IET Power Electronics

Research Article

IET Journals
The Institution of Engineering and Technology

Model, design and implementation of a low-cost HIL for power converter and microgrid emulation using DSP

ISSN 1755-4535
Received on 7th March 2019
Revised 14th August 2019
Accepted on 9th September 2019
E-First on 21st October 2019
doi: 10.1049/iet-pel.2019.0302
www.ietdl.org

Renan F. Bastos[1] ✉, Guilherme H. Fuzato[2], Cassius R. Aguiar[3], Rodolpho V.A. Neves[4], Ricardo Q. Machado[5]

[1]*Universidade Federal de Ouro Preto, Joao Monlevade, Minas Gerais, Brazil*
[2]*Instituto Federal de São Paulo, Campinas, São Paulo, Brazil*
[3]*Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brazil*
[4]*Universidade Federal de Viçosa, Viçosa, Minas Gerais, Brazil*
[5]*Universidade de São Paulo, São Carlos, São Paulo, Brazil*
✉ *E-mail: renanfb13@gmail.com*

Abstract: In this study, the authors propose a method to implement a low-cost hardware-in-the-loop (HIL) system for power converters and microgrids design, test and analysis. This approach uses a digital signal processor (DSP) Texas Instruments as the HIL core. All the differential equations of the power converters are solved in real-time by the DSP and displayed in the digital-to-analogue outputs. Three different converters are modelled in this study: boost converter, single-phase inverter connected to the grid and three-phase inverter connected to the grid. Experimental results are obtained and compared to the HIL response. These results were made triggering the real converter and the HIL with the same open-loop pulse width modulation signal, showing high fidelity between the digital models over the real systems. In a second moment, a microgrid is modelled in the proposed HIL and tested with a closed-loop controller. The experiments show that the proposed hardware supports time steps as low as 1 μs or 1 MHz update rate, depending on the model. The proposed technique has the potential to reduce testing time and cost, once commercial HIL devices such as Typhoon, dSPACE and RTDS have a significant cost, not affordable or available to all the research community

1 Introduction

In the last decade, there has been an exponentially growing interest for microgrids, bringing a lot of development in new control strategies and power conditioning structures. A big part of this fast advancement is in parts due to the rapid evolution of hardware-in-the-loop (HIL) control and modelling systems such as dSpace®, RTDS® and Typhoon®. The HIL technology allows researchers to implement and test their ideas faster, by building a digital prototype instead of a real experimental setup, which can take years in some projects to build. Furthermore, it allows researchers to perform exhaustive testing with total safety, using commercial control platform interacting with a virtual device.

Fig. 1 presents a real scenario with a digital control device, where the electrical/mechanical system is controlled by a closed-loop strategy. The real system's inputs/outputs can be voltage, current, pressure, temperature, speed, etc. The inputs are applied into the system by an actuator (power drive), while the outputs are read by the controller via analogue-to-digital converter. In most projects, building the system's prototype can be quite time-consuming and costly. Moreover, some experiments are impractical or unsafe in real life, for instance, short-circuit tests in power transmission systems. Therefore, a digital model interfaced with real controllers and actuators running in real-time can reduce costs and time for testing ideas. In addition, it provides the security of not working with real variables like high voltages, high or low temperatures, high speeds, etc.

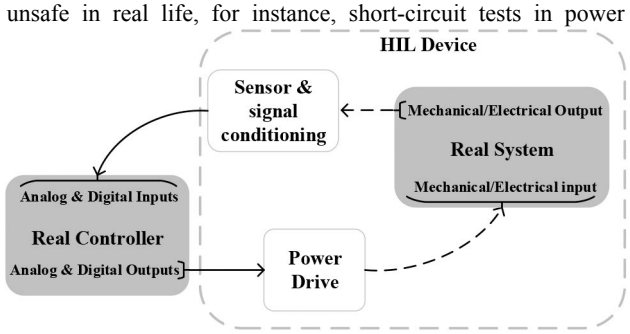

Fig. 1 *Flow chart of a real control system emulation*

In practical terms, a HIL simulation experiment is carried out as the second step of the prototyping process. The first step is the computational simulation (PSIM, MatLab and etc.) of the circuit and control strategies. Once they work effectively, we can proceed to the second step, HIL simulations. This phase is made in order to thoroughly test a commercial/designed controller using a virtual model of the power system, saving money in development and making it easier and safer to test critical conditions, such as short circuit, over voltages, etc. Only after extensive trials in a virtual platform that the real controllers are tested in real conditions, as in aerospace industry and high voltage systems.

In this context, Fig. 1 also presents the HIL device, where the real controller is connected in closed-loop to a digital representation of the real system. The digital model (HIL) is basically a set of differential equations representing the mathematical model of a real system and solved in a very short time step (T_s). A typical CPU-based *real time* (RT) simulation can only achieve a minimum time step of $T_s \geq 10\,\mu s$ caused by the large bus latencies in a CPU [1, 2].

In order to reproduce faithfully the system's behaviour, the simulation frequency ($f_s = (1/T_s)$) must be considerably higher than the system's input frequency. Thus, in order to model a HIL power converter for microgrid simulation purposes that is switching at some dozens of *kHz*, we naturally need a time step $T_s \ll 10\,\mu s$.

The main solution in the literature for solving the CPU-based HIL high time latency is based on the field-programmable gate arrays (FPGAs). This hardware has excellent parallel processing capacity and small bus latency, making it an ideal solution for RT. However, low-level programming is a huge disadvantage [1].

© The Institution of Engineering and Technology 2019

Several studies have been made using FPGA platform for real-time simulation in recent years [1, 3–10], part of this development is due to new tools that allow developers to program FPGAs in a more friendly environment, as an example we have the Xilinx® Tool provided in Matlab/Simulink [7]. In [1] the authors claim to have achieved time steps from 200 ns $\leq T_s \leq$ 650 ns depending on the complexity of the model. Just as a comparison, the commercial system Typhoon-HIL® can achieve a time step as low as $T_s \geq 500$ ns [11] ($T_s \geq 1\,\mu$s for Typhoon HIL402), and for the RTDS® $T_s \geq 3\,\mu$s [12].

Based on the aforementioned arguments, we propose the implementation of a HIL system for power converters and microgrid emulation in RT. Different from the systems presented in the literature [1–13], instead of using FPGA, we propose the use of a low-cost digital signalprocessor (DSP) LaunchPad C2000 Delfino F28377S designed by Texas Instruments. This device is a 32 bits floating-point unit and a 200 MHz clock speed, ensuring low bus latency with high-level programing (C++ using CCS by Texas Instruments).

The main contributions of this paper are: the development of a low-cost HIL device; the microgrid RT emulation based on DSP with high-level programing, which makes the modelling very easy and precise; the validation of the models with the comparison of HIL converter versus real converter; the reduced time step achieved (860 ns), comparable to that of the commercial systems.

This paper is organised as follows: in Section 2, we describe the implementation of the digital model on the DSP and mathematical model of a boost converter in state-space modelling. In Section 3, we describe the mathematical model of the single-phase inverter connected to the grid and the digital implementation of the model. In Section 4, we describe the mathematical model of the three-phase inverter connected to the grid and the digital implementation of the model. Section 5 presents the experimental setup and results, comparing the HIL models and experimental outputs. Section 6 presents the HIL device controlled in closed loop, proving this technique to be a powerful research tool. Section 7 presents the conclusions of the proposed approach.

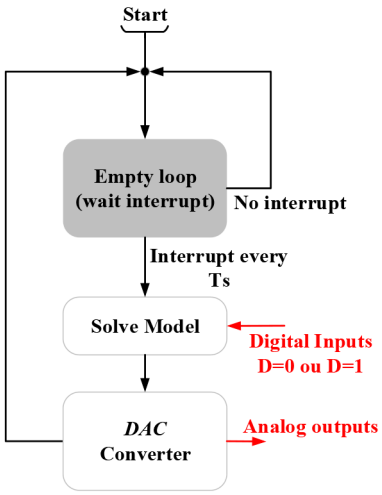

Fig. 2 *DSP configuration of HIL*

2 DSP implementation and converters modelling

Fig. 2 presents the basic algorithm of the DSP implementation of a general HIL model. The sample time is set by generating an interruption every T_s using the DSP epwm interruption. Thus, the code remains in an empty loop until the interruption is trigged, then the inputs are read and the equations of the model are solved as fast as possible before another interruption is called. In this paper we are only modelling switching converters (pulse width modulation (PWM) or sinusoidal pulse width modulation (SPWM) inputs), thus all the DSP inputs are set as digital inputs (0 or 1) as the PWM duty-cycle (D).

For modelling power converters, several papers have studied different approaches; between them, we can refer to well-known state–space modelling, modified nodal approach [13, 14] and Network Tearing [7]. In this paper, we focus on space–state modelling.

Figs. 3 and 4 show the basic boost converter topology, during which the active semiconductor is on ($D = 1$) or off ($D = 0$), respectively.

In this model, L_{dc} is the inductance, while R_L is the losses of inductance; C is the output capacitance; lastly, R is the load on the DC–DC converter terminal. The power semiconductor is modelled as resistance R_{on} for $D = 1$, and an open-circuit for $D = 0$. The diode is modelled as a 0.7 V voltage source in series with resistance R_d during $D = 0$ and an open circuit during $D = 1$. v_c is the output capacitor voltage and i_L is the inductance current, both being the state-variables.

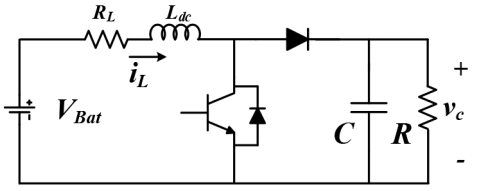

Fig. 3 *Boost converter model*

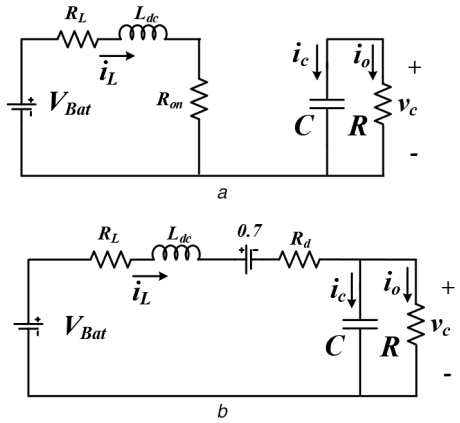

Fig. 4 *Boost model on*
(a) Closed switch ($D = 1$), ***(b)*** Open switch ($D = 0$)

© The Institution of Engineering and Technology 2019

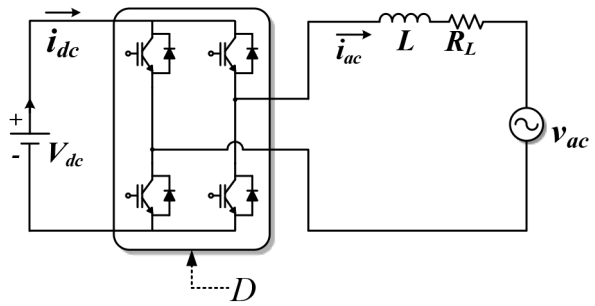

Fig. 5 *Single-phase inverter connected to the grid model*

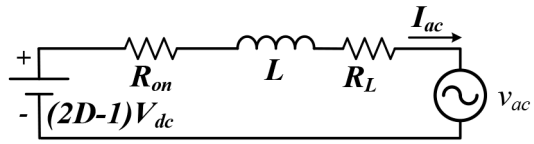

Fig. 6 *Single-phase inverter connected to the grid simplified model*

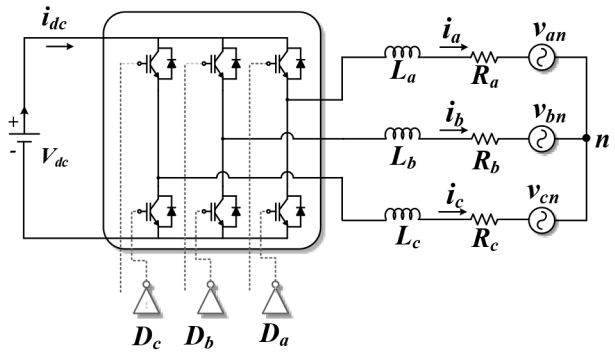

Fig. 7 *Three-phase inverter connected to the grid model*

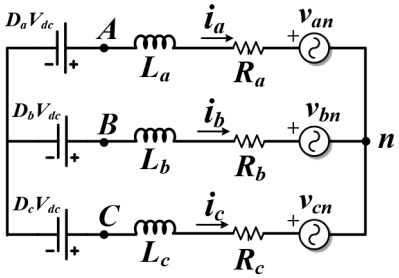

Fig. 8 *Three-phase inverter connected to the grid simplified model*

Modelling the boost converter when the semiconductor is switched on ($D = 1$) (Fig. 4*a*) we write (1) and (2), respectively. When the switch is turned off ($D = 0$), the complementary model is achieved (Fig. 4*b*), and as a result we get (3) and (4) as well

$$V_{\text{Bat}} - R_{\text{L}} i_{\text{L}}(t) - L_{\text{dc}} \frac{\text{d}i_{\text{L}}(t)}{\text{d}t} - R_{\text{on}} i_{\text{L}}(t) = 0 \tag{1}$$

$$C \frac{\text{d}v_{\text{c}}(t)}{\text{d}t} = -\frac{v_{\text{c}}(t)}{R} \tag{2}$$

$$V_{\text{Bat}} - R_{\text{L}} i_{\text{L}}(t) - L_{\text{dc}} \frac{\text{d}i_{\text{L}}(t)}{\text{d}t} - 0.7 - R_{\text{d}} i_{\text{L}}(t) - v_{\text{c}}(t) = 0 \tag{3}$$

$$C \frac{\text{d}v_{\text{c}}(t)}{\text{d}t} = i_{\text{L}}(t) - \frac{v_{\text{c}}(t)}{R} \tag{4}$$

Making D as an input (0 or 1), the discrete model of Fig. 4 can be defined as (5)–(8):

$$\frac{\text{d}i_{\text{L}}(k)}{\text{d}k} = (V_{\text{Bat}} - R_{\text{L}} i_{\text{L}}(k-1) - (0.7 + R_{\text{d}} i_{\text{L}}(k-1) + v_{\text{c}}(k-1) \\ (1-D) - DR_{\text{on}} i_{\text{L}}(k-1)) \frac{1}{L_{\text{dc}}} \tag{5}$$

$$\frac{\text{d}v_{\text{c}}(k)}{\text{d}k} = \frac{i_{\text{L}}(k-1)}{C}(1-D) - \frac{v_{\text{c}}(k-1)}{RC} \tag{6}$$

Finally, integrating (5) and (6) by the rectangular integration method (backward difference), we have

$$i_{\text{L}}(k) = i_{\text{L}}(k-1) + \frac{\text{d}i_{\text{L}}(k)}{\text{d}k} T_{\text{s}} \tag{7}$$

$$v_{\text{c}}(k) = v_{\text{c}}(k-1) + \frac{\text{d}v_{\text{c}}(k)}{\text{d}k} T_{\text{s}} \tag{8}$$

In this context, from the mathematical model (5)–(8), we define the discrete model implemented inside the DSP, Fig. 2. After solving the equations, they are displayed by means of a digital-to-analogue converter (DAC) built-in in the DSP evaluating board.

It is important to remark that solving the discrete equations using more sophisticated iteration methods would take too much processing time, making the use of DSP impracticable for real-time simulation of switching converter.

3 Single phase inverter model

Figs. 5 and 6 present the single-phase inverter connected to the AC-grid and its equivalent model. In both figures, L represents the inductive output filter, R_{L} the inductance losses and v_{ac} the single-phase grid voltage. V_{dc} and i_{dc} are the DC-link voltage and current, respectively, while i_{ac} is the current delivered by the voltage source inverter to the AC-grid. R_{on} is the parasite resistance of the semiconductors during on-state. Using the simplified model (Fig. 6), we can rewrite the dynamic model (9) given D as the input

$$V_{\text{dc}}(2D-1) - R_{\text{on}} i_{\text{ac}}(t) - L \frac{\text{d}i_{\text{ac}}(t)}{\text{d}t} - R_{\text{L}} i_{\text{ac}}(t) - v_{\text{ac}}(t) = 0 \tag{9}$$

Making D the input (0 or 1), the discrete model of Fig. 6 can be defined from (10) up to (12):

$$\frac{\text{d}i_{\text{ac}}(k)}{\text{d}k} = V_{\text{dc}}(2D-1) - R_{\text{L}} i_{\text{ac}}(k-1) \\ -R_{\text{on}} i_{\text{ac}}(k-1) - v_{\text{ac}}(k) \frac{1}{L} \tag{10}$$

Integrating (10) by the rectangular integration method (Backward difference), we have

$$i_{\text{ac}}(k) = i_{\text{ac}}(k-1) + \frac{\text{d}i_{\text{ac}}(k)}{\text{d}k} T_{\text{s}} \tag{11}$$

$$i_{\text{dc}}(k) = i_{\text{ac}}(k)(2D-1) \tag{12}$$

4 Three-phase inverter model

Additionally, in Figs. 7 and 8 we show the three-phase inverter connected to the grid and its simplified model. In the model aforementioned, L_a, L_b and L_c are the output inductive filters, while R_a, R_b, R_c are their losses. v_{dc} and i_{dc} are also the dc-link voltage and current, respectively. v_{an}, v_{bn}, v_{cn}, i_a, i_b, i_c and D_a, D_b, D_c represent the voltage, current and duty-cycle of the phases A, B and C, respectively.

Using the simplified model (Fig. 8) we can redefine (13)–(15).

© The Institution of Engineering and Technology 2019

$$V_{dc}D_a - L_a\frac{di_a(t)}{dt} - R_a i_a(t) - v_{an}(t) + v_{cn}(t) + R_c i_c(t) + L_c\frac{di_c(t)}{dt} - V_{dc}D_c = 0 \quad (13)$$

$$V_{dc}D_b - L_b\frac{di_b(t)}{dt} - R_b i_b(t) - v_{bn}(t) + v_{cn}(t) + R_c i_c(t) + L_c\frac{di_c(t)}{dt} - V_{dc}D_c = 0 \quad (14)$$

$$i_a(t) + i_b(t) + i_c(t) = 0 \quad (15)$$

Making D_a, D_b and D_c the inputs (0 or 1), and rearranging (13) and (14), the discrete model of Fig. 8 is defined from (16) up to (24)

$$\frac{di_a(k)}{dk} = [V_{dc}(D_a - D_c) + (v_{cn}(k) - v_{an}(k)) + R_c i_c(k-1) - R_a i_a(k-1)]K_1 - [V_{dc}(D_b - D_c) + (v_{cn}(k) - v_{bn}(k)) + R_c i_c(k-1) - R_b i_b(k-1)]K_3 \quad (16)$$

$$\frac{di_b(k)}{dk} = [V_{dc}(D_b - D_c) + (v_{cn}(k) - v_{bn}(k)) + R_c i_c(k-1) - R_b i_b(k-1)]K_2 - [V_{dc}(D_a - D_c) + (v_{cn}(k) - v_{an}(k)) + R_c i_c(k-1) - R_a i_a(k-1)]K_3 \quad (17)$$

Integrating (16) and (17) by the rectangular integration method (Backward difference), we have (18) and (19)

$$i_a(k) = i_a(k-1) + \frac{di_a(k)}{dk}T_s \quad (18)$$

$$i_b(k) = i_b(k-1) + \frac{di_b(k)}{dk}T_s \quad (19)$$

$$i_c(k) = -i_a(k) - i_b(k) \quad (20)$$

$$i_{dc}(k) = i_a(k)D_a + i_b(k)D_b + i_c(k)D_c \quad (21)$$

$$v_{An}(k) = L_a\frac{di_a(k)}{dk} + R_a i_a(k) + v_{an}(k) \quad (22)$$

$$v_{Bn}(k) = L_b\frac{di_b(k)}{dk} + R_b i_b(k) + v_{bn}(k) \quad (23)$$

$$v_{Cn}(k) = L_c\frac{di_c(k)}{dk} + R_c i_c(k) + v_{cn}(k) \quad (24)$$

K_1, K_2 and K_3 can be defined as

$$K_1 = \frac{L_b + L_c}{L_aL_b + L_aL_c + L_bL_c} = \frac{2}{3L}\Big|_{L_a = L_b = L_c = L} \quad (25)$$

$$K_2 = \frac{L_a + L_c}{L_aL_b + L_aL_c + L_bL_c} = \frac{2}{3L}\Big|_{L_a = L_b = L_c = L} \quad (26)$$

$$K_3 = \frac{L_c}{L_aL_b + L_aL_c + L_bL_c} = \frac{1}{3L}\Big|_{L_a = L_b = L_c = L} \quad (27)$$

5 Experimental setup and results

In order to compare the designed HIL models with the real power converters, we trigger the real converter with the same PWM signal as the HIL system. The PWM source we set an ARM processor (SAM3X8E), and the HIL models are stored in a DSP Texas Instruments, LaunchPad C2000 Delfino F28377S, presented in Figs. 9 and 10. The results are compared measuring the real output of the converter against the DACs outputs of the DSP-HIL system.

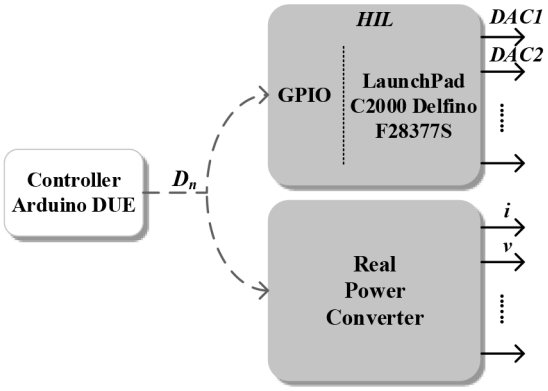

Fig. 9 *Flow chart of the proposed experimental setup*

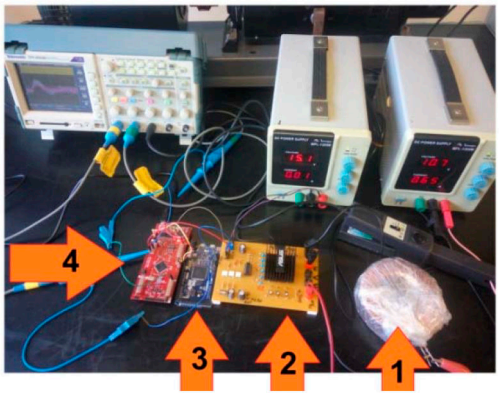

Fig. 10 *Experimental setup, 1: inductor (3.8 mH), 2: power converter, 3: Arduino Due controller (SAM3X8E ARM), 4: HIL (F28377S)*

Table 1 Boost circuit parameters

Symbol	Quantity
L_{dc}	3.8 mH
R_L	0.35 Ω
R_{on}	0.3 Ω
R_d	0.2 Ω
R	36 Ω
C	940 μF
V_{Bat}	10 V
f_{PWM}	1 kHz
T_s	5 μs

Table 1 presents the parameters stored in the DSP-HIL boost model and used in the real converter. The HIL time step is set $T_s = 5\ \mu s$ or interruption frequency 200 kHz.

The experimental comparison between the HIL and real boost converter during a transient in the duty-cycle (from 0.33 to 0.5) is shown in Fig. 11. We can clearly see the precision of the HIL model, overlapping almost perfectly the real converter's behaviour.

The small divergences mainly arise from the fact that the non-linear characteristics of the inductor were not modelled, as well as from the minor errors in estimating the internal losses of the components. Nevertheless, the accuracy of the HIL model is remarkable.

Table 2 presents the parameters of the single-phase inverter stored in the DSP-HIL and used in the real converter. In this case, we use a 370 Hz sinusoidal reference wave and a 2 kHz PWM to trigger the converter and the DSP-HIL. Fig. 12 shows the PWM wave and the inverter output current (i_{ac}) of the real converter and HIL. We can notice the perfect correspondence between the systems, just like in the boost converter.

The PWM and the DC current drained from the DC bus voltage source are shown in Fig. 13. As in the previous results, the model represents the real system with high fidelity and accuracy, proving

© The Institution of Engineering and Technology 2019

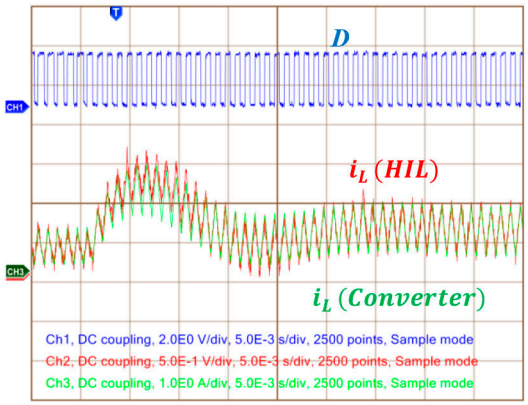

Fig. 11 *Zoom in the comparison between real boost converter and HIL during duty-cycle transient*

Table 2 Single phase inverter circuit parameters

Symbol	Quantity
L	3.8 mH
R_{L}	13.5 Ω
R_{on}	0.2 Ω
v_{ac}	0 V
V_{dc}	20 V
f_{AC}	370 Hz
f_{PWM}	2 kHz
T_{s}	5 µs

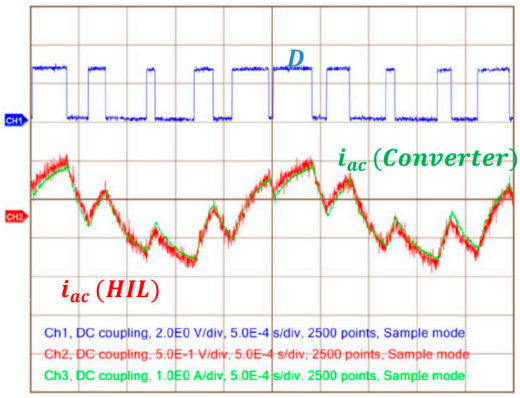

Fig. 12 *Comparison between real inverter and HIL in the real time (output current, i_{ac})*

this technique to be a powerful tool for developing and testing RT systems.

Table 3 presents the parameters of the three-phase inverter stored in the DSP-HIL and used in the real converter. In this experiment, we trigged the inverter with a $f_{\mathrm{sw}} = 60$ Hz square wave connected to RL load. Since we switch at a very low frequency, the load was chosen to have a very large inductance (130 mH) in order to filter the current.

The experimental results of the DSP-HIL versus the real three-phase inverter are presented in Figs. 14 and 15. The perfect concordance between the real converter and the HIL is clear, as all the signals overlap.

We can notice that even the variables with fast transitions (v_{An} and i_{dc}) are perfectly modelled in RT, (Fig. 15). Minor divergences can be caused by inaccuracy in estimation or measuring the components of the real setup.

6 Closed-loop HIL simulations

In order to show the capabilities of the proposed HIL in the design and to test the power of converters and microgrids, a system made

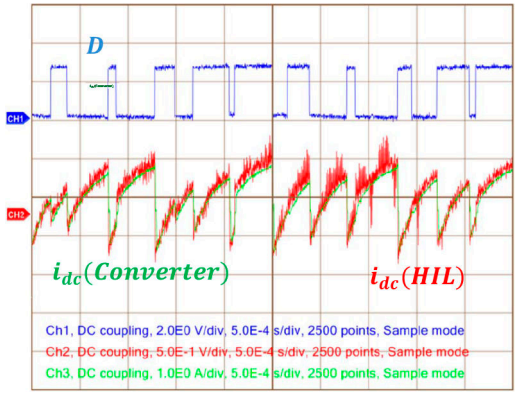

Fig. 13 *Comparison between real inverter and HIL in the real time (battery current, i_{dc})*

Table 3 Three phase inverter circuit parameters

Symbol	Quantity
$L_a = L_b = L_c$	130 mH
$R_a = R_b = R_c$	1.5 Ω
$v_{an} = v_{bn} = v_{cn}$	0 V
V_{dc}	50 V
f_{sw}	60 Hz
T_{s}	5 µs

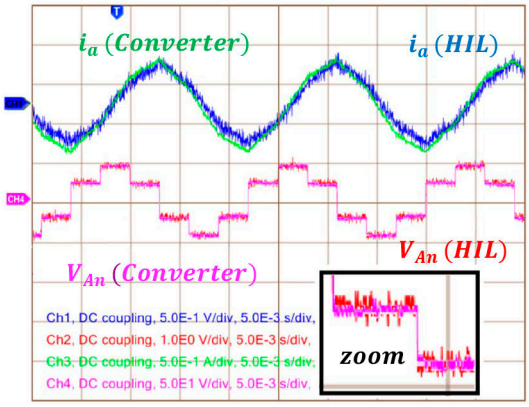

Fig. 14 *Comparison between the real inverter and HIL in the real time (output current, i_a and phase voltage, v_{An})*

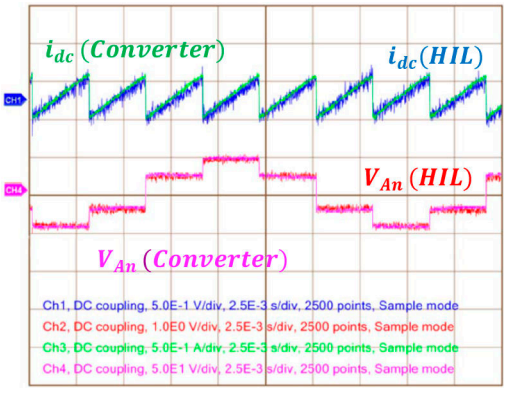

Fig. 15 *Comparison between the real inverter and HIL in the real time (battery current, i_{dc} and phase voltage, v_{An})*

of a boost converter connected to a DC bus and a single-phase DC–AC inverter is modelled, Fig. 16. Equations (28)–(35) presents the discrete model and Table 4 presents the parameters loaded in the digital HIL, where this model runs at $T_{\mathrm{s}} = 2.5\,\mu$s. The controller is designed in a SAM3X8E ARM processor, featuring a DC bus voltage loop in cascade with a current controller. In this case the

© The Institution of Engineering and Technology 2019

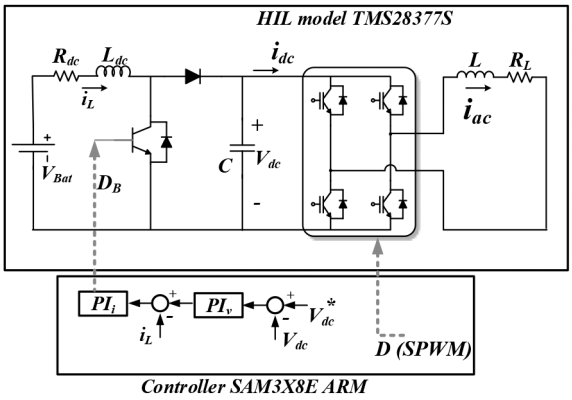

Fig. 16 *Power converter model with DC bus voltage control in cascade closed loop. The inverter operates in an open-loop SPWM*

Table 4 HIL model parameters

Symbol	Quantity
L_{dc}	10 mH
L	3 mH
R_L	20 Ω
R_{dc}	0.1 Ω
V_{Bat}	100 V
C	900 μF
f_{sw}	10 kHz
T_s	2.5 μs

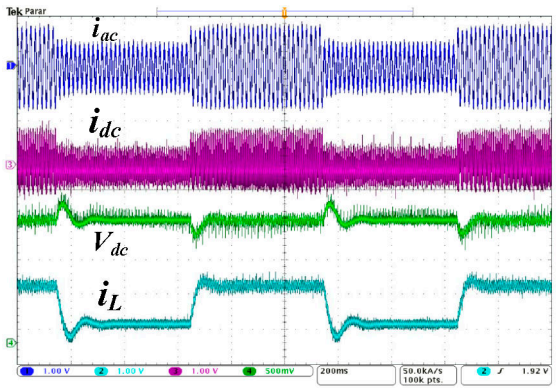

Fig. 17 *Power converter model (Fig. 16) with cascade voltage control during transient ($V_{dc}^* = 180\,V$). i_{ac} (6 A/div), i_{dc} (6 A/div), i_L (3 A/div), V_{dc} (50 V/div)*

boost converter is switching at $f_{sw} = 10\,\text{kHz}$. The inverter operates in open loop switched by a sinusoidal PWM (SPWM) signal.

$$\frac{di_{ac}(k)}{dk} = \frac{(V_{dc}(k-1)(2D-1) - R_L i_{ac}(k-1))}{L} \tag{28}$$

$$\frac{di_L(k)}{dk} = \frac{(V_{Bat} - V_{dc}(k-1)(1-D_B) - R_{dc}\, i_L(k-1))}{L_{dc}} \tag{29}$$

$$\frac{dV_{dc}(k)}{dk} = \frac{(i_L(k-1)(1-D_B) - i_{dc}(k-1))}{C} \tag{30}$$

Integrating (28)–(30) by the rectangular integration method (Backward difference), we have

$$i_{ac}(k) = i_{ac}(k-1) + \frac{di_{ac}(k)}{dk} T_s \tag{31}$$

$$i_L(k) = i_L(k-1) + \frac{di_L(k)}{dk} T_s \tag{32}$$

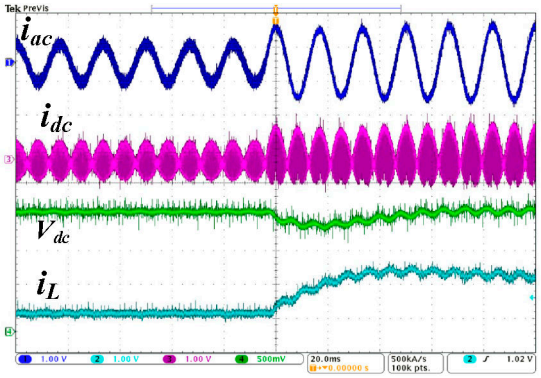

Fig. 18 *Power converter model (Fig. 16) with cascade voltage control ($V_{dc}^* = 180\,V$). Zoom during i_{ac} transient. i_{ac} (6 A/div), i_{dc} (6 A/div), i_L (3 A/div) and V_{dc} (50 V/div)*

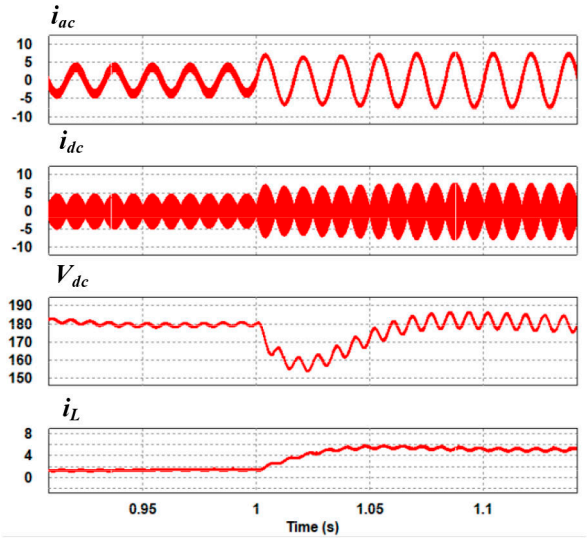

Fig. 19 *PSIM simulation for Fig. 16 circuit, showing perfect conformity with the HIL results from Fig. 18*

$$\text{IF}(i_L(k) \le 0)\,\text{then}\,\{i_L(k) = 0;\} \tag{33}$$

$$V_{dc}(k) = V_{dc}(k-1) + \frac{V_{dc}(k)}{dk} T_s \tag{34}$$

$$i_{dc}(k) = i_{ac}(k)(2D-1) \tag{35}$$

The voltage reference is set $V_{dc}^* = 180\,\text{V}$. In order to create power transients, the amplitude of the SPWM reference is changed in steps, from 40% of the maximum value to 80% of the maximum value, creating gradual changes in i_{ac} as well.

The transient behaviour of the HIL microgrid system driven by the closed-loop control is shown in Figs. 17 and 18. We can see V_{dc} stable at 180 V being adjusted by the DC current i_L as expected.

The closed-loop control computer simulation (PSIM®) of the system from Fig. 16 is shown in Fig. 19. When comparing Figs. 18 and 19, we can observe clear conformity between the HIL model and the computational simulation, showing the precision of the proposed approach.

As the last example to show the power of this approach, a microgrid made of two bidirectional DC–DC converters connected to a DC bus and connected to grid is modelled as shown in Fig. 20. The control structure for the inverter is made of a current controller, where i_{ac} is synchronised to the grid voltage V_{ac} by means of a classical PLL. The DC side is controlled by a DC bus voltage controller in cascade with two current controllers, one for each battery. The constants *K1* and *K2* are used to choose how much current each battery will deliver and were set as 1 and 0.7

© The Institution of Engineering and Technology 2019

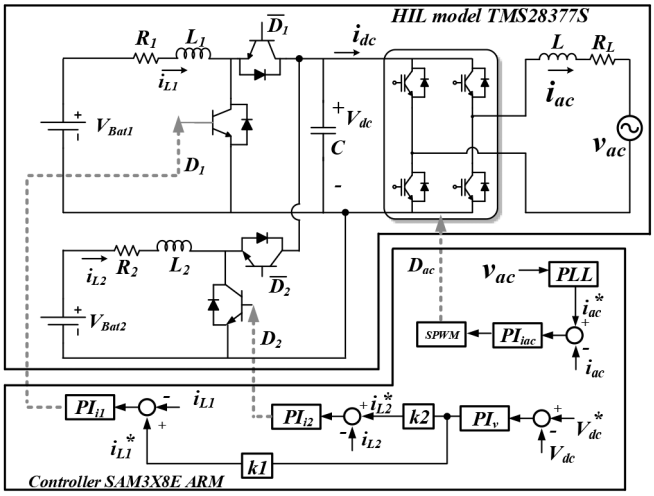

Fig. 20 *Microgrid circuit implemented in the proposed HIL and designed controllers*

respectively. The parameters of the HIL model are shown is Table 5.

Equations (36)–(46) show the discrete model implemented inside the HIL representing the microgrid from Fig. 20.

In order to create power transients in the microgrid, the current reference of the inverter (i^*_{ac}) is changed in step from 10 to −10 A and vice versa ($i_{as} = 10$ A to $i_{as} = -10$ A represents a phase shift of 180°). This power transient makes the batteries to change from discharging to charging mode (positive to negative current) and can be seen in Figs. 21 and 22. Since $K1 = 1$ and $K2 = 0.7$ the batteries share the DC current unevenly, making $i_{L2} = 0.7 i_{L1}$, as it can be seen in Figs. 21 and 22.

$$t(k) = t(k-1) + T_s \tag{36}$$

$$V_{ac}(k) = V_{ac\,max} \cos(2\pi 60\, t(k)) \tag{37}$$

$$\frac{di_{L1}(k)}{dk} = (V_{Bat1} - V_{dc}(k-1)(1-D_1) - R_1\, i_{L1}(k-1))\frac{1}{L_1} \tag{38}$$

$$\frac{di_{L2}(k)}{dk} = (V_{Bat2} - V_{dc}(k-1)(1-D_2) - R_2\, i_{L2}(k-1))\frac{1}{L_2} \tag{39}$$

$$\frac{di_{ac}(k)}{dk} = (V_{dc}(k-1)(2D_{ac}-1) - R_L i_{ac}(k-1) - v_{ac}(k))\frac{1}{L} \tag{40}$$

$$\frac{dV_{dc}(k)}{dk} = (i_{L1}(k-1)(1-D_1) + i_{L2}(k-1)(1-D_2) - i_{dc}(k-1))\frac{1}{C} \tag{41}$$

Integrating (38)–(41) by the rectangular integration method (Backward difference), we have

$$i_{ac}(k) = i_{ac}(k-1) + \frac{di_{ac}(k)}{dk} T_s \tag{42}$$

$$i_{L1}(k) = i_{L1}(k-1) + \frac{di_{L1}(k)}{dk} T_s \tag{43}$$

$$i_{L2}(k) = i_{L2}(k-1) + \frac{di_{L2}(k)}{dk} T_s \tag{44}$$

$$V_{dc}(k) = V_{dc}(k-1) + \frac{V_{dc}(k)}{dk} T_s \tag{45}$$

$$i_{dc}(k) = i_{ac}(k)(2D_{ac}-1) \tag{46}$$

Table 5 HIL model parameters

Symbol	Quantity
$L_1 L_2$	5 mH
L	3 mH
$R_L R_1 R_1$	0.1 Ω
R_{dc}	0.1 Ω
$V_{Bat1} V_{Bat2}$	100 V
V_{ac}	$127 V_{Rms}$
f_{ac}	60 Hz
C	900 μF
f_{sw} DC–DC	10 kHz
f_{sw} DC–AC	6 kHz
T_s	2.5 μs

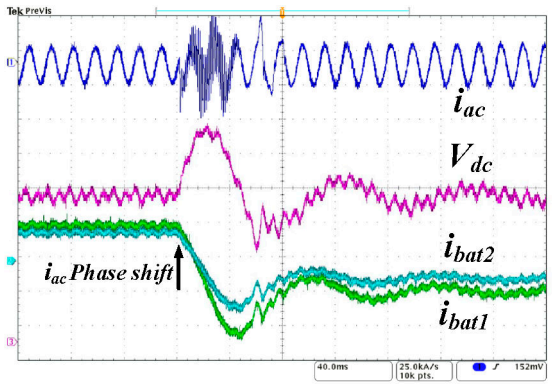

Fig. 21 *Microgrid HIL model (Fig. 20) during transient $i_{ac} = 10$ A to $i_{ac} = -10$ A. i_{ac} (20 A/div), i_{bat1} (5 A/div), i_{bat1} (5 A/div), V_{dc} (50 V/div)*

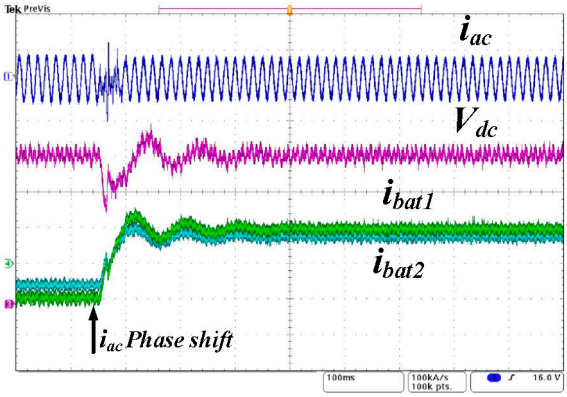

Fig. 22 *Microgrid HIL model (Fig. 20) during transient $i_{ac} = -10$ A to $i_{ac} = 10$ A. i_{ac} (20 A/div), i_{bat1} (5 A/div), i_{bat1} (5 A/div), V_{dc} (50 V/div)*

The computational simulation (PSIM) of the microgrid from Fig. 20 is shown in Fig. 23. Comparing Fig. 23 with Figs. 21 and 22 we can see perfect conformity for this very complex model in closed loop, proving again the precision and power of this approach.

The performance of the designed PLL synchronisation system in the HIL model can be seen in Fig. 24. The produced current (i_{ac}) locks on the grid voltage (V_{ac}) in feel cycles.

For the boost converter and single-phase inverter, the solving time of the model was measured around 860 ns, Fig. 25. For the three-phase inverter and for the microgrid systems the measured solving time was 2.1, 2.0 and 2.3 μs, respectively, making the last case (Fig. 20) the limit of the hardware when the time step is set 2.5 μs (400 kHz). In order to model a more complex system with the same hardware, the time step should be increased. However, in this case the switching frequency of the virtual power converter must be reduced as well, in order to keep the precision. The authors recommend a time step at least 40 times smaller than the switching

© The Institution of Engineering and Technology 2019

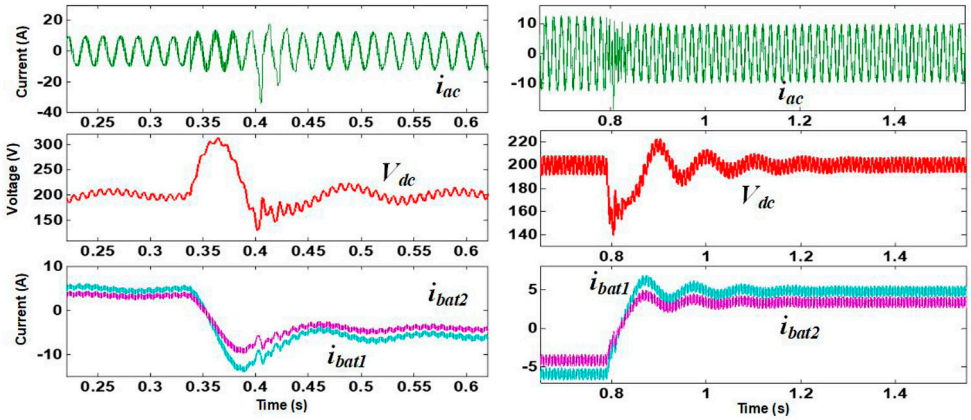

Fig. 23 *PSIM simulation for the microgrid from Fig. 20. The precision of the HIL can be seen by comparing this result with Figs. 21 and 22*

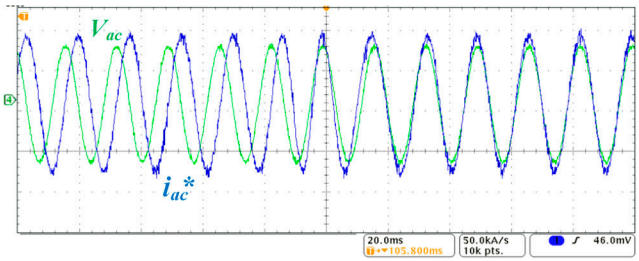

Fig. 24 *PLL synchronising the grid voltage (V_{ac}) and PLL current reference (i^*_{ac}) in the HIL model*

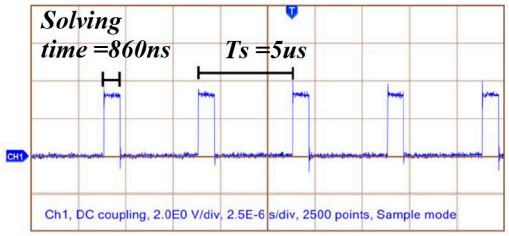

Fig. 25 *Time step (T_s = 5 µs) and solving time the proposed HIL*

period of the converter, ensuring good precision. Thus, for the most complex case of this paper (Fig. 20), the recommended maximum switching frequency for the converter is 10 kHz, since the time step was set in 2.5 µs (400 kHz).

With the current DSP technology, one way to improve the solving time is to use the newest dual-core DSP processor, such as TMS28379D. In this case, independent equations can be solved simultaneously in different cores, speeding up the processing time, maybe cutting the solving time in half, in the best-case scenario.

7 Conclusion

In this study, we proposed to build a HIL system based on a cheap DSP processor. Experimental results comparing the real case and the designed HIL models show high accuracy and fidelity. We showed that the proposed hardware can achieve time steps $T_s \geq 900$ ns (depending on the complexity of the model), slower than the FPGA approach present in the literature. However, it uses high-level programing language and can still model with fidelity a 10 to 12 kHz switching converter. The proposed hardware cannot replace the commercial hardware as Typhoon®, however it gives researchers with small resources the chance to test their ideas with small and cheap hardware. In practical terms, this approach is a great solution for test and design of small microgrid topologies, such as PV, battery and ultracapacitor connected by DC–DC converters and DC–AC inverter. Finally, this paper opens up new possibilities for researchers and students who lack the means to afford a commercial HIL system, since this HIL setup provides a less expensive, more comprehensive, fully repeatable, and faster test and verification approach.

8 References

[1] Herrera, L., Li, C., Yao, X., *et al.*: 'FPGA-based detailed real-time simulation of power converters and electric machines for EV HIL applications', *IEEE Trans. Ind. Appl.*, 2015, **51**, (2), pp. 1702–1712

[2] Jalili-Marandi, V., Zhou, Z., Dinavahi, V.: 'Large-scale transient stability simulation of electrical power systems on parallel GPUs'. Proc. IEEE Power Energy Society General Meeting, San Diego, CA, USA, 2012, pp. 1–11

[3] Matar, M., Iravani, R.: 'Massively parallel implementation of AC machine models for FPGA-based real-time simulation of electromagnetic transients', *IEEE Trans. Power Deliv.*, 2011, **26**, (2), pp. 830–840

[4] MacCleery, B., *et al.*: 'A new platform and methodology for system-level design of next-generation FPGA-based digital SMPS'. Proc. IEEE Energy Conversion Congress and Exposition (ECCE), Raleigh, NC, USA, 2012, pp. 1599–1606

[5] Bachir, T.O., Dufour, C., David, J., *et al.*: 'Floating-point engines for the FPGA-based real-time simulation of power electronic circuits'. Proc. IPST, Delft, Netherlands, 2011, pp. 1–7

[6] Bachir, T.O., David, J.P., Dufour, C., *et al.*: 'Effective FPGA based electric motor modeling with floating-point cores'. Proc. 36th IEEE IECON, Glendale, AZ, USA, 2010, pp. 829–834

[7] Ould-Bachir, T., Blanchette, H.F., Al-Haddad, K.: 'A network tearing technique for FPGA-based real-time simulation of power converters', *IEEE Trans. Ind. Electron.*, 2015, **62**, (6), pp. 3409–3418

[8] Bai, H., Liu, C., Rathore, A.K., *et al.*: 'An FPGA-based IGBT behavioral model with high transient resolution for real-time simulation of power electronic circuits', *IEEE Trans. Ind. Electron.*, 2019, **66**, (8), pp. 6581–6591

[9] Hadizadeh, A., Hashemi, M., Labbaf, M., *et al.*: 'A matrix-inversion technique for FPGA-based real-time EMT simulation of power converters', *IEEE Trans. Ind. Electron.*, 2019, **66**, (2), pp. 1224–1234

[10] Dagbagi, M., Hemdani, A., Idkhajine, L., *et al.*: 'ADC-based embedded real-time simulator of a power converter implemented in a low-cost FPGA: application to a fault-tolerant control of a grid-connected voltage-source rectifier', *IEEE Trans. Ind. Electron.*, 2016, **63**, (2), pp. 1179–1190

[11] Typhoon. Available at https://www.typhoon-hil.com/company/about-us, accessed 20 May 2018

© The Institution of Engineering and Technology 2019

[12] Forsyth, P.A., Maguire, T.L., Shearer, D., *et al.*: 'Testing firing pulse controls for a VSC-based HVDC scheme with a real time timestep <3 μs'. Proc. Int. Conf. Power Systems Transients, Kyoto, Japan, 2009, pp. 1–5

[13] Lucía, O., Urriza, I., Barragan, L.A., *et al.*: 'Real-time FPGA-based hardware-in-the-loop simulation test bench applied to multiple-output power converters', *IEEE Trans. Ind. Appl.*, 2011, **47**, (2), pp. 853–860

[14] Pejovic, P., Maksimovic, D.: 'A method for fast time-domain simulation of networks with switches', *IEEE Trans. Power Electron.*, 1994, **9**, (4), pp. 449–456

© The Institution of Engineering and Technology 2019