Modular Multi-level Converter Hardware-in-the-Loop Simulation on low-cost System-on-Chip devices

D. Tormo[§], R. Vidal-Albalate[*], L. Idkhajine[§], E. Monmasson[§] and R. Blasco-Gimenez[‡]
[§]Laboratoire SATIE, Université de Cergy-Pontoise, Cergy-Pontoise, France
[*]Department of Industrial Systems Engineering and Design, Univeristat Jaume I, Castelló de la Plana, Spain
[‡]Department of Systems Engineering and Control, Universitat Politècnica de València, València, Spain

Abstract*—*System-on-Chip (SoC) devices combine powerful general purpose processors, a *Field-Programmable Gate Array* (FPGA) and other peripherals which make them very convenient for *Hardware-in-the-Loop* (HIL) simulation. One of the limitations of these devices is that control engineers are not particularly familiarized with FPGA programming, which need extensive expertise in order to code these highly sophisticated algorithms using *Hardware Description Languages* (HDL). Notwithstanding, there exist *High-Level Synthesis* (HLS) tools which allow to program these devices using more generic programming languages such as C, C++ and SystemC. This paper evaluates SoC devices to implement a *Modular Multi-Level Converter* (MMC) model using HLS tools for being implemented in the FPGA fabric in order to perform HIL verification of control algorithms in a single low-cost device.

***Index Terms*—System-on-Chip (SoC), Hardware-in-the-Loop (HIL), Modular Multi-level Converter (MMC), Real-Time Simulation (RTS), High-Level Synthesis (HLS), Zynq**

I. Introduction

SoC platforms arrived years ago to the market starting a completely new paradigm, gathering in the same device a powerful processor, a fully configurable FPGA, and mixed analog-digital peripherals [1], [2], [3]. These promising platforms allow to build in a single device a complete control system mixing hardware and software functionalities which increases integration and eases the development of powerful and versatile controllers.

The common way of programming the FPGA fabric has been mainly using HDLs like Verilog, VHDL, or SystemC. However, the amount of time needed to code and debug the plant model using these languages is cumbersome. In this matter, HLS tools ease this task, allowing to use the benefits of hardware acceleration as speed and energy saving, without the mandatory need of building up extensive hardware expertise [4]. It allows designers to work at a higher abstraction level by specifying the behavior of an algorithm rather than its hardware definition. Notwithstanding, the hardware code these tools produce is still far from what can be achieved using HDL languages in terms of performance and FPGA resources [5]. However, when the complexity of the application increase, its use becomes convenient because it impacts significantly in the development time and cost in man-hours.

In a HIL simulation, the plant model is executed at the same pace as the real system, providing a similar dynamic response utilised for the testing of digital controllers in real-time [6], [7], [8], [9], [5]. When applied in the context of HIL validation of power electronics, these devices can bring important added value thanks to their potent processing speed, broad versatility, and high degree of parallelism [10], [11], [12]. Hence, if SoC platforms are used for this purpose, the whole control system could be verified using the same single device that will be used in the final application, facilitating significantly the HIL validation and speeding up the whole controller design process, reaching earlier its deployment on the real plant [13]. Another important point in favor is the relative low cost of such platforms, which makes them very attractive for cost-sensitive scenarios where the use of expensive commercial HIL platforms cannot be assumed.

The design of these *Intellectual Property* (IP) blocks must rigorously consider a set of constraints at different development stages: (i) during the modeling of the system to be real-time simulated; (ii) during the digital realization of the IP; and also (iii) during its final implementation in the digital platform. This paper focuses on the utilisation of the *Zynq-7000 All Programmable SoC* not only to control the real plant, but also for control validation using HIL techniques inside the same single device. The selected application is a controller for a MMC. The regulator will be implemented in the *Processing System* (PS) whereas the MMC model will be implemented in the *Programmable Logic* (PL). An important effort has been made in the hardware/software co-design in order to achieve good performance in the data transfer between the two entities, aiming always to achieve the fastest execution time of the plant model.

The rest of the paper is organized as follows: In Section II the MMC is explained. in Section III the controller under validation is presented. Section IV is devoted to the MMC discrete model used in the HIL validation. In Section V the HIL simulation is presented and results are outlined. Finally, the conclusions can be found in Section VI.

Corresponding author Dr Daniel Tormo {datorbor@upv.es}.

Dr Ricardo Vidal-Albalate and Dr Ramon Blasco-Gimenez would like to thank the support of the Spanish Ministry of Economy and EU FEDER funds to this work under Grant DPI2014-53245-R.

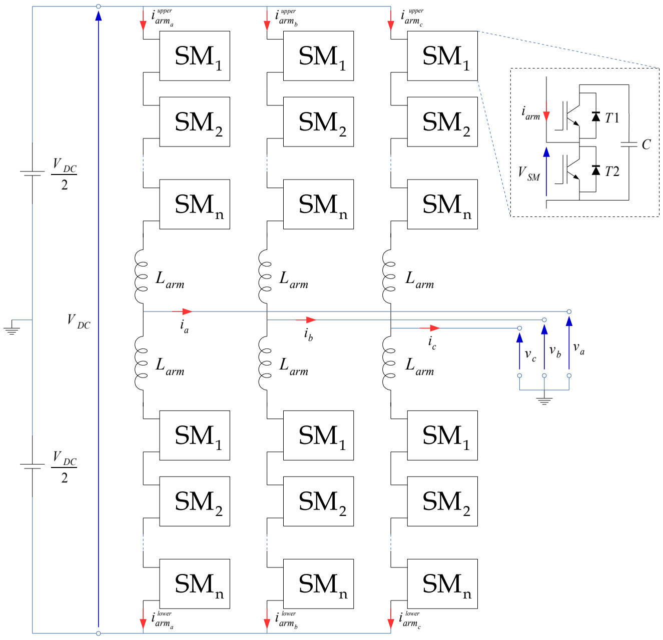


Fig. 1. Structure of the MMC

II. The Modular Multi-level Converter

The basic structure of an MMC is shown in Fig. 1. It is formed by N *Half-Bridge* (HB) *Submodules* (SM) per arm capable of producing a line-to-neutral voltage waveform of $N+1$ levels [14]. An inductor L_{arm} is added on each arm to limit current harmonics and the short circuit current in the event of a DC fault. Each SM includes a capacitor and two IGBTs with antiparallel diodes as shown in Fig. 1.

The output AC voltages of each converter phase can be computed using (1).

$$v_{abc} = \frac{V^l_{arm} - V^u_{arm}}{2} + \frac{R_{eq}}{2} i_{abc} + \frac{L_{arm}}{2}\frac{d}{dt} i_{abc} \qquad (1)$$

where V^u_{arm} and V^l_{arm} are the voltages to be inserted in the upper and lower arm ((2) and (3) respectively), R_{eq} the equivalent resistor considering the ON resistance of the IGBTs and anti-parallel diodes plus the parasitic resistance of the arm inductors, and i_{abc} are the phase AC output currents (4). Hence, the output current i_{abc} can be controlled by means of V^u_{arm} and V^l_{arm} when the MMC is connected to an AC grid of voltage v_{abc}.

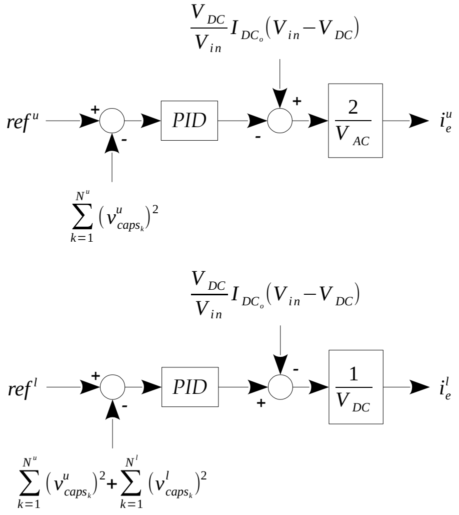
$$V^u_{arm} = \frac{V_{DC}}{2} - v^{ref}_{abc} - v^{ref}_{circ_{abc}} \qquad (2)$$

$$V^l_{arm} = \frac{V_{DC}}{2} + v^{ref}_{abc} - v^{ref}_{circ_{abc}} \qquad (3)$$

$$i_{abc} = i^u_{arm_{abc}} - i^l_{arm_{abc}} \qquad (4)$$

From the above equations, V_{DC} is the pole-to-pole DC voltage, v^{ref}_{abc} is the required output phase voltage, and $v^{ref}_{circ_{abc}}$ the voltage linked to the circulating currents which is computed using (5) according to [15].

$$v^{ref}_{circ_{abc}} = R_a\left(i^{ref}_{circ_{abc}} - i_{circ_{abc}}\right) - \hat{R} i^{ref}_{circ_{abc}} \qquad (5)$$



$\frac{V_{DC}}{V_{in}} I_{DC_o}(V_{in} - V_{DC})$
ref^u
+
-
PID
+
-
$\frac{2}{V_{AC}}$
i^u_e
$\sum_{k=1}^{N^u}(v^u_{caps_k})^2$
$\frac{V_{DC}}{V_{in}} I_{DC_o}(V_{in} - V_{DC})$
ref^l
+
-
PID
-
+
$\frac{1}{V_{DC}}$
i^l_e
$\sum_{k=1}^{N^u}(v^u_{caps_k})^2 + \sum_{k=1}^{N^l}(v^l_{caps_k})^2$

Fig. 2. Energy controllers

where R_a is the referred as the "active resistance" –which is in fact the gain parameter of the P regulator that controls the circulating current, $\hat{R}$ is an estimate of R_{eq}, and $i^{ref}_{circ_{abc}}$ is the circulating current reference which is calculated using (6). More information about this implementation can be found in [16].

$$i^{ref}_{circ_{abc}} = \frac{i^u_{abc} + i^l_{abc}}{2} \qquad (6)$$

In a perfectly balanced three-phase MMC, each arm would provide half of the AC output current plus a circulating current whose value would correspond to one third of the total DC current. Notwithstanding, capacitor voltage variations lead to additional circulating currents that increase not only the RMS value of the arm currents, but the capacitor voltage oscillations and the overall losses as well.

III. The controller

The controller used for this application is based on the *Modular Multi-level DC-DC Converter for HVDC grids* presented in [17]. Regarding the control function, it takes as inputs the currents and capacitor voltages and outputs the reference voltages to be applied to each HB cell. Hence, if there are 3 SMs in the upper arm, and 3 SMs in the lower arm, the function will require 6 input voltages, 2 input currents (one per arm), and will output 6 reference voltages (one per cell).

It is intended for the controller to be executed every $100\mu s$, using a double-update method being able to change the duty cycle in the highest and lowest value of the triangular waveform [18].

Focusing on a single phase, this regulator can be divided into three sections: (i) branch energy control, (ii) branch current control, and (iii) capacitor balancing. Fig. 2 shows the first loop where the energy of each arm is compensated by the currents $i^{u,l}_e$.

The lower arm is controlled to create a DC voltage plus an AC voltage (V_{DC} and V_{AC} respectively) [17]. V_{in} is

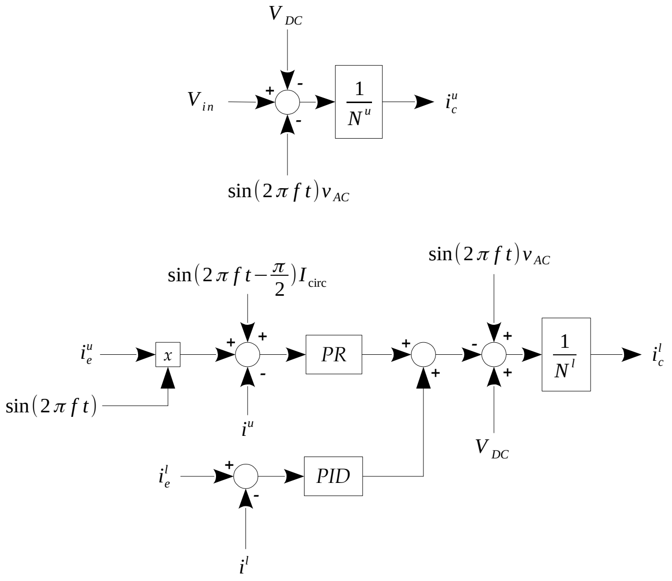


Fig. 3. Current controllers

the input DC voltage, $V^u_{caps_n}$ and $V^l_{caps_n}$ are the upper and lower capacitor voltages, ref^u and ref^l the reference values calculated using (7), where $V^{u,l}_{caps_{ref}}$ are the voltages at which the capacitors want to be kept.

$$ref^{u,l} = N^{u,l} \left(V^{u,l}_{caps_{ref}} \right)^2 \tag{7}$$

The control diagram utilised to control the arm currents $i_c^{u,l}$ is shown in Fig. 3, where f is the frequency of the desired output voltage ($100Hz$), t is the time in seconds, $N^{u,l}$ are the number of SMs in the upper and lower arms, and $i^{u,l}$ are the measured upper and lower currents.

Fig. 4 shows the upper and lower capacitor balancing controllers, which are replicated depending on the number of HB cells n. $V^{u,l}_{caps_n}$ are each measured capacitor voltage and $V^{u,l}_{ref_n}$ the voltage to be generated by each SM. This last value will be fed into the *PWM generator* to modulate the SM according to the required output voltage.

IV. MMC DISCRETE MODEL

The function that emulates the PSCAD detailed model is quite simple. It takes as inputs the total DC voltage V_{in} and the PWM signals, and from these it determines the currents through the upper and lower arms $i^{u,l}$, the capacitor voltages V_{caps_n} and the output voltage at the middle of the leg V_{out}.

The capacitor voltages are updated at each iteration using (8), where T_s is the time-step, and S_n is a logic function which depends on the corresponding PWM signal.

$$V^{u,l}_{caps_n}(k) = V^{u,l}_{caps_n}(k-1) + S_n(k-1) \frac{T_s}{C} i^{u,l}(k-1) \tag{8}$$

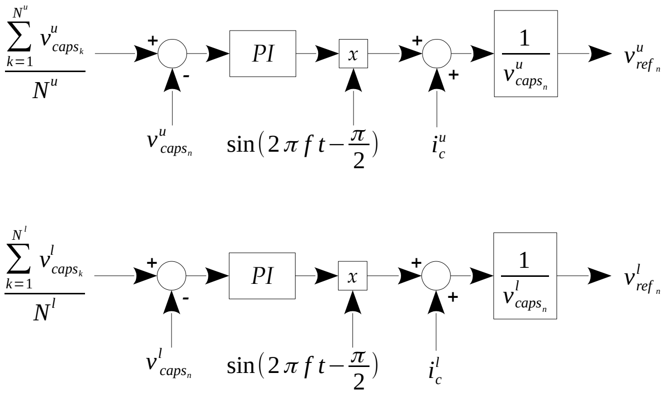

$\frac{\sum_{k=1}^{N^u} v^u_{caps_k}}{N^u}$
+
-
PI
x
+
+
$\frac{1}{v^u_{caps_n}}$
$v^u_{ref_n}$
$v^u_{caps_n}$
$\sin(2\pi f t - \frac{\pi}{2})$
i_c^u
$\frac{\sum_{k=1}^{N^l} v^l_{caps_k}}{N^l}$
+
-
PI
x
+
+
$\frac{1}{v^l_{caps_n}}$
$v^l_{ref_n}$
$v^l_{caps_n}$
$\sin(2\pi f t - \frac{\pi}{2})$
i_c^l

Fig. 4. Capacitor balancing controllers

Then, using these results, the equivalent arm voltages are calculated using (9), where $N^{u,l}$ is the number of SMs per arm.

$$V^{u,l}_{arm}(k) = \sum_{n=1}^{N^{u,l}} S_n(k) V^{u,l}_{caps_n}(k) \tag{9}$$

All systems subjected to a simulation in state-space representation [19] are written as:

$$\begin{aligned} \dot{x}(t) &= \mathrm{A}x(t) + \mathrm{B}u(t) \\ y(t) &= \mathrm{C}x(t) + \mathrm{D}u(t) \end{aligned}$$

Hence, if the the total DC voltage V_{in} and the arm voltages $V^{u,l}_{arm}$ are taken as inputs, arranging the system equations following the form presented above leads to (10) and (11).

$$\begin{bmatrix} \dot{i}^u(t) \\ \dot{i}^l(t) \end{bmatrix} = \mathrm{A} \begin{bmatrix} i^u(t) \\ i^l(t) \end{bmatrix} + \mathrm{B} \begin{bmatrix} V_{in}(t) \\ V^u_{arm}(t) \\ V^l_{arm}(t) \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} i^u(t) \\ i^l(t) \\ V_{out}(t) \end{bmatrix} = \mathrm{C} \begin{bmatrix} i^u(t) \\ i^l(t) \end{bmatrix} \tag{11}$$

where A, B, and C are:

$$\begin{aligned} \mathrm{A} &= \begin{bmatrix} {}^{-(R_{eq}+R_{load})}/_{L_{arm}} & {}^{R_{load}}/_{L_{arm}} \\ {}^{R_{load}}/_{L_{arm}} & {}^{-(R_{eq}+R_{load})}/_{L_{arm}} \end{bmatrix} \\ \mathrm{B} &= \begin{bmatrix} {}^{1}/_{L_{arm}} & -{}^{1}/_{L_{arm}} & 0 \\ 0 & 0 & -{}^{1}/_{L_{arm}} \end{bmatrix} \\ \mathrm{C} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ R_{load} & -R_{load} \end{bmatrix} \end{aligned}$$

and L_{arm} the arm inductance, R_{load} the charge connected to the middle point, and R_{eq} the equivalent value considering the ON resistance of the IGBTs and anti-parallel diodes plus the parasitic resistance of the arm inductors.

In order to alleviate computational burden, like both IGBTs and anti-parallel diodes have similar ON resistances, the R_{eq} present in the A matrix has been set to the constant value:

$$R_{eq} = N^{u,l} R_{on} + R_L$$

where $N^{u,l}$ is the number of HB cells per branch, R_{on} an average value of the ON resistance of the power electronics, and R_L the parasitic resistance of the arm inductors. By doing this, the A matrix remains constant and therefore there is no need to calculate it every time an HB changes its state.

Discrete-time state-space matrices have been obtained using the ZOH equivalent of their continuous-time counterparts using (12) and (13) for $T_s = 5\mu s$.

$$\mathrm{E} = e^{\mathrm{A}T_s} \tag{12}$$

$$\mathrm{F} = \mathrm{A}^{-1}\left(e^{\mathrm{A}T_s} - \mathrm{I}\right)\mathrm{B} \tag{13}$$

where I is the identity matrix. Performing these transformations, the discrete-time state-space model leads to:

$$\begin{aligned} x\left(k+1\right) &= \mathrm{E}x\left(k\right) + \mathrm{F}u\left(k\right) \\ y\left(k\right) &= \mathrm{C}x\left(k\right) \end{aligned}$$

The system parameters used are taken from an experimental prototype that is being built.

V. HIL SIMULATION

A. HLS implementation

The MMC C model was coded using Vivado HLS in order to generate an IP block containing the considered inputs and outputs of the system. Specifically, it takes as inputs the DC input voltage fed to the converter and the PWM signals and computes all the HB capacitor voltages, the currents of the upper and lower arms, and the output voltage in the middle connection point of the phase leg.

For an IP considering 6 SMs per leg, this block performs 13 multiplications, 2 divisions, 18 additions, and 1 subtraction in 88 cycles which correspond to $880ns$ with a running clock of $100MHz$. Regarding area utilisation, 1 BRAMs, 29 DSPs, 6.095 FF, and 6.583 LUT were needed, which correspond to 1%, 13%, 6% and 12% of the total hardware resources of the *Zynq-7020*. Therefore, there is still room in order to either implement all three phases or increment the number of SMs per arm. However, like the regulator under evaluation is only controlling one converter leg, there was no point on implementing all three phases. Verifying the proper functioning of one phase was sufficient in this case.

B. Hardware/Software co-design

A diagram of the whole system is shown in Fig. 5. As previously stated, the controller is placed in the PS whereas the MMC HIL IP and the PWM generator is left in the FPGA fabric. The input DC voltage is read from a file placed in the SDcard. This way, any voltage profile can be generated and used to test how the model and the controller responds to noisy signals, voltage blackouts, oscillations, etc. The MMC HIL IP has been configured with two *AXI4-Master* ports that store automatically its 9 results (6 capacitor voltages $V^{u,l}_{caps_n}$, 2 branch currents $i^{u,l}$, and the output voltage V_{out}) into two different memory spaces without any intervention of the processor. The DDR memory is utilised as data logger through the *High Performance* (HP) port, where the IP block increases automatically the writing address on every iteration in order to avoid the PS from doing it. Once the simulation has finished, this data is transferred into the SDcard for further verification of the results using MATLAB. The *On-Chip-Memory* (OCM) is also utilised but this time through the *Accelerator Coherency Port* (ACP). This way, the controller can access the results achieving minimum latency and keeping coherency with the *L1* chaches [20].

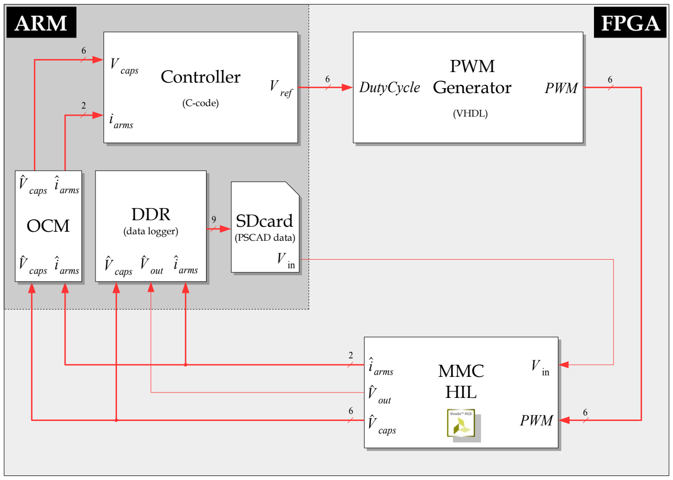


Fig. 5. MMC IP working as HIL

Regarding the controller, as said, it reads capacitor voltages and currents from the OCM, then it computes the control commands, and finally it writes the calculated duty cycle to be applied to every SM using an *AXI4-Lite* into the PWM Generator.

An interrupt was configured to halt every $5\mu s$ and start the MMC IP calculations, ensuring synchronization between all subsystems and hence a real-time execution. The controller function though was called every $100\mu s$. It then sends the duty cycles of every SM to the PWM Generator IP using 32-bit floating-point variables. Next, the PWM Generator waits the sync signal and applies the corresponding gate signals to the converter IP.

C. HIL control validation

Figs. 6 and 7 show a power up test, where the next stages can be seen:

1) from $0s$ to $1.5s$ the input voltage is raised up to $311V$ (Fig. 6 up), keeping the duty cycle of the PMW signals at 100%, thus the capacitors start charging (Fig. 7)
2) from $1.5s$ to $3s$ the duty cycle of all the cells is reduced gradually from 100% to 48% increasing the voltage of the capacitors til $100V$. Notice the increase of the arm currents in Fig. 6
3) from $3s$ to $4.5s$ the duty cycle is kept to 48% letting the capacitor voltages diverge as seen in Figs. 7 and 8.
4) at $4.5s$ and until $5.5s$ the controller is activated without the integral part. Hence, the voltages are kept balanced but not exactly at $100V$ as seen in Fig. 8. Fig. 6 shows

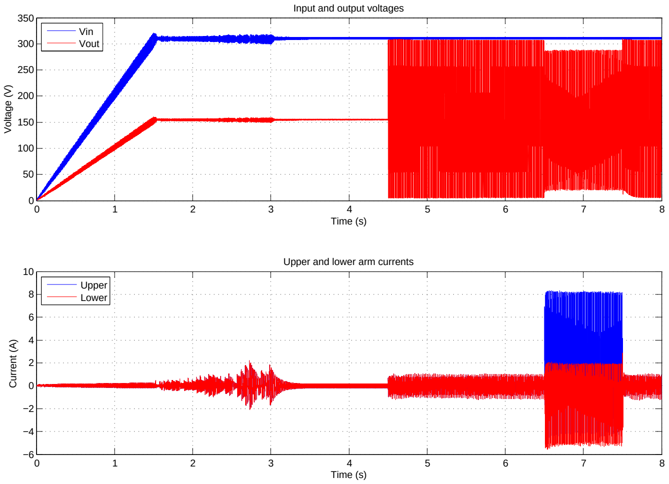


Fig. 6. Input voltage and HIL estimated output voltage and currents - Complete simulation

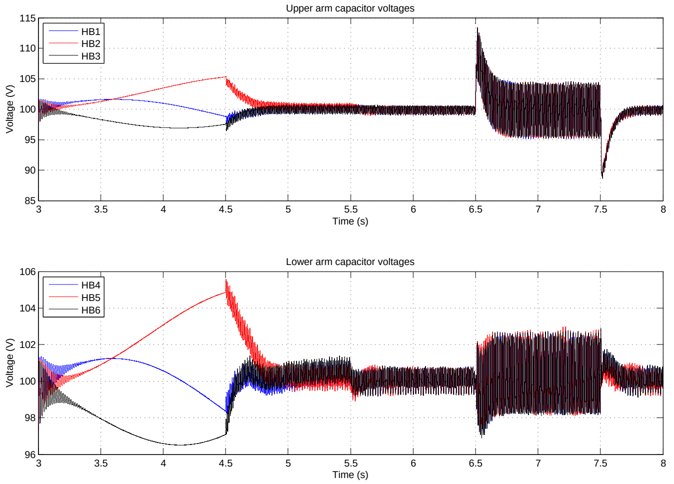


Fig. 8. HIL estimated capacitor voltages - Capacitor oscillations

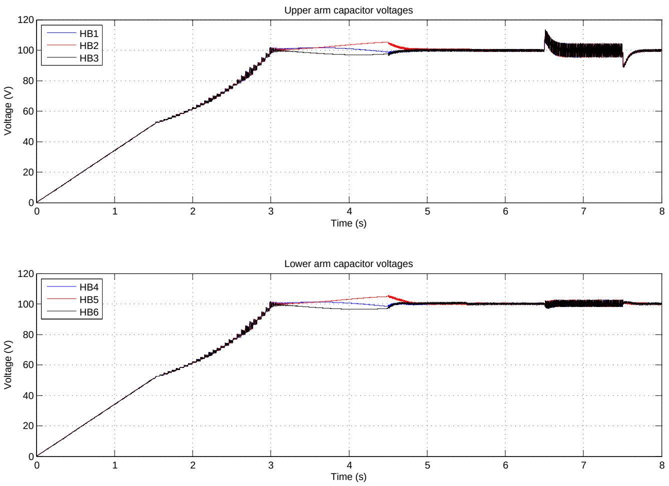


Fig. 7. HIL estimated capacitor voltages - Complete simulation

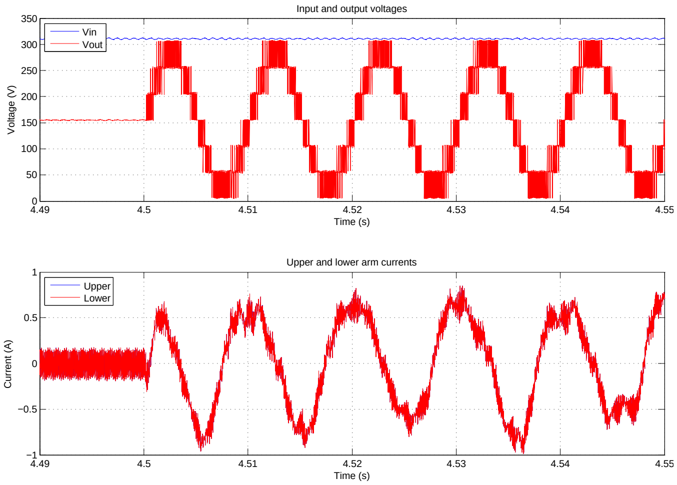


Fig. 9. Input voltage and HIL output voltage and currents. At $4.5s$ the control starts operating

how the output voltage changes from $155V$ DC to $125V$ AC plus $155V$ DC at $100Hz$

5) at $5.5s$ and until $6.5s$ the integral part of the regulators is activated driving all the capacitor voltages to $100V$
6) at $6.5s$ a 44Ω load is introduced (i.e. the HIL IP switches to *normal operation mode*). Fig. 6 down shows how the currents increase significantly. Observe as well in Figs. 7 and 8 how the capacitor ripple increase
7) at $7.5s$ the load is removed, the capacitor ripple decrease and the controller sets their voltages back to $100V$

Fig. 9 shows at $4.5s$ the exact moment when the control starts operating, showing the typical 7 levels waveform produced by the MMC configuration.

Fig. 10 presents the change at $6.5s$ from no-load to a 44Ω load. The smoothing of the output voltage when the load is connected is caused by the arm inductances L_{arm}. The voltage produced by the SMs is a PWM signal, but when the load current is increased, all the noise caused by the PWM infers an important voltage drop in the inductances. This latter, added to the voltage created by the HBs results in a sinusoidal-like waveform. Hence, these inductors, apart from filtering the output current, also filter the output voltage. In the same figure can be observed as well how the upper branch is providing much more current than the lower arm. Fig. 11 shows the opposite case, where at $7.5s$ the load resistor is removed, thus disappearing the voltage filtering and appearing again the 7-levels common in this kind of converters.

It is shown that the controller performs properly keeping the capacitor voltages under control and providing the required output voltage.

VI. Conclusions

In this paper, the use of SoC devices for HIL controller validation was presented. A complete MMC model which estimates output voltage, branch currents, and capacitor voltages based on the PWM signals and the input DC voltage was

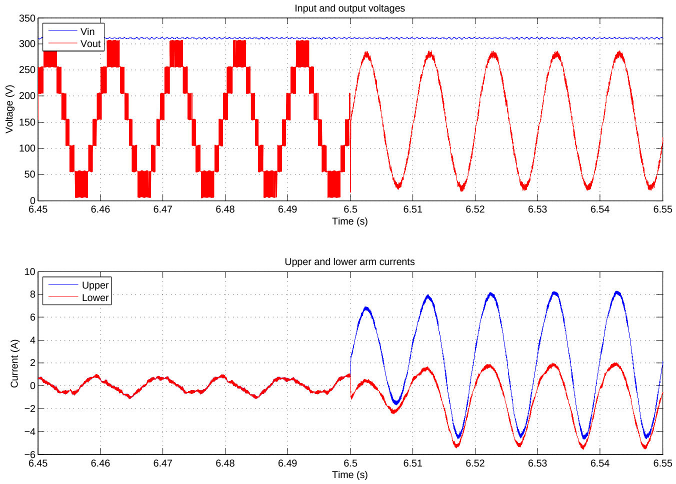


Fig. 10. Input voltage and HIL estimated output voltage and currents. Load switched in at $6.5s$

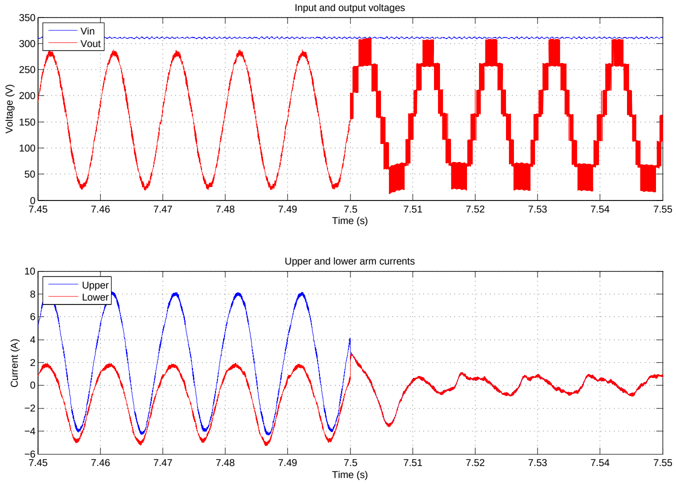


Fig. 11. Input voltage and HIL estimated output voltage and currents. Load switched out at $7.5s$

developed and implemented in the FPGA fabric side. The controller though was run in the ARM processor. Then, all the hardware/software co-design necessary to interconnect and communicate efficiently the MMC IP with the controller was set up. The PWM block, also placed in the PL, receives the duty cycle values of each SM using the *AXI4-Lite* protocol, whereas the MMC results are automatically written into the OCM for the lowest latency utilisation by the controller, and also into the DDR memory for data logging purposes. An interrupt was configured to halt every $5\mu s$ to ensure real-time operation.

This IP has been used in the HIL context in order to validate the converter's controller in real-time before its deployment on the real test rig. The controller performed properly keeping the system's state variables under control and providing the required output voltage to the load.

Even though the SoC device utilised in this experimental setup is considered as low/mid-range, the FPGA fabric resources utilised were all below 15%. This can give an idea of how suitable these systems are to be used as HIL validation of MMC with a low number of SMs.

References

[1] *DS190: Zynq-7000 All Programmable SoC Overview*, Xilinx Inc.

[2] *Intel User-customizable SoC FPGAs*, Intel Inc., 2018.

[3] *DS0112: SmartFusion Customizable System-on-Chip (cSoC)*, Microsemi Corp., mar 2015.

[4] *UG902: High-Level Synthesis with Vivado HLS*, Xilinx Inc., 2017.

[5] F. Montano, T. Ould-Bachir, and J.-P. David, "An evaluation of a High-Level Synthesis approach to the FPGA-based sub-microsecond real-time simulation of power converters," *IEEE Transactions on Industrial Electronics*, 2017.

[6] M. Matar and D. Iravani, "FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 852–860, apr 2010.

[7] J. Liu and V. Dinavahi, "A real-time nonlinear hysteretic power transformer transient model on FPGA," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3587–3597, jul 2014.

[8] H. Saad, T. Ould-Bachir, J. Mahseredjian, C. Dufour, S. Dennetiere, and S. Nguefeu, "Real-time simulation of MMCs using CPU and FPGA," *IEEE Transactions on Power Electronics*, vol. 30, no. 1, pp. 259 – 267, Jan 2015.

[9] M. Matar, D. Paradis, and R. Iravani, "Real-time simulation of Modular Multilevel Converters for controller Hardware-in-the-Loop testing," *IET Power Electronics*, vol. 9, no. 1, pp. 42–50, jan 2016.

[10] J. Sawma, F. Khatounian, E. Monmasson, R. Ghosn, and L. Idkhajine, "Evaluation of the new generation of System-on-Chip platforms for controlling electrical systems," *Industrial Technology (ICIT), 2015 IEEE International Conference on*, pp. 1570–1575, March 2015.

[11] M. Ricco, M. Gheorghe, L. Mathe, and R. Teodorescu, "System-on-Chip implementation of embedded real-time simulator for Modular Multilevel Converters," in *Energy Conversion Congress and Exposition (ECCE), IEEE*, 2017.

[12] D. Tormo, R. Vidal-Albalate, L. Idkhajine, E. Monmasson, and R. Blasco-Gimenez, "Study of System-on-Chip devices to implement embedded real-time simulators of Modular Multi-level Converters using High-Level Synthesis tools," in *IEEE International Conference on Industrial Technology*, 2018.

[13] T. Liang and V. Dinavahi, "Real-time device-level simulation of mmc-based mvdc traction power system on mpsoc," *IEEE Transactions on Transportation Electrification*, vol. 4, no. 2, pp. 626–641, jun 2018.

[14] A. Lesnicar and R. Marquardt, "An innovative Modular Multilevel Converter topology suitable for a wide power range," in *Power Tech Conference Proceedings, IEEE Bologna*, 2003.

[15] L. Angquist, A. Antonopoulos, D. Siemaszko, K. Ilves, M. Vasiladiotis, and H. Nee, "Open-loop control of Modular Multilevel Converter using estimation of stored energy," *IEEE Transactions on Industry Applications*, vol. 47, no. 6, pp. 2516–2524, 2011.

[16] L. Harnefors, A. Antonopoulos, S. Norrga, L. Angquist, and H. Nee, "Dynamic analysis of Modular Multilevel Converters," *IEEE Transactions in Industrial Electronics*, vol. 60, no. 7, pp. 2526–2537, 2013.

[17] R. Vidal-Albalate, J. Barahona, D. Soto-Sanchez, E. Belenguer, R. Pena, and R. Blasco-Gimenez, "A modular multi-level DC-DC converter for HVDC grids," in *Industrial Electronics Society , IECON - 42nd Annual Conference of the IEEE*, 2016.

[18] N. Hoffmann, F. W. Fuchs, and J. Dannehl, "Models and effects of different updating and sampling concepts to the control of grid-connected PWM converters - A study based on discrete time domain analysis," in *Power Electronics and Applications (EPE 2011), Proceedings of the 2011-14th European Conference on*, 2011.

[19] W. L. Brogan, *Modern Control Theory*, 1st ed. Quantum Publishers, 1974.

[20] M. Sadri, C. Weis, N. Wehn, and L. Benini, "Energy and performance exploration of accelerator coherency port using xilinx Zynq," in *FPGA-World'13 Copenhagen and Stockholm*, 2013.