

Real-time Electromagnetic Transient Simulation of Multi-Terminal HVDC-AC Grids based on GPU

Jingfan Sun, *Student Member, IEEE*, Suman Debnath, *Senior Member, IEEE*, Maryam Saeedifard, *Senior Member, IEEE*, and Phani R.V. Marthi, *Member, IEEE*

Abstract—High-fidelity electromagnetic transient (EMT) simulation plays a critical role in understanding the dynamic behavior and fast transients involved in operation, control, and protection of Multi-Terminal dc (MTdc) grids. This paper proposes a cost-effective high-performance real-time EMT simulation platform for large-scale cross-continental MTdc grids based on graphics processing unit (GPU). Fast dynamic transients from both ac and dc networks are captured in real time with $5\ \mu\text{s}$ time step, using advanced hybrid-discretized modular multilevel converter (MMC) model, frequency-dependent transmission line model, and EMT-type model of synchronous generators. The proposed simulation platform: i) assembles detailed EMT models of all components within an MTdc-ac grid into a single platform. This setup provides a complete simulation solution to capture fast transient signals required for high-bandwidth controller design and protection studies without any compromise; ii) implements the first GPU-based simulation architecture and corresponding algorithms for MTdc-ac grids with real-time performance at scales of 1 s; iii) is highly-efficient and balances the high utilization of GPU resources and low latency required for the simulation; and iv) outperforms the existing central processing unit (CPU)- or digital signal processor (DSP)/field-programmable gate array (FPGA)-based simulators in terms of its higher scalability on large-scale MTdc-ac grids and superior price-performance ratio on the hardware. Accuracy and performance of the proposed platform are evaluated with respect to the reference results from PSCAD/EMTDC environment.

Index Terms—Multi-terminal HVdc systems, Real-time simulation, Graphics processing unit

I. INTRODUCTION

HIGH Voltage dc (HVdc) transmission is a mature technology with many installations around the world [1]–[3]. Over the past few years, significant breakthroughs in Voltage-Sourced Converters (VSCs) along with their attractive features have made the HVdc technology even more promising. The VSCs provide enhanced reliability with reduced cost and power losses compared to Line-Commutated Converters (LCC). Concomitantly, significant changes in generation,

transmission, and loads have emerged. These changes include integration and tapping renewable energy generation in remote areas, relocation or retiring older conventional and/or nuclear power plants, increasing transmission capacity, and urbanization [2]. These new trends have called for VSC-based Multi-Terminal dc (MTdc) systems, which when embedded inside the ac grid, can enhance stability, reliability, and efficiency of the present power grid [3].

The strategic importance of MTdc grids is evidenced by the number of worldwide projects currently in their planning stage, e.g., European “Supergrids” and the Baltic Sea project along with a few projects in China. In the US, the cross-continental dc/ac grid is expected to enable the connection of three primary interconnections [4], i.e., the Western Interconnection (WI), Eastern Interconnection (EI), and Electric Reliability Council of Texas (ERCOT). To understand the operation, control, and protection of such a sophisticated system, fast transients and dynamic behavior of both power system and power electronics devices present in the system have to be investigated [5]. This requires the use of detailed high-fidelity electromagnetic transient (EMT) models to address the extensive appearance of nonlinear devices in the simulation [6], [7]. EMT studies have been comprehensively conducted in both ac and dc networks, including transmission lines [8], modular multilevel converters (MMCs) [9]–[13], and transformers and generators [14]–[16]. These models provide building blocks for the detailed analysis, like the study of fault transients, stability, frequency support, etc., of MTdc-ac grids.

Based on case specific requirements, EMT simulations can be performed either offline or in real time. Real-time simulation synchronizes the program execution with the real-world clock, enabling the interaction with actual control and protection equipment. This makes it a good choice for the design and evaluation of control and protection systems. To meet real-time requirements in simulation of an MTdc-ac grid, each component should be modelled with high fidelity while staying computationally efficient. The MMCs typically consist of hundreds of submodules (SMs) in each arm. The large number of SMs in an MMC result in thousands of states [25], [26]. These states include arm currents, capacitor voltages, and switching signals, and controller states. To capture the fast transients and cope with high bandwidth of MMC controllers, time steps in the order of few microseconds should be applied [19]. Real-time calculation of large number of states within small time intervals requires sufficient computation

Jingfan Sun and Maryam Saeedifard are with the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA 30332-0250, USA (e-mails: jingfan@gatech.edu; maryam@ece.gatech.edu). Suman Debnath is with Oak Ridge National Laboratory, Knoxville, TN 37932, USA (email: debnaths@ornl.gov). Phani R.V. Marthi is with Missouri University of Science and Technology and Oak Ridge National Laboratory (email: pm226@mst.edu).

TABLE I
THE EXISTING AND PROPOSED REAL-TIME SIMULATION SPECIFICATIONS OF THE MTDC GRID

Paper	Year	Step size	MMC model	SMS/arm	Transmission line	Generator	Terminals	Platforms
[17]	2018	5 μ s	detailed	400	N/A	ideal source	1	DSP
[18]	2014	2.5 μ s (MMC), 50 μ s (rest)	detailed	200	Bergeron	ideal source	3	CPU & FPGA
[19]	2015	9 μ s	detailed	400	distributed	ideal source	1	CPU & FPGA
[20]	2016	8 μ s (MMC), 30 μ s (rest)	detailed	100	N/A	ideal source	2	CPU & FPGA
[21]	2019	20 μ s	device & equivalent	24	distributed	ideal source	3	CPU & FPGA
[22]	2018	2.7 μ s (MMC), 51 μ s (rest)	detailed	400	N/A	ideal source	2	CPU & FPGA
[23]	2017	5 μ s (MMC), 25 μ s (rest)	detailed	500	N/A	ideal source	2	CPU & FPGA
[24]	2018	20 μ s	arm equivalent	200	freq. dependent	ideal source	11	multiple CPUs
Proposed Method		5 μ s	detailed	400	freq. dependent	EMT model	3+	CPU & GPU

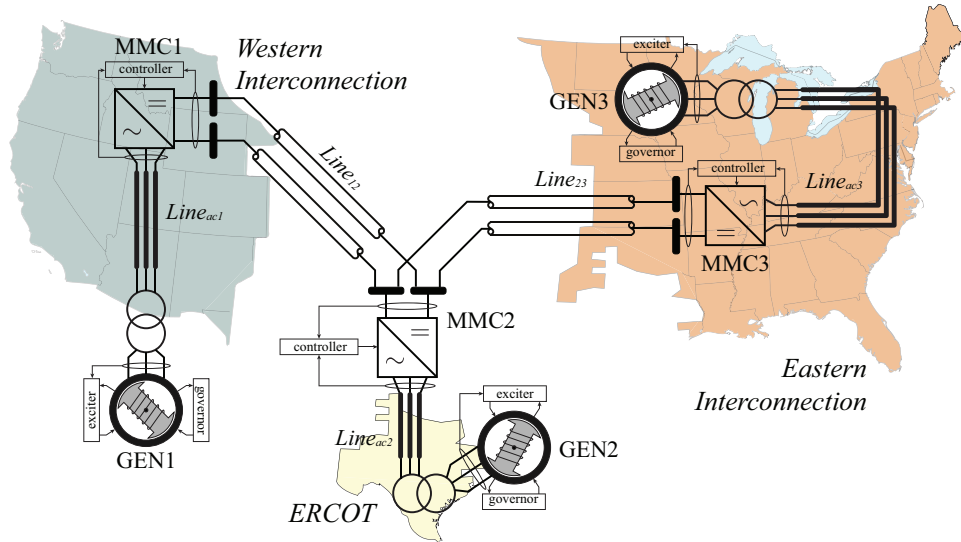


Fig. 1. Diagrams of the US cross-continental MTdc grid connecting the three major interconnections, i.e., WI, EI, and ERCOT [4].

power and precise calibration of parallelism [17]. In the transmission networks, propagation of travelling waves and frequency dependent effects are critical for analysis of dc/ac-side faults [27], [28]. Fast transients should also be captured for study of controller stability [7]. To these ends, both ac and dc lines/cables should be modelled with high fidelity and based on frequency-dependent models. Additionally, the ac grid connecting to the MTdc system consists of synchronous generators, transformers, and ac lines/cables [26]. The computational complexity of simulating MTdc-ac systems with models of components mentioned above, makes it difficult to achieve real-time simulation at small time-steps. It is also challenging to keep the simulation system scalable as more ac and dc terminals are introduced while expanding an MTdc system.

The real-time simulation platforms of MTdc systems have been implemented on different computing hardware with various level of details, as summarized in Table I. Most of these platforms are built on central processing units (CPUs) and field-programmable gate arrays (FPGAs) [18]–[23], where the simulation of MMC is performed on FPGAs. A high-end FPGA board is capable of simulating 1500 to 3000 SMS, which is the number of SMS in a single MMC [18], [20], [23]. This number varies depending on the SM circuit configurations and FPGA models. Consequently, the number of FPGAs has to be scaled out linearly as the size of MTdc system grows, which

is not an economic option for simulation of a large MTdc-ac grid. Additionally, extensive matrix-matrix or matrix-vector computation is required for simulating frequency-dependent transmission line models. It is challenging to implement these data-intensive operations efficiently on an FPGA. The MMC models adopted in existing real-time platforms include detailed equivalent models [17]–[20], [22], [23], device-level models [21], and arm equivalent models [24]. To study fault transients and advanced control methods (like frequency, voltage support to ac grids) in a large MTdc system, the detailed equivalent models are preferable. These models strike a reasonable balance between modelling details and the computational load [7]. Real-time simulations of these MMC models and their controllers have been performed with time steps less than ten microseconds. However, the rest of the system, i.e., the transmission lines in dc and ac systems, and transformers and generators in ac systems, have been simulated at much larger steps using the simplified models. The highly simplified ac/dc systems' model limits the application of these real-time platforms on frequency-sensitive high-bandwidth controller and protection system designs.

To address modelling and scalability issues, in this paper, a cost-effective high-performance real-time EMT simulation platform is proposed for large-scale cross-continental MTdc grids. This platform is built on graphics processing unit (GPU), using hybrid-discretized MMC detailed equivalent model [13],

[17], universal line model (ULM) for lines/cables [8], [29], and EMT-type model for synchronous generators and transformers [14]–[16]. The GPU-based EMT simulation methods have been developed in previous research for MMC device-level modelling [30], ac grids [31]–[33], and MTdc grids [34]. Parallel massive-thread methods are designed in [30] to simulate the nonlinear device-level power transistors within the MMCs. Significant speed-up is achieved using the GPU-based solution. EMT simulations and transient stability studies are performed on GPU platforms for the applications of pure ac grids as well [31]–[33], which have shown large improvements in the speed of such numerical studies in ac grids. In [34], EMT simulation of an MTdc grid connected to wind farms is performed on GPU. However, these simulation methods and platforms are not designed to achieve real-time performance. The proposed simulation platform: i) incorporates detailed EMT models of all components of an MTdc-ac grid within a single platform. This setup provides a complete simulation solution to capture fast transient signals required for high-bandwidth controller design and protection studies, without any compromise; ii) implements the first GPU-based simulation architecture and corresponding algorithms for MTdc-ac grids with real-time performance at scales of 1 s; iii) is highly-efficient and balances the high utilization of GPU resources and low latency required for the simulation; and iv) outperforms the existing CPU- and DSP/FPGA-based simulators in terms of its higher scalability on large-scale MTdc-ac grids and superior price-performance ratio on the hardware.

II. UNITED STATES CROSS-CONTINENTAL MTDC GRID

Fig. 1 shows the layout of the cross-continental MTdc system originated from one of the Grid Modernization Laboratory Consortium (GMLC) projects [4]. This MTdc system, which is embedded across multiple asynchronous ac interconnections in US, enables economic power transfer, mitigation of congestion and loop flow, cross-interconnection frequency support, and loss reduction in the national-scale power grid.

The MTdc system consists of three dc terminals. Two ± 320 kV dc transmission lines/cables are used to connect the three dc terminals in the radial configuration. The dc terminals, which are based on MMCs, are represented by their detailed equivalent models using the hybrid discretization method as described in [13]. Solid grounding of dc capacitor neutral is employed in the system. On the ac sides of the MMC terminals, the ac interconnections are modelled as aggregated generations. They are modeled by the dynamic models of synchronous generators, transformers, and ac transmission lines. The exciters and governors of these synchronous generators are also modelled in detail. The ac interconnection models are based on generic parameters that represent the dynamics of ac grids. The MTdc-ac grid studied in this paper takes inspiration from the above project in [4]. The ac grid parameters have been generalized for applicability of these models to any similar systems.

III. MODELLING OF THE MTDC-AC GRID

In the demonstrated MTdc-ac grid, there are four major components modelled in detail, i.e., the MMCs, transmission

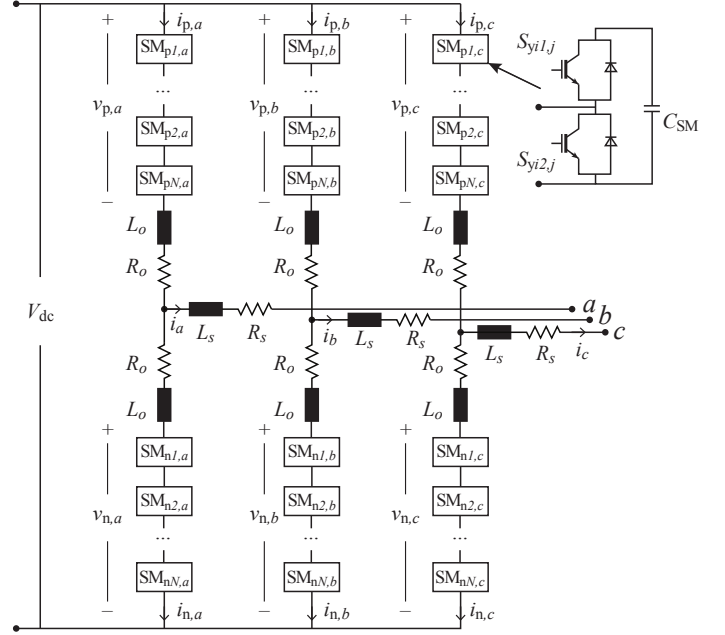


Fig. 2. Circuit diagram of a three-phase MMC.

lines, generators, and transformers. The modelling assumptions and techniques applied are described in this section.

A. MMC Modelling

A three-phase MMC consists of six arms, each of which has an inductor and N series-connected half-bridge SMs, as shown in Fig. 2. Basics of operation and control of MMC are summarized in [9] and are not elaborated here.

The dynamics of MMC arm currents are given by

$$\begin{aligned} (L_o + L_s) \frac{di_{p,j}}{dt} - L_s \frac{di_{n,j}}{dt} &= -(R_o + R_s) i_{p,j} + R_s i_{n,j} \\ &+ \frac{V_{dc}}{2} - v_j - v_{cm} - v_{p,j}, \forall j \in \{a, b, c\}, \\ (L_o + L_s) \frac{di_{n,j}}{dt} - L_s \frac{di_{p,j}}{dt} &= -(R_o + R_s) i_{n,j} + R_s i_{p,j} \\ &+ \frac{V_{dc}}{2} - v_j - v_{cm} - v_{n,j}, \forall j \in \{a, b, c\}, \end{aligned} \quad (1)$$

where the arm voltages $v_{y,j} = \sum_{l=1}^N v_{sm,y,l,j}$ can be further factorized as functions of switching states and SM capacitor voltages $v_{cy,i,j}$, as provided by

$$\begin{aligned} v_{y,j} &= \sum_{i=1}^N \{ \{ (1 - S_{yi1,j})(1 - S_{yi2,j}) v_{cy,i,j} \text{sgn}(i_{y,j}) \} \\ &+ S_{yi1,j} v_{cy,i,j} \}, \forall y \in \{p,n\}, \forall j \in \{a, b, c\}. \end{aligned} \quad (2)$$

The dynamics of SM capacitor voltages are represented by

$$\begin{aligned} C_{SM} \frac{dv_{cy,i,j}}{dt} &= -\frac{v_{cy,i,j}}{R_p} + S_{yi1,j} i_{y,j} + \text{sgn}(i_{y,j})(1 - S_{yi1,j}) \\ &\times (1 - S_{yi2,j}) i_{y,j}, \forall y \in \{p,n\}, \forall j \in \{a, b, c\}. \end{aligned} \quad (3)$$

By assigning zero to the gating signals $S_{yi1,j}$ and $S_{yi2,j}$ of all SMs, equations (2) and (3) can be applied to simulate the

blocking status of MMC, which is critical to understand the fast transients during dc faults, ac startup charging, and dc startup charging [13]. Equations (1)-(3) are discretized with different algorithms, i.e., backward Euler and forward Euler, based on their numerical stiffness. The discretization method applied in this work is the hybrid approach detailed in [13]. The MMC system equations are formed with this approach in the real-time simulation.

B. Transmission Line Modelling

The ac and dc transmission lines/cables in the MTdc system are modelled using the frequency-dependent models [8]. The use of frequency-dependent model enables the detailed characterization of the system dynamics over a wide range of frequency. Such representation is critical for the study of protections and high-bandwidth control methods in the system.

Derived from the well-known transmission line equations,

$$-\frac{dV}{dx} = ZI, \quad -\frac{dI}{dx} = YV, \quad (4)$$

where the voltage V and current I at both ends of the transmission line are correlated in the travelling wave form in frequency domain. This correlation is given by

$$\begin{aligned} I_1 &= Y_c V_1 - H(I_2 + Y_C V_2), \\ I_2 &= Y_c V_2 - H(I_1 + Y_C V_1), \end{aligned} \quad (5)$$

where $Y_c = Z^{-1}\sqrt{YZ}$ is the characteristic admittance matrix, $H = e^{-\sqrt{YZ}l}$ is the propagation matrix.

These two frequency-dependent matrices are sampled at a large number of frequency points across a wide range. To transform (5) into the time domain, Y_c and H are fitted based on these frequency samples and replaced by their time-domain rational function equivalents of lower order (normally less than 20) [35]. These rational functions are discretized and convoluted with their corresponding terminal voltages/currents, and as a result, the time-domain solutions at both ends are obtained, given by

$$\begin{aligned} i_k &= G \cdot v_k - i_{\text{hist},k}, \quad \forall k \in \{1, 2\} \\ i_{\text{hist},k} &= \sum_{i=1}^m \left(\sum_{j=1}^{N_H} h_{j,i,k} \right) - \sum_{i=0}^{N_Y} y_{i,k}, \quad \forall k \in \{1, 2\} \end{aligned} \quad (6)$$

where $i_{\text{hist},k}$ is the history current from the other end, and m is the size of mode groups, as defined in [8]. i_k and v_k are currents and voltages from both ends of line, expressed in time domain. G gives the admittance. N_H and N_Y represent the fitted order of H and Y_C , respectively. $h_{j,i,k}$ and $y_{i,k}$ are state variables derived from the fitted rational functions. Equation (6) can be cast onto an equivalent circuit, as shown in Fig. 3. The coupling between two ends of the transmission line is enabled through the history currents, which represent the currents transmitted from the other end after a certain time delay. This circuit interface is used to represent the transmission lines in MTdc-ac grid.

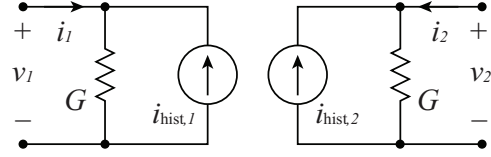


Fig. 3. Time domain transmission line equivalent circuit.

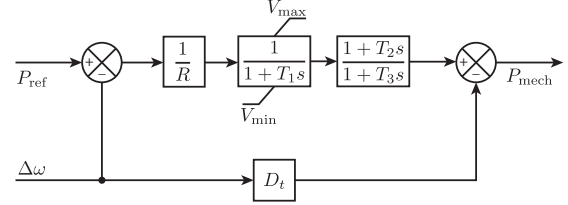


Fig. 4. Generator steam turbine governor model.

C. Generator and Transformer Modelling

The basic dynamics of a symmetrical, three-phase synchronous generator are derived based upon the physical principles within and between different windings using Kirchoff's, Faraday's, Newton's laws, and Park transformation. The detailed derivation can be found in [15], [16] and are not repeated here. The set of differential-algebraic equations (DAEs) describing the dynamics is discretized using a combination of forward Euler and backward Euler based on stiffness in the system, given by

$$(\mathbf{M} - \mathbf{A}h) \cdot \mathbf{x}[k] = \mathbf{M} \cdot \mathbf{x}[k-1] + h\mathbf{B}\mathbf{u}[k] \quad (7)$$

where

$$\begin{aligned} \mathbf{x} &= \{\psi_d, \psi_q, E'_q, \psi_{1d}, E'_d, \psi_{2q}, I_d, I_q\}' \\ \mathbf{u} &= \{V_d, V_q, E_{fd}\}' \end{aligned}$$

The matrices \mathbf{A} , \mathbf{M} , \mathbf{B} are coefficient matrices derived from the generator dynamic equations.

The governor model of the generator is shown in Fig. 4 using the steam turbine governor model (TGOV1) [36], representing the turbine-governor droop and motions of steam valve and reheater in multiple stages.

The transformer is modelled using the classical non-ideal equivalent circuit [16]. Detailed derivation is not repeated here. The transformer employs a wye-delta configuration grounded on wye side. Combined with synchronous generator models and transmission line models, the electromagnetic transient of ac-side system is captured in the proposed real-time platform with respect to generic interconnections.

IV. REAL-TIME IMPLEMENTATION OF MTDC-AC GRIDS

The task of real-time simulation of a cross-continental MTdc-ac grid is partitioned into two that are driven by CPUs and GPUs.

Compared to CPUs, which are good for serial processing with low latency, the GPUs, composed of thousands of cores running at lower frequency, are capable of handling massive amount of threads simultaneously. As a result, GPUs are good for tasks which can be broken into a large number of mutually

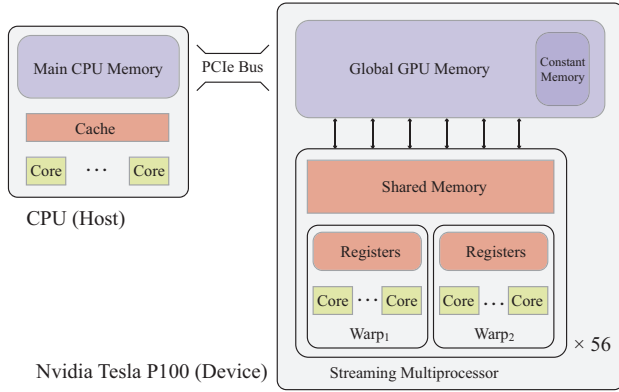


Fig. 5. Architecture of CPU and GPU simulation platform.

independent parts. These parts can be processed at a much higher speed in parallel.

A. The Platform Architecture

The GPU adopted in this paper is Nvidia's Tesla P100, using Pascal GP100 architecture [37]. Equipped with 3584 CUDA cores and 16 Gigabytes of High Bandwidth Memory 2 (HBM2) memory, P100 delivers 5.3 TFLOPS of double precision floating point (FP64) and 10.6 TFLOPS of single precision (FP32) performance.

The GPU device is connected to the host system via PCI-Express (PCIe) bus, as shown in Fig. 5. Data is exchanged between main CPU memory and global GPU memory through this connection. The minimum computation unit on a GPU is a CUDA core, on which a parallel thread is executed. These cores are organized into groups called warps, with a fixed size of 32 cores. The same instruction is executed in parallel by all CUDA cores within the same warp. Above this level of architecture is the streaming multiprocessor. There are 56 streaming multiprocessors in total in a P100. The streaming multiprocessor incorporates 2 warps, or 64 CUDA cores equivalently. There are 64 Kilobytes of shared memory allocated to each streaming multiprocessor. The shared memory is accessible from the CUDA cores within the same streaming multiprocessor, at a much higher speed (less than ten clock cycles) compared to global GPU memory access (a few hundreds of clock cycles). The optimum use of this on-chip shared memory is critical in the optimization of MMC and transmission line parallelism.

B. Implementation of MMC on GPU

The major step of the GPU acceleration design is the partition of simulation algorithm based on its execution logic. It is preferable to move the parallel-friendly part to the GPU, while keep the heavily-serial part in CPU. However, since the CPU and GPU programs have to exchange data within each simulation step, the communication latency should also be considered. Therefore, this partition should be well balanced to take advantage of the computational power of both CPU and GPU and hide the communication latency as much as possible.

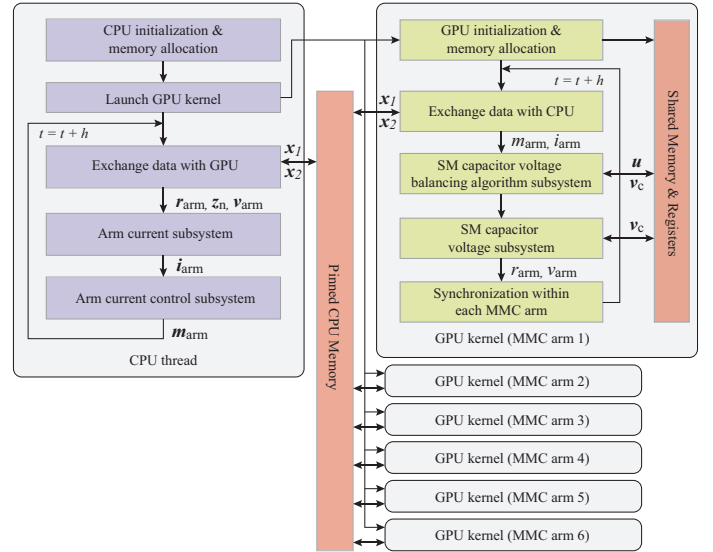


Fig. 6. Summary of MMC hybrid simulation algorithm.

The CPU implementation of MMC simulation and control algorithms have been described in [13], [17]. The simulation and control of MMC within each time step can be divided into four subsystems:

- 1) Arm current control subsystem, which comprises qd ac-side current control and qd circulating current control of the second- and fourth-order harmonics.
- 2) SM capacitor voltage balancing algorithm subsystem, which calculates the gating signals for each IGBT in the MMC.
- 3) SM capacitor voltage subsystem, which solves the dynamics given in (3).
- 4) Arm current subsystem, which computes the currents of each arm based on DAEs provided in (1) and (2).

Among these four subsystems, SM capacitor voltage balancing algorithm and SM capacitor voltage subsystem are suitable to be computed on GPU. This separation is presented in Fig. 6. Assuming the number of SMs in each arm is N (400 in this work), six persistent GPU kernels are launched onto streaming multiprocessors, one for each arm. In each step h , arm current control subsystem and arm current subsystem are computed in series in CPU while the other two subsystems are calculated in parallel on GPU. In each GPU kernel, the gating signals \mathbf{u} (400 integer values) and SM capacitor voltages \mathbf{v}_c (400 double precision values) are stored in shared memory for each arm. Together with other variables that are not as large as \mathbf{u} and \mathbf{v}_c , less than 10 Kilobytes of shared memory is allocated for each kernel, which is well below the limit. The GPU kernels are launched to available streaming multiprocessors and scheduled into warps for execution.

For each MMC arm, the SM capacitor voltage balancing algorithm [17], [38] first calculates the minimum number of SMs which should change their states within this step, either turning ON or turning OFF, given the modulation index of this arm, m_{arm} . One additional SM may change its state if the corresponding SM capacitor voltage exceeds a pre-set

Algorithm 1: Parallel SM capacitor voltage balancing

Input: $m_{\text{arm}}, i_{\text{arm}}, v_c$
Output: u
 Mask out unrelated SMs and convert v_c if needed
 $Warp_{\text{modified}} \leftarrow -1$ /* initialize warp index */
 Compute ΔN_{on} for current step
while $\Delta N_{\text{on}} > 0$ **do**
 if $Warp_{\text{modified}} = -1$ **then** /* first run */
 Find top one within each warp
 Copy top one to shared memory
 else /* only re-calculate the modified warp */
 Find top one within warp of index $Warp_{\text{modified}}$
 Update top one in shared memory
end
 Find top one in shared memory (on thread 0)
 $\Delta N_{\text{on}} \leftarrow \Delta N_{\text{on}} - 1$
end

value. The number of SMs changing their states is denoted as ΔN_{on} . Then, based on the direction of the arm current i_{arm} , the first ΔN_{on} SMs with either the maximum or minimum capacitor voltages are selected and their states are changed and saved into variable u in the shared memory. Majority of the computation in the balancing algorithm is performed to find the top ΔN_{on} SMs with extreme capacitor voltages. Finding minimum voltages can be converted to finding maximum with a sign change. So this problem can be simplified into finding the top one SM with maximum capacitor voltage, and repeating this process for ΔN_{on} times, as shown in Algorithm 1. To find the top one among all N SMs, these N capacitor voltages are partitioned into groups of 32. Each group is scheduled to a warp, within which the top one of the 32 voltages can be calculated efficiently using CUDA warp-level primitives [39]. These top one candidates ($\lceil 400/32 \rceil = 13$ double precision values) are copied to the shared memory by the first thread in each warp. Then, the global maximum can be picked from these candidates with another run of warp-level maximum selection on one of the warps. Instead of simply rerunning the same program multiple times, on subsequent runs other than the very first one, the same operations can be performed only on the warp where the modified SM on previous run is located. Over 50% speed up has been achieved using this optimization technique in a test with $\Delta N_{\text{on}} = 3$.

Another parallel part on GPU is the SM capacitor voltage subsystem, which updates v_c and calculates equivalent arm resistance r_{arm} and arm voltage v_{arm} . The most time consuming step in this subsystem is to compute the summation of all SM voltages within an arm. This can be solved by parallel reduction using the warp-level primitives, based on the same idea implemented on the SM capacitor voltage balancing algorithm.

With the computation of four subsystems distributed onto CPU and GPU, the data generated by each partition should be exchanged by the end of each simulation step. This communication between CPU and GPU is performed using the mapped pinned memory, as shown in Fig. 6, where a pinned allocation of CPU memory is created and two pointers are generated

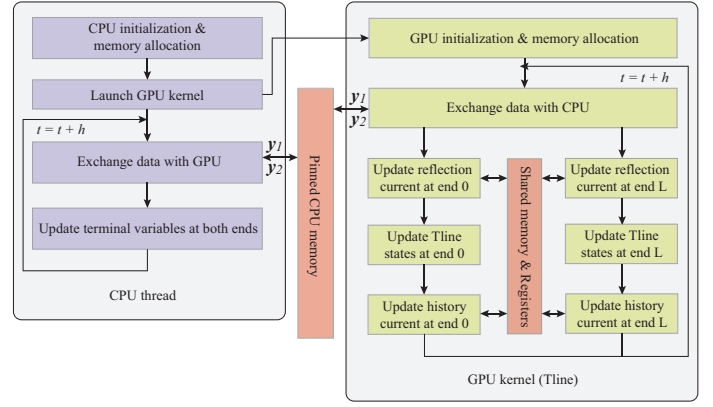


Fig. 7. Summary of transmission line simulation algorithm.

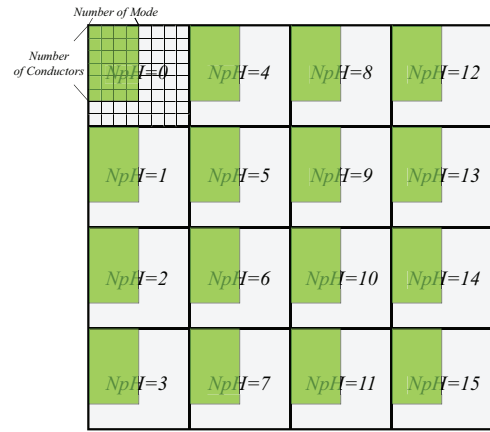


Fig. 8. Distribution of transmission line variables across a thread block.

for host and device separately. On launching the GPU kernel, the device pointer is passed to the GPU, which can be used to exchange data with CPU efficiently. x_1 and x_2 are data sent from GPU and CPU, respectively. x_1 consists of r_{arm} and v_{arm} , x_2 consists of $m_{\text{arm}}, i_{\text{arm}}$, and ac-side voltages v_s . A ping-pong style communication [39] is applied where two additional variables are used to indicate successful completion of read/write on the other side. Once the indicators are verified, data will be exchanged and indicators will be reset. Both CPU and GPU will run in parallel until the next communication is issued.

C. Implementation of Transmission Line on GPU

The simulation of transmission line involves extensive operations of matrices. These matrix-related tasks can be better performed on GPU given its parallelism potential. The simulation algorithm, partitioned on both CPU and GPU, is summarized in Fig. 7. Using the same idea as in MMC simulation, a persistent GPU kernel is launched for each transmission line. The calculation of reflection and history currents is performed on GPU while the terminal variables are updated on CPU. The data exchange is implemented using the same strategy as in MMC simulation.

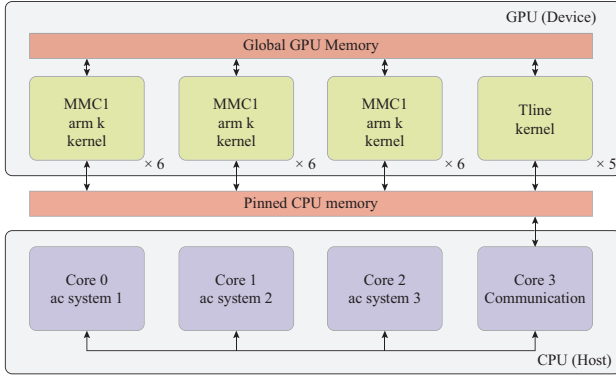


Fig. 9. Summary of the complete simulation platform.

The reflection currents on both ends of the line can be calculated based on (6). However, these currents will be propagated to the other ends after a certain delay, which is determined by the length of the line. Therefore, a buffer is allocated in the shared memory to store the reflection currents and a variable index pointer is used to function as the delay. As described in Section III.B, \mathbf{Y}_c and \mathbf{H} matrices are fitted using time-domain rational functions. Assuming the number of poles for \mathbf{Y}_c and \mathbf{H} are $N_p Y_c$ and $N_p H$, respectively, the poles, residues, and constants for the fittings can be packed into 2- and 3-dimensional matrices [29] and have to be allocated in shared memory. Another sets of variables allocated in shared memory are the states variables used in discretization of time-domain expressions of \mathbf{Y}_c and \mathbf{H} [8], [35]. The size of these shared memory variables are largely dependent on the number of conductors in this transmission line. An estimate of 5464, 9264, and 25752 bytes are to be allocated for lines with two, three, and six conductors, respectively. So the shared memory space in a streaming multiprocessor is large enough for a typical transmission line.

The matrix-related operations in transmission line simulation include matrix-vector multiplication and one- and 2-dimensional slicing of higher dimensional matrices. To take a better advantage of the parallel capability in GPU, the computation of these matrices are distributed to the CUDA cores, as shown in Fig. 8. Each thread block (consisted of 32×32 threads) is evenly divided into 16 parts, which is beyond the typical number of poles obtained in the fitting. The green area indicates the active CUDA cores. Warp-level primitives are easier to be adopted given the proposed data assignment.

D. Complete Setup

As shown in Fig. 1, there are three MMCs, two dc transmission lines, three generators, transformers, and ac transmission lines in the system. The complete simulation platform is depicted in Fig. 9.

On GPU, 18 kernels are launched for each arm in the three MMCs, while 5 kernels are launched for the 5 transmission lines. The kernel-to-kernel communication is performed using the global GPU memory. Data exchange between CPU and GPU is implemented using the pinned CPU memory, as

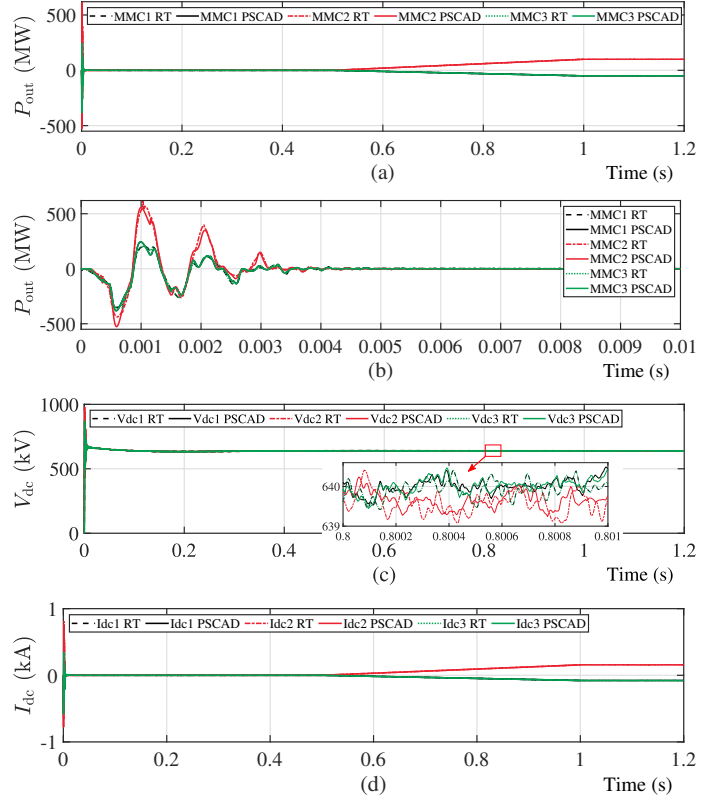


Fig. 10. Comparison of results simulated by real-time platform and PSCAD benchmark: (a) active power P_{out} , (b) zoomed-in view of active power, (c) dc terminal voltages V_{dc} , and (d) dc terminal currents I_{dc} .

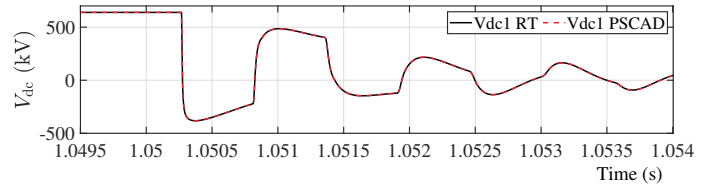


Fig. 11. Comparison of results simulated by real-time platform and PSCAD benchmark under dc fault.

introduced in previous sections, using a dedicated CPU core. The three other CPU cores are used to compute the ac system dynamics including generators and transformers. These four CPU cores are executed in parallel and communicated through Message Passing Interface (MPI).

V. RESULTS AND EVALUATION

The proposed real-time simulation platform has been tested on the demonstration three-terminal dc network shown in Fig. 1. The computed results are compared to the reference system in the PSCAD software under two cases, i.e., normal operation and dc fault propagation, to validate the accuracy and real-time performance of the proposed platform.

A. Comparison with Reference Results

The real-time simulator is first compared to the PSCAD benchmark system under normal operation where an active

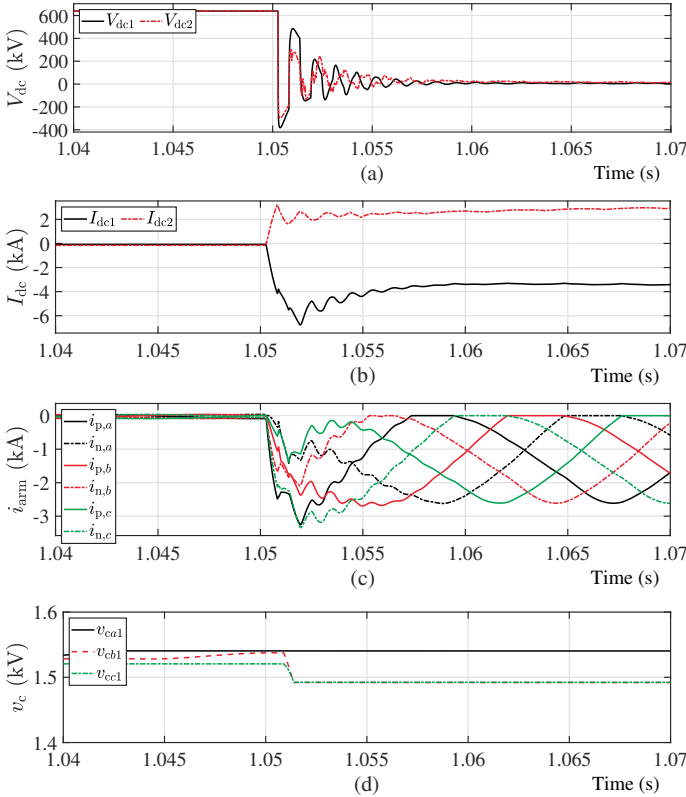


Fig. 12. The measured signals under a pole-to-pole fault located at the MMC1 terminals: (a) dc voltages at MMC1 and MMC2 terminals, (b) dc currents at MMC1 and MMC2 terminals, (c) arm currents of six arms of MMC1, and (d) SM capacitor voltages, one from each phase within MMC1.

power dispatch command of 100 MW is applied to MMC2 and equally distributed to MMC1 and MMC3. The comparison of active power P_{out} , dc terminal voltages V_{dc} , and dc terminal currents I_{dc} are presented in Fig. 10. The simulation error of the proposed real-time platform is less than 1% as compared to the reference results.

The comparison is also conducted under fault scenario where a pole-to-pole dc-side fault is introduced in the middle of dc cable between MMC1 and MMC2. The dc terminal voltage from terminal 1 is shown in Fig. 11. It is verified that the transients and propagation of travelling waves on dc cables are fully captured in the proposed real-time platform.

B. Results on Various Fault Types and Locations

To demonstrate the performance of the proposed simulator, various fault scenarios have been tested, including the pole-to-pole and pole-to-ground faults at MMC terminals. The plots of dc voltages, dc currents, MMC arm currents, and SM capacitor voltages are presented in Figs. 12 and 13 for pole-to-pole and pole-to-ground faults located at the MMC1 terminals, respectively. Given the small simulation error as shown in Figs. 10 and 11, only the results from the proposed simulator are presented. Both faults are triggered at $t = 1.05$ s. Subsequent to the fault occurrence, MMC SMs are blocked [13]. The blocking states of SMs can be observed from the plots of

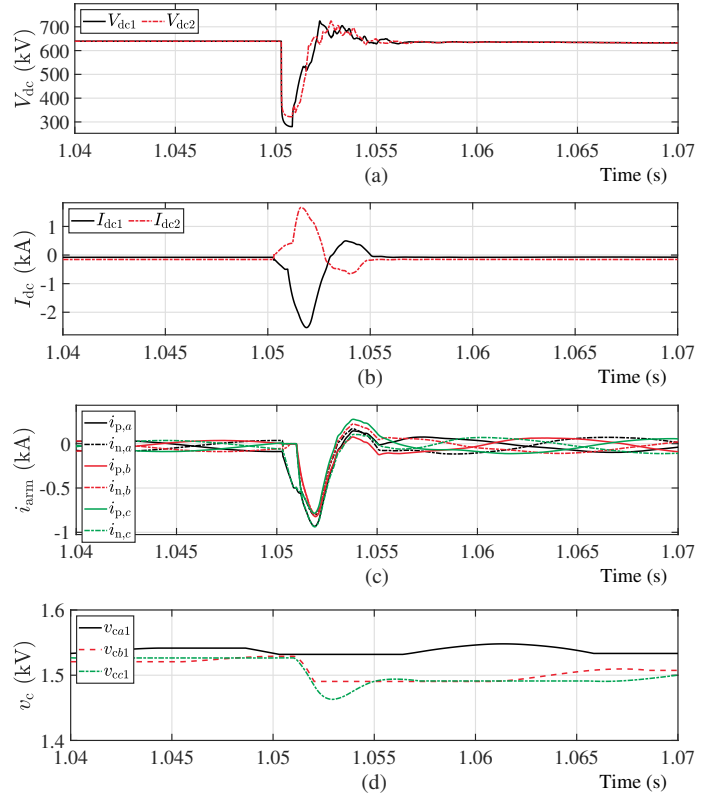


Fig. 13. The measured signals under a pole-to-ground fault located at the MMC1 terminals: (a) dc voltages at MMC1 and MMC2 terminals, (b) dc currents at MMC1 and MMC2 terminals, (c) arm currents of six arms of MMC1, and (d) SM capacitor voltages, one from each phase within MMC1.

TABLE II
PERFORMANCE OF 1 s REAL-TIME SIMULATION

CPU	GPU	Time Spent
IBM POWER8	NVIDIA Tesla P100	0.94 s
Intel Core i5	NVIDIA GeForce GTX 1080 Ti	0.97 s

arm currents and SM capacitor voltages in Figs. 12(c-d), and 13(c-d), respectively.

C. Evaluation of Real-time Performance

The proposed GPU-accelerated real-time simulation platform has been deployed on two hardware setups, one commercial line setup (Tesla P100) and one consumer line setup (GeForce GTX 1080 Ti), as shown in Table II. Both these hardware platforms achieve faster than real-time simulations at scales of 1 s. The result does not show any apparent difference between the performance of commercial and consumer lines.

The GPU-based platform can simulate up to 1,000 SMs/arm per streaming multiprocessor with a time-step of $5 \mu\text{s}$ in real-time, compared to 425 SMs/arm/core using DSP [17] and the best CPU implementation of 230 SMs/arm per CPU core given the same time step [18]–[20]. Compared to the CPU setup, which is hard to be scaled vertically due to the limitation on the number of cores, the GPU-based platform can be scaling horizontally by adding more GPUs in parallel. Additionally,

considering the per unit cost metric defined as the number of SMs simulated per arm divided by the simulation time-step, the cost of the GPU-based implementation is only 1/2 and 1/10 of the CPU- and FPGA-based platform. These advantages greatly benefit the real-time simulation of large-scale MTdc systems.

VI. CONCLUSION

In this paper, a GPU-based cost-effective high-performance real-time EMT simulation platform is proposed for large-scale cross-continental MTdc-ac grids. Fast dynamic transients from both ac and dc networks are captured with 5 μ s time step, using an advanced hybrid-discretized MMC model, frequency-dependent transmission line model, and EMT model of transformers and synchronous generators incorporated in a single platform. This setup is important for high-bandwidth controller design and protection studies. Additionally, the proposed simulation platform achieves better price-performance ratio compared to the existing CPU- or DSP/FPGA-based simulators, and is highly scalable when applied to larger MTdc-ac grids, which is critical in continental-level grid studies. Accuracy and performance of the proposed platform are evaluated and verified with respect to the reference results from the PSCAD/EMTDC software. As the future work, the proposed simulation platform is expected to be extended to a real-time hardware-in-the-loop (HIL) platform for MTdc-ac grids, which will contribute to the design, evaluation, and testing of MTdc-related projects.

ACKNOWLEDGMENT

The authors would like to acknowledge the support from Kerry Cheung, Program Manager at U.S. Department of Energy.

REFERENCES

- [1] N. Chaudhuri, B. Chaudhuri, R. Majumder, and A. Yazdani, *Multi-terminal direct-current grids: Modeling, analysis, and control*. John Wiley & Sons, 2014.
- [2] D. Van Hertem and M. Ghandhari, "Multi-terminal VSC HVDC for the European Supergrid: Obstacles," *Renewable and sustainable energy reviews*, vol. 14, no. 9, pp. 3156–3163, 2010.
- [3] CIGRE WG B4.52, "HVDC Grid Feasibility Study," *CIGRE Technical Brochure 533*, Apr. 2013.
- [4] S. Debnath, M. S. Chinthavali, J. Sun, P. R. V. Marthi, S. Chinthavali, S. M. Lee, M. Elizondo, Y. Markarov, Q. Huang, M. Vellam, H. Kirkham *et al.*, "Models and Methods for Assessing the Value of HVDC and MVDC Technologies in Modern Power Grids," Oak Ridge National Lab.(ORNL), Oak Ridge, TN, United States, Tech. Rep., 2019.
- [5] CIGRE WG B4.38, "Modelling and Simulation Studies to be Performed During the Lifecycle of HVDC Systems," *CIGRE Technical Brochure 563*, Dec. 2013.
- [6] CIGRE WG B4.57, "Guide for the Development of Models for HVDC Converters in a HVDC Grid," *CIGRE Technical Brochure 604*, Dec. 2014.
- [7] S. Debnath and J. Sun, "Fidelity Requirements with Fast Transients from VSC-HVdc," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 6007–6014, Oct. 2018.
- [8] A. Morched, B. Gustavsen, and M. Tartibi, "A Universal Model for Accurate Calculation of Electromagnetic Transients on Overhead Lines and Underground Cables," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1032–1038, Jul. 1999.
- [9] S. Debnath, J. Qin, B. Bahrani, M. Saeedifard, and P. Barbosa, "Operation, Control, and Applications of the Modular Multilevel Converter: A Review," *IEEE Transactions on Power Electronics*, vol. 30, no. 1, pp. 37–53, Jan. 2015.

- [10] C. Dufour, J. Mahseredjian, and J. Belanger, "A Combined State-Space Nodal Method for the Simulation of Power System Transients," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 928–935, Apr. 2011.
- [11] U. N. Gnanarathna, A. M. Gole, and R. P. Jayasinghe, "Efficient Modeling of Modular Multilevel HVDC Converters (MMC) on Electromagnetic Transient Simulation Programs," *IEEE Transactions on Power Delivery*, vol. 26, no. 1, pp. 316–324, Jan. 2011.
- [12] H. Saad, S. Denetiere, J. Mahseredjian, P. Delarue, X. Guillaud, J. Peralta, and S. Nguefeu, "Modular Multilevel Converter Models for Electromagnetic Transients," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1481–1489, Jun. 2014.
- [13] S. Debnath and M. Chinthavali, "Numerical-Stiffness-Based Simulation of Mixed Transmission Systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9215–9224, Dec. 2018.
- [14] D. W. Olive, "Digital Simulation of Synchronous Machine Transients," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, no. 8, pp. 1669–1675, Aug. 1968.
- [15] J. Weber, "Description of machine models GENROU, GENSAL, GENTPF and GENTPJ," *PowerWorld Corporation*, 2015.
- [16] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*, vol. 101. Prentice hall Upper Saddle River, NJ, 1998.
- [17] S. Debnath, "Real-Time Simulation of Modular Multilevel Converters," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 5196–5203, Sep. 2018.
- [18] K. Ou, H. Rao, Z. Cai, H. Guo, X. Lin, L. Guan, T. Maguire, B. Warkentin, and Y. Chen, "MMC-HVDC Simulation and Testing Based on Real-Time Digital Simulator and Physical Control System," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 4, pp. 1109–1116, Dec. 2014.
- [19] H. Saad, T. Ould-Bachir, J. Mahseredjian, C. Dufour, S. Denetiere, and S. Nguefeu, "Real-Time Simulation of MMCs Using CPU and FPGA," *IEEE Transactions on Power Electronics*, vol. 30, no. 1, pp. 259–267, Jan. 2015.
- [20] W. Li and J. Bélanger, "An Equivalent Circuit Method for Modelling and Simulation of Modular Multilevel Converters in Real-Time HIL Test Bench," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2401–2409, Oct. 2016.
- [21] Z. Shen and V. Dinavahi, "Real-Time MPSoC-Based Electrothermal Transient Simulation of Fault Tolerant MMC Topology," *IEEE Transactions on Power Delivery*, vol. 34, no. 1, pp. 260–270, Feb. 2019.
- [22] M. Ashourloo, R. Mirzahosseini, and R. Iravani, "Enhanced Model and Real-Time Simulation Architecture for Modular Multilevel Converter," *IEEE Transactions on Power Delivery*, vol. 33, no. 1, pp. 466–476, Feb. 2018.
- [23] T. Ould-Bachir, H. Saad, S. Denetiere, and J. Mahseredjian, "CPU/FPGA-Based Real-Time Simulation of a Two-Terminal MMC-HVDC System," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 647–655, Apr. 2017.
- [24] S. Denetiere, B. Bruned, H. Saad, and E. Lemieux, "Task Separation for Real-Time Simulation of the CIGRE DC Grid Benchmark," in *2018 Power Systems Computation Conference (PSCC)*, pp. 1–7, Jun. 2018.
- [25] J. Peralta, H. Saad, S. Denetiere, J. Mahseredjian, and S. Nguefeu, "Detailed and Averaged Models for a 401-Level MMC?HVDC System," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1501–1508, Jul. 2012.
- [26] S. Debnath and M. Chinthavali, "MMC-HVDC: Simulation and control system," in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 1–8, Sep. 2016.
- [27] Y. Song, J. Sun, M. Saeedifard, S. Ji, L. Zhu, A. P. S. Meliopoulos, and L. Graber, "Reducing the Fault-Transient Magnitudes in Multiterminal HVdc Grids by Sequential Tripping of Hybrid Circuit Breaker Modules," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7290–7299, 2019.
- [28] J. Sun, M. Saeedifard, and A. P. S. Meliopoulos, "Backup Protection of Multi-Terminal HVDC Grids Based on Quickest Change Detection," *IEEE Transactions on Power Delivery*, vol. 34, no. 1, pp. 177–187, Feb. 2019.
- [29] O. Ramos-Leaños, J. L. Naredo, and J. A. Gutierrez-Robles, "An Advanced Transmission Line and Cable Model in MATLAB for the Simulation of Power-system Transients," in *MATLAB-A Fundamental Tool for Scientific Computing and Engineering Applications-Volume 1*. IntechOpen, 2012.
- [30] S. Yan, Z. Zhou, and V. Dinavahi, "Large-Scale Nonlinear Device-Level Power Electronic Circuit Simulation on Massively Parallel Graphics Processing Architectures," *IEEE Transactions on Power Electronics*, vol. 33, no. 6, pp. 4660–4678, Jun. 2018.

- [31] V. Jalili-Marandi, Z. Zhou, and V. Dinavahi, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1255–1266, Jul. 2012.
- [32] Z. Zhou and V. Dinavahi, "Parallel Massive-Thread Electromagnetic Transient Simulation on GPU," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1045–1053, Jun. 2014.
- [33] Y. Song, Y. Chen, S. Huang, Y. Xu, Z. Yu, and W. Xue, "Efficient GPU-Based Electromagnetic Transient Simulation for Power Systems With Thread-Oriented Transformation and Automatic Code Generation," *IEEE Access*, vol. 6, pp. 25 724–25 736, 2018.
- [34] N. Lin and V. Dinavahi, "Exact Nonlinear Micro-Modeling for Fine-Grained Parallel EMT Simulation of MTDC Grid Interaction with Wind Farm," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2018.
- [35] B. Gustavsen and A. Semlyen, "Rational Approximation of Frequency Domain Responses by Vector Fitting," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999.
- [36] Task Force on Turbine-Governor Modeling, "Dynamic Models for Turbine-Governors in Power System Studies," IEEE Power & Energy Society, Tech. Rep., Aug. 2013.
- [37] NVIDIA, "Nvidia tesla p100 whitepaper," *NVIDIA Corporation*, 2016.
- [38] Q. Tu, Z. Xu, and L. Xu, "Reduced switching-frequency modulation and circulating current suppression for modular multilevel converters," *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 2009–2017, Jul. 2011.
- [39] N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Pearson Education, 2013.