Hardware-in-the-Loop Validation of an FPGA-Based Real-Time Simulator for Power Electronics Applications

R. Razzaghi, F. Colas, X. Guillaud, M. Paolone, and F. Rachidi

***Abstract* – This paper presents the hardware-in-the-loop (HIL) validation of a proposed FPGA-based real-time simulator for power electronics applications. The proposed FPGA-based real-time simulation platform integrates the Modified Nodal Analysis (MNA) method, Fixed Admittance Matrix Nodal Method (FAMNM) and an optimization technique to assess the optimal value of the switches conductance in order to minimize the relevant errors. Moreover, the proposed platform includes an automatic procedure to translate the *netlist* user-defined circuit schemes to the relevant equations to be solved in the FPGA. The proposed simulator is validated first by comparing the FPGA-based simulation results with offline ones performed by EMTP-RV. Then, further validation is presented by means of a dedicated HIL experimental setup composed of a controller connected to an actual two-level, three-phase inverter and its corresponding FPGA real-time model.**

***Keywords*: Hardware-in-the-loop, real-time simulation, field programmable gate array, modified nodal analysis, fixed admittance matrix nodal method.**

I. Introduction

Real-time simulation is a way to couple replica models of a given hardware or system, with real-scale monitoring and control devices/systems. Such simulations are referred to as hardware-in-the-loop (HIL) and allow performing different operational or control experimental tests which might not be possible to be conducted on the real hardware/system (e.g., [1], [2]).

For industrial applications, there are two main types of hardware used to develop a real-time simulator for the HIL tests: (i) CPU-based simulators, (ii) FPGA-based ones. In general, CPU based real-time simulators represent a better option to simulate bulk power networks since they can achieve acceptable simulation time steps (e.g., in the order of few tens of microseconds) and represent relatively complex systems. Additionally, existing CPU-based real-time simulators are typically linked to well-established programming environments (e.g., MATLAB SimPowerSystems (SPS)) that allow a more straightforward way to model components and run the simulation. However, the achievable integration time steps of CPU-based real-time simulators have a lower bound associated with the partial sequential operations that the CPU architectures need to deploy. As a consequence, the relatively large simulation time steps required by these simulators do not allow to model high frequency phenomena such as electromagnetic transients in power converters. With particular reference to this last item, as indicated in [3], the simulation time-step should be at least 20 times smaller than the switching frequency. Therefore, further techniques (e.g., interpolation ones) are required to be employed to qualify CPU-based simulators for high PWM power electronics [4].

During the past years, the size and computational power of FPGAs have been dramatically increased. As a consequence, FPGA-based real-time simulation has emerged as a leading trend for the Electromagnetic Transient (EMT) simulations of power systems and HIL simulations (e.g., [5], [6]).

Concerning the HIL simulation of power electronics applications, FPGA-based real-time simulators provide several advantages over CPU-based ones (e.g., [2], [4]). In particular, the parallel processing hardwired in FPGAs enables the implementation of specific methodologies that dramatically reduce the sequencing of the operations taking place in CPUs. FPGA-based real-time simulators provide lower sampling rate, higher frequency bandwidth and lower I/O latency [4].

However, FPGA-based real-time simulations suffer from some limitations. In particular, the model development requires, in general, the knowledge of the Hardware Description Language (HDL) which limits the implementation of complex models.

Moreover, the matrix manipulation operations are limited in FPGAs and, as a consequence, the simulation of switching devices such as power electronics requires special care. In this respect, the most straightforward method to represent topology-variable circuits in FPGA real-time simulators is the so-called Fixed Admittance Matrix Nodal Method (FAMNM) [6]. This method, irrespective of the number of the switches and their states, allows for obtaining a fixed nodal admittance matrix during switching transitions. However, it introduces artificial oscillations and errors in the simulation results (e.g., [7]).

Recently, several studies have been performed in the

R. Razzaghi and M. Paolone are with the Distributed Electrical Systems Laboratory, F. Rachidi is with the Electromagnetic Compatibility Laboratory. (EMC). Both laboratories are with the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, (e-mail: reza.razzaghi@epfl.ch, phone: +41789075556, mario.paolone@epfl.ch, farhad.rachidi@epfl.ch).
X. Guillaud and F. Colas are members of the L2EP laboratory. X. Guillaud is with Ecole Centrale de Lille, France (e-mail: xavier.guillaud@ec-lille.fr). F. Colas is with Arts et Métiers ParisTech, France (e-mail: frederic.colas@ensam.eu).

Paper submitted to the International Conference on Power Systems Transients (IPST2015) in Cavtat, Croatia, June 15-18, 2015.

literature on the applicability of the FPGA-based real-time simulators for HIL simulation of power electronics applications (e.g., [2], [6], [8], [9], [10]). These studies are mainly based on the use of FAMNM approach which enables simulation of power electronics within very low time steps. However, they do not take into account the tuning of the discrete-time switch conductance value and its effect on the simulation results accuracy. Moreover, the obtained FPGA-based real-time simulation results are validated by comparing them with offline simulations (e.g., SPS or EMTP-RV).

Within this context, this paper briefly illustrates a method to develop FPGA-based real-time simulation for power electronics applications that integrates the Modified Nodal Analysis (MNA) method, FAMNM and an optimization technique proposed in [7] to find the optimal value of the switches conductance in order to minimize the relevant errors. The proposed method includes an automatic procedure to translate the *netlist* user-defined circuit schemes to the relevant equations to be solved in the FPGA. Then, the paper mainly focuses on illustrating the validation of the proposed simulator by means of a dedicated HIL experimental setup composed of a controller connected to an actual two-levels, three-phase inverter and its corresponding FPGA real-time model.

The structure of this paper is as follows. Section II provides a brief overview of the EMT simulation. Section III describes the proposed real-time simulation platform. Section IV illustrates the experimental HIL setup, the FPGA model of the two-level three-phase inverter, its preliminary validation by comparing its results with offline simulations, and comparison with an actual three-phase inverter. Section VI concludes the paper with final remarks.

II. Overview of EMT Simulations

A. Circuit solvers and numerical integration methods

In general, two main types of solution methods are used in power systems and power electronics electromagnetic simulations: (i) nodal, and (ii) state-space [11]. In this paper we have adopted the first one since it allows a straightforward formulation of the power electronics systems equations and, in particular, it enables the FAMNM approach. MNA is represented by the general equation of (1) [12]:

$$[A_n][x_n]=[b_n] \tag{1}$$

where matrix $[A_n]$ is formed by the discrete representation of the network elements; $[x_n]$ is the vector of unknowns including the network's node voltages and branch currents; and $[b_n]$ is a vector of the independent sources and current history terms related to the network components. For EMT simulation applications, trapezoidal and backward-Euler methods are the most popular numerical integration methods [11]. For the case of switching devices, it is preferred to use the latter one since backward-Euler rule gives better damping to numerical oscillations introduced by switches [13].

B. Discrete models of simple network components

1) Lumped elements (*L*,*C*)

The most common approach for discrete-time representation of the network elements is the one proposed in [14] where the circuit elements are converted into their Norton equivalent. In particular, the lumped elements (*R*, *L*, *C*) connected between nodes *k* and *m* are described by [12], [14]:

$$G_{eq}\left(v_k(t)-v_m(t)\right)=i_{km}(t)+I_{hist}(t-\Delta t) \tag{2}$$

where G_{eq} is the equivalent conductance, and $I_{hist}(t-\Delta t)$ is the history current source associated with the time-discretized element. The values for the equivalent conductance and the history current are determined by the element type (i.e., *R*, *L*, *C*) together with the adopted numerical integration method [14].

2) Switches

Accurate and efficient switch modeling is a challenging issue for EMT simulators, especially when real-time constraints need to be achieved. In general, detailed switch models are too much sophisticated and not suitable for real-time applications. Therefore, behavioral switch models have been proposed for EMT real-time applications [15]. Among them, the simplest ones are the ideal switch model and the so-called two-valued resistor model where two resistors, characterized by large differences of their resistances, are associated with each state of the switch (R_{off}, R_{on}). However, as well described by the literature on the subject, for both models the system's admittance matrix needs to be updated and re-factorized after each switching change generating major issues to satisfy the FPGA computational time constraints.

On the contrary, the use of the discrete-time switch model allows defining the so-called fixed nodal admittance matrix method (FAMNM). In this case, the switch is represented by a relatively small inductance when its state is 'closed' and by a relatively small capacitance when its state is 'open' (e.g., [13],[16]). As a consequence, in view of (2), the switch is replaced by an equivalent conductance (G_s) in parallel with a controlled current source.

The main drawback of such representation is that it introduces artificial parameters in the circuit and, consequently, oscillations and errors to the results [7], [17]. Therefore, an optimal tuning of the switch conductance value is needed to achieve accurate results. An efficient method for the optimal selection of this parameter has been proposed in [7] and it is the method adopted in this paper to properly select this parameter (see [7] for further details).

III. The Proposed Real-Time Simulation Platform

The overall structure of the proposed and developed real-time simulator is schematically represented in Fig.1. In what follows the various blocks appearing in this figure will be explained.

With reference to the adopted hardware platform, the proposed FPGA-based real-time simulator is based on the National Instruments compactRIO-9033, an industrial

reconfigurable real-time embedded hardware platform combining an Intel Atom dual-core processor, a Xilinx Kintex-7 FPGA, and reconfigurable I/O modules. This embedded system is based on NI Linux Real-Time OS and is programmed by using the NI LabVIEW-FPGA environment. The reason to choose this hardware platform is that it provides reconfigurable platform including the CPU and the FPGA as well as reconfigurable I/O modules.

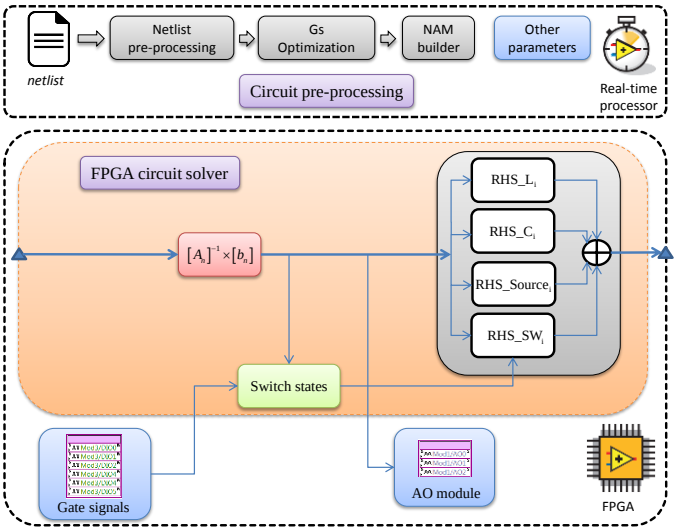

Fig. 1. The FPGA solver architecture together with the real-time processor tasks.

A. Circuit Pre-Processing

In the proposed real-time simulation platform, the EMTP-RV simulation environment is used as a Graphical User Interface (GUI) to define the circuit under study and its parameters. Then, the designed circuit is analyzed by this software to generate the so-called *netlist* file. It contains all the information about the types of the circuit components, their values, and their interconnections.

The *netlist* file is then used by *netlist pre-processing* unit to extract the relevant information to be passed to the FPGA solver. Since this is an offline process, it is done by the CPU of the real-time hardware platform. This unit is shown in Fig. 2.

According to the type of the element indicated in the *netlist* file, the algorithm extracts the relevant information (e.g., topological connections, values, etc.). The structure of the data pre-processing unit is shown in Fig. 1.

The output arrays of the *netlist* pre-processing unit are used by *NAM builder block* to form the nodal admittance matrix (NAM) in the simulator. This matrix is inverted in the CPU level based on the floating point numerical representation and the double-precision.

The inverted matrix is transformed to the fixed-point numerical representation form by proper selection of the fixed point in order to provide good accuracy. Then, the matrix is stored in the memory blocks in order to be transferred to the FPGA solver.

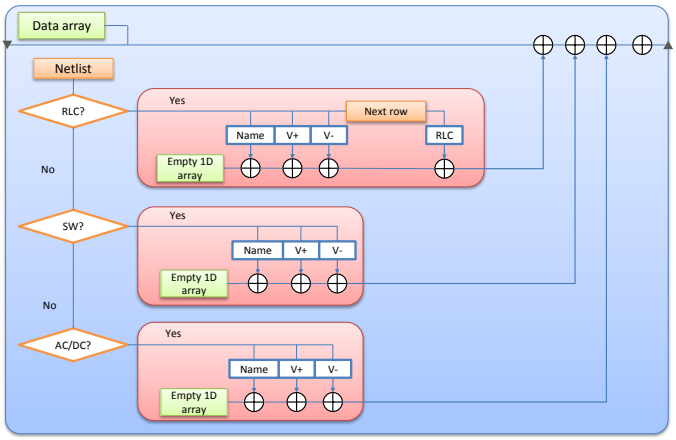

Fig. 2. Architecture of the *netlist* pre-processing unit.

It is worth noting that, the optimization problem to find the optimal switches conductance values is performed in the CPU of the real-time hardware platform by the G_s *optimization unit*. Then, the calculated values are used to build the nodal admittance matrix and, also, are transferred to the FPGA to be used in the switches RHS computations.

Moreover, additional data are calculated in the *other parameters* unit and transferred to the FPGA. These data include the desired simulation time step, independent voltage/current sources information, the number of elements in the network per element type, and controller variables (in our specific case, the converter PWM setup variables).

B. FPGA Circuit Solver

In order to take advantage of the parallel processing capability of the FPGAs, the adopted solver architecture is based on several parallel sub-tasks. In particular, in order to achieve very low simulation time steps, the two main steps of solving the MNA equation (1) are decoupled to several parallel and independent tasks. Among them, the RHS vector update for different elements is done independently and in parallel. Namely, dedicated RHS computation units are considered for inductors, capacitors, and switches.

1) RHS_L_i , RHS_C_i

For the case of inductors and capacitors, the RHS variables are function of corresponding node voltages and branch currents in the previous time step (the RHS element for the resistor is zero). Therefore, the required values to compute RHS elements are stored in the FPGA memory to be accessed in the next iteration. Then, the stored variables are used to update the RHS elements. It is worth observing that, in the CPU-based real-time simulators, the update of the RHS elements is done sequentially for different types of elements. However, thanks to the inherent parallel processing capability of the FPGA, these tasks are done in parallel. In particular, for inductors and capacitors, a dedicated computational unit has been coded. The RHS elements for the inductors and capacitors can be expressed by the general equation (3):

$$I_{His\_L_i,C_i}^{n+1} = \gamma_{L_i,C_i} x_i^n \tag{3}$$

where $I_{His\_L_i,C_i}^{n+1}$ is the history element for the i^{th} inductor or

capacitor in the current time step, γ_{L_i,C_i} is the coefficient corresponding to the i^{th} inductor (or capacitor) and x_i^n is the inductor (or capacitor) state variable in the previous time step. For the case of inductors, $\gamma_{L_i,C_i}=1$, $x_i^n=i_{L_i}^n$, and for capacitors $\gamma_{L_i,C_i}=-\frac{C_i}{\Delta t}$, $x_i^n=v_{C_i}^n$. This equation is solved independently for every capacitor and inductor to realize the highest level of parallelism.

2) RHS_Sw$_i$

The RHS elements of the switches are calculated using another dedicated sub-module. In particular, after calculating the optimal conductance values in the offline pre-processing, these values are transferred to the FPGA to be used in this sub-module. Then, according to the switches states and by accessing to their voltages and currents, the RHS elements are calculated as [13]:

$$J_{sw_i}^{n+1}=\begin{cases}-i_{s_i}^n & s^{n+1}=1\\ G_{s_i}v_{s_i}^n & s^{n+1}=0\end{cases};i=1,...,N_{sw} \quad (4)$$

where $J_{sw_i}^{n+1}$ is the RHS variable for the i^{th} switch, $i_{s_i}^n$ is the i^{th} switch current, $v_{s_i}^n$ is the i^{th} switch voltage, G_{s_i} is the i^{th} switch optimal conductance value, s^{n+1} is the switch current state, and N_{sw} is the number of switches. Similar to the inductors and capacitors, the RHS elements for different switches are calculated independently.

In order to determine the switch state, *Switch states* block is considered. The switch state is determined by its type (e.g., diode, IGBT-diode pair, etc.) [13], and the switches commands can be determined by the digital input modules (*Gate signals* block in Fig. 1) or user-defined logics.

For the case of an IGBT in parallel with an anti-parallel diode, the switch current state is determined based on (5):

$$s^{n+1}=c^{n+1}+s^n\left(i^n\le 0\right)+s^n\left(v^n<0\right) \quad (5)$$

where s^{n+1} is the switch current state, c^{n+1} is the switch gate command, and s^n is the switch previous state.

Concerning the matrix-to-vector multiplication, thanks to FAMNM switch representation, the NAM is constant during the simulation. Thus, it is computed once in the pre-processing unit. In principle, the matrix-to-vector multiplication process consists of two loops where the outer loop is associated with the number of the matrix rows and the inner loop corresponds to the number of elements within each row (i.e., number of columns).

In order to accelerate the multiplication, different levels of parallelism and techniques can be applied. In particular, the multiplication is done by splitting the matrix into individual rows and doing the dot-product and accumulation for each row, individually. Then, within each dot product operation, the multiplication is done in parallel. To this end, NI LabVIEW FPGA *IP Builder* tool is used to optimize the multiplication algorithm based on the requested latency and the throughput and by considering the available FPGA resources [18].

It is worth observing that, the FPGA-based calculations are based on fixed point numerical representation. In general, floating point offers higher precision for the numerical representation compared to the fixed point one. However, fixed-point representation is more efficient from the hardware resources usage and performance points of view. By carefully selecting the fixed-point representation, good accuracy values can be achieved.

Apart from the circuit solver engine, additional logics concerning the PWM controller are implemented in FPGA in order to provide higher precision for the high frequency PWM signals. The internal PWM controller logic can be used to verify the performance of the simulator without the need of using an external one.

IV. Experimental HIL Setup and FPGA Model of the System Under Study

A. Description of the HIL Setup

To validate the performance of the developed FPGA-based real-time simulator, we have adopted the experimental test setup in which power components are composed of a two-level three-phase inverter connected to an inductive filter (10 mH) and a resistive load (20 Ω). The global setup is depicted in Fig. 3. It shows three main parts: the system under study (which can be a FPGA based real time model or a real inverter), the DS 1104 controller board and the HMI (Human Machine Interface).

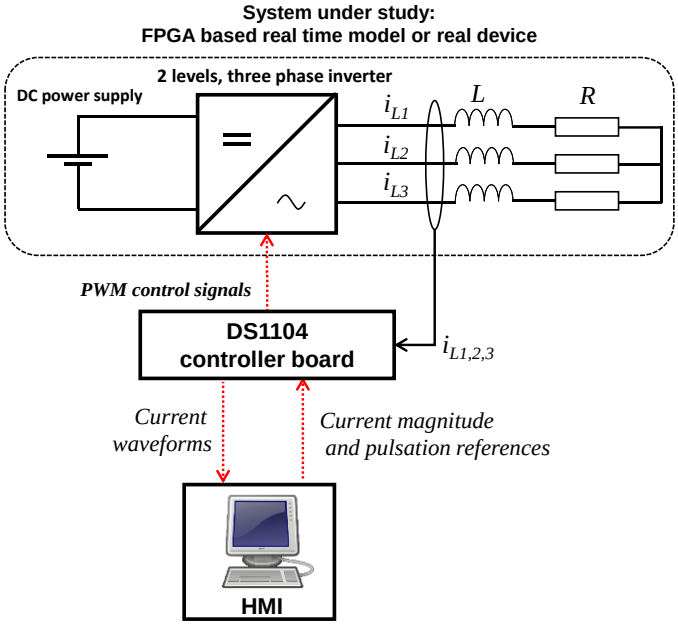

Fig. 3. Schematic representation of the HIL setup.

A classical *dq* synchronous frame current controller (e.g. [20]) has been used and implemented in a DS1104 controller board with a PowerPC 603e@250 MHz, 16bit ADC and a

sampling rate of 2μs (time step of controller has been fixed to 50μs). The real inverter is a three-phase two-level inverter based on SEMIKRON SKM75GB123d, and the current sensors are E3n sensors from fluke, with a bandwidth of 100kHz and a precision of +/- 3% (-3dB). A picture of this setup is shown in Fig. 4.

B. Description of the FPGA Model for the System under Study

The schematic representation of the modeled circuit with the MNA variables is shown in Fig. 5. Since the FPGA solver is based on fixed point numerical representation and as a consequence, it limits the amplitude of the simulation variables, the per-unit model of this circuit is derived based on the following base values:

$$\begin{aligned} V_{base} &= 60 \text{ V} \\ I_{base} &= 10 \text{ A} \end{aligned} \quad (6).$$

Then, the pre-processing unit analyzes the generated *netlist* file and builds the corresponding NAM as equation (7). This matrix is inverted and converted to the fixed point representation based on 40 bits for the word length and 19 bits for the integer part. It is worth observing that the model provided by (5) does not explicitly appear in (7) since it is a logic determining the status of this aggregated IGBT+diode device.

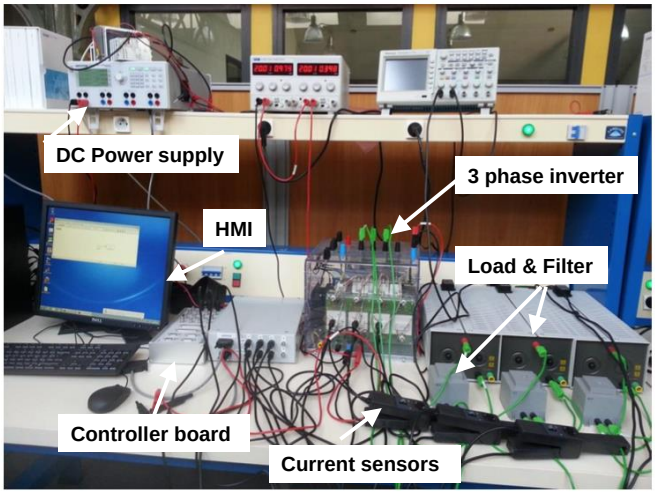

Fig. 4. Three phase inverter and controller board.

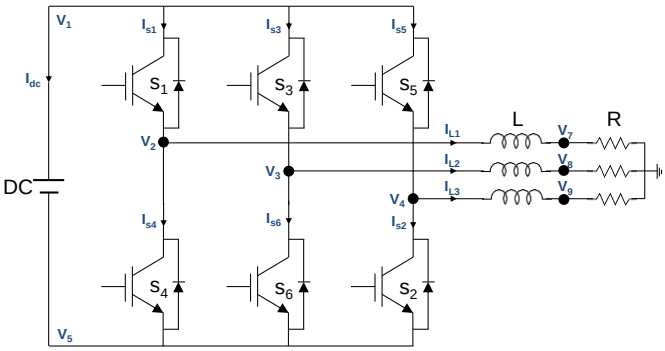

Fig. 5. Schematic representation of the two-level three-phase inverter.

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{R} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{R} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{R} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -G_{eq_{L1}} & 0 & 0 & 0 & G_{eq_{L1}} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -G_{eq_{L2}} & 0 & 0 & 0 & G_{eq2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -G_{eq_{L3}} & 0 & 0 & 0 & G_{eq3} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G_{s1} & -G_{s1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & G_{s2} & -G_{s2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
G_{s3} & 0 & -G_{s3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & G_{s4} & 0 & 0 & -G_{s4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
G_{s5} & 0 & 0 & -G_{s5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & G_{s6} & 0 & -G_{s6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
V_1^{n+1} \\ V_2^{n+1} \\ V_3^{n+1} \\ V_4^{n+1} \\ V_5^{n+1} \\ V_6^{n+1} \\ V_7^{n+1} \\ V_8^{n+1} \\ I_{L1}^{n+1} \\ I_{L2}^{n+1} \\ I_{L3}^{n+1} \\ I_{S1}^{n+1} \\ I_{S2}^{n+1} \\ I_{S3}^{n+1} \\ I_{S4}^{n+1} \\ I_{S5}^{n+1} \\ I_{S6}^{n+1} \\ I_{dc}^{n+1}
\end{bmatrix} =
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ I_{His_L1}^{n+1} \\ I_{His_L2}^{n+1} \\ I_{His_L3}^{n+1} \\ J_{SW1}^{n+1} \\ J_{SW2}^{n+1} \\ J_{SW3}^{n+1} \\ J_{SW4}^{n+1} \\ J_{SW5}^{n+1} \\ J_{SW6}^{n+1} \\ V_{dc}
\end{bmatrix} \quad (7).$$

The optimization process to find the optimal values for the switch conductances is performed in the pre-processing phase. By considering the switching modes where each switch conducts for 180 degrees of a cycle, there are eight possible switching permutations: (S1,S2,S6), (S1,S2,S3), (S2,S3,S4), (S3,S4,S5), (S4,S5,S6), (S1,S5,S6), (S1,S3,S5), and (S2,S4,S6). Therefore, we obtain eight ANAM corresponding to each status of the ideal switches. Among them, six switching patterns generate a non-zero voltage across the load and two of them (the upper or lower switches are conducting) generate zero voltage across the load.

According to the method presented in [7], the first step to calculate the optimal value for the switch conductance is to find the sets of eigenvalues corresponding to the possible switching permutations and also, the ones of the FAMNM. It is worth noting that, since the load and filter parameters are identical for all the phases, the eigenvalues for the two sets of patterns are equal for each set.

Therefore, in view of the symmetrical nature of the circuit, one identical conductance value can be assigned to the six switches. By applying the optimization method presented in [7], the objective function exhibits an optimal value of G_s=0.51 (see [7] for further details about the objective function definition). This value is used to build the NAM and also update the switches RHS elements.

In the first step, the performance of the proposed FPGA-based real-time simulator is validated by comparing its results with offline simulations carried out in EMTP-RV [21], [22]. Fig. 6 illustrates the three-phase load currents obtained by the FPGA-based real-time simulator, and by EMTP-RV off-line simulation environment, respectively. In this figure, the PWM carrier frequency is 1 kHz. Fig. 7 shows the error between the load currents of the benchmark model and those of the FPGA simulator. The error is calculated based on the pu values of the load currents. It can be observed that the FPGA-based results are characterized by a maximum error of 0.0002 pu with respect to the benchmark simulation. The reasons behind this small error are two: (i) the truncation realized by the fixed-point simulation calculations, and (ii) the approximations introduced by the discrete-time switch model.

Concerning the achieved integration time step, the simulation needs 6 FPGA ticks per time step. Consequently, by considering the 40-MHz FPGA clock, it results into an integration time step of 150 ns. Therefore, the availability of a faster FPGA clock will directly enable to further reduce the integration time step.

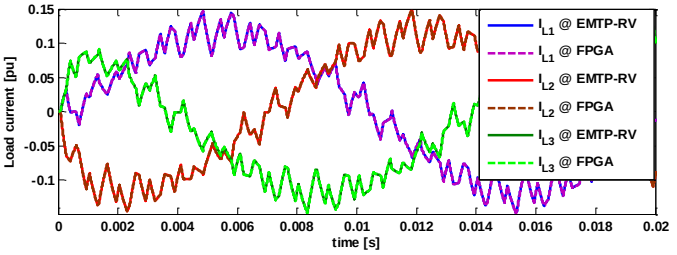

Fig. 6. Comparison of the FPGA-based real-time solver results with the corresponding EMTP-RV ones (three-phase load currents).

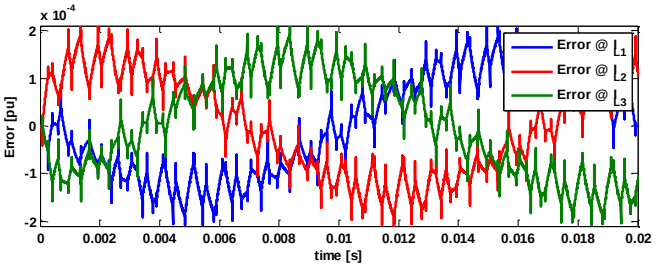

Fig. 7. Error of the load currents in pu. (reference values of Fig. 6).

The total utilized FPGA resources (based on the fixed point calculations) are: slice registers: 10.1%, slice LUTs: 28%, block RAMs: 1.5%, DSP48s: 85.3%

C. Validation by Means of HIL Simulation Test

In the previous section, the validation of the proposed FPGA-based real-time simulator was presented by comparing its obtained results with offline simulations which exhibits excellent simulation accuracy together with very low simulation time step. In this section, a further validation test is presented by making reference to a HIL simulation test performed by the proposed FPGA-based real-time simulator. To this end, first, the experimental setup explained in the previous section is used to perform the HIL test with the proposed real-time simulation platform. The external PWM controller (the PWM frequency is 1 kHz) is coupled with the simulator by using digital input modules. In particular, the switches gate signals are determined by NI-9401 which is a high speed digital I/O module and the gate signals loop and the simulation one are synchronized. To export the simulation generated signals, NI-9263 is used which is an analog output module. Since, the maximum sampling rate of this module is 100 kHz, the generated load current signals are down-sampled by this frequency to be monitored in the oscilloscope.

Then, the same controller is coupled with a physical inverter which is connected to the physical inductive filter and resistive load with the same value of the HIL simulation. The controller type and parameters are identical to the ones of the HIL test. The load currents are measured by using the current sensors described earlier and are observed by the oscilloscope.

Fig. 8 shows the comparison of the HIL simulation results and the measured waveforms for the three phase load currents in a half a period. Fig. 9 shows the errors between the HIL simulation results and the measured waveforms. It can be observed that the HIL simulation results are in very good agreement with the measurements with obvious higher errors compared to the offline simulations. The reasons for this error are: (i) the presence of noise in the measurements, (ii) the limited current sensors bandwidth, (iii) the error associated with adopted models used for the converter filter and load, and (iv) the non-linear behavior of the switches in the real inverter compared to the linear switch model in the HIL simulation. However, is spite of the above-listed sources of errors, the comparison appears satisfactory providing a good experimental validation of the proposed FPGA simulation platform.

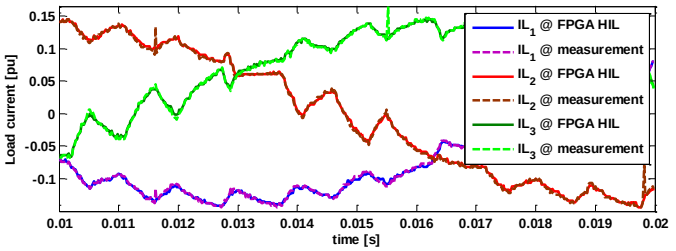

Fig. 8. Comparison of the FPGA-based HIL test results with the measured ones (three-phase load currents).

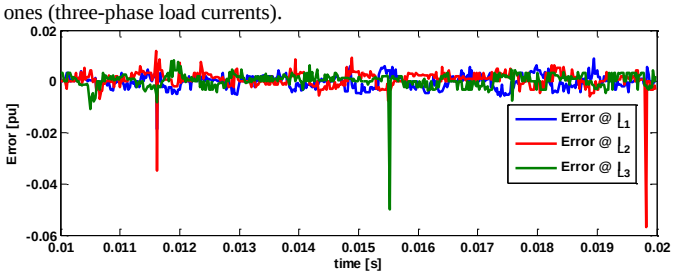

Fig. 9. Error of the load currents in pu. (reference values of Fig. 8).

V. Conclusion

The paper presented the HIL validation of the proposed FPGA-based real-time simulator for the specific case of power electronics applications. The proposed real-time simulator was implemented in an industrial real-time embedded system (National Instruments CompactRio Xilinx Kintex-7 platform) and has the following features: (i) it makes use of the Modified Nodal Analysis (MNA) method, (ii) it integrates the Fixed Admittance Matrix Nodal Method (FAMNM) together with an optimal selection of the switch conductance parameter, (iii) it accurately reproduces electromagnetic switching transients taking place in power electronic switching devices, and (iv) it enables to reach extremely low integration time steps (in the order of hundreds of ns) and avoids the need of redesigning and recompiling the FPGA code.

The accuracy of the proposed real-time simulation platform has been assessed by making reference to a real test case composed of a two-level three-phase inverter connected to an RL load.

The results obtained using the proposed FPGA simulator have been first compared with those inferred from the corresponding offline benchmark model running in the EMTP-RV simulation environment. Then, a further validation has been presented through a dedicated HIL experimental setup in which the real inverter represented in the FPGA model has been connected to the same physical controller. The FPGA results have been then compared with those measured on the experimental bench. Very good agreement has been found between the FPGA simulations and experimental results.

Future applications of the proposed solution are ongoing and refer to more sophisticated converter topologies (e.g., modular multi-level converters).

References

[1] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 919–931, 2007.

[2] T. Ould-Bachir, C. Dufour, J. Belanger, J. Mahseredjian, and J.-P. David, "Effective floating-point calculation engines intended for the FPGA-based HIL simulation," *2012 IEEE Int. Symp. Ind. Electron.*, pp. 1363–1368, May 2012.

[3] A. M. Gole, A. Keri, C. Kwankpa, E. W. Gunther, H. W. Dommel, I. Hassan, J. R. Marti, J. A. Martinez, K. G. Fehrle, L. Tang, M. F. McGranaghan, O. B. Nayak, P. F. Ribeiro, R. Iravani, and R. Lasseter, "Guidelines for modeling power electronics in electric power engineering applications," *IEEE Trans. Power Deliv.*, vol. 12, no. 1, pp. 505–514, 1997.

[4] C. Dufour, S. Cense, V. Jalili-Marandi, and J. Belanger, "Review of state-of-the-art solver solutions for HIL simulation of power systems, power electronic and motor drives," in *2013 15th European Conference on Power Electronics and Applications (EPE)*, 2013, pp. 1–12.

[5] M. Matar and R. Iravani, "Massively Parallel Implementation of AC Machine Models for FPGA-Based Real-Time Simulation of Electromagnetic Transients," *IEEE Trans. Power Deliv.*, vol. 26, no. 2, pp. 830–840, Apr. 2011.

[6] C. Dufour, S. Cense, T. Ould-Bachir, L. A. Gregoire, and J. Belanger, "General-purpose reconfigurable low-latency electric circuit and motor drive solver on FPGA," in *IECON Proceedings*, 2012, pp. 3073–3081.

[7] R. Razzaghi, C. Foti, M. Paolone, and F. Rachidi, "A method for the assessment of the optimal parameter of discrete-time switch model," *Electr. Power Syst. Res.*, vol. 115, pp. 80–86, Oct. 2014.

[8] M. Matar and R. Iravani, "FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients," *IEEE Trans. Power Deliv.*, vol. 25, no. 2, pp. 852–860, Apr. 2010.

[9] M. Dagbagi, L. Idkhajine, E. Monmasson, and I. Slama-Belkhodja, "FPGA implementation of Power Electronic Converter real-time model," *Int. Symp. Power Electron. Power Electron. Electr. Drives, Autom. Motion*, pp. 658–663, Jun. 2012.

[10] R. Razzaghi, M. Paolone, and F. Rachidi, "A general purpose FPGA-based real-time simulator for power systems applications," in *2013 4th IEEE/PES Innovative Smart Grid Technologies Europe, ISGT Europe 2013*, 2013.

[11] J. Mahseredjian, "Computation of power system transients: overview and challenges," in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–7.

[12] L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms & Computational Techniques*. Prentice-Hall, 1975, p. 737.

[13] P. Pejovic and D. Maksimovic, "Method for fast time-domain simulation of networks with switches," *IEEE Trans. Power Electron.*, vol. 9, no. 4, pp. 449–456, 1994.

[14] H. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single-and Multiphase Networks," *IEEE Trans. Power Appar. Syst.*, vol. PAS-88, no. 4, pp. 388–399, Apr. 1969.

[15] H. Jin, "Behavior-mode simulation of power electronic circuits," *IEEE Trans. Power Electron.*, vol. 12, no. 3, pp. 443–452, May 1997.

[16] S. Y. R. Hui and C. Christopoulos, "A discrete approach to the modeling of power electronic switching networks," *IEEE Trans. Power Electron.*, vol. 5, no. 4, pp. 398–403, 1990.

[17] Q. Mu, J. Liang, X. Zhou, Y. Li, and X. Zhang, "Improved ADC Model of Voltage-Source Converters in DC Grids," *IEEE Trans. Power Electron.*, vol. 29, no. 11, pp. 5738–5748, Nov. 2014.

[18] "Using NI LabVIEW FPGA IP Builder to Optimize and Port VIs for Use on FPGAS." available: http://www.ni.com/white-paper/14036/en/.

[19] M.P. Kazmierkowski, L. Malesani, "Current control techniques for three-phase voltage-source PWM converters: a survey," IEEE Transactions on Industrial Electronics, vol. 45, p. 691-703, 1998.

[20] J. Mahseredjian, S. Lefebvre and X.-D. Do, "A new method for time-domain modelling of nonlinear circuits in large linear networks," Proc. of 11th Power Systems Computation Conference PSCC, August 1993.

[21] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian and L. Gérin-Lajoie: "On a new approach for the simulation of transients in power systems," Electric Power Systems Research, vol. 77, issue 11, Sept. 2007, pp. 1514-1520.