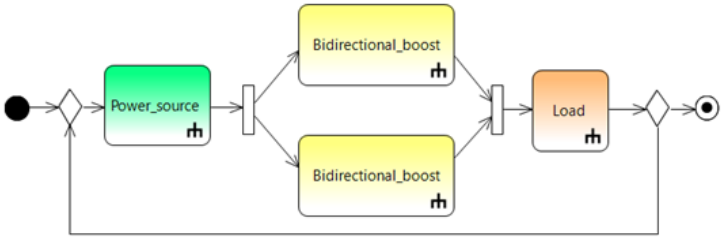


Figure 11: Activity Diagram for the boost inverter model

Petri nets and VHDL code [32]. The main advantages of this transformation are the structural and temporal verification of the developed models before implementation.

The HiLeS framework automatically transforms each element of the SysML model into an equivalent HiLeS formalism element [31]. The HiLeS formalism integrates in a Petri net representation the Internal Block Diagrams and the Activity Diagrams. Fig.12 depicts an illustrative example of these transformations. First, the Internal Block Diagrams are represented by operational blocks which are controlled by a Petri net according to the Activity Diagram sequence. Then, the control Petri net and the operational blocks are described as a synthesizable VHDL code. At this transformation level, the control Petri net is synthesized in digital components and the operational blocks are represented in VHDL code by Entities and Signals. The transfor-

mation rules are presented in Reference [46].

The VHDL code automatically generated by the tool takes advantage of the Petri net features to provide a fully parallel architecture intended to decrease the processing time. However, the main constraints of the HiLeS tool are the needs of the algorithm complexity simplification, the code optimization, and the allocation control of the synthesized components. These aspects will be considered in further research on the tool development.

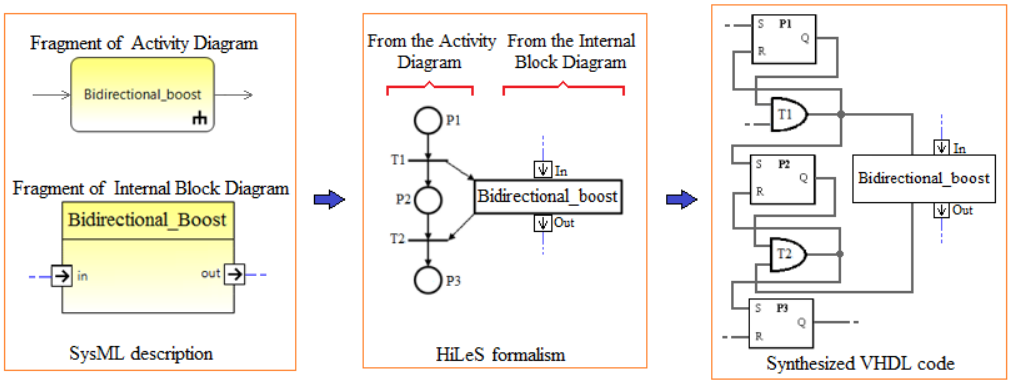


Figure 12: Illustrative example of SysML to VHDL code transformation

Fig.13 shows the SysML transformation chain used by the HiLeS framework. In Fig.13, T1 transforms SysML diagrams to the HiLeS formalism. Stage T2 extracts the Petri net representations from the HiLeS model. Finally, transformation T3 is used to transform the HiLeS description in VHDL code for implementation in FPGA [33]. An illustrative example is described through this section using the bidirectional boost converter model developed in section 2.4.

In Fig.14, the Activity Diagram partially represents the flow of operations to calculate the bidirectional boost converter model from eq.(22) to eq.(24). Fig.14 shows the advantage of using Activity Diagrams because these diagrams allow defining the concurrency and interdependency of operations to take advantage of the parallel processing and adaptability of FPGAs. Furthermore, the HiLeS framework transforms automatically the Activity Diagrams in Petri nets according to the rules in [32].

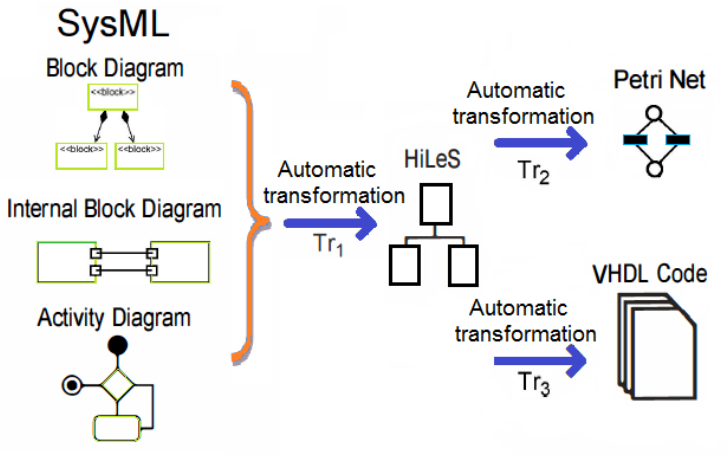


Figure 13: SysML transformations using the HiLeS framework

The Petri nets are a graphical and mathematical modeling language for the description of distributed systems. The Petri net elements are the places P_n (circles), transitions T_n (rectangles), and arcs (arrows) [45]. Places are connected to transitions by mean of arcs. According to [32], each element from the Activity Diagram is transformed to an equivalent Place-Arc-Transition branch. Fig.15a shows the associated Petri net of Fig.14. As example, the sequence of operations P_5 highlighted in the Activity Diagram is translated to a Petri net branch in Fig.15a. A repetitive loop with the transition T_6 is added to the Petri net of Fig.15a to analyze the iterative properties of the equivalent Petri net. Fig.15b shows a simplified version of the Petri net of Fig.15a to illustrate in the next section the formal mathematical validation.

4.2.1. Petri nets analysis

The Petri nets and the linear algebra theories allows analyzing the Petri net properties. The main properties of Petri nets are the boundedness, liveness, invariance, and consistency. To accomplish these properties ensure the Petri net stability and the feasible use of computational resources to avoid unexpected behavior after implementation [45].

From the simplified Petri net of Fig.15b, the graph of markings is shown Fig.16a. In Petri nets, a change of state is denoted by the movement of tokens or marks ($\bullet$) from one place to another. A graph of markings represents

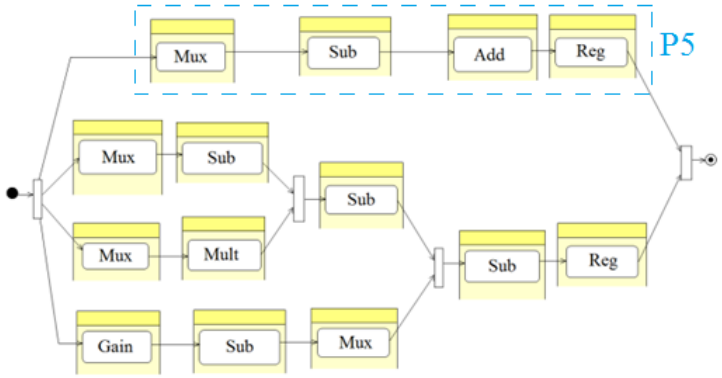
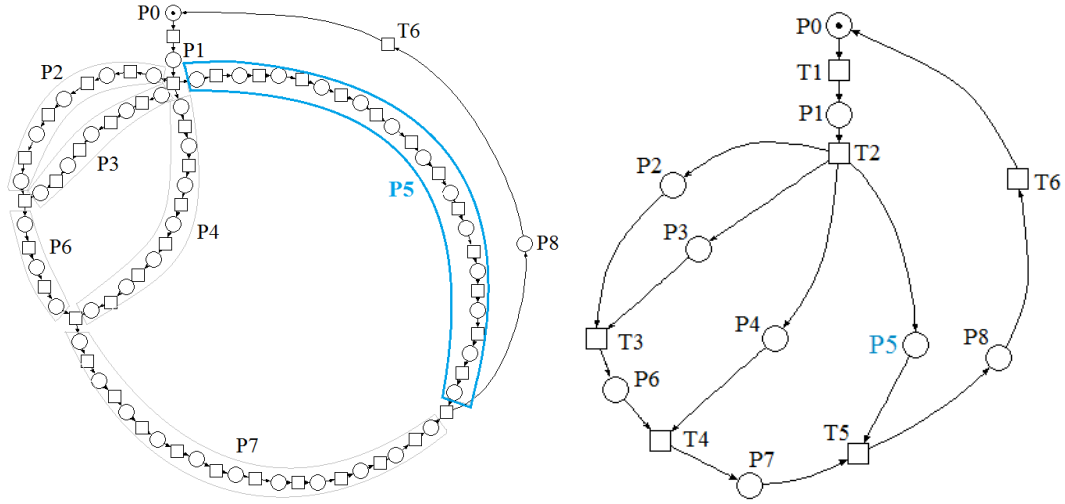


Figure 14: Partial Activity Diagram for calculation of the bidirectional boost model



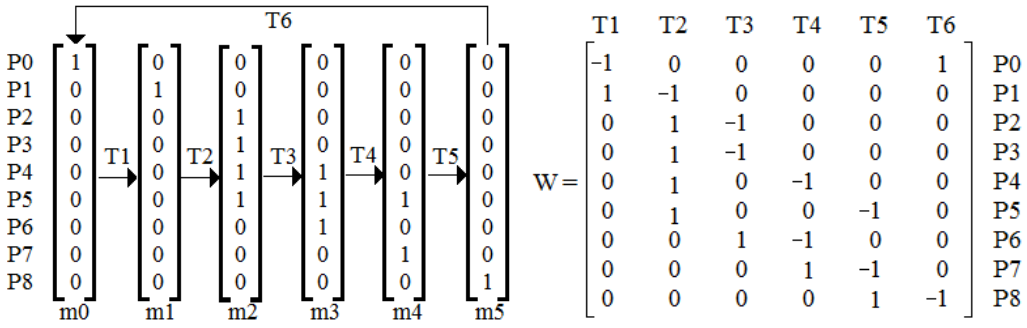
(a) Petri net for bidirectional boost model calculation

(b) Simplified Petri net

Figure 15: Petri net for the bidirectional boost converter model

the possible markings for the analyzed Petri net after transition firing [45]. A marking is a column vector whose components indicate the number of tokens or marks in each place (P_n). Fig.16a shows that each place has maximum one token. Then, the simplified Petri net is bounded because it has a limited number of tokens in each place. The interpretation of this property is that the Petri net requires a finite memory space and a finite use of hardware re-

sources. Furthermore, the graph of markings shows that is possible to reach any marking from any other marking; then, the Petri net is live without deadlocks [45]. An incident matrix **W** denotes in columns the corresponding marking modification when a transition T_n is firing and to add (1) or remove (-1) tokens [45]. Fig.16b represents the incident matrix of the bidirectional boost converter model.



(a) Graph of markings (b) Incidex matrix

Figure 16: Petri net representation with linear algebra

A vector of places x, $x = (p_1, p_2, ..., p_n)$ *with* $0 \leq p_i$, which is a solution of $x^T \cdot W = 0$ is known as a P-invariant and represents the invariant property of Petri nets [45]. Given the incident matrix **W** for the analyzed Petri net and the vector of places x, there is a finite solution for $x^T \cdot W = 0$ in $x^T = [4\ 4\ 1\ 1\ 1\ 1\ 2\ 3\ 4]$. Then, this Petri net is invariant [45].

A vector y associated with a firing sequence and solution of $W \cdot y = 0$ is known as a T-invariant. y represents the consistent property of Petri nets [45]. Given the incident matrix **W** for the analyzed Petri net and the vector of transitions y, there is a finite solution for $W \cdot y = 0$ in $y = [1\ 1\ 1\ 1\ 1\ 1]^T$. Then, this Petri net is consistent [45].

As a result, the Petri net for the bidirectional boost converter model is bounded, live, invariant, and consistent [45]. These properties ensure the structural and behavioral stability of the proposed model before implementation in FPGA.

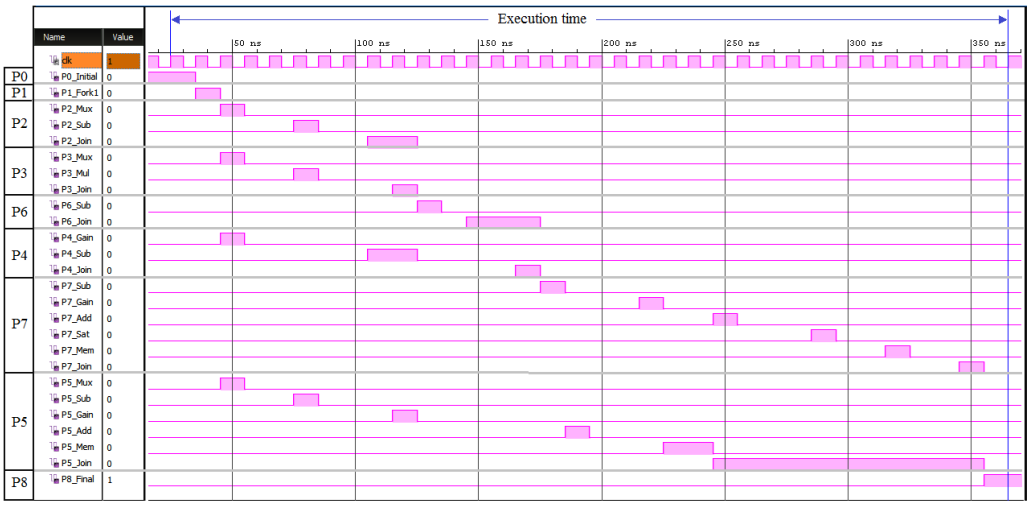


Figure 17: Time graph for the Petri net evolution

The formal validation of Petri nets by means of linear algebra methods is usually validated using computational tools [47]. The use of these automatic analysis tools, such as the integrated tool to HiLeS (TINA - TIme Petri Net Analyzer), allows the formal validation of Petri nets with a large number of places and transitions [47]. Then, developed power converter models in SysML can be transformed to Petri nets for formal Petri net validation without constraint in the Petri net size.

In addition, the HiLeS framework automatically transforms the developed models to VHDL code for FPGA implementation. Then, the Petri net of the bidirectional boost converter model is analyzed by simulating the generated VHDL code. Fig.17 shows the temporal evolution of the model operations. Results in this figure depict the Petri net temporal evolution and the delays cause by the calculus in each stage. The simulation of Fig.17 allows validating the real-time constraints for the model and identifying the delay problems. In this case, the step time for model calculation is suitable for real-time application given their range in nanoseconds. After validation of the bidirectional boost model characteristics, the entire boost inverter model is integrated and the simulation results are analyzed.

Fig.18 depicts the simulation of the VHDL code for the inductor currents

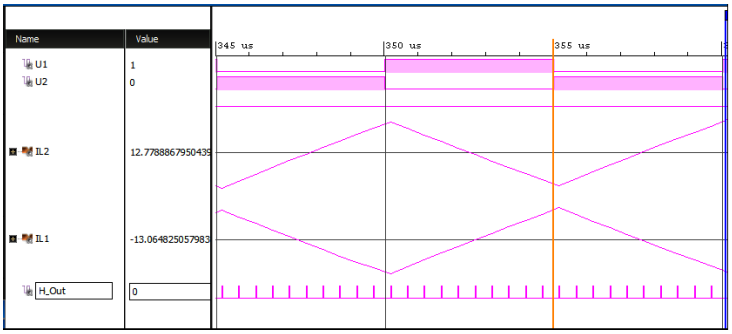


Figure 18: Simulation results for inductor current with time step around 500nS

(I_{L1},I_{L2}) of the boost inverter model. Fig.18 shows that the step time for the boost inverter model is around 500nS (sampling signal H_Out). This step time is significantly lower that the expected switching time $U = 10\mu s$. Then, the developed model is able to emulate the boost inverter in real-time given this expected switching time U.

Furthermore, Fig.19 outlines the whole behavior for the boost inverter model driven by a control signal with a Sinusoidal Pulse Width Modulation (SPWM). Simulation of Fig.19 confirms that the developed real-time model is able to follow the dynamic and nonlinear behavior of the boost inverter given a sinusoidal reference. After this suitable results, the next section will describe the experimental setup to compare the developed emulation with an actual boost inverter prototype.

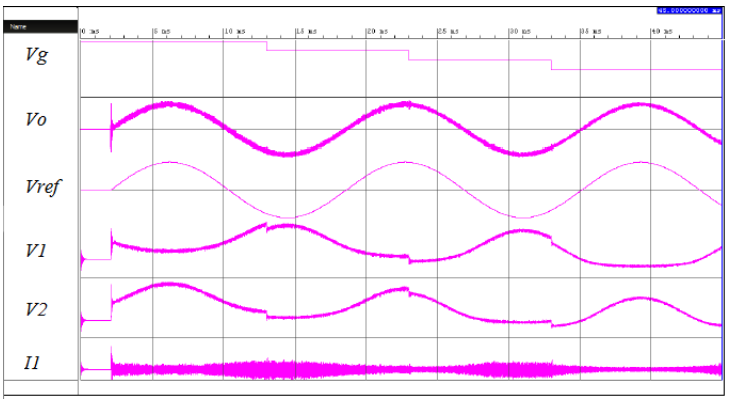


Figure 19: VHDL simulation of boost inverter real-time model

5. Experimental test to validate the proposed approach

The developed boost inverter model is implemented in FPGA and validated in real-time against the experimental test of a physical boost inverter (Fig.20). For validation, a Sinusoidal Pulse Width Modulation (SPWM) controls at the same time the boost inverter model and the boost inverter prototype. The digital signals from the FPGA model are transformed in analogical signals using high speed Digital-to-Analogical converters. Then, the analogical signals are scaled and compared with the actual output sensor signals. Table 1 lists the parameters for the experimental setup.

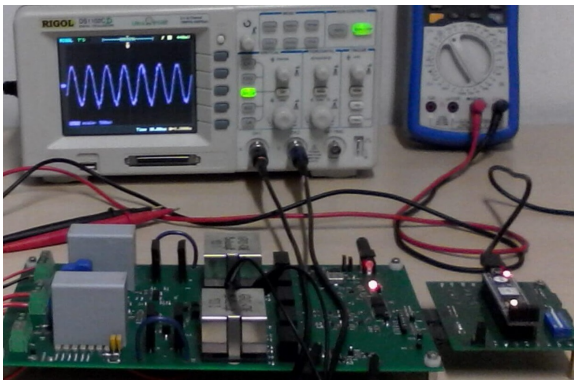


Figure 20: Experimental test of boost inverter prototype

The VHDL code automatically generated from the SysML models is synthesized in FPGA. In this study, a Zynq-7Z020 - Artix7 series FPGA is used to implement the developed model. The Vivado Design Suite WebPACK™ of Xilinx provides the VHDL synthesis, simulation, and configuration file for model implementation in FPGA. Fig.21 summaries the resource utilization after synthesis with an important use of the dedicated Digital Signal Processors (DSPs) given the floating-point operations for high model accuracy. Synthesis results also showed that the execution time is around 350ns which agrees with the expected Petri net analysis. In addition, the ratio between the execution time and the time step is approximately 70% where the difference is reserved to the high speed DAC communication.

Fig.22 depicts the DC comparison between the gain for the FPGA-based emulation and actual boost inverter. Comparison of emulated and physi-

Table 1: Parameters for the experimental validation

Parameter	Value
$L_1 - L_2$	47 μH
R_L	6 $m\Omega$
$C_1 - C_2$	10 μF
R_c	50 $m\Omega$
$S_1 - S_4$	C3M0065090D - SiC MOSFET
R_{on}	65 $m\Omega$
$DriverMOSFET$	1EDI20N12AF
$SPWM\ signal$	F_m = 50Hz F_c = 100KHz
DAC	High speed - AD5541
$FPGA$	Zynq-7Z020 - XC7Z020

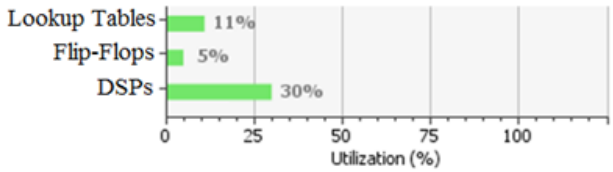


Figure 21: FPGA utilization after synthesis

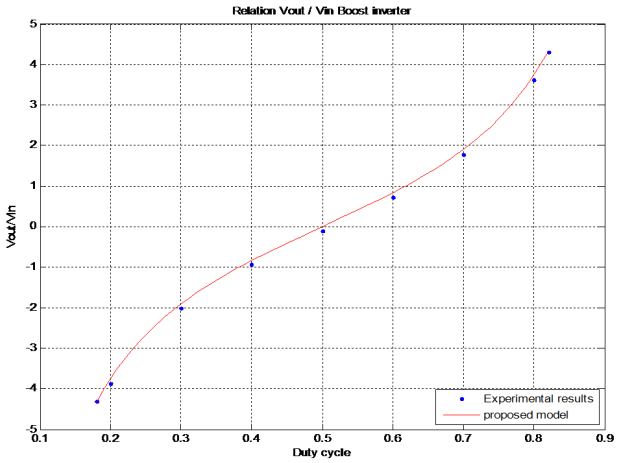


Figure 22: Boost inverter gain (red line - emulation, blue dot - prototype)

cal boost inverter gains shows the FPGA-based emulation accuracy. These

results also agree with the theoretical gain of eq.(16). Emulation results of Fig.23 allow verifying the experimental time step speed in comparison with the switching time for the emulated inductor currents. The boost inverter power results are listed in Table 2 with suitable agreement between the emulation and the experimental results. Finally, results in Fig.24 show the synchronization between the emulated and the physical boost inverter for two percentages of modulation SPWM. In both cases, the results allow confirming the suitable performance of developed emulator to follow the nonlinear and complex behavior of the boost inverter. In the next section, the results are discussed.

Table 2: Boost inverter power results

	SPWM Modulation 10%		SPWM Modulation 25%	
	Emulation	Experimental	Emulation	Experimental
Input power (W)	33.4	33.3	187.6	186.5
Output power (W)	30.8	30.3	167.0	164.1
Efficiency η	0.92	0.91	0.89	0.88

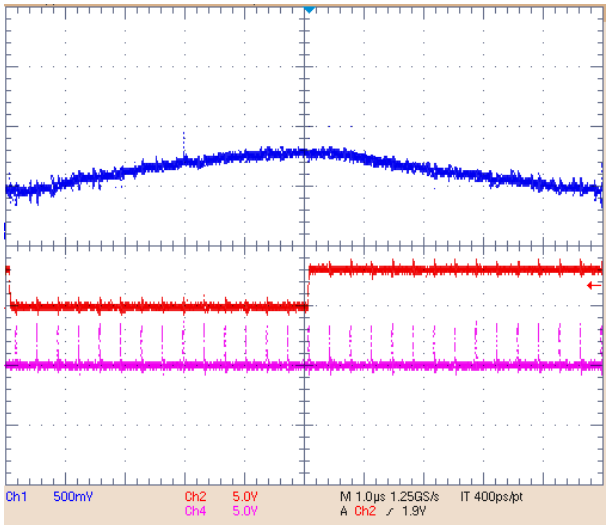
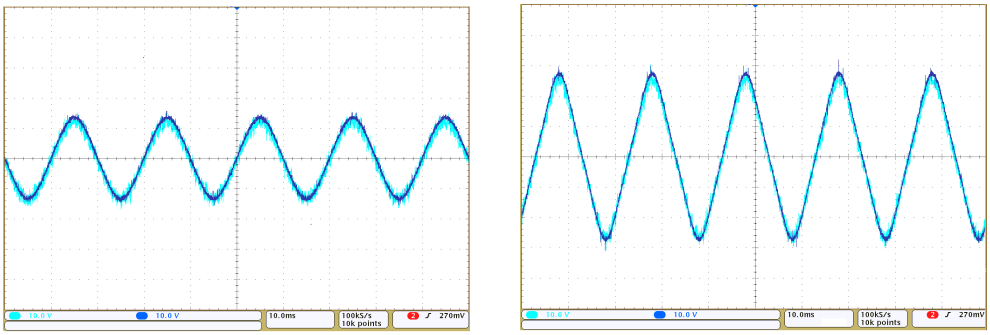


Figure 23: Time step for emulation of inductor current (blue - inductor current, red - control signal, magenta - time step)



(a) Comparison output voltage SPWM modulation = 10%

(b) Comparison output voltage SPWM modulation = 25%

Figure 24: Comparison of emulated and experimental output voltages of the boost inverter

6. Results and discussion

Fig.22 shows the gain relation for the proposed emulation model and the actual boost inverter. For this gain comparison, the Medium Square Error $MSE = \left(\sum_{i=1}^{n} (e_i)^2\right)/n$ is MSE=0.01. The Medium Absolute Error $MSE = \left(\sum_{i=1}^{n} |e_i|\right)/n$ is MAE=0.093. Then, the gain results validate the suitable accuracy of the real-time emulation model.

Fig.23 depicts the inductor current, which is the faster signal, the control signal, and the model sampling signals. These results show that the time for the sampling signal allows calculating the model for the faster signal of the system. This case of study assumes a control signal with switching time of $10\mu S$ and a time step 500nS suitable to evaluate the model.

In Fig.24 both emulated and physical signals are synchronized and compared. Fig.24a and Fig.24b show that the emulated signal is able to follow the behavior of the actual boost inverter in a wide range of SPWM variation. These results confirm that the proposed methodology is suitable to generate models for real-time emulation with a high degree of accuracy.

These experimental and emulated results have shown that the proposed methodology based on SysML and Petri nets can contribute to the research challenges discussed in section 3. The time step and accuracy achieved in this study of case are in the expected region of research proposed by Lauss *et .al* [42] with suitable results. Finally, the proposed approach has shown

its potential to decrease the gap between the real-time modeling and the complex implementation in FPGA as suggested by Andina *et .al* [44].

7. Conclusion

This paper has presented a modeling methodology based on SysML and Petri nets able to model power converters. This study of case was focused on the boost inverter with appropriate results. This methodology identified the boost inverter functional components and their interactions by means of a component-based approach. Furthermore, the bidirectional boost converter, as main functional component of the boost inverter, was analyzed in detail and subsequently integrated to the boost inverter for a real-time emulation. The real-time emulation and experimental results provided a suitable model accuracy and time performance. As a consequence, the proposed methodology has shown its potential to model power converters. In future work, this methodology can be applied to other power systems with different types of energy resources, converter topologies, and nonlinear loads.

Acknowledgements

The authors acknowledge the support provided by the Andes University, the Laboratory for Analysis and Architecture of Systems (LAAS-CNRS), and the Paul Sabatier University in the framework of an international joint supervision PhD. thesis.

8. References

[1] M. Jamil, B. Hussain, M. Abu-Sara, R. J. Boltryk, and S. M. Sharkh, "Microgrid power electronic converters: State of the art and future challenges," in *2009 44th International Universities Power Engineering Conference (UPEC)*, pp. 1–5, Sept 2009.

[2] S. M. Sharkh, M. A. Abusara, G. I. Orfanoudakis, and B. Hussain, *Power electronic converters for microgrids*. John Wiley and Sons, 2014.

[3] S. Danyali, S. H. Hosseini, and G. B. Gharehpetian, "New extendable single-stage multi-input dc-dc-ac boost converter," *IEEE Transactions on Power Electronics*, vol. 29, pp. 775–788, Feb 2014.

[4] R. Caceres and I. Barbi, "A boost dc-ac converter: operation, analysis, control and experimentation," in *Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on*, vol. 1, pp. 546–551 vol.1, Nov 1995.

[5] Y. Fang and X. Ma, "A novel pv microinverter with coupled inductors and double-boost topology," *IEEE Transactions on Power Electronics*, vol. 25, pp. 3139–3147, Dec 2010.

[6] H. Renaudineau, D. Lopez, F. Flores-Bahamonde, and S. Kouro, "Flatness-based control of a boost inverter for pv microinverter application," in *2017 IEEE 8th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 1–6, April 2017.

[7] E. Hofreiter and A. M. Bazzi, "Single-stage boost inverter reliability in solar photovoltaic applications," in *2012 IEEE Power and Energy Conference at Illinois*, pp. 1–4, Feb 2012.

[8] R. O. Caceres and I. Barbi, "A boost dc-ac converter: analysis, design, and experimentation," *IEEE Transactions on Power Electronics*, vol. 14, pp. 134–141, Jan 1999.

[9] P. Sanchis, A. Ursaea, E. Gubia, and L. Marroyo, "Boost dc-ac inverter: a new control strategy," *IEEE Transactions on Power Electronics*, vol. 20, pp. 343–353, March 2005.

[10] D. B. W. Abeywardana, B. Hredzak, and V. G. Agelidis, "A rule-based controller to mitigate dc-side second-order harmonic current in a single-phase boost inverter," *IEEE Transactions on Power Electronics*, vol. 31, pp. 1665–1679, Feb 2016.

[11] R. J. Wai, Y. F. Lin, and Y. K. Liu, "Design of adaptive fuzzy-neural-network control for a single-stage boost inverter," *IEEE Transactions on Power Electronics*, vol. 30, pp. 7282–7298, Dec 2015.

[12] D. B. W. Abeywardana, B. Hredzak, and V. G. Agelidis, "A fixed-frequency sliding mode controller for a boost-inverter-based battery-supercapacitor hybrid energy storage system," *IEEE Transactions on Power Electronics*, vol. 32, pp. 668–680, Jan 2017.

[13] D. Lopez, F. Flores-Bahamonde, S. Kouro, M. A. Perez, A. Llor, and L. Martnez-Salamero, "Predictive control of a single-stage boost dc-ac photovoltaic microinverter," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6746–6751, Oct 2016.

[14] S. Danyali, S. H. Hosseini, and G. B. Gharehpetian, "New extendable single-stage multi-input dc-dc/ac boost converter," *IEEE Transactions on Power Electronics*, vol. 29, pp. 775–788, Feb 2014.

[15] F. Flores-Bahamonde, H. Valderrama-Blavi, J. M. Bosque, A. Leon-Masich, and L. Martinez-Salamero, "Grid-connected boost inverter for small-wind urban integration: Analysis and design," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 433–439, Oct 2012.

[16] M. Jang and V. G. Agelidis, "A boost-inverter-based, battery-supported, fuel-cell sourced three-phase stand-alone power supply," *IEEE Transactions on Power Electronics*, vol. 29, pp. 6472–6480, Dec 2014.

[17] D. B. W. Abeywardana, B. Hredzak, and V. G. Agelidis, "Single-phase grid-connected lifepo4 battery-supercapacitor hybrid energy storage system with interleaved boost inverter," *IEEE Transactions on Power Electronics*, vol. 30, pp. 5591–5604, Oct 2015.

[18] A. Gutierrez, *Study of Photovoltaic System Integration in Microgrids through Real-Time Modeling and Emulation of its Components Using HiLeS*. PhD thesis, Université Toulouse III - Universidad de los Andes, 2017.

[19] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," in *2007 IEEE Power Engineering Society General Meeting*, pp. 1–1, June 2007.

[20] A. Myaing and V. Dinavahi, "Fpga-based real-time emulation of power electronic systems with detailed representation of device characteristics," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 358–368, Jan 2011.

[21] . Lucia, I. Urriza, L. A. Barragan, D. Navarro, . Jimenez, and J. M. Burdio, "Real-time fpga-based hardware-in-the-loop simulation test bench

applied to multiple-output power converters," *IEEE Transactions on Industry Applications*, vol. 47, pp. 853–860, March 2011.

[22] D. Ocnasu, *Modélisation, Commande et Simulation Temps-Réel Hybride des Systémes de Génération Non Conventionnels*. PhD thesis, Université Joseph Fourier, France, 2008.

[23] S.-J. Oh, C.-H. Yoo, I.-Y. Chung, and D.-J. Won, "Hardware-in-the-loop simulation of distributed intelligent energy management system for microgrids," *Energies*, vol. 6, no. 7, pp. 3263–3283, 2013.

[24] S. Friedenthal and A. M. and, *A Practical Guide to SysML, Third Edition: The Systems Modeling Language*. MK, 3rd ed., 2015.

[25] M. Hause, F. Thom, and A. Moore, "Inside sysml," *Electronics Systems and Software*, vol. 3, pp. 20–25, June 2005.

[26] D. Dori, *Model-Based Systems Engineering with OPM and SysML*, vol. 1. Springer, first ed., 2016.

[27] K. Nasraoui, N. Lakhoua, and L. E. Amraoui, "Study and analysis of micro smart grid using the modeling language sysml," in *2017 International Conference on Green Energy Conversion Systems (GECS)*, pp. 1–8, March 2017.

[28] A. Gutierrez, H. Chamorro, and J. Jimenez, "Hardware-in-the-loop based sysml for model and control design of interleaved boost converters," in *Control and Modeling for Power Electronics (COMPEL), 2014 IEEE 15th Workshop on*, pp. 1–6, June 2014.

[29] D. Rosenberg and S. Mancarella, *Embedded Systems Development using SysML*, vol. 1. OMG, first ed., 2010.

[30] E. Monmasson, L. Idkhajine, and M. W. Naouar, "Fpga-based controllers," *IEEE Industrial Electronics Magazine*, vol. 5, pp. 14–26, March 2011.

[31] F. Jimenez, *Specification et Conception de Micro-systemes Bases sur des Circuits Asynchrones.* PhD thesis, Institut National des Sciences Appliquees (Toulouse, Fra), LAAS-CNRS (Toulouse, Fra), Uniandes (Bogota, Col)., 2000.

[32] H. Hoyos, F. Jimenez, R. Casallas, and D. Correal, "Tucan - virtual prototype generation and time constraints analysis of real-time embedded systems," *Design Automation for Embedded Systems*, vol. 17, no. 1, p. 129165, 2013.

[33] A. Gutierrez, H. R. Chamorro, L. F. L. Villa, J. F. Jimenez, and C. Alonso, "Sysml methodology for hil implementation of pv models," in *Power Electronics and Applications (EPE'15 ECCE-Europe), 2015 17th European Conference on*, pp. 1–7, Sept 2015.

[34] A. Gutierrez, M. Bressan, F. Jimenez, and C. Alonso, "Development of boost inverter sysml model for hardware-in-the-loop simulations," in *ELECTRIMACS*, 2017.

[35] R. W. Erickson, *Fundamentals of Power Electronics*. Chapman and Hall, second ed., 1997.

[36] F. Flores-Bahamonde, H. Valderrama-Blavi, J. M. Bosque-Moncusi, G. Garcia, and L. Martinez-Salamero, "Using the sliding-mode control approach for analysis and design of the boost inverter," *IET Power Electronics*, vol. 9, no. 8, pp. 1625–1634, 2016.

[37] D. B. W. Abeywardana, B. Hredzak, and V. G. Agelidis, "A rule-based controller to mitigate dc-side second-order harmonic current in a single-phase boost inverter," *IEEE Transactions on Power Electronics*, vol. 31, pp. 1665–1679, Feb 2016.

[38] D. Maksimovic, A. M. Stankovic, V. J. Thottuvelil, and G. C. Verghese, "Modeling and simulation of power electronic converters," *Proceedings of the IEEE*, vol. 89, pp. 898–912, Jun 2001.

[39] M. D. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. 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, pp. 63–73, June 2015.

[40] N. Lin and V. Dinavahi, "Behavioral device-level modeling of modular multilevel converters in real time for variable-speed drive applications,"

IEEE Journal of Emerging and Selected Topics in Power Electronics, vol. 5, pp. 1177–1191, Sept 2017.

[41] T. Liang and V. Dinavahi, "Real-time system-on-chip emulation of electro-thermal models for power electronic devices via hammerstein configuration," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. PP, no. 99, pp. 1–1, 2017.

[42] G. F. Lauss, M. O. Faruque, K. Schoder, C. Dufour, A. Viehweider, and J. Langston, "Characteristics and design of power hardware-in-the-loop simulations for electrical power systems," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 406–417, Jan 2016.

[43] A. Myaing and V. Dinavahi, "Fpga-based real-time emulation of power electronic systems with detailed representation of device characteristics," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–11, July 2011.

[44] J. J. Rodriguez-Andina, M. D. Valdes, and M. J. Moure, "Advanced features and industrial applications of fpgas a review," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 853–864, Aug 2015.

[45] D. Rene and H. Alla., *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2005.

[46] C. Arcos, "Hiles-pl : un framework para la construcción de líneas especializadas en el diseño de sistemas embebidos," Master's thesis, Universidad de los Andes, 2012.

[47] B. Berthomieu and F. Vernadat, "Time petri nets analysis with tina," in *Quantitative Evaluation of Systems, 2006. QEST 2006. Third International Conference on*, pp. 123–124, Sept 2006.