

Parallel-in-time-and-space electromagnetic transient simulation of multi-terminal DC grids with device-level switch modelling

Tianshi Cheng  | Ning Lin  | Tian Liang  | Venkata Dinavahi 

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada

Correspondence

Tianshi Cheng, Department of Electrical and Computer Engineering, University of Alberta, T6G 1H9, Edmonton, Alberta, Canada.
Email: tcheng1@ualberta.ca

Funding information

Natural Science and Engineering Research Council of Canada (NSERC)

Abstract

The electromagnetic transient (EMT) simulation of multi-terminal DC (MTDC) grids requires a detailed device-level modular multilevel converter (MMC) model, which can have thousands of state variables and complex internal structures. The fast device-level insulated gate bipolar transistor (IGBT) transient requires a very small time-step, making the computational overhead prohibitive. Based on the analysis of the parallel-in-time (PiT) implementation of detailed modelled MMCs, this paper proposes a task-based hybrid PiT algorithm to achieve high parallel efficiency and speed-up of MMC with device-level modelling. Moreover, a transmission line model (TLM)-based parallel-in-time-and-space (PiT+PiS) method is proposed to connect PiT grids to conventional or other PiT grids and exploit the maximum parallelism. Simulation results show greater than 30× speed-up and 60% parallel efficiency on a 48 cores computer for the hybrid PiT method in a 201-level three-phase MMC test case, and 20× speed-up in the transient simulation of CIGRÉ B4 DC grid test system for the PiT+PiS method.

1 | INTRODUCTION

The modular multilevel converter (MMC) has gained tremendous attention in high-voltage direct current (HVDC) transmission systems and multi-terminal DC (MTDC) system [1], where the electromagnetic transient (EMT) simulation is paramount for every step of the design, testing, commissioning, operation, control, and protection [2]. Due to a large number of states determined by the multilevel nature and the pursuit for a small time-step to accurately reflect the non-linear characteristics of power electronic devices, high computation efforts are always required. Therefore, several approaches have been proposed to accelerate the simulation. At the system level, degrees of freedom reduction technique [3, 4] and variable time-stepping methods [5, 6] are proposed to improve the simulation efficiency; whilst the graphic processing unit (GPU) shows its potential in accelerating large-scale DC grid simulations with power-semiconductor-level modelling [7–9]; the field-programmable gate array (FPGA) and heterogeneous multi-processor system-on-chip (MPSoC) are also resorted to performing the simulation [10–13]. In addition, multirate simulation methods are widely used. In 1993, [14] proposed the early

version of multirate EMT simulation. With the rapid development of parallel hardware, recently multirate methods become more popular, especially for the co-simulation of heterogeneous system models [15, 16] and hybrid CPU-GPU computation [16, 17]. Although these computing methods achieved satisfactory speed-ups, they are based on the physical topology reconfiguration, and consequently, the efficiency is limited by the spatial structure of systems.

Time-domain parallelism methods such as parallel-in-time (PiT) provides a new perspective regarding multi-threading computation and have shown effectiveness in fields involving ordinary differential equations (ODEs) [18]. Although the PiT algorithms use multiple time-steps, which is similar to multirate methods, their theoretical foundations are different. The multirate methods are still considered as PiS methods since the systems are often decomposed into different subsystems with multiple time-step. For PiS algorithms, the single system is decomposed on multiple time-grids, which means the system scale can be as small as a single MMC or several simple RLCs but still get considerable acceleration from PiT computing. A popular PiT algorithm is the Parareal algorithm, which solves initial value problems iteratively by using two ODE integration methods.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial License](https://creativecommons.org/licenses/by-nc/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2021 The Authors. *IET Generation, Transmission & Distribution* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

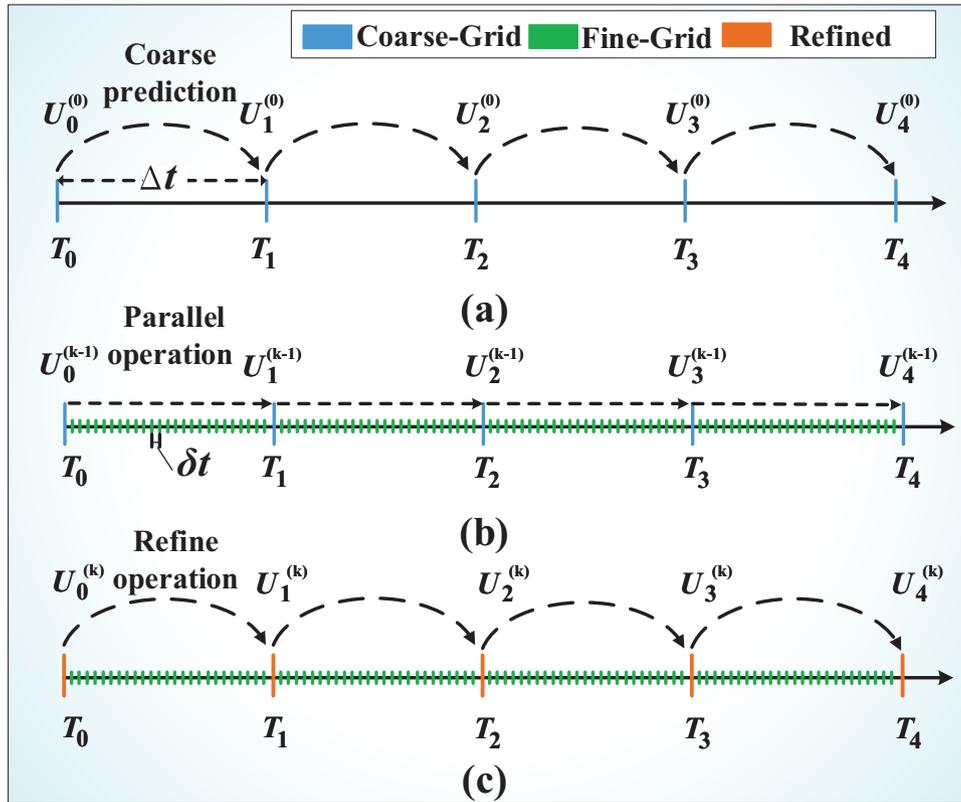


FIGURE 1 Progression of steps in the Parareal algorithm. (a) Initialize $U_j^{(0)}$, which equals $G_j^{(0)}$. (b) Produce fine-grid solution F_j^k . (c) Refine $U_j^{(k)}$ with $U_j^{(k)} = G_j^{(k)} + F_j^k - G_j^{(k-1)}$

It has become one of the most widely studied PiT integration methods [19]. It has also been proven to be able to solve semi-differential algebraic equation (DAE) or full-DAE problems in system-level dynamic simulation [20, 21], and eddy current calculation [22], yet its application in EMT simulation of large-scale HVDC grids with detailed device-level modelling is to be explored noticing the apparent computational burden by current methods. The demand for detailed submodules to verify the control scheme in more realistic scenarios, as well as the desire of power semiconductor transients to facilitate the selection of a proper power switch type, further justifies the adoption of PiT. Besides, it is currently not practical to solve the whole power system using the PiT method because all system elements must be converted to PiT models. Therefore, a practical approach to connect PiT systems to conventional systems is needed.

In this work, the Parareal implementation of the ideal switch model and the transient curve-fitting insulated gate bipolar transistor (IGBT) model for device-level simulation are investigated and based on the two implementations, a new PiT method is proposed by combining the ideal switch and transient curve-fitting model, which can significantly improve the performance with flexible accuracy control by adjusting the coarse predictor. Furthermore, in order to handle transmission line models (TLMs) in EMT simulation and take advantages of the traditional TLM-decoupling spatial parallel computing methods [23],

a queue-based connection method for PiT systems is investigated to extend the practical value of the PiT method. Following the proposal of TLM decoupling, a twofold parallelism concept, that is parallel-in-time-and-space (PiT+PiS), is proposed for the EMT simulation of the CIGRÉ B4 DC grid composing of 11 201-level MMCs.

The paper is organized as follows: Section 2 briefly describes the fundamentals of the Parareal algorithm; Section 3 introduces MMC modelling, including the MMC class architecture, IGBT ideal switch model, and the transient curve-fitting model; Section 4 introduce the PiT implementation for ideal switch model and hybrid PiT method for the device-level model, and transmission line based decoupling method; Section 5 presents the case study and efficiency analysis of the proposed methods; Section 6 is the Conclusion.

2 | THE PARAREAL PIT ALGORITHM

Parallel-in-time methods are parallel methods for temporal discretization. The Parareal algorithm can be derived as both a multi-grid method or as multiple shooting along the time axis [19]. It relies on parallel operation in the time domain to achieve acceleration.

The Parareal algorithm procedure is shown in Figure 1. The entire simulation duration $[t_0, t_{end}]$ is decomposed into N

sub-intervals denoted as $I_j = [T_{j-1}, T_j]$, where the start-time $t_0 = T_0$, and end-time $t_{end} = T_N$.

At every time instant T_j , the system has a unique solution for its state variables \mathbf{U}_j produced by a fine solution operator $\mathcal{F}(T_j, T_{j-1}, \mathbf{U}_{j-1})$. Therefore, for an overall N intervals, following non-linear equations can be established:

$$\mathbf{W}(\mathbf{U}) := \begin{cases} \mathbf{U}_1 - \mathcal{F}(T_1, T_0, \mathbf{U}_0) & = 0, \\ \mathbf{U}_2 - \mathcal{F}(T_2, T_1, \mathbf{U}_1) & = 0, \\ \vdots & \\ \mathbf{U}_N - \mathcal{F}(T_N, T_{N-1}, \mathbf{U}_{N-1}) & = 0, \end{cases} \quad (1)$$

where \mathbf{U}_0 is the known initial value.

The system (1) can be solved by Newton's method. For every $\mathbf{U}_j \in \mathbf{U} = \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_{N-1}\}$, the individual update formula is given by

$$\begin{aligned} \mathbf{U}_j^{(k)} &= \mathcal{F}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}) \\ &+ \frac{\partial \mathcal{F}}{\partial \mathbf{U}}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}) (\mathbf{U}_{j-1}^{(k)} - \mathbf{U}_{j-1}^{(k-1)}) \end{aligned} \quad (2)$$

where k represents the iteration number. The $\frac{\partial \mathcal{F}}{\partial \mathbf{U}}$ is approximated by a computationally cheap operator \mathcal{G} in sequential, which can produce a close approximation to \mathcal{F} . Therefore, the pre-requirement and fundamental principle of the Parareal algorithm is as

$$\begin{aligned} \mathcal{F}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k)}) &\approx \mathcal{G}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k)}), \\ \mathcal{F}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}) &\approx \mathcal{G}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}). \end{aligned} \quad (3)$$

Substituting the $\frac{\partial \mathcal{F}}{\partial \mathbf{U}}$ entries in (2) with the approximation in (3), the following can be derived:

$$\begin{aligned} \mathbf{U}_j^{(k)} &= \mathcal{F}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}) \\ &+ \mathcal{G}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k)}) - \mathcal{G}(T_j, T_{j-1}, \mathbf{U}_{j-1}^{(k-1)}), \end{aligned} \quad (4)$$

which becomes a Quasi-Newton method [24].

Implicit integration methods are required to solve circuit systems as DAEs, which restricts the method choices and brings more complexity to adapt them in Parareal. The Backward-Euler and Trapezoidal methods are generally sufficient for EMT simulation. In this work, the \mathcal{G} and \mathcal{F} are both based on the Trapezoidal integration method with a large time-step Δt and a small time-step δt respectively, so traditional EMT programs can be reused for the PiT purpose.

Usually power system components are expressed by differential equations. Taking the capacitor as an example, the discrete-time EMT model under Trapezoidal Rule of a capaci-

tor is expressed by:

$$I(n) = V(n)G_{eq} + I_{eq}(n), \quad (5)$$

$$I_{eq}(n) = V(n-1)G_{eq} + I(n-1), \quad (6)$$

$$G_{eq} = \frac{2C}{dt}, \quad (7)$$

where I is the current going through the capacitor, V is the voltage across the capacitor, I_{eq} is a virtual equivalent current source and G_{eq} is a equivalent admittance under time-step dt . For PiT simulation, the state variables are $\mathbf{U} = \{V, I\}$ and (5), (6) form up \mathcal{G} and \mathcal{F} under a large time-step and a small time-step, respectively. The $\{V, I\}$ produced by \mathcal{G} and \mathcal{F} participate in the iteration process of (4). I_{eq} is considered as a output so no need to store it. The other basic components like transformers and generators also need to be adapted to this form for PiT simulations. Some components like non-linear diodes do not have differential equations so that their model remain the same in the PiT simulation. The TLMs like Bergeron Line Model also has no differential equations, but they contain delay items which can bring delay differential equations (DDEs) to the system. These principles were explained in [25] for the first time.

3 | MMC PIT MODELLING

3.1 | Three-phase MMC modelling

As shown in Figure 2a, the submodules of the MMC are solved as individual sub-circuits with the arm current of the previous time-step as the injection $i_{arm}(n-1)$ after applying the V - I decoupling method [26, 27], which is valid since the time constant of the submodule is much larger than the simulation time-step – normally a few microseconds.

The fact that pure voltage sources cannot be used in nodal analysis directly prompts the merging of an arm inductor into the corresponding arm and results in the reduced equivalent model in Figure 2(b). The reduced equivalent current source and admittance are used to construct the single-phase or three-phase MMC topology shown in Figure 2(c), which finally yields a 5×5 element in nodal analysis. The nodal voltages of Nodes 1–5 in Figure 2(c) are the basic state variables that should participate in Parareal iteration.

All fundamental modules are wired up in a monolithic top-level MMC class in Figure 3(a) for the final use in the EMT simulation program. A typical d - q frame-based control scheme for grid-connected converters is adopted to regulate the DC voltage or power as the outer-loop, which generates the three-phase modular index \mathbf{m}_{abc} , as shown in Figure 3(b). The nearest-level-modulation (NLM) [28] is in charge of MMC internal current flow so as to maintain stable submodule capacitor voltages as shown in Figure 3(c). Based on the proposed MMC

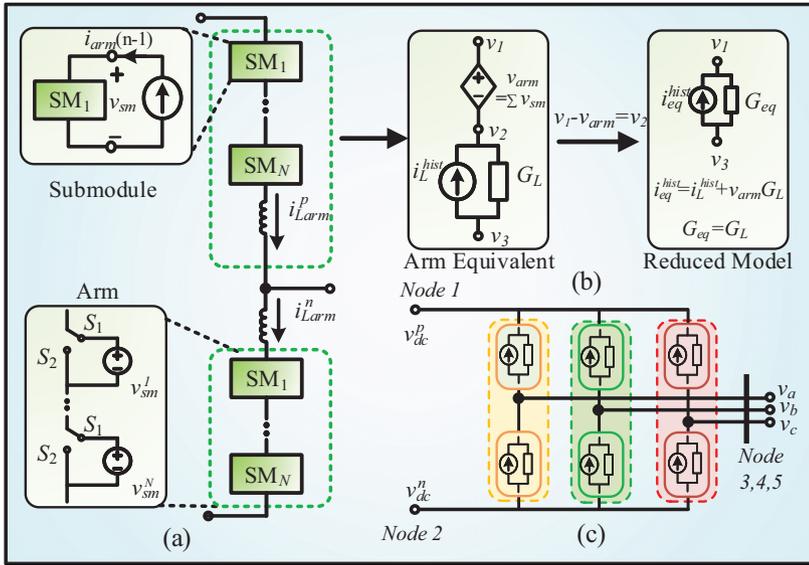


FIGURE 2 The three-phase MMC model for nodal analysis. (a) MMC arm simplification by v - i coupling, (b) Arm nodal reduction. (c) Three-phase MMC equivalent circuit

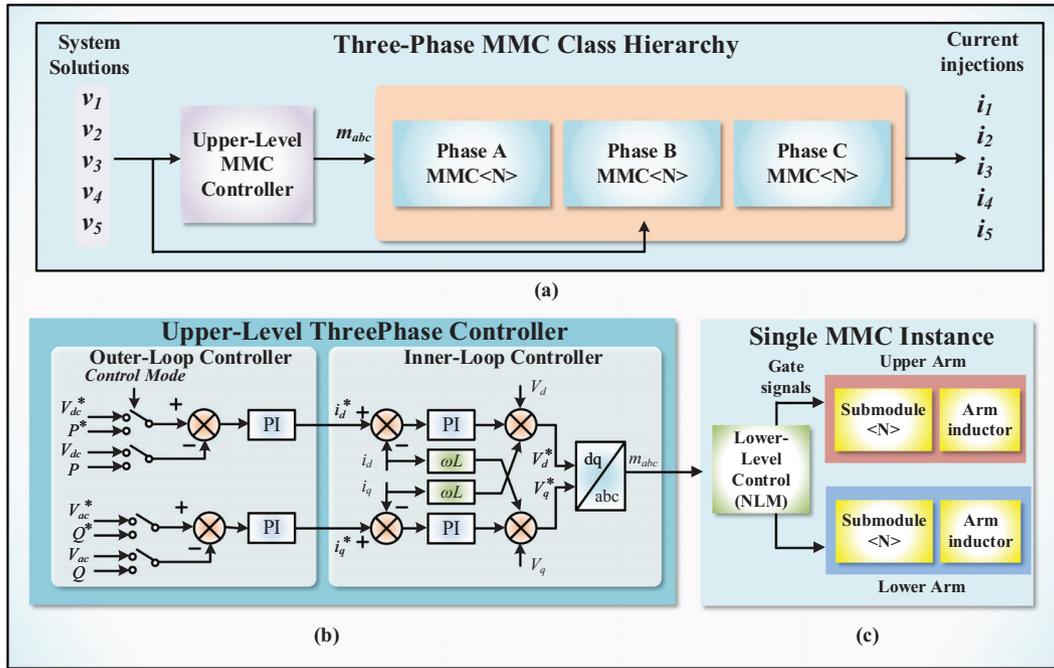


FIGURE 3 Object-oriented three-phase MMC class hierarchy. (a) Top-level module of the three-phase MMC. (b) Upper-level controller for DC voltages or PQ power control. (c) Single-phase MMC module implementation

architecture, the ideal switch and device-level transient curve-fitting half-bridge submodules (HBSMs) are investigated in this work.

3.2 | Ideal switch model

The HBSM has two IGBTs and two diodes, which is shown in Figure 4. The IGBTs and diodes are represented by fixed on-state and off-state resistors so it is considered as an ideal switch model. In normal operation, the control signals $S_2S_1 = 01$ means the SM is inserted and the capacitor voltage appears in

the main circuit; $S_2S_1 = 10$ means the capacitor is bypassed and the capacitor stops charging or discharging; $S_2S_1 = 00$ is the blocking state which may be used for charging up the capacitor in the initialization stage; The discrete numerical integration methods cause overshoots at the switching instant due to the discrete nature, which cause numerical oscillation under the Trapezoidal Rule. Therefore, the blocking state requires additional measures to handle the numerical oscillations. Some measures are presented in [27], but here a common interpolation method used by PSCAD[®] is introduced [29]. As shown in Figure 5, the interpolation method takes four steps to avoid the oscillation: (1) the system detects the zero-crossing event

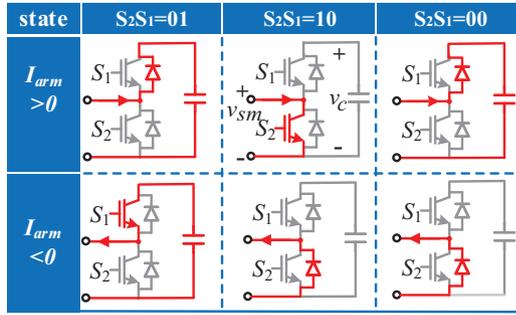


FIGURE 4 Six valid conduction states under different current directions and control signals for HBSM

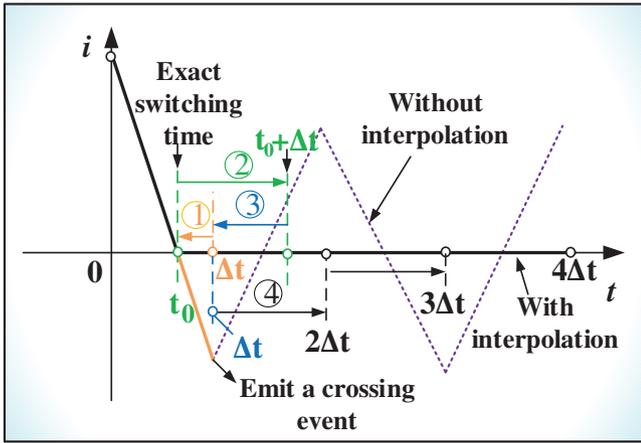


FIGURE 5 Interpolation approach to eliminate numerical oscillation caused by uncontrolled diode switching in the MMC mode [29, 34]

at $t = \Delta t$, then the system move backwards and all the state variables such as voltages and currents are interpolated to the zero-crossing time instant $t = t_0$, using the linear interpolation; t_0 should be the exact switching time between $t = 0$ and $t = \Delta t$; (2) using the system state approximated at $t = t_0$, the system triggers the diode switching and solve the solution at $t = t_0 + \Delta t$; (3) since the $t_0 + \Delta t$ is not at the original discrete-time grid, the system performs another linear interpolation with solutions at $t = t_0$ and $t = t_0 + \Delta t$ to get the solution at $t = \Delta t$; (4) with the corrected solution at $t = \Delta t$, the system can go back to the normal solution steps.

As mentioned in Section 3.1, the individual submodule is solved independently from the main circuit, which means for ideal switch model it needs to solve a bunch of 2×2 matrices to get the capacitor voltages v_{csm} and submodule port voltages v_{sm} , which can be expressed by following discretized equations under nodal analysis:

$$\begin{bmatrix} g_{sw1} + g_{sw2} & -g_{sw1} \\ -g_{sw1} & g_{sw2} \end{bmatrix} \begin{bmatrix} v_{csm}(n) \\ v_{sm}(n) \end{bmatrix} = \begin{bmatrix} i_{ceq}(n) \\ i_{arm}(n-1) \end{bmatrix}, \quad (8)$$

$$i_{ceq}(n) = v_{csm}(n-1)G_{ceq} + i_{csm}(n-1),$$

where G_{ceq} , i_{ceq} denotes the capacitor equivalent admittance and current source, $g_{sw1,2}$ denote the equivalent admittance of ideal switches, i_c is the capacitor current, and n denotes the discretized time-step index. Equation (8) forms up the basic \mathcal{G} or \mathcal{F} with the global circuit solution, where the capacitor currents and voltages i_{csm} , v_{csm} and arm current i_{arm} are the state variables required in PIT iterations. Since i_{arm} can be inferred from MMC nodal voltages, the final state vector of ideal switch MMC is $U = \{v_{dc}^p, v_{dc}^n, v_a, v_b, v_c, v_{csm}, i_{csm}\}$.

3.3 | Transient curve-fitting model

The transient curve-fitting model provides device-level information while also maintaining higher computational efficiency than the non-linear physical model. The datasheet-driven model contains two stages [7]:

- (1) *Steady-State*: it is represented by a resistor similar to the ideal switch model with its value being determined by static $i_c - v_{CE}$ curves, which can be expressed by

$$\begin{aligned} r_s(i_c, v_g, T_{vj}) &= \frac{V_{CE}}{i_c} \\ &= a_1(i_c, v_g, T_{vj}) + a_2(i_c, v_g, T_{vj}) \cdot I_C^{-1}, \end{aligned} \quad (9)$$

where i_c is the collector current, a_1 and a_2 are coefficients dependent on factors such as the gate voltage v_g , and junction temperature T_{vj} .

- (2) *Transient-State*: it models the turn-on and turn-off transient behaviours of collector current i_c and v_{CE} . The rising and falling time t_r, t_f are, based on Stone–Weierstrass theorem, approximated by a polynomial function

$$\begin{aligned} t_{r,f}(s_1, s_2, s_3) &= \kappa_0 \cdot \prod_{i=1}^2 s_i + \\ &\sum_{i,j=1,2,3}^{i \neq j} \kappa_i s_i s_j + \sum_{i=1}^3 b_i s_i + b_0, \end{aligned} \quad (10)$$

where the variable s_i represents one of the three factors such as i_c , T_{vj} , and R_g provided in the datasheet; κ_i, b_i are constants. The turn-off and turn-on transients are modelled by a virtual RC circuit which has $\tau = r * C_g, C_g = 1nF$. The τ is related to $t_{r,f}$ and the virtual capacitor voltage v_{Cg} must be initialized according to the transient type. The turn-on transient is triggered by high-level (rising edge) control signals which charges the v_{Cg} to 1V; on the other hand, the turn-off transient is triggered by low-level (falling edge) control signals and the RC circuit discharges from v_{Cg} to 0. During the RC circuit calculation, a current source is computed for generating transient i_c across the collector and emitter

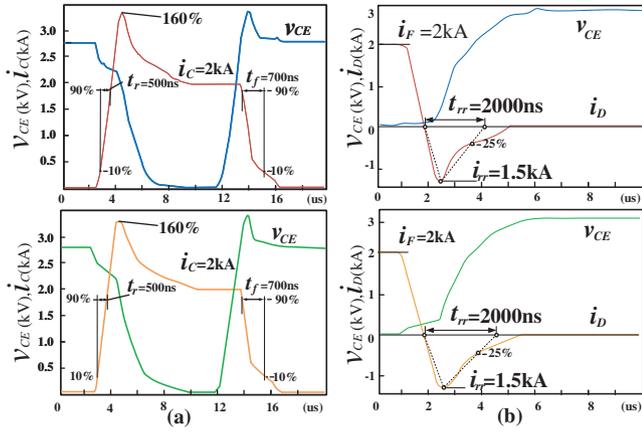


FIGURE 6 Comparison between experimental (top) and curve-fitting model simulation (bottom) of ABB® 5SNA2000K450300 StakPak IGBT module. (a) IGBT currents and voltages. (b) Diode reverse recovery

poles of the IGBT, which is computed by

$$i_{G_s}(n) = \begin{cases} 0, & (v_{G_g} = 1), \\ i_{G_s}(n-1) + k_1 \cdot \Delta t / \tau, & \left(v_{G_g} \leq 1 - e^{-\frac{-2t_r}{\tau}} \right), \\ \left(i_{G_s}(n-1) - 1 \right) \cdot e^{-\frac{-t}{k_2}} + i_C, & \left(v_{G_g} > 1 - e^{-\frac{-2t_f}{\tau}} \right), \end{cases} \quad (11)$$

where i_{G_s} is the current source, k_1 and k_2 denote the rise and fall rates of the current, and i_C is the steady-state current. In this work the ABB® 5SNA2000K450300 StakPak IGBT module is selected for device-level HBSMs [30]. The comparison between the transient curve-fitting model and experimental transient waveforms is shown in Figure 6 [31].

Although it can perform PiT computation by registering all internal states into the iteration, the model causes great difficulties to the PiT iteration due to these internal transient states and time-step limitations, making the PiT method slower than the sequential one. Therefore, a hybrid PiT solution is proposed in Section 4 B by using the ideal switch model as a coarse predictor. In this case, the additional state variable from the curve-fitting HBSM is the gate signals v_g from the last time-step, which can determine the next IGBT stage (turn-on or turn-off) along with the new control signals.

4 | PARALLEL-IN-TIME IMPLEMENTATIONS

4.1 | Parareal implementation

The Parareal algorithm has four stages shown in Figure 7. The system state vector \mathbf{G} is obtained from coarse operator \mathcal{G} and \mathbf{F} is obtained from fine operator \mathcal{F} . The final refined state vec-

tors \mathbf{U} are obtained through iterations. This iterative algorithm is applied to all system states including system voltages and the internal states for transient elements according to their model representation in state space. For the ideal-switch-based MMC, arm current, gate signals, and capacitor voltage of each HBSM should appear in the state vector. The control system's states are also included.

As shown in Figure 7, the algorithm has 4 major stages: (1) *Initialization*; (2) *Fine-grid Operation*; (3) *Sequential Update*; (4) *Finalization*. More details about Parareal implementation can be found in [25]. To save system memory, the simulation is divided into multiple small time windows. The error of each iteration is evaluated by

$$error = \sum_{j=k+1}^{N-1} \frac{\|\mathbf{U}_j^{(k+1)} - \mathbf{U}_j^{(k)}\|}{\|\mathbf{U}_j^{(k)}\|}, \quad (12)$$

where N is the number of parallel fine-grid workers, k is the iteration index. The synchronization of controllers with discrete switching events plays a significant role in the algorithm. If the controller uses the simulation time-step of their worker, the fine-grid and coarse-grid results differ greatly due to the different firing signals generated by controllers under different time-steps. In practice, even a tiny difference can produce inconsistent results between coarse-grid and fine-grid. The solution is to define a fixed sampling time for the controller in both grids, and it is better to be the same as the coarse-grid time-step.

This method works fine with the ideal switch model. However, the iteration process becomes much slower when the MMC scale increases. As more state variables are involved in the iteration, the algorithm becomes harder to converge, and the overhead increases a lot so that the speed-up cannot be achieved. This becomes worse with the more complex device-level HBSM model. Thus, a hybrid PiT model is proposed.

4.2 | Hybrid PiT model

The curving-fitting model reflects the behaviour of the IGBT devices by using simple RC circuits to generate device-level turn-on and turn-off transients. It is almost equivalent to the ideal switch model in the steady-state. The controller events can be captured on the coarse-grid so that the transient-state will function normally within the fine-grid just like the serial implementation. The core PiT concept of the Parareal is to find a computationally cheap coarse predictor to make an initial guess, so combining the system and device-level model may be a good solution, which is considered a hybrid model. A similar approach can be found in [32] for FPGA real-time simulation without PiT.

The basic concept of the hybrid PiT model is shown in Figure 8. The coarse-grid arm current i_{arm} , gate signals v_g , capacitor voltages v_c obtained from ideal switch models are passed to state interpreter first; the state interpreter determines the device-level IGBT states and initial the variables for the detailed

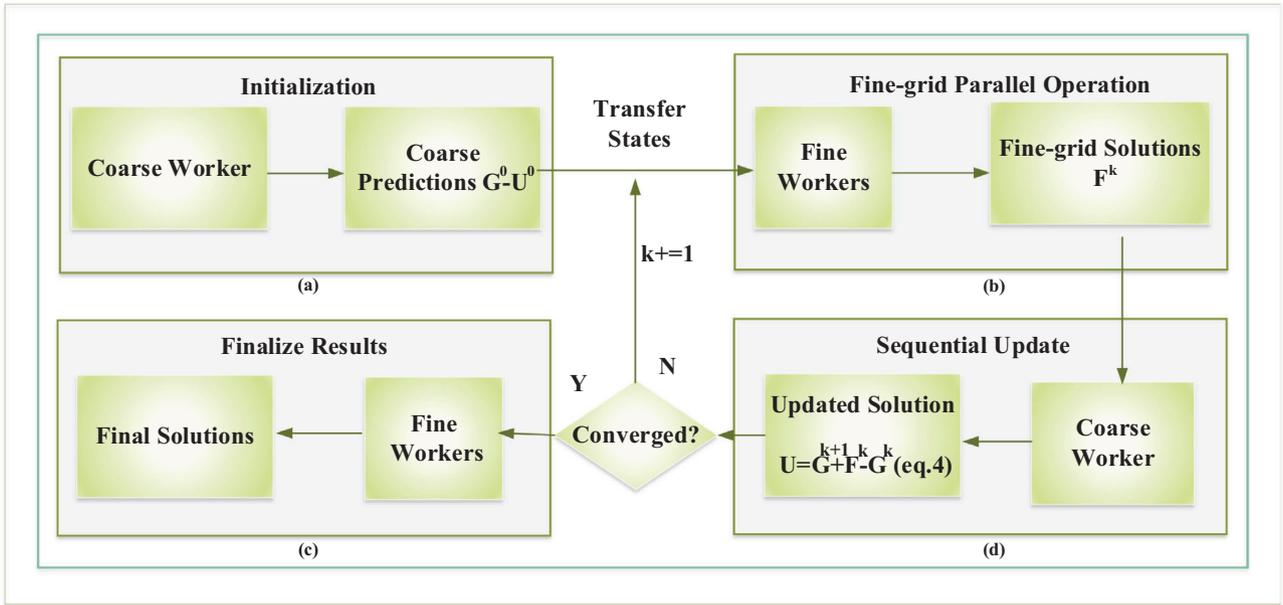


FIGURE 7 Four stages of PiT EMT simulation (a) Initialize variables and produce the initial prediction $G^0 = U^0$. (b) Fine-grid workers solve for fine-grid solutions F . (c) Coarse worker performs Equation (4) to get refined solution U . (d) Fine-grid workers generate final solutions after convergence

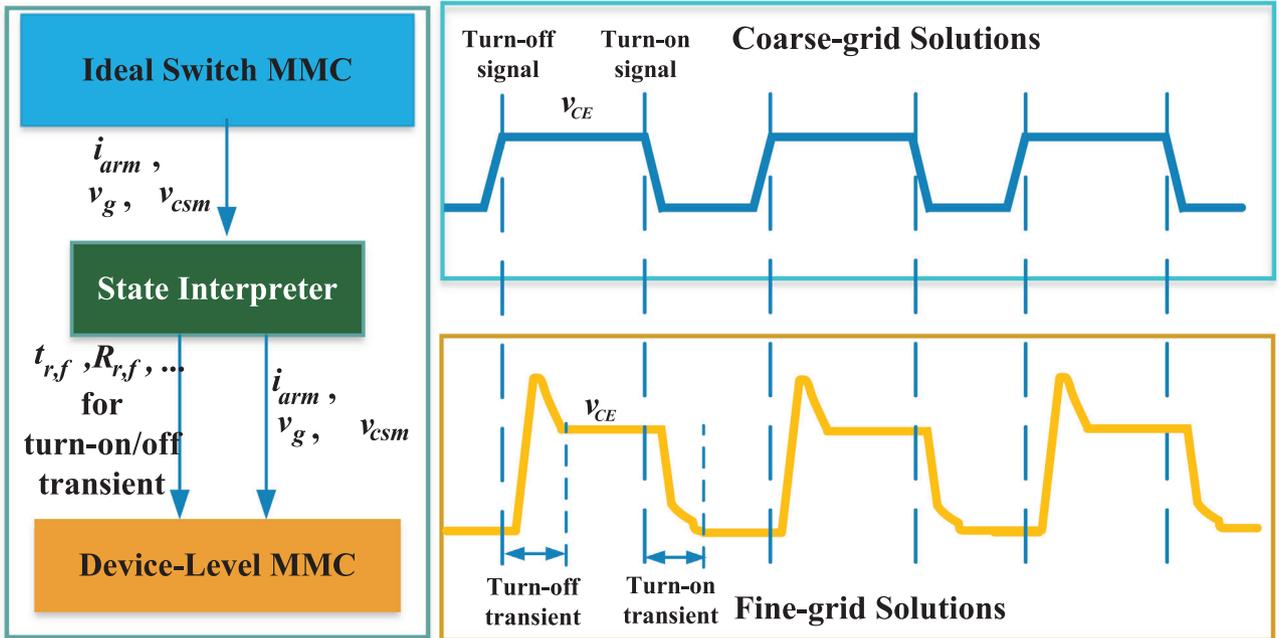


FIGURE 8 State interpretation from the coarse-grid ideal switch model to the fine-grid device-level model

transient; finally, the fine-grid device-level curve-fitting model produces the transients in parallel. The control system only operates on the coarse-grid so there is no different behaviour between coarse-grid and fine-grid control signals, making the synchronization between coarse-grid and fine-grid much easier. From the experimental results, the ideal switch model produces accurate enough system-level results so it is possible to skip the PiT iteration process for extreme performance. The

PiT algorithm is generally a trade-off between accuracy and performance, so if some fault transients require high accuracy or cause diverge, the algorithm can always use a smaller time-step or revert to sequential to overcome these difficulties and go back to PiT in the steady-state.

The hybrid PiT implementation is based on OpenMP® API. OpenMP® pragmas are directly compiled by the C/C++ compiler which masks the complexity to call system-specific

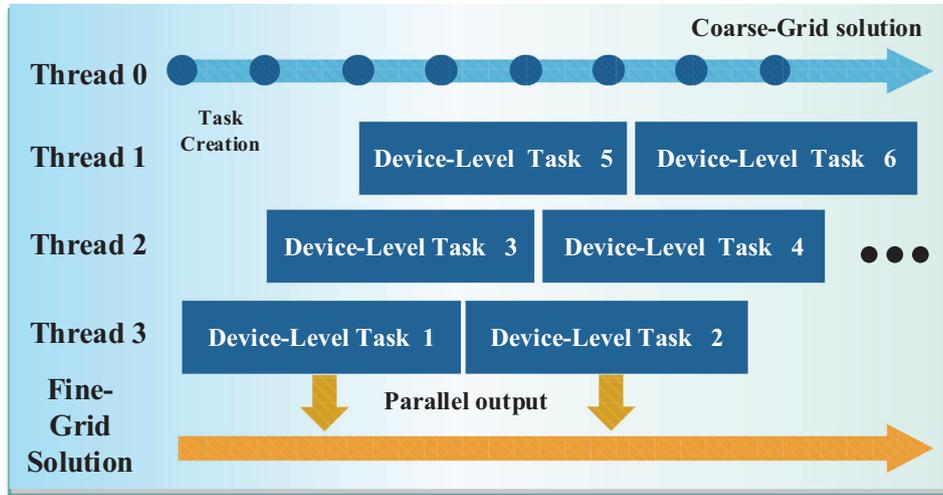


FIGURE 9 OpenMP[®] task-based parallelism for hybrid PiT method for MMC

API functions. Moreover, the OpenMP[®] API provides many useful functionalities such as thread-pooling, load-balancing, tasking and even heterogeneous hardware off-loading in the OpenMP 5.0 standard, which are extremely useful for parallel computing [33]. In Figure 9, the coarse-grid predictor launches a fine-grid parallel task when a solution is ready, which is implemented by the OpenMP[®] tasking feature. This task-based parallel scheme can hide the states translating latency and benefit from the workload balancing by the task scheduler. A fixed-size block of memory is assigned to each task for the outputs; each task writes to the final result vector in parallel and overrides the original coarse-grid time points.

4.3 | TLM-based PiT+PiS method

Different kinds of elements exist in power systems, while not all of them currently have a PiT implementation. Transmission lines modelled based on travelling wave theory bring DDEs to the systems of EMT simulation [25], which needs extra treatments in PiT methods. The transmission lines bring DDE challenges, but it also makes it possible to divide a large PiT system into decoupled PiT subsystems. Therefore, this paper provides a more practical and efficient method than in [25] to handle TLMs, which can also achieve the PiT+PiS goal. The TLM used in this work is Bergeron Line Model, while other travelling-wave-based TLMs can also be used with the same principles.

Figure 10 shows the overview of a PiT+PiS simulation architecture. Because of the usage of TLMs and PiT adapters, the PiT and traditional PiS methods can be used in subsystems without modifications. The only difference is that all subsystems should synchronize at the minimal time window required by PiT systems.

The transmission line elements connect two systems via a sending-end and a receiving-end queue, which are usually implemented by ring buffers. All queues are managed by a global queue hub. The queue hub is a hash map in which all trans-

mission line elements can look up their queue objects by the unique designated names in a string. For multi-thread computing, these queues are implemented as atomic lock-free single-reader-single-writer queues.

The coarse-grid and fine-grid line models in the parallel-in-time implementation must use the same fine-grid history data to get correct solutions due to the TLM limitation. Besides, the iteration process requires restarting the simulation. Therefore, a PiT adapter is used to decouple the queue hub from PiT TLM and traditional TLM. The PiT adapter is a simple memory buffer for PiT systems to repeatedly read the history vector from other systems so that the iteration process can proceed normally. The size of time windows is limited by the propagation delay and the memory size of buffers should be pre-determined according to the system configurations.

In the conventional implementation, a pair of transmission lines will send and receive one packet of history data via the queues at each simulation step. For PiT implementation, it is necessary to synchronize per time window. It also has a PiT adapter as a buffer between conventional and PiT line models since the PiT transmission line models may have different interpretation schemes.

5 | RESULTS AND DISCUSSION

Figure 11 shows the CIGRÉ B4 DC test system for case studies. Two cases are presented to evaluate the accuracy and performance of proposed methods. The performance is evaluated by the theoretical speed-up, actual speed-up, and parallel efficiency. The theoretical speed-up depends on recording function calls to coarse-grid and fine-grid functions:

$$S_{theor} = \frac{C_{coarse} + C_{fine} * R_{fine}}{C_{seq}}, \quad (13)$$

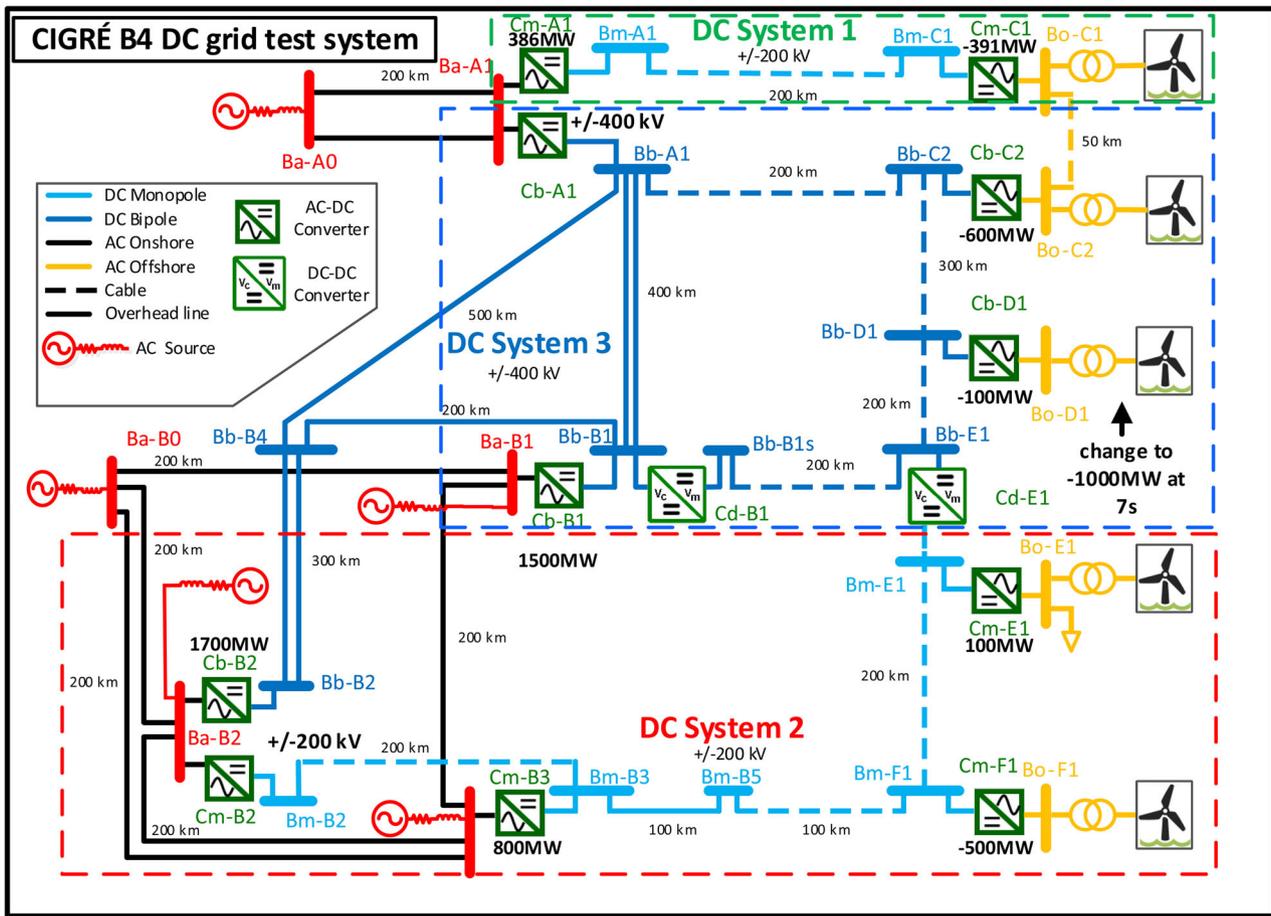


FIGURE 11 CIGRÉ HVDC grid test system for the case study

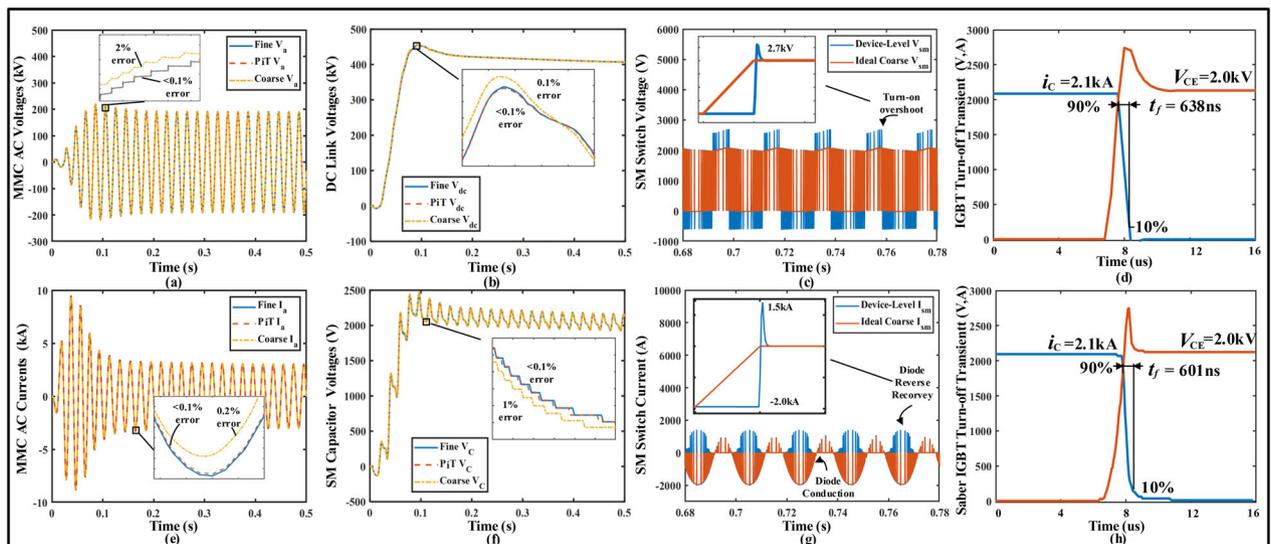


FIGURE 12 Simulation results of Case 1 with PiT and serial methods (a) Three-phase voltages of MMC. (b) DC link voltages. (c) Voltages across the IGBT S_2 of SM-0. (d) IGBT turn-off transient from PiT method. (e) Three-phase currents of MMC. (f) Capacitor voltages of SM-0. (g) Currents through the IGBT S_2 of SM-0. (h) IGBT turn-off transient from fourth-order IGBT model igbt1_3x in SaberRD[®]

TABLE 1 Performance comparison of ideal switch parareal algorithm with different MMC scales over simulation duration=1s, Thread Number=6, $\Delta t = 50\mu\text{s}$ and $\Delta t = 1\mu\text{s}$

SM Num. N_{sm}	Parareal (s)	Traditional (s)	Theoretical speed-up	Actual speed-up	Average iterations
8	0.31	0.39	2.33	1.28	2.26
16	0.42	0.55	2.20	1.29	2.41
32	0.54	0.67	2.06	1.24	2.57
64	0.92	1.05	1.93	1.13	2.77
100	1.40	1.60	1.88	1.14	2.84
200	2.84	2.74	1.84	0.96	2.91

TABLE 2 Performance comparison of hybrid algorithm with different MMC scales over simulation duration = 1 s, thread number = 6, $\Delta t = 50\mu\text{s}$ and $\Delta t = 1\mu\text{s}$

SM Number N_{sm}	Hybrid PiT (s)	Serial curve-fitting Model (s)	Speed-up	Parallel efficiency
8	0.168	0.803	4.78	0.80
16	0.267	1.511	5.65	0.94
32	0.471	2.695	5.72	0.96
64	0.872	5.062	5.80	0.97
100	1.714	7.662	4.47	0.75
200	3.491	15.50	4.44	0.74

variable time-step up to 10ns. Despite the curve-fitting model is computed with a time-step of $1\mu\text{s}$, its maximum voltage and transient duration are close to SaberRD[®].

The performance of PiT methods is compared in Tables 1 and 2, where the speed-up is defined as the proposed parallel execution time over that of the conventional method, and efficiency is calculated as speed-up divided by the number of threads. Table 1 is the traditional Parareal implementation with the ideal switch model. With SM numbers lower than 100 per arm, the Parareal program can be 110-120% faster than the serial program. When the SM number increases to 200, the Parareal method loses speed-up. Compared to the highly optimized serial program, although the Parareal gets accurate results, its iteration overhead is so large that it is difficult to get a reasonable speed-up with a large SM number.

In Table 2, the hybrid model gets much better performance with a trade-off of accuracy. The parallel efficiency is greater than 70% in all cases. When the number of SMs N_{sm} is between 16-64, almost full parallel efficiency is obtained. This can be inferred from Figure 9 as when the conventional program finishes the solution whose length equals Task 1, the PiT method has almost finished 6 tasks since the coarse prediction in Thread 0 is much faster than the conventional model, and the Thread 0 can also handle tasks when it is idle. If the predictor is not fast enough or the task executes much longer than the conventional implementation, which is similar to the $N_{sm} = 8$ or 200, the efficiency drops. Thus, this method sacrificed a little accuracy but gets a huge performance boost.

Since the single MMC station has no limit on the time window, the thread number can go up to a maximum of 48. Table 3 shows the multi-thread performance of the hybrid algorithm

with a 201-level MMC. High parallel efficiency greater than 60% is obtained in all test cases. Notice that the simulation for 1s only takes 0.527s when using 48 threads (30 \times speed-up) and $\delta t = 1\mu\text{s}$.

5.2 | CASE 2: CIGRÉ HVDC Grid

The full-scale CIGRÉ HVDC grid test system is used to verify the connection among PiT systems. All 11 AC-DC converters are modelled by the MMC model mentioned in Section 3. DC-DC converter Cd-B1 is bypassed and Cd-E1 is disconnected following the guide of the CIGRÉ B4 test case. All MMC parameters including controller set-points are set according to the standard power flow as shown in Figure 11, except Cb-D1, which has only 100 MW generating power at the beginning. The Cb-D1 power will change to 1000 MW at 7s, and the standard

TABLE 3 Performance test of hybrid algorithm with different number of threads over $N_{sm} = 200$, Simulation Duration = 1s, $\Delta t = 50\mu\text{s}$ and $\Delta t = 1\mu\text{s}$

Threads	Time (s)	Speed-up	Efficiency
48	0.527	30.42	0.63
24	0.950	16.87	0.70
12	1.838	8.72	0.73
6	3.612	4.44	0.74
4	5.478	2.93	0.73
2	10.998	1.46	0.73

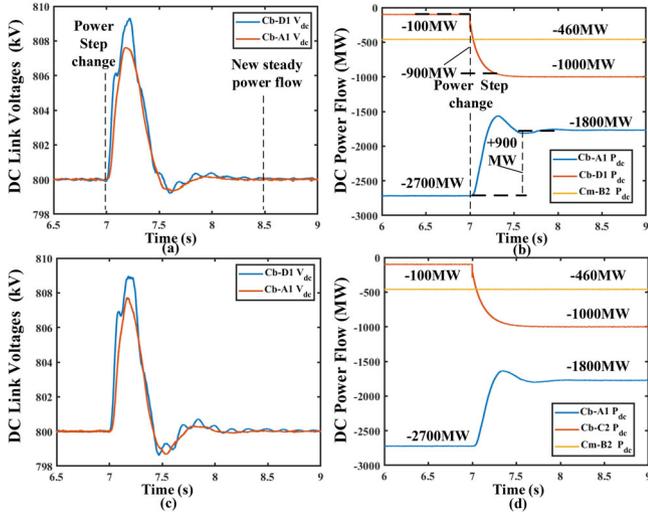


FIGURE 13 Simulation results of CIGRÉ B4 DC grid test system (a) DC-link voltages when power step-change at 7s from the PiT+PiS program. (b) Real power flow at MMC stations from the PiT+PiS program. (c) DC-link voltages from PSCAD/EMTDC[®]. (d) Real power flow from PSCAD/EMTDC[®]

power flow should be observed at Cb-A1. All MMCs have 200 curve-fitting device-level HBSMs per arm.

The results are shown in Figure 13. Figure 13(b) shows the power flow at Cb-A1 and Cb-D1. As the power output of Cb-D1 increases, the Cb-A1 returns the same amount of power to the AC system; the final power flow is -1800 MW while the standard power flow file is around -2000 MW, which is a little different. This is because the DC-DC station Cd-E1 is supposed to transmit 300 MW power from DCS3 to DCS2 in the standard case. In the simplified case, Cd-E1 is disconnected so the surplus 300 MW power is supplied by Cm-B2, and with the surplus real power, the Cb-A1's power generation decreases.

Another scenario for Case 2 is a DC fault that occurs at 6s in DCS 2. The phase-to-phase fault resistance is 1Ω , which is located in the middle of Bm-B2 and Bm-B3. The results are shown in Figure 14; the Figure 14(a) and (b) are DC voltages and currents respectively of MMCs in DCS 2 with no change to the SM states. Figure 14c,d is the result of MMCs with SMs changed to blocking state $100\mu\text{s}$ after the phase-to-phase fault. The Cm-B2 and Cm-B3 have a larger impact on voltages and currents since they are closer to the fault location. Comparing the blocked SMs to the unblocked SMs, the fault voltages are similar but their transients become different as the blocked version has smaller oscillations. The currents are more different and the maximum fault current of blocked SMs is about half of the unblocked SMs' maximum fault current. The fault current reaches the peak value within 50ms in both cases, which indicates that faster protection equipment is required to effectively protect the power system from the potential damage of DC faults.

Only hybrid PiT program performance tests are presented. For the PiT program, $2.81\times$ speed-up and 70% efficiency can be obtained with four threads. The performance test on the CIGRÉ system is different from the previous ones. Since mul-

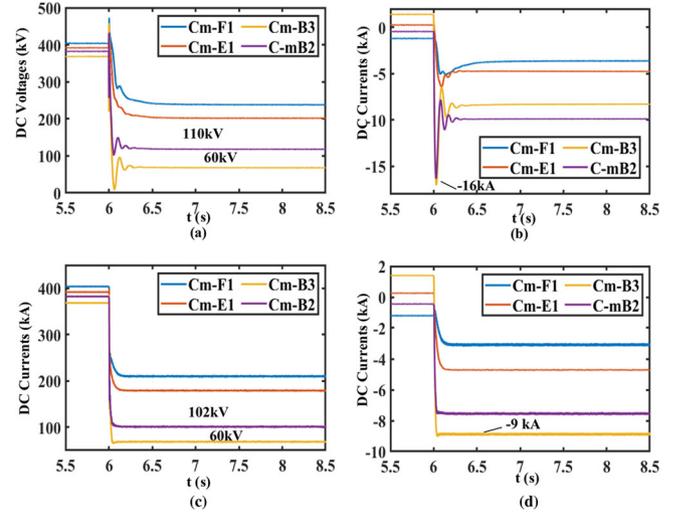


FIGURE 14 Simulation results of a DC line fault in CIGRÉ B4 DCS 2 system (a) MMC DC voltages without blocked SMs. (b) MMC DC currents without blocked SMs. (c) MMC DC voltages with blocked SMs. (d) MMC DC currents with blocked SMs

tipple MMC stations can be executed in parallel with high efficiency, it is worthy to check the performance of PiT+PiS against the traditional TLM decoupled PiS method. However, it creates two levels of parallelism with a nested relationship, which brings extra complexity and overhead. In this test, a number of PiS threads are assigned to the first level of parallelism; each PiS thread launches its own team of 4 threads for PiT tasks at the second level; different teams cannot communicate with each other.

Results are given in Table 4. The PiS method achieves around 80% efficiency when the thread number is equal to or less than 11 with an exception of $N = 8$. When $N = 8$, it may be affected by the load-imbalance so that the efficiency is lower than the balanced case $N = 11$. When the thread number is greater than 11, the speed-up stops at $10\times$ since there are only 11 MMC stations. For the PiT+PiS program, the speed-up and efficiency are generally lower than pure PiS or PiT cases due to the overhead brought by the nested parallelism. However, it extends the boundary of overall speed-up and can utilize more threads to accelerate the simulation. Therefore, greater than $20\times$ speed-up is obtained with $N > 24$.

6 | CONCLUSION

EMT simulation of MTDC grids especially with device-level models for MMC converters is time-consuming not only due to the accuracy constraint but also due to the excessive computational burden. This paper investigated a joint parallel scheme that takes both advantages of traditional TLM-based spatial and Parareal-based temporal parallel techniques. To realize the device-level IGBT transient simulation under PiT, a hybrid PiT method that utilizes ideal switch as the coarse-grid predictor and curve-fitting IGBT model as fine-grid corrector is proposed and implemented on multi-core CPU using

TABLE 4 Performance comparison of TLM PiS, PiT+ PiS and serial program, over simulation duration=1s, $\Delta t = 50\mu\text{s}$ and $\Delta t = 1\mu\text{s}$

PiS single level parallel computing				PiT+PiS nested parallel computing			
Threads N	Time (s)	Speed-up	Efficiency	Threads N	Time (s)	Speed-up	Efficiency
1	294.60	1.00	1.00	$1 \times 4 = 4$	105.21	2.81	0.70
2	171.57	1.72	0.86	$2 \times 4 = 8$	69.79	4.22	0.53
4	86.55	3.40	0.85	$4 \times 4 = 16$	34.53	8.53	0.53
6	58.38	5.05	0.84	$6 \times 4 = 24$	15.60	18.89	0.79
8	58.85	5.01	0.63	$8 \times 4 = 32$	12.92	22.80	0.71
11	30.57	9.64	0.88	$11 \times 4 = 44$	14.12	20.86	0.47
16	29.49	9.99	0.62	$16 \times 4 = 64$	12.70	23.20	0.36
32	29.88	9.86	0.31	$32 \times 4 = 128$	12.69	23.21	0.18
48	29.80	9.88	0.21	$48 \times 4 = 192$	13.58	21.70	0.11

OpenMP[®] tasking to achieve maximum efficiency. A TLM propagation-delay-based method is integrated into the PiT systems to enable flexibility regarding forming an MTDC grid. Test results show a 30 \times speed-up of the hybrid PiT model for device-level simulation of a 201-level three-phase MMC with 48 threads, with an error of less than 2%. High parallel efficiency from 60% to 97% is achieved with the proposed hybrid model. For the CIGRÉ HVDC grid test system, the PiT+PiS method shows a 20 \times greater speed-up than the pure PiT or PiS method. The case studies on a single MMC and CIGRÉ B4 system show a significant speed-up of the proposed methods and indicate great potentials for hybrid PiT+PiS methods for device-level MTDC EMT simulation. Future applications can be extended to non-linear physical power electronic simulation, the power system with renewable energy sources, and other applications with complex device-level transients.

NOMENCLATURE

DAE	Differential-algebraic equation
DDE	Delay differential equation
EMT	Electromagnetic transient
HVDC	High-voltage direct current
IGBT	Insulated gate bipolar transistor
ODE	Ordinary differential equation
PiT	Parallel-in-time
PiS	Parallel-in-space
TLM	Transmission line model
MTDC	Multi-terminal direct current
MMC	Modular multilevel converter
\mathcal{F}	Fine solution operator
\mathcal{G}	Coarse solution operator
F	Result vector of fine solution operator
G	Result vector of coarse solution operator
t_r	Rising time
t_f	Falling Time
U^k	System state vector of k -th iteration
Δt	Coarse-grid time-step
δt	Fine-grid time-step

ACKNOWLEDGEMENTS

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC).

DATA AVAILABILITY STATEMENT

Data available on request from the authors

ORCID

Tianshi Cheng  <https://orcid.org/0000-0001-7919-7368>

Ning Lin  <https://orcid.org/0000-0002-7220-2109>

Tian Liang  <https://orcid.org/0000-0002-1501-9789>

Venkata Dinavahi  <https://orcid.org/0000-0001-7438-9547>

REFERENCES

- Vrana, T., Dennetiere, S., Yang, Y., Jardini, J.A., Jovic, D., Saad, H.: The CIGRÉ B4 DC grid test system. *CIGRÉ Electra* 270 (2013)
- Guo, G., Wang, H., Song, Q., Zhang, J., Wang, T., Ren, B., et al.: HB and FB MMC based onshore converter in series-connected offshore wind farm. *IEEE Trans. Power Electron.* 35(3), 2646–2658 (2020)
- Xu, J., Zhao, Y., Zhao, C., Ding, H.: Unified high-speed EMT equivalent and implementation method of MMCs with single-port submodules. *IEEE Trans. Power Delivery* 34(1), 42–52 (2019)
- Xu, J., Fan, S., Zhao, C., Gole, A.M.: High-speed EMT modeling of MMCs with arbitrary multiport submodule structures using generalized norton equivalents. *IEEE Trans. Power Delivery* 33(3), 1299–1307 (2018)
- Lin, N., Dinavahi, V.: Variable time-stepping modular multilevel converter model for fast and parallel transient simulation of multiterminal dc grid. *IEEE Trans. Ind. Electron.* 66(9), 6661–6670 (2019)
- Duan, T., Dinavahi, V.: Variable time-stepping parallel electromagnetic transient simulation of hybrid ac–dc grids. *IEEE J. Emerging Selected Topics Ind. Electron.* 2(1), 90–98 (2021)
- Lin, N., Dinavahi, V.: Parallel high-fidelity electromagnetic transient simulation of large-scale multi-terminal dc grids. *IEEE Power Energy Technol. Syst. J.* 6(1), 59–70 (2019)
- Lin, N., Dinavahi, V.: Exact nonlinear micromodeling for fine-grained parallel emt simulation of mtcd grid interaction with wind farm. *IEEE Trans. Ind. Electron.* 66(8), 6427–6436 (2019)
- Shu, D., Wei, Y., Dinavahi, V., Wang, K., Yan, Z., Li, X.: Cosimulation of shifted-frequency/dynamic phasor and electromagnetic transient models of hybrid LCC-MMC DC grids on integrated CPU–GPUs. *IEEE Trans. Ind. Electron.* 67(8), 6517–6530 (2020)
- Dinavahi, V., Lin, N.: *Real-Time Electromagnetic Transient Simulation of AC-DC Networks*. Wiley-IEEE Press, Hoboken (2021)

11. Chen, Y., Dinavahi, V.: Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems. *IET Gener., Transm. & Distrib.* 7(5), 451–463 (2013)
12. Shen, Z., Dinavahi, V.: Real-time device-level transient electrothermal model for modular multilevel converter on FPGA. *IEEE Trans. Power Electron.* 31(9), 6155–6168 (2016)
13. Liang, T., Dinavahi, V.: Real-time device-level simulation of MMC-based MVDC traction power system on MPSoC. *IEEE Trans. Transp. Electrific.* 4(2), 626–641 (2018)
14. Semlyen, A., de Leon, F.: Computation of electromagnetic transients using dual or multiple time steps. *IEEE Trans. Power Syst.* 8(3), 1274–1281 (1993)
15. Lin, N., Liu, P., Dinavahi, V.: Component-level thermo-electromagnetic nonlinear transient finite element modeling of solid-state transformer for dc grid studies. *IEEE Trans. Ind. Electron.* 68(2), 938–948 (2021)
16. Shu, D., Xie, X., Jiang, Q., Guo, G., Wang, K.: A multirate EMT co-simulation of large AC and MMC-based MTDC systems. *IEEE Trans. Power Syst.* 33(2), 1252–1263 (2018)
17. Lin, N., Cao, S., Dinavahi, V.: Adaptive Heterogeneous Transient Analysis of Wind Farm Integrated Comprehensive AC/DC Grids. *IEEE Transactions on Energy Conversion.* 36(3), 2370–2379 (2021). <http://doi.org/10.1109/tec.2020.3043307>
18. Tai, G.C., Korman, C.E., Mayergoyz, I.D.: A parallel-in-time method for the transient simulation of SOI devices with drain current overshoots. *IEEE Trans. Comp.-Aided Design Integr. Circuits Syst.* 13(8), 1035–1044 (1994)
19. Carraro, T., Geiger, M., Rorkel, S., Rannacher, R.: *Multiple Shooting and Time Domain Decomposition Methods*. Springer, New York (2015)
20. Gurrula, G., Dimitrovski, A., Pannala, S., Simunovic, S., Starke, M.: Parareal in time for fast power system dynamic simulations. *IEEE Trans. Power Syst.* 31(3), 1820–1830 (2016)
21. Lecouvez, M., Falgout, R.D., Woodward, C.S., Top, P.: A parallel multigrid reduction in time method for power systems. In: 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, USA, pp. 1–5 (2016)
22. Schöps, S., Niyonzima, I., Clemens, M.: Parallel-in-time simulation of eddy current problems using parareal. *IEEE Trans. Magn.* 54(3), 1–4 (2018)
23. Zhou, Z., Dinavahi, V.: Parallel massive-thread electromagnetic transient simulation on gpu. *IEEE Trans. Power Delivery* 29(3), 1045–1053 (2014)
24. Gander, M.J., Hairer, E.: *Domain decomposition methods in science and engineering XVII*. Springer, New York (2008)
25. Cheng, T., Duan, T., Dinavahi, V.: Parallel-in-time object-oriented electromagnetic transient simulation of power systems. *IEEE Open Access J. Power Energy* 7, 296–306 (2020)
26. Champagne, R., Dessaint, L.A., Fortin, Blanchette, H., Sybille, G.: Analysis and validation of a real-time ac drive simulator. *IEEE Trans. Power Electron.* 19(2), 336–345 (2004)
27. Saad, H., Dennetière, S., Mahseredjian, J., Delarue, P., Guillaud, X., Peralta, J., et al.: Modular multilevel converter models for electromagnetic transients. *IEEE Trans. Power Delivery* 29(3), 1481–1489 (2014)
28. Tu, Q., Xu, Z.: Impact of sampling frequency on harmonic distortion for modular multilevel converter. *IEEE Trans. Power Delivery* 26(1), 298–306 (2011)
29. Kuffel, P., Kent, K., Irwin, G.: The implementation and effectiveness of linear interpolation within digital simulation. *Int. J. Electrical Power Energy Syst.* 19(4), 221–227 (1997)
30. Faruque M.O., Dinavahi V., Xu W.: Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power-Electronic Systems. *IEEE Transactions on Power Delivery.* 20(2), 1157–1167 (2005). <http://doi.org/10.1109/tpwrd.2004.834672>
31. ABB. ABB 5SNA-2000K450300 StakPak IGBT module. <https://new.abb.com/products/5SNA2000K450300/stakpak-igbt-module> (2020). Accessed 4 June 2020
32. Dugal, F., Tsyplakov, E., Baschnagel, A., Storasta, L., Clausen, T.: IGBT press-packs for the industrial market. In: *PCIM Europe proceedings, Nuremberg, Germany*, pp. 800–807 (2012)
33. Bai, H., Liu, C., Rathore, A.K., Paire, D., Gao, F.: An FPGA-based IGBT behavioral model with high transient resolution for real-time simulation of power electronic circuits. *IEEE Trans. Ind. Electron.* 66(8), 6581–6591 (2019)
34. The OpenMP API specification for parallel programming. <https://www.openmp.org/> (2020). Accessed 4 June 2020

How to cite this article: Cheng, T., Lin, N., Liang, T., Dinavahi, V.: Parallel-in-time-and-space electromagnetic transient simulation of multi-terminal DC grids with device-level switch modelling. *IET Gener. Transm. Distrib.* 16, 149–162 (2022). <https://doi.org/10.1049/gtd2.12285>

APPENDIX

Test Environment: Compute Canada Cedar cluster; Two Intel® Xeon® Platinum 8260 processors (24 Cores, base frequency: 2.4GHz, cache L3: 35.75MB, cache L2: 24MB). Software configuration: Operating System: CentOS 7.7, Linux 3.10 kernel; Compiler: GCC/G++ 7.3, OpenMP 4.5.

TABLE A.1 System Parameters of Study Case 1

Parameter	Symbol	Value
Submodule number per arm	N	200
Capacitor size	C_{sm}	20mF
Arm inductance	L_{arm}	29mH
System frequency	f	50Hz
Controller sample-time	T_s	50 μ s
DC bus voltage	V_{dc}	+/-200kV
Active Power	P	800 MW