POLYPUBLIE

Inter-step PWM interpolation for real-time FPGA simulation: a 1 MHz buck converter case study

Kevin Garnier, Nader El Zarif, Mostafa Amer, Tarek Ould-Bachir, Yvon Savaria

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2026
Type	Communication de conférence / Conference or Workshop Item Communication écrite/Paper
Version	Version finale avant publication / Accepted version
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/84945/
Évaluation par les pairs Peer reviewed	Révisé par les pairs / Refereed
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Garnier, K., El Zarif, N., Amer, M., Ould-Bachir, T., & Savaria, Y. (octobre 2026). Inter-step PWM interpolation for real-time FPGA simulation: a 1 MHz buck converter case study [Communication écrite]. 52nd Annual Conference of the IEEE Industrial Electronics Society (IECON 2026), Doha, Qatar (6 pages). https://publications.polymtl.ca/84945/

Publication officielle
Official publication

Nom de la conférence Conference name	52nd Annual Conference of the IEEE Industrial Electronics Society (IECON 2026)
Date et lieu Date and location	2026-10-18 - 2026-10-21, Doha, Qatar
Maison d'édition Publisher	IEEE
URL officielle Official URL	
Mention légale Legal notice	© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

Inter-Step PWM Interpolation for Real-Time FPGA Simulation: A 1 MHz Buck Converter Case Study

Kevin Garnier*, Nader El-Zarif†, Mostafa Amer†, Tarek Ould-Bachir*, Yvon Savaria†
*MOTCE Laboratory, DGIGL, Polytechnique Montréal, Canada
†Electrical Engineering Department, Polytechnique Montréal, Canada
{kevin.garnier, nader-2.el-zarif, mostafa.amer, t.ould-bachir, yvon.savaria}@polymtl.ca

***Abstract*—The global electrification of transport, energy, and industry is driving the widespread deployment of power converters. To maximise power density and minimise passive component size, switching frequencies are moving well beyond 100 kHz. FPGAs have become the platform of choice for real-time HIL simulation of such converters. However, the pursuit of ever-smaller simulation time steps leads to narrow fixed-point word lengths, accuracy issues, and beating effects. This paper proposes an inter-step PWM interpolation method that breaks this race to the bottom: by resolving the PWM switching instant to 4 ns within each 200 ns simulation step via linear state interpolation, the beating effect is suppressed. The methodology is described and implemented on the NOA Arch 1 open FPGA platform (Zynq UltraScale+, 125 MSPS I/O) and validated on a 1 MHz synchronous buck converter equipped with a reinforcement-learning-optimised controller. The proposed approach allows using a 50× larger time step than the 4 ns implementation, relaxing the timing closure constraints associated with the latter's narrow fixed-point word length, while achieving equivalent PWM resolution and simulation accuracy.**

***Index Terms*—Real-time simulation, FPGA, Hardware-in-the-loop, PWM interpolation, High switching frequency converter.**

I. Introduction

The global push toward decarbonisation is reshaping the energy landscape: electric vehicles, photovoltaic inverters, on-board fast chargers, and more-electric aircraft are becoming ubiquitous, placing power electronic converters at the heart of modern energy systems [1]. To maximise power density and reduce passive component size, switching frequencies are steadily moving above 100 kHz and increasingly toward the megahertz range [2]. Hardware-in-the-loop (HIL) simulation has become standard practice for validating converter designs and control systems under conditions that physical prototypes cannot safely or economically replicate. Real-time simulation above 100 kHz requires submicrosecond time steps, which has made FPGAs the technology of choice [3]. This has driven simulation time steps to ever lower values: implementations at 25 ns and 15 ns [4]–[6], down to a few nanoseconds on dedicated commercial simulators [7], have been reported.

This race, however, comes at a cost. On one hand, very small time steps force the use of narrow fixed-point word lengths to meet FPGA timing closure, restricting the simulator to a limited set of operating conditions [8]. On the other hand, they do not resolve the fundamental *beating effect*: when the simulation clock and the controller's PWM generator are asynchronous, spurious subharmonic components corrupt the simulated waveforms [9], [10]. Reducing the time step shifts these sub-harmonics in frequency but cannot eliminate them.

Several methods have been proposed to mitigate the beating effect. Time-average methods [11] and sub-cycle averaging [12] average the PWM signal over a single simulation step and then feed it to an averaged converter model, thereby suppressing spectral aliasing but sacrificing instantaneous waveform accuracy. Integration-oversampling methods [13] reconstruct an equivalent PWM pattern that minimizes the integrated duty-cycle reading error at each step. Interpolation-based techniques resolve the switching instant within each step by linearly interpolating the system states [14], [15]. However, none of these methods has been demonstrated on an open, academically documented FPGA platform. Commercial simulators may implement similar techniques, but without disclosing implementation details.

The present paper addresses this gap by extending the interpolation-based approach [14], [15] to the FPGA HIL context, detecting the switching instant of an external asynchronous PWM signal and resolving it to 4 ns. To the best of our knowledge, no open FPGA platform has yet demonstrated real-time HIL simulation of a converter switching at 1 MHz. The main contributions of this work are as follows:

- A case study showing that, for a 1 MHz buck converter equipped with a reinforcement-learning-optimized (RLO) controller, fixed-time-step FPGA simulation excites circuit resonances during input-voltage transients.
- The assessment of an inter-step PWM interpolation method suitable for FPGA implementation, achieving 4 ns effective PWM resolution with much larger time steps while suppressing these oscillations.
- Two FPGA implementations of a 1 MHz buck converter HIL simulator, based on a conventional 4 ns fixed-step solver and a 200 ns interpolated solver, validated in real time with the RLO controller.

This paper is organized as follows. Section II analyses the beating effect in a 1 MHz synchronous buck converter. Section III presents the proposed inter-step PWM interpolation method and its FPGA implementation. Section IV reports HIL validation results and FPGA resource utilization, and Section V concludes the paper.

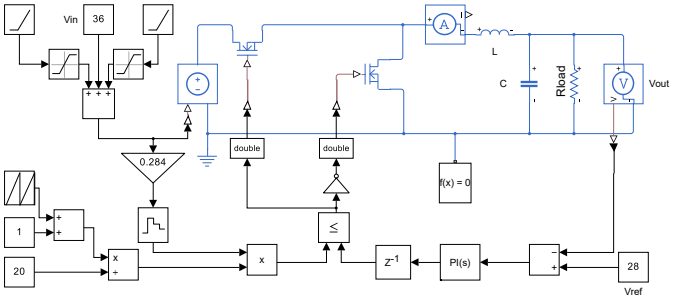


Fig. 1: Synchronous Buck converter with the reinforcement-learning-optimised (RLO) controller used in this work.

TABLE I: Buck Converter Parameters

Parameter	Value	Parameter	Value
V_{out}	28 V	V_{in}	36–60 V
L	24 µH	r_L	37 mΩ
C	33 µF	r_C	2.7 mΩ
r_{ds}	25 mΩ	R_L	56 Ω
f_{sw}	1 MHz	T_{sw}	1 µs

II. Case Study and Beating Effect Analysis

A. Buck Converter and RLO Controller

The case study considers the synchronous buck DC–DC converter shown in Fig. 1, operating at a switching frequency of $f_{sw} = 1$ MHz. Its electrical parameters are summarized in Table I. The converter regulates a 28 V output derived from an input voltage ranging from 36 V to 60 V, representative of a regulated avionics DC bus.

Figure 2 illustrates the simulation scenario over a 45 ms interval. The input voltage v_{in} undergoes two large-signal disturbances: A 36 V $\rightarrow$ 60 V step at $t = 15$ ms followed by a 60 V (Zone 1) $\rightarrow$ 36 V step at $t = 30$ ms (Zone 2). Throughout these transients, the controller is designed to maintain the output voltage within 1% of the 28 V reference.

The controller combines two complementary control paths, as shown in Fig. 1. The first is a voltage-feedback (VFB) loop based on a PI regulator acting on the voltage error $v_o - V_{\text{ref}}$. The second is an input-voltage feedforward (IVFF) path that compensates for variations in v_{in} by directly adjusting the nominal duty cycle [16]. The gains of the resulting PI-FF controller are optimized offline using a reinforcement-learning (RL) agent based on the Soft Actor-Critic (SAC) algorithm. The optimization simultaneously targets an output overshoot below 1% and a settling time shorter than 2.2 ms over the entire input-voltage range.

Offline simulations show that the optimized controller achieves excellent transient performance, exhibiting fast recovery and negligible overshoot following both input-voltage disturbances. Such aggressive closed-loop dynamics, however, make the controller particularly sensitive to timing errors introduced by real-time fixed-step simulation.

B. Beating Effect in Fixed-Step Simulation

In a fixed-step simulator, the PWM command is sampled once every simulation time step Δt. Because the PWM generator and the simulation kernel operate on independent clocks, switching transitions occur at arbitrary positions within each simulation interval, resulting in duty-cycle errors. These errors generate discrete low-frequency sub-harmonics. This phenomenon is known as the *beating effect* [9], which can excite converter resonances and compromise controller validation [10]. Simply reducing the simulation time step shifts these sub-harmonics toward higher frequencies without eliminating them.

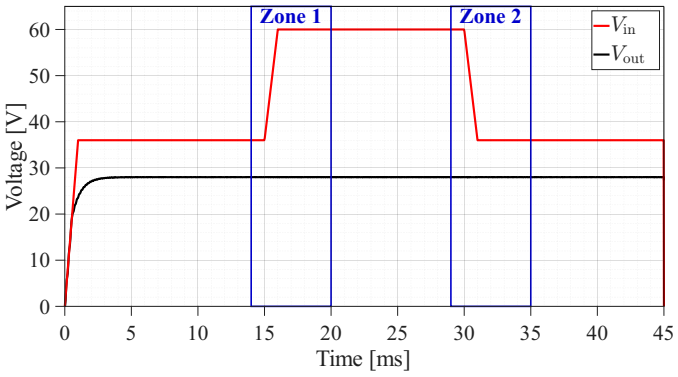


Fig. 2: Input-voltage transient scenario: the converter regulates a 28 V output while V_{in} steps from 36 V to 60 V (Zone 1) and back (Zone 2).

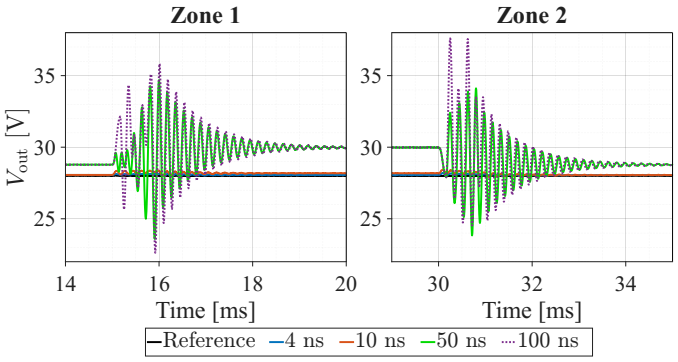


Fig. 3: Influence of a fixed simulation time step on the converter output voltage during the input-voltage transients.

Figure 3 illustrates the simulated output voltage during the two input-voltage transients (Zones 1 and 2) for four fixed simulation time steps: 100 ns, 50 ns, 10 ns, and 4 ns. The ideal variable-step result from Simscape is used as a reference. To isolate the modelling error from the controller dynamics, the PWM gate signals are generated by the variable-step Simscape reference model and then replayed identically to each fixed-step model. As a result, any discrepancy between the fixed-step and reference waveforms originates solely from the numerical discretization rather than from differences in the closed-loop controller behavior.

This procedure also explains the steady-state offset observed for the 100 ns and 50 ns models: because the switching events are quantized to the simulation time step, the effective duty cycle differs from the reference value, shifting the output

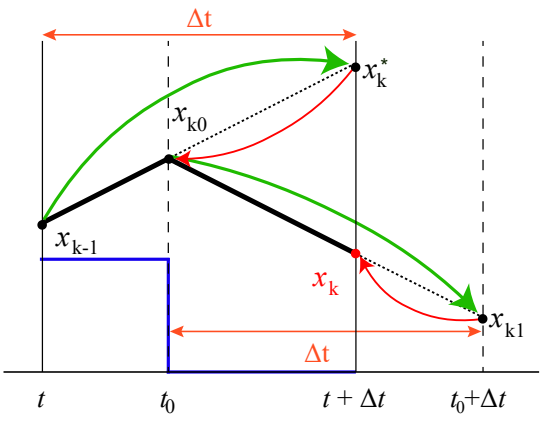


Fig. 4: Inter-step state interpolation principle, adapted from [14].

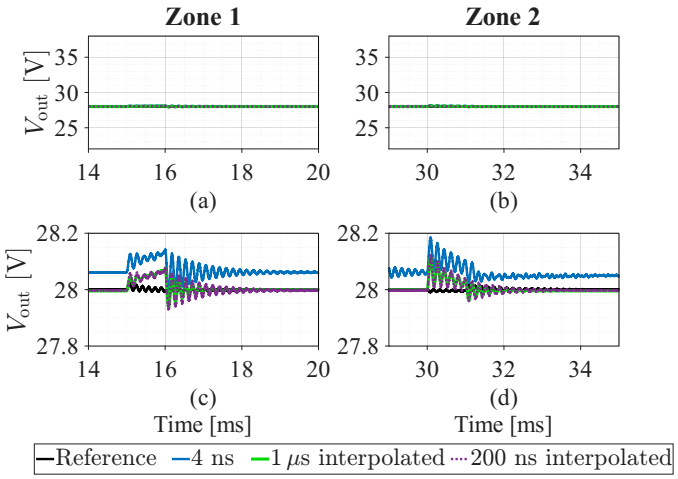


Fig. 5: Offline assessment of the proposed interpolation method: (a) Zone 1 and (b) Zone 2 overviews; (c) and (d) Zoomed views (27.8–28.2 V) (resp. Zone 1 Zone 2).

voltage away from the nominal 28 V. The offset becomes much smaller at 10 ns and is nearly eliminated at 4 ns. More importantly, at $\Delta t = 100$ ns and 50 ns, severe beating-induced oscillations make the simulation unsuitable for controller validation. Reducing the simulation time step to 10 ns, and finally 4 ns, progressively decreases the oscillation amplitude.

III. Proposed Inter-Step PWM Interpolation

A. Principle

The root cause of the beating effect is that a single switch state is assigned to each simulation step of duration Δt, regardless of when within that step the PWM transition actually occurs. If the transition happens at time $\tau \in (t_n,\, t_n + \Delta t)$, the simulator either misses it entirely or applies it one full step late. For $f_{sw} = 1$ MHz and $\Delta t = 100$ ns, this timing error can reach 10% of the switching period.

The proposed method locates the PWM switching instant with a 4 ns resolution by subdividing each simulation step into $N = \Delta t/\delta t$ sub-intervals. The state equations are integrated only once over Δt, while the intermediate trajectory is reconstructed by linear interpolation. The method builds upon the two-stage interpolation principle of Sybille *et al.* [14]. Unlike the original floating-point implementation, which handles multiple switching events iteratively, the proposed FPGA implementation assumes at most one PWM transition per simulation interval. This simplification eliminates the iterative procedure, enables a fully pipelined deterministic architecture, but limits the simulation time step to one PWM period.

The interpolation principle is illustrated in Fig. 4. Starting from the state x_{k-1} at time t, the system value is first propagated over the entire simulation interval assuming that no switching occurs, yielding the state x_k^* (upper green trajectory). When a PWM transition is detected at time t_0, the state at the switching instant, x_{k0}, is obtained by linearly interpolating between x_{k-1} and x_k^*. The converter topology is then updated, and the system is integrated from t_0 to $t_0 + \Delta t$, producing the state x_{k1} (lower green trajectory). Finally, the corrected state at the end of the original simulation interval x_k is obtained by interpolating between x_{k0} and x_{k1}. The resulting solution accounts for the actual switching instant with a 4 ns resolution while preserving the 200 ns integration step.

B. Assessment of the Proposed Method

Fig. 5 compares the proposed interpolated solver using main simulation time steps of 200 ns and 1 μs against the direct 4 ns fixed-step solver and the variable-step Simscape reference. As in the previous experiment, the PWM gate signals are generated by the reference model and replayed in open loop to all fixed-step models, ensuring that the observed differences originate solely from the numerical simulation. Figures 5(a) and (b) present the complete transient responses for Zones 1 and 2, respectively, while Figs. 5(c) and (d) provide magnified views of the output voltage between 27.8 V and 28.2 V.

The zoomed waveforms reveal that the direct 4 ns simulation exhibits a small steady-state voltage offset with respect to the variable-step reference. In contrast, both interpolated models almost perfectly overlap with the reference throughout the transient and steady-state responses. Moreover, the 200 ns and 1 μs interpolated responses are virtually indistinguishable, despite the fivefold increase in the main simulation time step.

The 1 μs case corresponds to the largest main simulation time step allowing at most one PWM transition per simulation interval. These results demonstrate that accurately reconstructing the switching instant preserves the accuracy of the 4 ns reference while increasing the main simulation time step by a factor of 250. Although equally accurate in offline simulation, the 1 μs implementation is not retained for HIL validation because the associated PWM-to-V_{out} latency becomes undesirable for the aggressive bandwidth of the RL-optimised controller. Consequently, the remainder of this paper focuses on the 200 ns implementation.

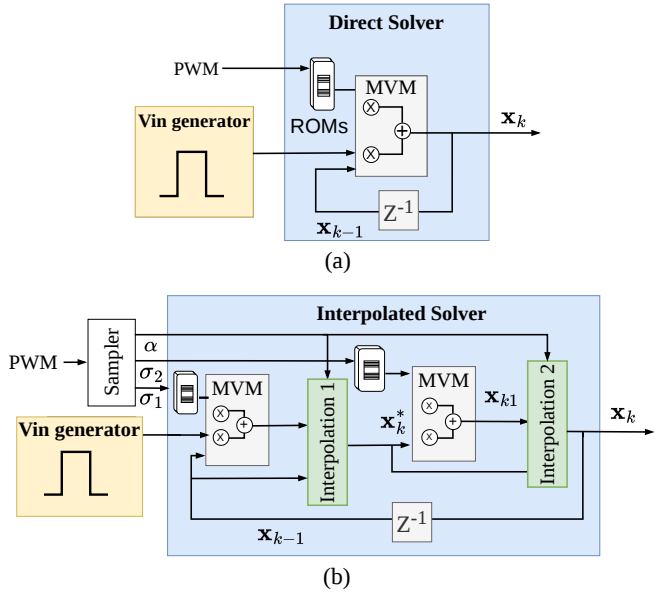


Fig. 6: FPGA implementation of the proposed state-space solvers. (a) Direct solver. (b) Interpolated solver.

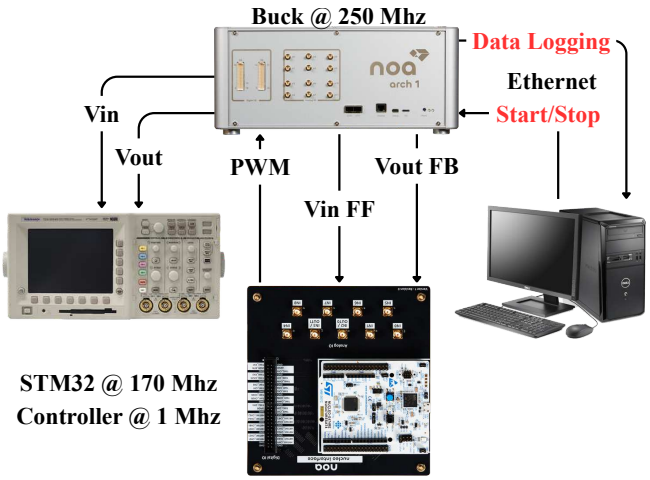


Fig. 7: Experimental HIL setup for closed-loop validation.

C. Circuit Equations

Both the direct and interpolated solvers implement the same discrete-time state-space model. For a given switching configuration σ, the state update is

$$\mathbf{x}_k = \mathbf{A}^{\sigma}\mathbf{x}_{k-1} + \mathbf{B}^{\sigma}\mathbf{u}_k, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{u}$ denote the state and input vectors, respectively. The state vector consists of the inductor current (i_L) and capacitor voltage ($v_C = V_{\text{out}}$). Since the synchronous buck converter has two switching configurations, two sets of discrete-time state-space matrices are required.

The direct solver evaluates (1) using a fixed simulation time-step of $\Delta t_D = 4$ ns. The interpolated solver instead uses a simulation time-step of $\Delta t_I = 200$ ns and accounts for the switching instant through the following sequence:

$$\mathbf{x}_k^* = \mathbf{A}^{\sigma_1}\mathbf{x}_{k-1} + \mathbf{B}^{\sigma_1}\mathbf{u}_k, \tag{2}$$
$$\mathbf{x}_{k0} = (1-\alpha)\mathbf{x}_{k-1} + \alpha\mathbf{x}_k^*, \tag{3}$$
$$\mathbf{x}_{k1} = \mathbf{A}^{\sigma_2}\mathbf{x}_{k0} + \mathbf{B}^{\sigma_2}\mathbf{u}_{k0}, \tag{4}$$
$$\mathbf{x}_k = \alpha\mathbf{x}_{k0} + (1-\alpha)\mathbf{x}_{k1}, \tag{5}$$

where $\alpha \in [0,1]$ denotes the normalized switching instant within the simulation step and $\mathbf{u}_{k0}$ is the input vector sampled at the switching instant.

D. FPGA Architecture

Fig. 6 presents the FPGA implementations of the direct and interpolated solvers. The direct solver (Fig. 6.a) consists of a single matrix–vector multiplication (MVM) stage implementing (1). The sampled PWM signal selects the switching configuration σ, which addresses the ROMs storing the corresponding state-space matrices $\mathbf{A}^{\sigma}$ and $\mathbf{B}^{\sigma}$. The MVM stage updates state vector $\mathbf{x}_k$, and provides it as the solver output.

The interpolated solver (Fig. 6.b) extends the direct architecture by cascading two MVM stages separated by two interpolation stages. The PWM sampler, located upstream of the solver, determines the normalized switching instant α together with the switching configurations σ_1 and σ_2. These quantities are provided as inputs to the solver and control the sequence of computations described by (2)–(5). The resulting state is stored for the next simulation step and simultaneously provided as the solver output.

The normalized switching instant α is represented as an 18-bit fixed-point value. It is computed by the PWM sampler as a running accumulator: at each $\delta t = 4$ ns sub-step, α is incremented by $1/N = 1/50$ for every sub-step during which the PWM signal has been high (resp. low) since the beginning of the main simulation step.

IV. Experimental Results

A. HIL Experimental Setup

The experimental HIL setup is illustrated in Fig. 7. The power converter model executes in real time on the NOA Arch 1 platform [17], an open FPGA platform based on the AMD/Xilinx Zynq UltraScale+ (XCZU5EV) featuring 14-bit, 125 MSPS digital-to-analog converters. The external controller is implemented on an STM32G474 microcontroller [18], operating at a PWM frequency of 1 MHz.

The STM32 generates the PWM gate signal, which is transmitted to the NOA Arch 1 through its digital I/O interface. The FPGA executes the real-time converter simulation and returns the simulated input and output voltages to the microcontroller through analog outputs, thereby closing the control loop. The same analog signals are simultaneously monitored using a digital oscilloscope for experimental observation. A host computer is connected to the NOA Arch 1 through

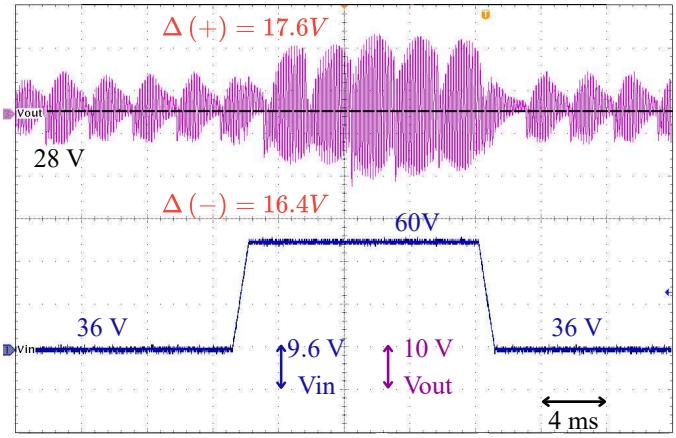
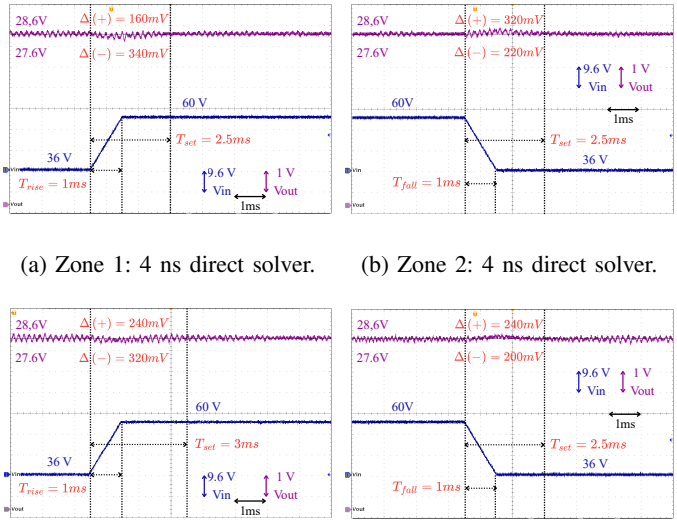


Fig. 8: Closed-loop HIL response obtained with the conventional 100 ns direct solver.



(a) Zone 1: 4 ns direct solver.

(b) Zone 2: 4 ns direct solver.

(c) Zone 1: 200 ns interpolated solver.

(d) Zone 2: 200 ns interpolated solver.

Fig. 9: Closed-loop hardware-in-the-loop (HIL) validation. The top row shows the 4 ns direct solver, while the bottom row presents the 200 ns interpolated solver.

Ethernet to configure the experiment, start and stop the real-time simulation, and log the measured data.

B. Closed-Loop HIL Validation with the RL Controller

The complete HIL chain is validated in closed loop using the RL-optimised PI–FF controller of [16]. Figure 8 illustrates the behaviour of a conventional 100 ns direct solver. Severe beating-induced oscillations are present throughout the simulation, both during steady-state operation and following the input-voltage transitions. The oscillation amplitude reaches nearly 10 V around the nominal 28 V operating point for the 36 V input voltage and increases to approximately 18 V after the transition to 60 V, making the HIL simulation unsuitable for controller validation.

Figure 9 compares the responses obtained with the 4 ns direct solver and the 200 ns interpolated solver. The left column of Fig. 9 corresponds to the 36 V→60 V input-voltage transition (Zone 1), whereas the right column corresponds to the 60 V→36 V transition (Zone 2). The top row presents the reference 4 ns direct solver, while the bottom row shows the 200 ns interpolated solver. In both operating conditions, the 200 ns interpolated solver closely reproduces the expected closed-loop behaviour obtained with the 4 ns direct solver, while completely suppressing the beating-induced oscillations observed with the conventional 100 ns implementation.

These experimental results demonstrate that the proposed interpolation enables reliable closed-loop HIL validation of MHz-class power converters using a 200 ns simulation time step, providing the same effective PWM timing resolution as the 4 ns direct solver at a 50× larger simulation time step.

C. FPGA Resource Utilization

Table II summarizes the FPGA resource utilization of the direct and interpolated solvers implemented on the NOA Arch 1 platform based on a Xilinx Zynq UltraScale+ (XCZU5EV-2SFVC784I). The interpolated solver requires only a modest

TABLE II: FPGA Resource Utilization and Accuracy on Zynq UltraScale+

Resource	**Direct Solver**	**Interpol. Solver**	**Available**
LUT	744 (0.6%)	1442 (1.2%)	117,120
FF	1079 (0.5%)	1790 (0.8%)	234,240
DSP	8 (0.6%)	25 (2.0%)	1 248
PWM resolution	4 ns	4 ns	-
Time-step	4 ns	200 ns	-
PWM-V_{out} Latency	4 ns	224 ns	-
2-norm error V_{out}	0.087 %	0.225 %	-
2-norm error i_L	2.66 %	2.23 %	-

increase in logic resources compared with the direct implementation. LUT and flip-flop utilization increase by approximately 50%, while DSP usage rises from 8 to 25 blocks to accommodate the additional arithmetic associated with the interpolation stages. Despite this increase, the design still occupies less than 2% of the available FPGA resources and does not require any BRAM.

The reported PWM-to-V_{out} latency corresponds to the delay between a PWM sampling on the main time-grid and the availability of the updated converter state. For the interpolated solver, the PWM interpolation stage introduces one simulation-step delay before the switching instant can be identified. The state-space solver itself is fully pipelined and produces a result after six clock cycles at 250 MHz (24 ns). Consequently, the total latency is 224 ns. By contrast, the direct solver is not pipelined and produces updated states within 4 ns.

D. Accuracy Assessment

Table II also reports the relative 2-norm errors of the output voltage V_{out} and inductor current i_L with respect to the

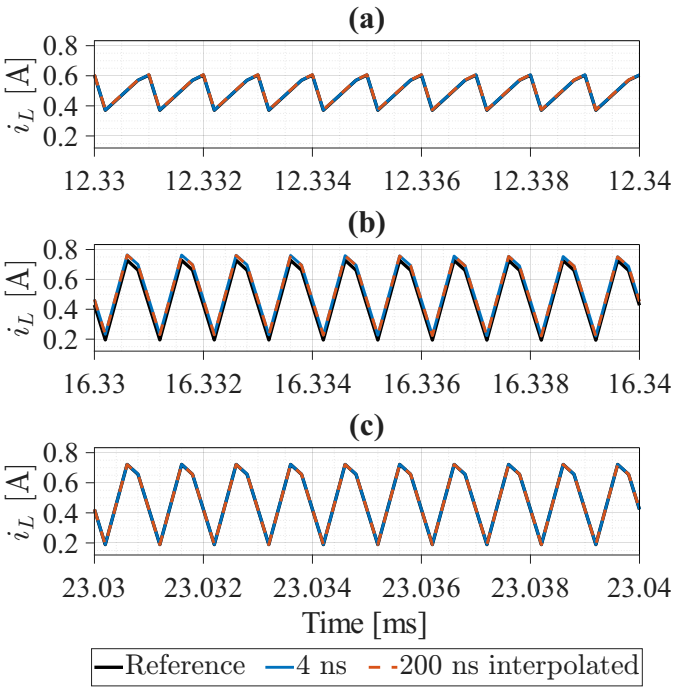


Fig. 10: (a) Before Zone 1. (b) During Zone 1. (c) In between Zone 1 and Zone 2.

variable-step Simscape reference. Both solvers achieve errors of the same order of magnitude, confirming that the 200 ns interpolated solver preserves the accuracy of the 4 ns direct solver. As expected, the error is larger for i_L than for V_{out}, since the inductor current is more sensitive to residual PWM timing quantization.

Figure 10 further compares the inductor current before, during, and after the first input-voltage transient. The 4 ns direct and 200 ns interpolated solvers produce nearly identical waveforms and exhibit the same small offset with respect to the reference after the transient. This common offset results from the residual 4 ns PWM timing quantization rather than from the interpolation method or the fixed-point implementation, confirming that the proposed interpolated solver introduces no additional modelling error.

V. Conclusion

This paper presented an inter-step PWM interpolation method for the real-time FPGA simulation of high-switching-frequency power converters. By reconstructing the PWM switching instant within each main simulation step, the proposed approach suppresses the beating effect without requiring a reduction of the simulation time step. The method was assessed on a 1 MHz synchronous buck converter equipped with a reinforcement-learning-optimised controller, showing that the accuracy of a 4 ns direct solver can be preserved with a 200 ns main simulation time step in real-time HIL, and up to 1 μs in offline simulations. The FPGA implementation requires only a modest increase in hardware resources while maintaining deterministic real-time execution, demonstrating that accurate PWM timing reconstruction offers a practical alternative to continuously reducing the simulation time step in HIL simulations of MHz-class power converters.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery Grants Program.

References

[1] F. Blaabjerg, Y. Yang, K. Ma, and X. Wang, "Power electronics – the key technology for renewable energy system integration," in *International Conference on Renewable Energy Research and Applications (ICRERA)*, 2015, pp. 1618–1626.

[2] H. A. Soomro, M. H. Bin Md Khir, S. A. B. M. Zulkifli, G. E. M. Abro, and M. M. Abualnaeem, "Applications of wide bandgap semiconductors in electric traction drives: Current trends and future perspectives," *Results in Engineering*, vol. 26, p. 104679, 2025.

[3] X. Guo, Y. Tang, M. Wu, Z. Zhang, and J. Yuan, "FPGA-based hardware-in-the-loop real-time simulation implementation for high-speed train electrical traction system," *IET Electric Power Applications*, vol. 14, no. 5, pp. 850–858, 2020.

[4] H. Bai, H. Luo, C. Liu, D. Paire, and F. Gao, "Real-time modeling and simulation of electric vehicle battery charger on FPGA," in *International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 1536–1541.

[5] H. Chalangar, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, "A direct mapped method for accurate modeling and real-time simulation of high switching frequency resonant converters," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6348–6357, 2021.

[6] K. Meddah, H. Chalangar, and T. Ould-Bachir, "Real-time simulation of a fast charger using a low-cost FPGA platform," in *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2022, pp. 1–6.

[7] Plexim GmbH, "RT Box – real-time hardware-in-the-loop simulator," https://www.plexim.com/products/rt_box, 2025, accessed: 2026-06-27.

[8] A. Sánchez, E. Todorovich, and A. de Castro, "Exploring the limits of floating-point resolution for hardware-in-the-loop implemented with FPGAs," *Electronics*, vol. 7, no. 10, p. 219, 2018.

[9] E. Zamiri, A. Sanchez, M. S. Martínez-García, and A. de Castro, "Analysis of the aliasing effect caused in hardware-in-the-loop when reading PWM inputs of power converters," *International Journal of Electrical Power & Energy Systems*, vol. 136, p. 107678, 2022.

[10] T. Robert, T. Ould-Bachir, V. Combet, M. Kenzi, and R. Monthéard, "Accurate real-time simulation of CLLLC converters on FPGA: A study of sampling resolution and beating effect," in *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2025, pp. 1–6.

[11] K. Lian and P. Lehn, "Real-time simulation of voltage source converters based on time average method," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 110–118, 2005.

[12] J. Allmeling and N. Felderer, "Sub-cycle average models with integrated diodes for real-time simulation of power converters," in *IEEE Southern Power Electronics Conference (SPEC)*, 2017, pp. 1–6.

[13] E. Zamiri, A. Sánchez, and A. de Castro, "Four-period integration over-sampling method (4PIOM) for hardware-in-the-loop power converters with complementary switches," *Electrical Engineering*, 2024.

[14] G. Sybille, H. Le-Huy, R. Gagnon, and P. Brunelle, "Analysis and implementation of an interpolation algorithm for fixed time-step digital simulation of PWM converters," in *IEEE International Symposium on Industrial Electronics (ISIE)*, 2007, pp. 793–798.

[15] Y. Li, P. Deng, J. Zhang, D. Liu, and Z. Hao, "A study of interpolation compensation based large step simulation of PWM converters," *Energies*, vol. 13, no. 12, p. 3069, 2020.

[16] M. Amer, A. Abuelnasr, K. Garnier, A. Ragab, N. El-Zarif, M. Ould-Bachir, Tarek, M. Sawan, and Y. Savaria, "Reinforcement learning based optimal controller design for effective dynamic regulation of DC–DC converters: Case studies on buck and zeta topologies," IEEE Access, in preparation.

[17] Rexys, "NOA Arch 1 – Open FPGA Platform," https://rexys.io/products/noa-arch-1/, 2025, accessed: 2026-06-27.

[18] STMicroelectronics, *STM32 Nucleo-64 Boards (MB1367) User Manual*, 2024, accessed: 2026-06-30. [Online]. Available: https://www.st.com/resource/en/user_manual/um2505-stm32g4-nucleo64-boards-mb1367-stmicroelectronics.pdf