Validation of a Multicore Microcontroller-Based HIL Using an OSV-MPC Controller ⋆

Bernardo Figueiredo de Andrade Campos * **Matheus Sotero** **
Luís Guilherme Barbosa Rolim * **Lucas Frizera Encarnação** ***

* *Universidade Federal do Rio de Janeiro (e-mails: bernardo.campos@coppe.ufrj.br, rolim@poli.ufrj.br)*
** *Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (e-mail: matheus.ribeiro@cefet-rj.br)*
*** *Universidade Federal do Espírito Santo (e-mail: lucas.encarnacao@ufes.br)*

Abstract: In this work, a multicore microcontroller-based hardware-in-the-loop (HIL) simulation scheme is implemented, using both cores of a Texas Instruments F-28379D microcontroller: one core holds the control algorithm, while the other holds the power system mathematical model. Steady- and transient-state real-time simulations of a three-phase two-level inverter feeding a set of linear loads through an LC filter are carried out, and the results obtained are compared with Simulink offline results in terms of output voltage mean absolute error (MAE). An optimal switching vector model predictive control (OSV-MPC) algorithm is implemented to control the electrical plant.

Keywords: Real-time simulations, microcontroller-based hardware-in-the-loop, model predictive control.

1. INTRODUCTION

Power electronics simulation techniques have been dramatically improved with the advances in computing over the last decades, so that the average simulation tools are sophisticated enough to accurately represent most of the transient and steady state phenomena involved in an electrical system simulation. Digital simulation has reached small sampling times, enabling the resolution of increasingly complex problems. At the same time computer technologies have experienced a decrease in cost, making basic offline simulation tools more accessible (J. Bélanger and Paquin, 2010).

In this scenario, real-time simulations have emerged as a mandatory step in order to reliably test a wide variety of systems before final commissioning, such as controllers in electrical systems or mechanical systems of cars and aircrafts. Differently from the offline case, in which the simulation time does not coincide with real physical time, in real-time simulations the tested hardware and software must be able to produce accurate results within the defined sampling time, actuating as they would on a practical application (Dinavahi, 2000).

Currently, the hardware-in-the-loop (HIL) scheme dominates real-time simulations. For testing controllers in electrical systems with HIL schemes, a microcontroller holding the control algorithm is connected to a virtual plant run in a real-time environment. By the real-time approach, it is possible to arbitrarily stress control systems without damage, perform any kind of factory testing and test black-box algorithms (Dicler, 2021).

HIL simulation has been widely adopted for power electronics and is mainly done with schemes that use central processing unities (CPUs), field programmable gate arrays (FPGAs) or a combination of both. Commercial equipment are usually CPU-based, such as Typhoon HIL or OPAL-RT, but depending on the simulator, FPGA modules may also be utilized if needed. The main differences between those two technologies reside in the fact that FPGA provides a higher computational power than CPU, thus suiting better for applications that require extremely low sampling times, such as high frequency electromagnetic transient, as in power electronics simulations.

Sotero et al. (2021), for example, presents an FPGA-based real-time simulator for power electronics converters with sampling times of a few microseconds, having achieved good accuracy in relation to PSCAD offline simulations for a DC-DC boost converter and for a three-phase PWM inverter. In Bieber et al. (2023), the authors develop an universal equivalent model of hybrid cascaded multilevel converters for FPGA/CPU-based real-time simulations, and in Qi et al. (2021) the authors design a custom high fidelity HIL test platform to study a grid-forming wind turbine.

A lesser used scheme is the microcontroller-based HIL, in which the controller and the real-time simulator are implemented in microcontrollers. Microcontrollers are also CPU-based, but they have significantly less computational power and are much more accessible compared to the aforementioned. Such scheme may suit well for the modeling

⋆ This research was supported in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under Grant 001.

and test of small systems with sampling times within the range of dozens of microseconds. Other convenient features of this scheme are the possible peripherals, as PWMs, analog-to-digital converters (ADCs) and serial communication interfaces.

In this work, a multicore HIL simulation scheme using both cores of a Texas Instruments F28379D microcontroller is implemented: one core holds the control algorithm, while the other holds the power system model. This configuration is selected due to the fact that it uses only one easily accessible microcontroller, being very low cost and constituting a scheme that makes it easier for more researchers to carry out their HIL simulations. To validate the results obtained, Simulink offline simulations were also carried out.

2. MICROCONTROLLER-BASED HARDWARE-IN-THE-LOOP

Microcontroller-based simulations have emerged as viable low-cost alternatives for real-time validation of small systems since the late 1990s and the beggining of the 2000s, as evidenced by some works like Jack et al. (1998), Crosbie et al. (2004), Lu et al. (2007) and Liu et al. (2006). Each of them employed a setup featuring multiple microcontrollers (and occasionally FPGAs) with the objective of conducting a HIL simulation for power electronics converters.

Over the time, the improvement of microcontroller technologies has allowed improved arrangements to real-time simulations based in this technology, as in Choe et al. (2014), Castro et al. (2019), de Sá Santos Ribeiro et al. (2019), Ranjan et al. (2021), Mohammadi et al. (2022), and Sotero and Rolim (2023). These arrangements use less devices and can also achieve faster sampling times.

Considering these recent advancements, this work builds upon the arrangement proposed by Sotero and Rolim (2023) to enable a multicore F28379D-based HIL simulation. This setup, as explained in Sotero and Rolim (2023), utilizes the two available CPU cores in the F28379D. One core is dedicated to executing the simulated model in real-time, while the other handles the implementation of the control algorithm undergoing commissioning.

In addition to the two CPUs, four DSP peripheral modules are also employed to implement this microcontroller-based HIL: ePWM, IPC, GPIO and DAC modules. In the subsections below, each component's role is explained, and Figure 1 presents a functional diagram of the general operation of the proposed system.

2.1 CPU01 - Controller

The code designated for CPU01 execution solely comprises a control algorithm, specifically the Optimal Switching Vector Model Predictive Control (OSV-MPC) in this work. Alternative controller types can also undergo testing within this framework.

Consequently, since it employs OSV-MPC, the CPU01 interrupt program essentially encompasses five stages. Initially, system measurements are acquired, and voltage references for the three-phase bus are computed. Subsequently, the OSV-MPC algorithm is invoked to determine the optimal converter switching state. Lastly, the initial interruption-measured signals are exported, and the simulation time on CPU01 is updated. Specific information regarding controller modeling is given in Section 3.

2.2 CPU02 - System Model

In the second core of the microcontroller, at each interruption of the core it is necessary to execute a time step of the model that solves the system being simulated in real time. In this work, the system under consideration is a three-phase inverter feeding a set of linear loads through an LC filter.

To execute this simulation step upon CPU02 interruption, four essential steps are taken. Initially, the controller's issued switching state is retrieved. Next, the system solution is determined by employing the Modified Nodal Analysis (MNA) algorithm. Subsequently, the simulated time step results are extracted from the simulator core, followed by an update to the simulation time.

For modeling the three-phase inverter, a simplified switching function model is employed, obviating the need for Modified Augmented Nodal Analysis (MANA). Further details regarding the electrical system model are expounded upon in Section 3.

2.3 ePWM Module

The objective of the ePWM module is to periodically trigger the interrupt functions that are responsible for executing the control (in CPU01) and the system model (in CPU02), since both use discretization mathematical models and these simulation time steps need to be synchronized with real time.

To achieve this synchronization, two units (ePWM6 and ePWM7) are configured to operate independently. This setup ensures that the control and system model function autonomously without temporal interdependence. Specifically, ePWM7 triggers the control interrupt routine on CPU01, while ePWM6 triggers the model solution interrupt routine. Each ePWM is configured to generate one interrupt trigger per period.

2.4 IPC Module

The LAUNCHXL-F28379D provides only two accessible digital-analog converters (DACs) for utilization. Consequently, any hardware-in-the-loop setup requiring more than two feedback signals for the control loop cannot be executed with only the microcontroller DACs. This limitation arises as only two signals originating from the system model solution can be exported from CPU02 via DACs.

To overcome this problem, the interprocessor communication (IPC) peripheral is utilized. Within this module, the three IPC registers responsible for transmitting data from CPU02 to CPU01 (`DATA`, `ADDR`, and `COM`) are adjusted to export all necessary signals from the system model to the control system.

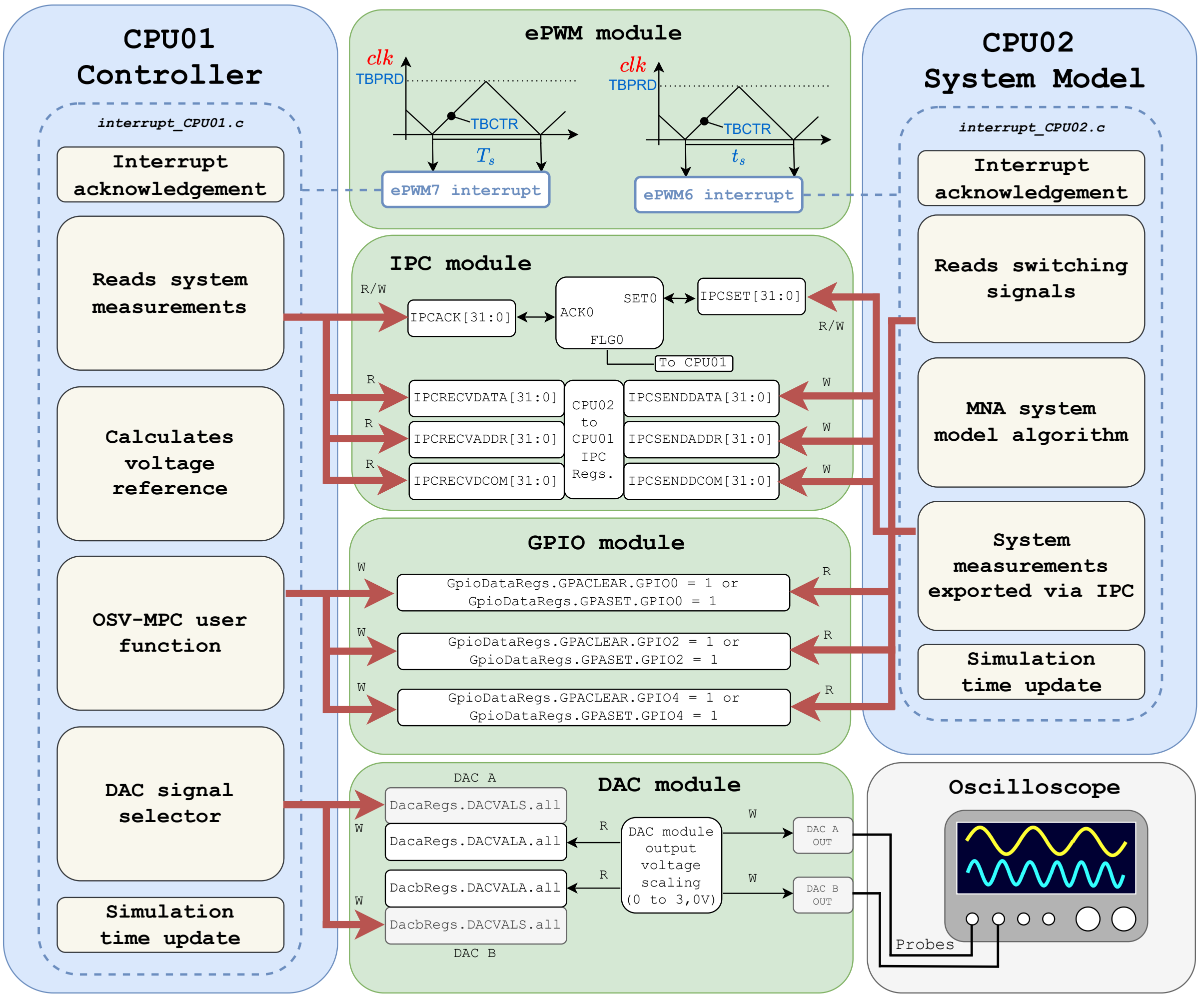


Figure 1. Functional diagram of the proposed multicore microcontroller-based HIL.

2.5 GPIO Module

To control the three-phase inverter, the system model must receive the controller's switching state order as an input signal. To send the switching signals from the controller to the plant, the GPIO peripheral is configured. Within this module, three I/O pins (`GPIO0`, `GPIO2`, and `GPIO4`) are configured as outputs from CPU01 (for writing) and simultaneously as inputs from CPU02 (for reading).

2.6 DAC Module

To visualize HIL results on an oscilloscope, it is necessary to export the desired signals through the DAC peripheral configured on CPU01. However, as previously noted, the hardware features only two DAC channels (DAC1 and DAC2) available. Thus, each one of these channels is set alongside a variable selector. This enables the visualization of multiple signals two by two, by changing the variable selector in the microcontroller debug interface.

3. CONTROL AND SYSTEM MODEL

With the development of more powerful hardware, the Optimal Switching Vector Model Predictive Control (OSV-MPC) has emerged as a possible alternative to classic controllers for the control of power electronic converters and has been widely investigated for this purpose since the early 2000s (Rodriguez et al., 2005, 2007).

OSV-MPC uses the discrete nature of power converters to evaluate the controlled variables on a set of possible switching states over a given predictive time horizon, based on a model of the system. In every sampling time, the switching state that synthesizes a signal with minimum error compared to the reference signal is directly applied to the converter, a step that results from the minimization of a cost function.

The works in Young et al. (2014); Geyer (2011) demonstrated that the OSV-MPC provides faster dynamic responses than classic controllers, while achieving similar steady-state performance. It may be noted, however, that the dynamics on the OSV-MPC are limited by the system complexity, because high complexity systems may require either larger sampling times or model simplifications. This control strategy also has its drawbacks, such as variable switching frequency, non-zero stationary error and the great amount of calculating required.

In this work, the OSV-MPC is implemented for voltage control in a system composed of a three-phase two-level inverter feeding a load through an LC filter. The DC link is considered to be a constant voltage source.

ISSN: 2525-8311

DOI: 10.20906/CBA2024/4424

A great computational power is required for simulating the switches of the three-phase two-level inverter in real time. This is because several large matrices are needed to represent the possible system configurations that result from the different switching states, and these matrices have to be stored and processed within a few microseconds by the simulator.

Since in this work the entire simulation is run in a microcontroller-based HIL, the inverter switching function model, Figure 2, proposed by Yazdani and Iravani (2010), is used to reduce the computational complexity of the system, at the expense of losing the representation of the switches' dynamics.

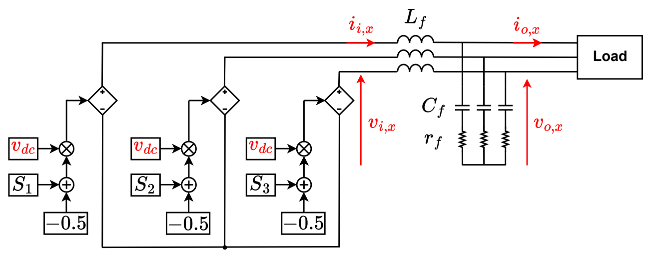


Figure 2. System model representation using the inverter switching function model.

In Figure 2, x is the phase index ($x = a, b, c$), $v_{i,x}$ represents the inverter three-phase voltages, $i_{i,x}$ represents the inverter three-phase currents, $v_{o,x}$ represents the three-phase output voltages to be regulated over the LC filter capacitor and $i_{o,x}$ are the three-phase load currents. L_f and C_f are the filter inductance and capacitance, respectively, and r_f is the filter damping resistance. v_{dc} is the DC link voltage.

A two-step forward predictive model is implemented, to compensate the one sampling time delay associated with the digital implementation of the OSV-MPC, extending the control predictive horizon to two sampling times, as done in Zheng et al. (2021):

$$\begin{bmatrix} \vec{i}^{\,p}_{i(k+1)} \\ \vec{v}^{\,p}_{o(k+1)} \end{bmatrix} = \boldsymbol{G_{2x2}} \begin{bmatrix} \vec{i}_{i(k)} \\ \vec{v}_{o(k)} \end{bmatrix} + \boldsymbol{H_{2x2}} \begin{bmatrix} \vec{v}^{\,p}_{i(k+1)} \\ \vec{i}_{o(k)} \end{bmatrix}, \quad (1)$$

$$\begin{bmatrix} \vec{i}^{\,N}_{i(k+2)} \\ \vec{v}^{\,N}_{o(k+2)} \end{bmatrix} = \boldsymbol{G_{2x2}} \begin{bmatrix} \vec{i}^{\,p}_{i(k+1)} \\ \vec{v}^{\,p}_{o(k+1)} \end{bmatrix} + \boldsymbol{H_{2x2}} \begin{bmatrix} \vec{v}^{\,N}_{i(k+1)} \\ \vec{i}_{o(k)} \end{bmatrix}. \quad (2)$$

In (1) and (2), $\boldsymbol{G_{2x2}}$ and $\boldsymbol{H_{2x2}}$ are the coefficient matrices, resulting from the system's discrete state-space formulation, p denotes the index of the optimal switching state selected on the previous control iteration, and $\vec{v}^{\,p}_{i(k+1)}$ is the corresponding voltage vector, that is used to calculate the $[k+1]$ predictions $\vec{i}^{\,p}_{i(k+1)}$ and $\vec{v}^{\,p}_{o(k+1)}$ at the beginning of each control iteration.

$\vec{i}^{\,p}_{i(k+1)}$ and $\vec{v}^{\,p}_{o(k+1)}$ are then applied to the model, and the eight possible voltage vectors for a three-phase inverter, $\vec{v}^{\,N}_{i(k+1)}$, are evaluated (N=8). $\vec{i}^{\,N}_{i(k+2)}$ and $\vec{v}^{\,N}_{o(k+2)}$ are the eight $[k+2]$ predictions for the inverter currents and output voltages, respectively. The formulation of the predictive model is done in the $\alpha\beta$ frame, hereby represented by complex vector notation. The mathematical development of this control strategy is out of the scope of this work, and may be found in detail in Zheng et al. (2021). The two-step forward prediction delay compensation technique is originally proposed in Rodriguez and Cortes (2012).

In this book chapter, the authors also present a method to compensate the reference tracking delay, by estimating the signal value at $[k+2]$ for smoothly changing reference signals, the Lagrange extrapolation, which is also implemented. The resulting cost function to be minimized is:

$$g_i^N = |\vec{v}^{\,*}_{o(k+2)} - \vec{v}^{\,N}_{o(k+2)}|. \quad (3)$$

In (3), $\vec{v}^{\,*}_{o(k+2)}$ is the extrapolated reference, which is sinusoidal and constant in amplitude and frequency. The switching state that minimizes the error vector g_i^N is directly applied to the converter.

4. SIMULATION RESULTS

Table 1 summarizes the system voltage and power levels we used in the real-time simulations, as well as the controller and solver sampling times.

Table 1. System voltage and power levels and simulation sampling times.

Parameter	Symbol	Value	Unit
DC-link voltage	V_{dc}	450	V
Base load power (three-phase)	$S^{3\phi}_{base}$	10	kVA
Base voltage (line-to-line, RMS)	V_{base}	220	V
Base load current (RMS)	I_{base}	26.24	A
Base load impedance	Z_{base}	4.84	Ω
AC grid fundamental frequency	f	50	Hz
Controller sampling time	T_s	50	μs
Maximum switching frequency	f_{max}	10	kHz
Solver sampling time	t_s	50	μs

No load, resistive (R), and unbalanced RL load conditions are considered for the steady- and transient-state analysis. For the unbalanced load scenario, a large resistor was connected to phase b, a series RL inductive load was applied to phase a and a resistor to phase c, remaining no neutral wire connected. To avoid numerical instabilities due to a poor conditioning of the system's solution matrix, large resistors are used to simulate the no load condition. Table 2 summarizes the apparent, active and reactive powers associated to each load.

Table 2. Load powers.

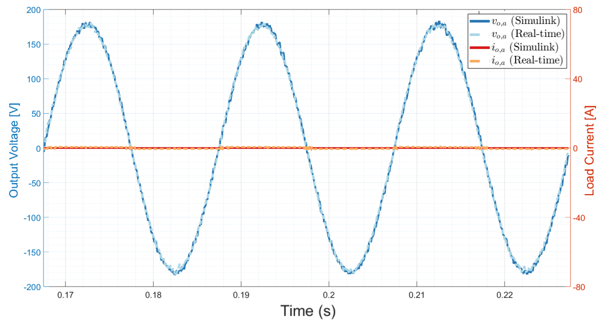
Load type	S_{load} [kVA]	P_{load} [kW]	Q_{load} [kvar]
No load	0.00	0.00	0.00
Resistive	10.00	10.00	0.00
Unbalanced (phase a)	2.31	1.85	1.38
Unbalanced (phase c)	3.33	3.33	0.00

4.1 Steady-state simulation results

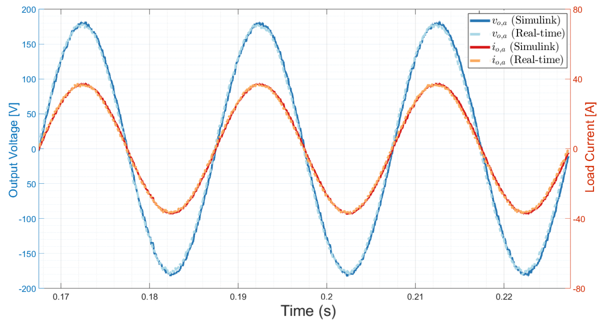
Simulations comprising the three load conditions are implemented, to evaluate the steady-state performance of the multicore F28379D-based HIL. The real-time results are

ISSN: 2525-8311

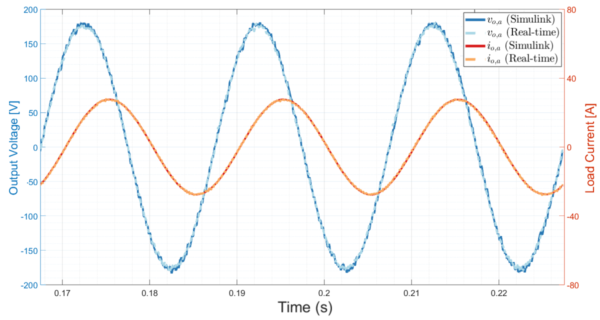
DOI: 10.20906/CBA2024/4424



(a) No load

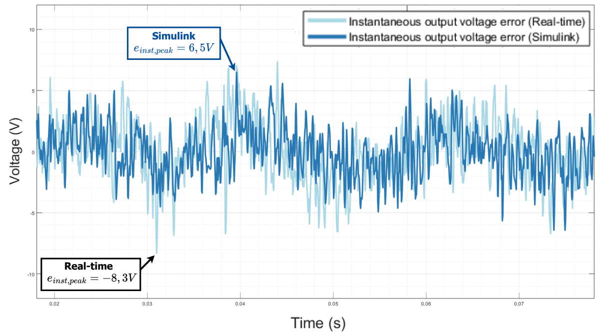


(b) R load

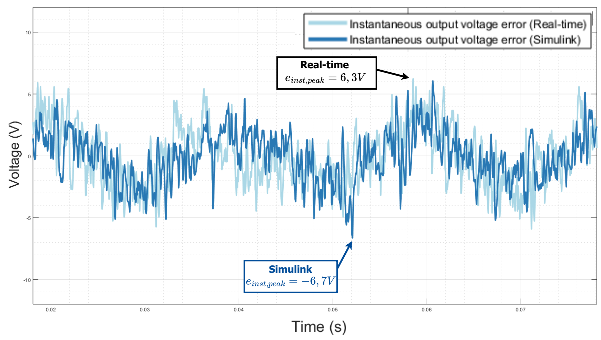


(c) Unbalanced RL load

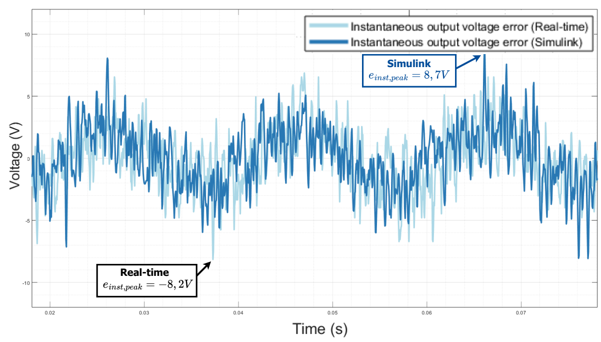
Figure 3. Output voltage and load current (phase a) synthesized by the controller in Simulink and in the multicore F28379D-based HIL for different load scenarios.



(a) No load



(b) R load



(c) Unbalanced RL load

Figure 4. Output voltage and load current (phase a) synthesized by the controller in Simulink and in the multicore F28379D-based HIL for different load scenarios.

compared to Simulink offline simulation results, in terms of output voltage mean absolute error (MAE). Figure 3 presents a visual comparison of the (phase a) output voltage and load current synthesized by the controller for the no load, R, and unbalanced RL scenarios, respectively, in both simulation environments.

One can see that there is a good adherence between the real-time and offline results in all load cases, which is confirmed by looking at the instantaneous output voltage errors from Figure 4. This error is calculated in relation to the sinusoidal reference signal.

The instantaneous voltage error produced by the OSV-MPC in the F28379D-based HIL matches the profiles verified in Simulink for all load conditions. They do not match each other exactly, because the data acquired corresponds to different time intervals for the different simulation schemes, and the OSV-MPC synthesizes slightly different waveforms every cycle.

The peak values of the instantaneous voltage errors are similar for both simulation schemes in all load conditions. The output voltage MAEs were calculated from the instantaneous error data, and are presented in Table 3.

Table 3. Output voltage MAE comparison for each load scenario in Simulink and multicore F28379D-based HIL simulation environments.

Load type	Simulink MAE [V]	Real-time MAE [V]
No load	1.68	1.66
Resistive	1.69	1.50
Unbalanced	1.91	1.94

The difference between the voltage MAE obtained in Simulink and in the real-time simulation is always less than 0.2 V, result that is considered to validate the real-time implementation performed.

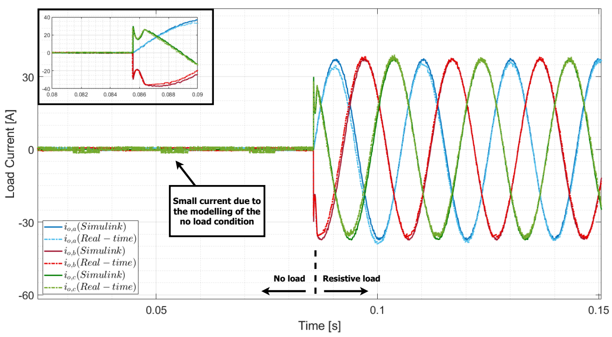


Figure 5. Current transient (no load to R load).

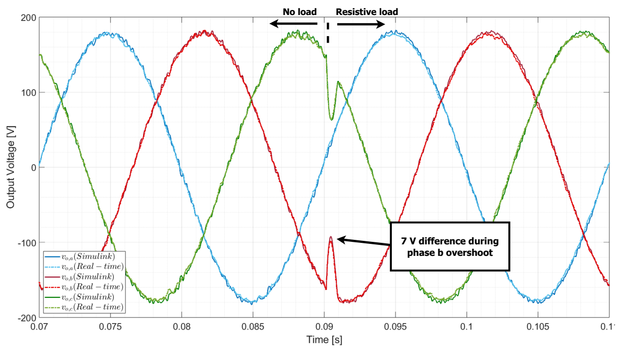


Figure 6. Voltage transient (no load to R load).

4.2 Transient-state simulation results

Two load transient scenarios are considered: no load to R and R to unbalanced RL. To simulate the load commutation, an ideal switch is implemented.

Figure 5 shows the current transient and Figure 6 shows the voltage transient when the system is on no load and an R load is commuted. In both cases, the real-time results are in adherence with the Simulink results, even during the transient event.

In Figure 5, one may notice a small current flowing during the no load condition, because of the resistors used to model this scenario in the real-time platform. This current should be sinusoidal, but it appears distorted due to the quantization process of the 12-bit DACs. The real-time results obtained validate the implementation performed for the no load to resistive load transient.

Figures 7 and 8 present, respectively, the current and voltage transient responses during an R load to unbalanced RL load commutation event. The total transient event lasts 2 ms. During the transient, phases b and c output voltages present an overshoot and then an undershoot behavior. For the offline simulation, no significant transient was perceived in phase a output voltage, differently from the real-time simulation, in which a small peak occurs due to the non-zero phase b load current observed. Except for the aforementioned differences, the dynamics obtained with the real-time simulation are in accordance with the offline simulation.

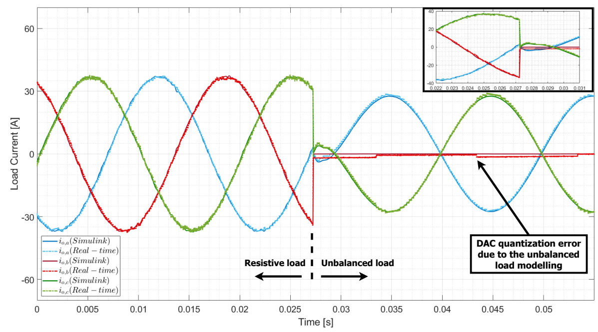


Figure 7. Current transient (R load to unbalanced RL load).

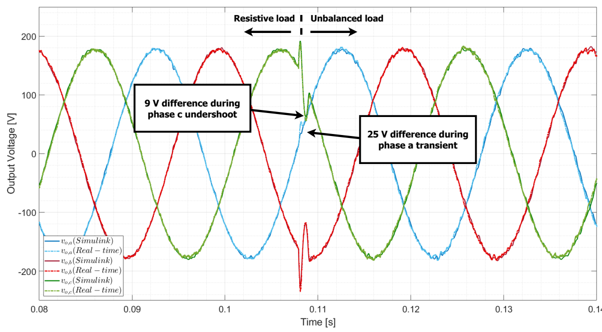


Figure 8. Voltage transient (R load to unbalanced RL load).

5. CONCLUSIONS

In this work, real-time simulations of an OSV-MPC controlled three-phase two-level inverter feeding a load through an LC filter were successfully carried out, using a multicore F28379D-based HIL scheme.

In the steady-state real-time simulations, the output voltage MAE calculated from the instantaneous output voltage error matched the values obtained offline using Simulink for the three load conditions tested. Also, the transient-state real-time simulations showed dynamics corresponding to what was expected according to the offline simulations.

Despite lacking computational power, this scheme is much cheaper than commercial solutions, and may suit well for small systems with a few electrical nodes and with sampling times within the range of dozens of microsecond. The transient-state simulations, however, still have aspects that must be improved, such as the modeling of the no-load and unbalanced load scenarios and problems associated with the DACs signal exportation for these scenarios.

REFERENCES

Bieber, L., Wang, L., Jatskevich, J., and Li, W. (2023). Universal equivalent model for real-time cpu/fpga co-simulation of hybrid cascaded multilevel converters. *IEEE Access*, 11, 4228–4241. doi:10.1109/ACCESS.2023.3235272.

Castro, D.S., Magossi, R.F.Q., Bastos, R.F., Oliveira, V.A., and Machado, R.Q. (2019). Low-cost hardware in the loop implementation of a boost converter. In *2019*

18th European Control Conference (ECC), 423–428. doi: 10.23919/ECC.2019.8796228.

Choe, S., Ko, S., Kim, S.Y., and Sul, S.K. (2014). Small scaled power hardware-in-the loop and control method of ship integrated power system with active front end converter and battery energy storage system using low cost multicore dsp. In *2014 16th European Conference on Power Electronics and Applications*, 1–10. doi:10.1109/EPE.2014.6910818.

Crosbie, R., Zenor, J., Bednar, R., Word, D., Hingorani, N., and Ericsen, T. (2004). High-speed, scalable, real-time simulation using dsp arrays. In *18th Workshop on Parallel and Distributed Simulation, 2004. PADS 2004.*, 52–59. doi:10.1109/PADS.2004.1301285.

de Sá Santos Ribeiro, L.F.C., Dicler, F.N.F., Rolim, L.G.B., and Aredes, M. (2019). Real-time implementation of a dc converter using modified nodal analysis, sparsity handling and parallelism on a dsp platform. In *2019 IEEE 15th Brazilian Power Electronics Conference and 5th IEEE Southern Power Electronics Conference (COBEP/SPEC)*, 1–6. doi:10.1109/COBEP/SPEC44138.2019.9065368.

Dicler, F.N.F. (2021). *Contributions to the FPGA and CPU Implementation of Frequency-Dependent Network Equivalents for Real-Time and Offline Electromagnetic Transient Power System Simulators.* Ph.D. dissertation, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.

Dinavahi, V. (2000). *Real-time Digital Simulation of Switching Power Circuits.* Ph.D. thesis, University of Toronto, Toronto, ON.

Geyer, T. (2011). A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus pwm-based schemes. *IEEE Transactions on Industry Applications*, 47(3), 1380–1389. doi:10.1109/TIA.2011.2127433.

J. Bélanger, P.V. and Paquin, J.N. (2010). The what, where and why of real-time simulation. *IEEE PES General Meeting*, 25–29.

Jack, A.G., Atkinson, D.J., and Slater, H.J. (1998). Real-time emulation for power equipment development. part 1: Real-time simulation. *IEE Proceedings - Electric Power Applications*, 145(2), 92–97. doi:10.1049/ip-epa:19981753.

Liu, Z., Song, Q., Zhang, H., and Liu, W. (2006). Real-time digital simulation for a 50mvar cascaded multilevel statcom. In *2006 International Conference on Power System Technology*, 1–6. doi:10.1109/ICPST.2006.321764.

Lu, B., Wu, X., Figueroa, H., and Monti, A. (2007). A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *IEEE Transactions on Industrial Electronics*, 54(2), 919–931. doi:10.1109/TIE.2007.892253.

Mohammadi, F., Bok, R., and Hajian, M. (2022). Real-time controller-hardware-in-the-loop testing of power electronics converters. In *2022 13th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC)*, 398–402. doi:10.1109/PEDSTC53976.2022.9767506.

Qi, C., Qin, S., Miao, F., Li, S., and Wang, R. (2021). Test and assessment of grid forming wind turbine based on controller hardware-in-the-loop. In *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*, 2718–2723. doi:10.1109/EI252483.2021.9713532.

Ranjan, R.K., Upamanyu, K., and Narayanan, G. (2021). Real-time simulation of wind turbine driven grid-connected induction generator on dsp platform. In *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, 1–6. doi:10.1109/IPRECON52453.2021.9640901.

Rodriguez, J., Pontt, J., Correa, P., Lezana, P., and Cortes, P. (2005). Predictive power control of an ac/dc/ac converter. In *Fourtieth IAS Annual Meeting. Conference Record of the 2005 Industry Applications Conference, 2005.*, volume 2, 934–939 Vol. 2. doi:10.1109/IAS.2005.1518458.

Rodriguez, J., Pontt, J., Silva, C.A., Correa, P., Lezana, P., Cortes, P., and Ammann, U. (2007). Predictive current control of a voltage source inverter. *IEEE Transactions on Industrial Electronics*, 54(1), 495–503. doi:10.1109/TIE.2006.888802.

Rodriguez, J. and Cortes, P. (2012). *Delay Compensation*, 177–189. doi:10.1002/9781119941446.ch12.

Sotero, M., Fontenele, G., Dicler, F., Neves, M., Corrêa, L.F., and Aredes, M. (2021). An fpga-based hardware-in-the-loop implementation of power electronics circuits using a generic real-time simulator. In *2021 Brazilian Power Electronics Conference (COBEP)*, 1–8. doi:10.1109/COBEP53665.2021.9684098.

Sotero, M. and Rolim, L.G.B. (2023). Hardware-In-the-Loop Simulation of a Step-Up Converter Using the GDS Oversampling Technique in a Microcontroller. In *2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP)*, 1–5.

Yazdani, A. and Iravani, R. (2010). *Two-Level, Three-Phase Voltage-Sourced Converter*, 115–126. doi:10.1002/9780470551578.ch5.

Young, H.A., Perez, M.A., Rodriguez, J., and Abu-Rub, H. (2014). Assessing finite-control-set model predictive control: A comparison with a linear current controller in two-level voltage source inverters. *IEEE Industrial Electronics Magazine*, 8(1), 44–52. doi:10.1109/MIE.2013.2294870.

Zheng, C., Dragičević, T., Zhang, Z., Rodriguez, J., and Blaabjerg, F. (2021). Model predictive control of lc-filtered voltage source inverters with optimal switching sequence. *IEEE Transactions on Power Electronics*, 36(3), 3422–3436. doi:10.1109/TPEL.2020.3015540.