# Parallel Electromagnetic Transient Simulation of Power Electronic Systems on Advanced Digital Hardware

by

Bingrong Shang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Energy Systems

Department of Electrical and Computer Engineering
University of Alberta

# Abstract

Electromagnetic Transient (EMT) simulation is an essential tool for the analysis and design of power system components, such as power electronic converters, transformers, and transmission lines. It enables investigation of the dynamic behavior of power systems and power electronic converters under transient conditions, including circuit faults and interruptions in the system, as well as device stress and thermal behavior of power electronic converters during switching events. This thesis presents two contributions toward the efficient simulation of power electronic systems at the device-level and at the system-level based on the heterogeneous adaptive compute acceleration platform (ACAP) and graphics processing unit (GPU). Both contributions were validated and compared against traditional simulation methods, providing effective and accurate means of simulating power electronic systems, and paving the way for the efficient and reliable design of modern energy systems.

First, the nonlinear high-order electro-thermal model of the Insulated-gate bipolar transistor (IGBT) is developed and then deployed onto the heterogeneous digital hardware for real-time implementation. As the complexity of the nonlinear behavioral model (NBM) of the IGBT poses a significant computational burden on real-time hardware emulation, machine learning (ML) methodology is utilized so that the trained model can reproduce the characteristics of its original counterpart as accurately as possible and then it is implemented on the ACAP, which comprises of the processing system (PS), programmable logic (PL), and Artificial Intelligent Engine (AIE). The vector multiplication feature of the AIE caters to mathematical operations of the ML-based model particularly well and consequently enables it to be executed in real-time with remarkable speedup over the original model with which matrix inversion is otherwise mandatory. Finally, the validation for real-time device-level results and system-level results of a multi-converter system is provided by SaberRD® and MATLAB/Simulink®.

Second, a high-voltage direct current (HVDC) link model based on the modular multilevel converter with embedded energy storage (MMC-EES) is proposed and, utilizing

the massively parallel computing feature of the GPU, its efficacy in compensating a varying wind energy generation is studied. Constant power is oriented in the inverter control by incorporating a DC-DC converter with EES into its submodules. High-fidelity EMT modeling is conducted for insight into converter control and energy management. A fully iterative solution is carried out for the nonlinear model for high accuracy. Since the sequential data processing manner of the central processing unit (CPU) is prone to an extremely long simulation following an increase of component quantity with even one order of magnitude, the massively concurrent threading of the GPU is exploited. The computational challenges posed by the complexity of the MMC circuit are effectively tackled by circuit partitioning which separates nonlinearities. In the meantime, components of an identical attribute are designed as one kernel despite inhomogeneity. The proposed modeling and computing method is applied to a multi-terminal DC system with wind farms, and the accuracy is validated by offline simulation.

# Preface

The material presented in this thesis is based on original work by Bingrong Shang. Both the algorithm development and hardware implementation present in this thesis were conducted in the Real-Time Experimental Laboratory (RTX-Lab) at the University of Alberta. Dr. Ning Lin participated in discussions about concept visualization and algorithm formation while providing constructive suggestions. As detailed in the following, material from some chapters of this thesis has been published or submitted as journal articles under the supervision of Prof. Venkata Dinavahi in concept formation and by providing comments and corrections to the article manuscript.

Chapter 3 includes contents published in the following paper:

- B. Shang, T. Cheng, T. Liang, N. Lin and V. Dinavahi, "Real-time nonlinear behavioral electrothermal device-level emulation of IGBT on heterogeneous adaptive compute acceleration platform," *IEEE Open Journal of the Industrial Electronics Society,* vol. 3, pp. 663-673, Nov. 2022.

Chapter 4 contains contents from the following paper that is currently under review:

- B. Shang, N. Lin and V. Dinavahi, "High-fidelity parallel transient modeling of modular multilevel converter with embedded energy storage for wind farm grid integration," *under review, IEEE Transactions on Energy Conversion,* pp. 1-10, Mar. 2023.

*To my parents,*
*for their endless love and encouragement.*

# Acknowledgements

I would like to express my sincere gratitude to my supervisor *Prof. Venkata Dinavahi* for his guidance, support and encouragement throughout my research. His expertise, insight, and patience have been invaluable to my work.

I want to thank *Dr. Ning Lin*, for his help, advice, and stimulating discussions. I am also grateful to my colleagues in the RTX-Lab: *Tianshi Cheng*, *Weiran Chen*, *Songyang Zhang* and *Chengzhang Lyu*. Their support and friendship have made my graduate studies a truly enriching experience.

Finally, I would like to thank my family and friends for their constant love, encouragement, and support, and for always believing in me.

# Table of Contents

# List of Tables

# List of Figures

# List of Acronyms

| | |
|---|---|
| **ACAP** | Adaptive Compute Acceleration Platform |
| **AIE** | AI Engine |
| **ANN** | Artificial Neural Network |
| **API** | Application Programming Interface |
| **APU** | Application Processing Unit |
| **BRAM** | Block Random-Access Memory |
| **CCM** | Continuous Conduction Mode |
| **CPU** | Central Processing Unit |
| **CUDA** | Compute Unified Device Architecture |
| **DCM** | Discontinuous Conduction Mode |
| **DSP** | Digital Signal Processing (Processor) |
| **EMT** | ElectroMagnetic Transients |
| **FPGA** | Field-Programmable Gate Array |
| **GPU** | Graphic Processing Unit |
| **HBSM** | Half-Bridge Submodule |
| **HVDC** | High Voltage Direct Current |
| **IGBT** | Insulated Gate Bipolar Transistor |
| **LUT** | Look-Up Table |
| **MAE** | Mean Absolute Error |
| **ML** | Machine Learning |
| **MMC** | Modular Multi-Level Converter |
| **MMC-EES** | MMC with Embedded Energy Storage |
| **MOSFET** | Metal Oxide Semiconductor Field Effect Transistor |
| **NBM** | Nonlinear Behavioral Model |
| **NN** | Neural Network |

| | |
|---|---|
| **NoC** | Network on Chip |
| **PCC** | Point of Common Coupling |
| **PS** | Processing System |
| **PL** | Programmable Logic |
| **ReLU** | Rectified Linear Unit |
| **RNN** | Recurrent Neural Network |
| **RPU** | Real-Time Processing Unit |
| **SIMD** | Single Instruction Multiple Data |
| **SIMT** | Single Instruction Multiple Thread |
| **SM** | Submodule |
| **SM-ES** | Submodule with Energy Storage |
| **SoC** | State of Charge |
| **TLM** | Transmission Line Model |
| **VLIW** | Very Long Instruction Word |

# 1 Introduction

Electromagnetic transient (EMT) studies are a type of analysis in power systems used to model and understand the behavior of electromagnetic phenomena that occur during transient events, such as switching operations, faults, and lightning strikes. During these events, high levels of electromagnetic energy are generated and propagated throughout the power system, resulting in voltage and current transients, oscillations, and other disturbances that can lead to component damage, equipment failure, and even power outages. By modeling the electromagnetic behavior of the power system under different conditions, potential problems can be identified and strategies can be developed to mitigate these problems. Therefore, in order to design and operate reliable, efficient and safe power systems, electromagnetic transient simulation has become a powerful tool for studying the dynamic behavior of power systems under various operating conditions.

The simulation of power systems using EMT presents several challenges due to the high level of detail and complexity required to accurately model the behavior of power system components. One of the main challenges arises from the nonlinear behavior exhibited by many components, such as insulated gate bipolar transistors (IGBTs) and diodes. EMT simulations must solve large systems of nonlinear differential equations, which can be computationally intensive and time-consuming. To achieve accurate modeling of these components, more complex mathematical models are needed, which further increase the computational complexity of the simulation. Moreover, the traditional Newton-Raphson method used for solving nonlinear problems involves repetitive computations for com-

bining and decomposing large matrix systems. This approach requires solving a set of high-order matrix equations at each iteration, which can be particularly computationally intensive, especially for large-scale systems like Modular Multilevel Converters (MMC). Another significant challenge in EMT simulation arises from the complex network topologies of modern power systems, such as multi-terminal high-voltage direct current (HVDC) systems. To accurately model these systems and their corresponding control systems, advanced modeling techniques and software tools are necessary. Therefore, EMT simulations require substantial computational resources and specialized software to provide accurate results.

This chapter provides an introduction to EMT simulation, highlighting its significance in the field of power systems, as well as the challenges it faces. To address these challenges, research directions are identified for the development of EMT simulation at both the device level and system level, along with corresponding model design methods and simulation acceleration platforms. These efforts aim to solve the challenges associated with nonlinear behavior, large-scale matrix computations, and complex network topologies. Finally, the structure and components of the thesis are presented to give an overview of the subsequent chapters.

## 1.1 IGBT Device-Level Modeling and Simulation

The focus of device-level electromagnetic transients is to analyze the behavior of electromagnetic phenomena that occur within individual electronic devices, such as voltage spikes, electromagnetic interference, or thermal effects. Among the many power electronic devices, IGBT is a power semiconductor device that is widely used in power systems due to its high power handling capability, fast switching speed and low on-resistance.

For example, in motor drives and renewable energy systems, the high power handling capability of IGBTs makes them ideal for high-power applications [1]. In addition, IGBTs have a low on-resistance, which means when they are operated at high current levels, power losses can be reduced, thereby increasing the efficiency of the power system. In applications that require fast switching, such as motor control and power converters, IGBTs can be turned on and off very quickly, allowing efficient and precise control of the power flow in the system. IGBTs are also known for their high reliability and long operating life, making them a popular choice for critical applications such as high-speed train traction inverters [2] and HVDC transmission systems [3].

Various device-level IGBT models have been developed and widely used, such as the ideal model [4], the associated discrete circuit model [5], and the Thévenin equivalent model [6]; previous studies have widely used these models to represent the switching characteristics of IGBT. These models are simplistic and computationally efficient, however, they are not sufficient to describe realistic IGBT switching transients. Dynamic models based on the transient behavior of IGBT are necessary for high-accuracy device-level simulations. There are several different dynamic models, compact model [7] and EMT model, while EMT models are usually divided into physical-based models and behavioral models.

Physical-based models are based on physical principles and nonlinear equations, which account for the physical phenomena occurring inside the IGBT, such as carrier generation, recombination, drift, and diffusion [8]. Although these models are highly accurate, they require significant computational resources and time to simulate due to their computational intensity. Numerical solution methods usually involve finite element or finite difference methods [9]. For the behavioral models based on empirical data and simplified equations that capture the essential behavior of the IGBT during transient events, their faster and more efficient simulation makes them a popular choice for power system simulations that involve large numbers of IGBTs [10].

To improve the efficiency of EMT simulation, several approaches can be taken, such as simplifying the models used in the simulation and employing computational technology and hardware acceleration measures.

One method of simplifying the model is to reduce the number of elements or use more approximate models, which can reduce the computational requirements and improve simulation performance. Machine learning (ML) algorithms such as offline training of artificial neural network (ANN) and recurrent neural network (RNN), can be used to develop alternative models that approximate the behavior of complex power system components. ANNs have been successfully employed in power electronic systems for system fault diagnosis [11], converter and controller modeling [12, 13], and predictive control for MMCs emulation [14]. The required training data can be collected from an actual power electronics device or system, or from a simulation model of the device or system. After training and validation, the ML model will provide the desired output given the appropriate inputs during the simulation. Since ANN models involve only basic algebraic operations, a well-trained ANN is able to optimize the simulation process to a large extent.

In addition to model simplification, measures in computational technology and hard-

ware acceleration can also be employed. Parallel computing techniques can help distribute the workload across multiple processors or computers, reducing the simulation time and increasing the number of simulations performed. Another approach is to use hardware acceleration techniques such as Field-Programmable Gate Arrays (FPGAs) [15] to accelerate the simulation process, which can provide significant performance improvements over traditional CPU-based simulations. Compared to traditional FPGAs, Xilinx® Adaptive Compute Acceleration Platform (ACAP) has several notable advantages, for example, ACAP provides higher computational performance and energy efficiency due to its heterogeneous architecture that combines the processing system (PS), programmable logic (PL), and innovative AI Engine (AIE). Additionally, the availability of various optimized processors and interfaces, such as the Single Instruction Multiple Data (SIMD) vector unit and GMIO, helps to speed up data processing efficiency and maximize processing throughput.

This thesis employs a combination of these approaches to address the computational complexity and resource requirements in IGBT EMT device-level simulations.

## 1.2 MMC-EES System-Level Modeling and Simulation

Modular Multilevel Converter (MMC) is a power electronic converter topology consisting of multiple submodules (SMs), each containing a set of power electronic switches. The structure of a common submodule can usually be divided into a half-bridge submodule (HB-SM) consisting of two switches connected in series, or a submodule (FB-SM) consisting of four switches arranged in a full-bridge structure. Compared to other converters such as Voltage Source Converter (VSC), MMC has become a popular converter topology for high-power and medium/high-voltage applications in the field of power systems [16], The outstanding features of MMC include its modularity and scalability, making it suitable for any voltage level, the superior harmonic performance due to stacking a large number of identical low-voltage-level SMs, and high power density, which is important for high-voltage applications. These features make it suitable for applications such as HVDC transmission systems [17], flexible AC transmission systems (FACTS) [18], and renewable energy integration.

As power systems continue to transition towards more renewable energy sources to reduce greenhouse gas emissions and combat climate change, MMC is playing an increasingly important role in enabling efficient and reliable power delivery. Some of the commonly used renewable energy sources in the MMC power system include wind farm

power [19], solar photovoltaic (PV) [20], and hydroelectric power [21]. Wind power is one of the most widely used renewable energy sources and can be harnessed through the use of wind turbines to convert wind energy into electricity. In areas with high wind speeds, such as offshore wind farms, wind energy has a huge potential for deployment.

The integration of renewable energy sources, such as wind power, into power grids presents several challenges, one of which is the intermittent nature of wind energy. Since wind energy is highly dependent on weather conditions and highly susceptible to changes in wind speed and direction, power output is prone to fluctuations. In this case, energy storage systems (ESS) can help mitigate the variability and uncertainty associated with wind power as a valuable solution. MMC with energy storage system (MMC-ESS) has been researched extensively as a means of integrating wind power into power grids, which can help to smooth out these fluctuations and provide grid stability by storing excess wind energy during periods of high output and releasing it during periods of low output or high demand. Examples of MMC-ESS applied to HVDC wind farm systems can be found in [22] and [23].

Simulation of MMC poses challenges due to its complex topology and control scheme, requiring significant computational resources for the simulation of large-scale MMC systems with multiple submodules. In this thesis, Graphics Processing Unit (GPU) is utilized as a hardware platform for simulating MMC-EES. Compared with the CPU, GPU has much higher parallel processing capabilities, which enables it to perform thousands of computations simultaneously, and a larger number of cores to handle more data in parallel, resulting in improved performance. GPU is also more cost-effective than FPGA due to the fact that it is more flexible and can be easily programmed or modified, making it a more readily available option. For example, an accurate electromagnetic-thermal simulation of the multi-terminal DC (MTDC) grid is proposed on the architecture of the GPU in [24], and a variable time-stepping MMC model was presented in [25] to accelerate the parallel EMT simulation of a detailed MTDC grid on the GPU.

## 1.3 Motivation and Objectives of This Thesis

The field of power electronics is constantly evolving, driven by the growing demand for high-performance, efficient, and reliable power electronic systems. In this context, the accurate modeling and simulation of power electronic devices and systems is essential for the design, analysis, and optimization of modern power electronics applications. How-

ever, existing simulation tools often suffer from limitations in terms of accuracy, speed, and flexibility, especially for nonlinear and time-varying systems. Therefore, there is a need for advanced simulation techniques that can address these challenges and enable more efficient and effective development of power electronics applications.

The motivation of this thesis is to develop advanced simulation techniques for power electronics, with a focus on real-time, nonlinear behavioral electro-thermal device-level emulation and high-fidelity parallel transient modeling of modular multilevel converters with embedded energy storage.

The main objectives of this thesis are to:

- Propose a novel heterogeneous adaptive compute acceleration platform (ACAP) for real-time, high-accuracy, and flexible device-level emulation of insulated gate bipolar transistors (IGBTs) under nonlinear and thermal effects. Provide a detailed description of the proposed platform, including the hardware and software components and the algorithms used for the IGBT emulation. Present the performance results of the platform and compare them with other simulation tools.

- Develop a high-fidelity parallel transient model for MMC with embedded energy storage, which can accurately capture the dynamic behavior of MMC systems under various operating conditions. Provide a detailed description of the proposed model, including the mathematical formulations, control strategies, and implementation on the parallel computing platform GPU. Present the validation results of the model and demonstrate the effectiveness and efficiency through the multi-terminal wind farm system simulation.

## 1.4 Thesis Outline

This thesis contains 5 chapters. The rest of the chapters are outlined as follows:

- **Chapter 2** - Introduces the background of the ACAP and GPU hardware architecture used in this thesis and the programming languages of AI Engine and CUDA.

- **Chapter 3** - The IGBT device-level nonlinear behavior electrothermal model is introduced. Then, the implementation and performance of the IGBT NBM in the PS, PL, and AIE domains of the Versal$^{TM}$ ACAP are presented. Machine learning models, training methods, and vectorization implementations are also specified. Finally, the verification and hardware simulation results of the ML model are shown.

- **Chapter 4** - This chapter first introduces the topology and control strategy of MMC with embedded energy storage. After that, the EMT model of MMC-EES is presented and the design of GPU parallelism is provided. The results of the implementation of the four-terminal HVDC system are also given to verify the accuracy of the simulation and the effectiveness of the embedded energy storage system in the MMC.
- **Chapter 5** - The contributions and some future work of the thesis are given in this chapter.

# 2

# Background on Adaptive Compute Acceleration Platform and The GPU

The increasing demand for artificial intelligence, machine learning, and data-intensive applications has driven the development of powerful computing hardware such as Xilinx® ACAPs and NVIDIA® GPUs. These platforms have become essential tools for implementing high-performance computing tasks. This chapter provides a detailed overview of the hardware platform architecture of the Xilinx® Versal™ ACAP VCK190 and NVIDIA® GPU V100, as well as the programming features of the AI Engine and CUDA C++. The characteristics of these two hardware platforms are compared in terms of implementing power system simulation, which is the focus of this thesis.

## 2.1 Heterogeneous Adaptive Compute Acceleration Platform

### 2.1.1 Xilinx® Versal™ VCK190 Hardware Architecture

Versal™ devices are the first ACAPs based on the TSMC 7 nm FinFET process technology developed by Xilinx®. Fig. 2.1 (a) depicts the architecture of ACAP, which consists of a scalar engine (PS), an adaptable engine (PL), and an intelligent engine, all of which are connected together via a series of high-speed and integrated horizontal and vertical paths NoC to achieve remarkable performance and meet design timing, speed, and logic utilization requirements. The ACAP can be divided into three specific components: AI Engine (AIE), Processing System (PS) and Programmable Logic (PL), as described below.

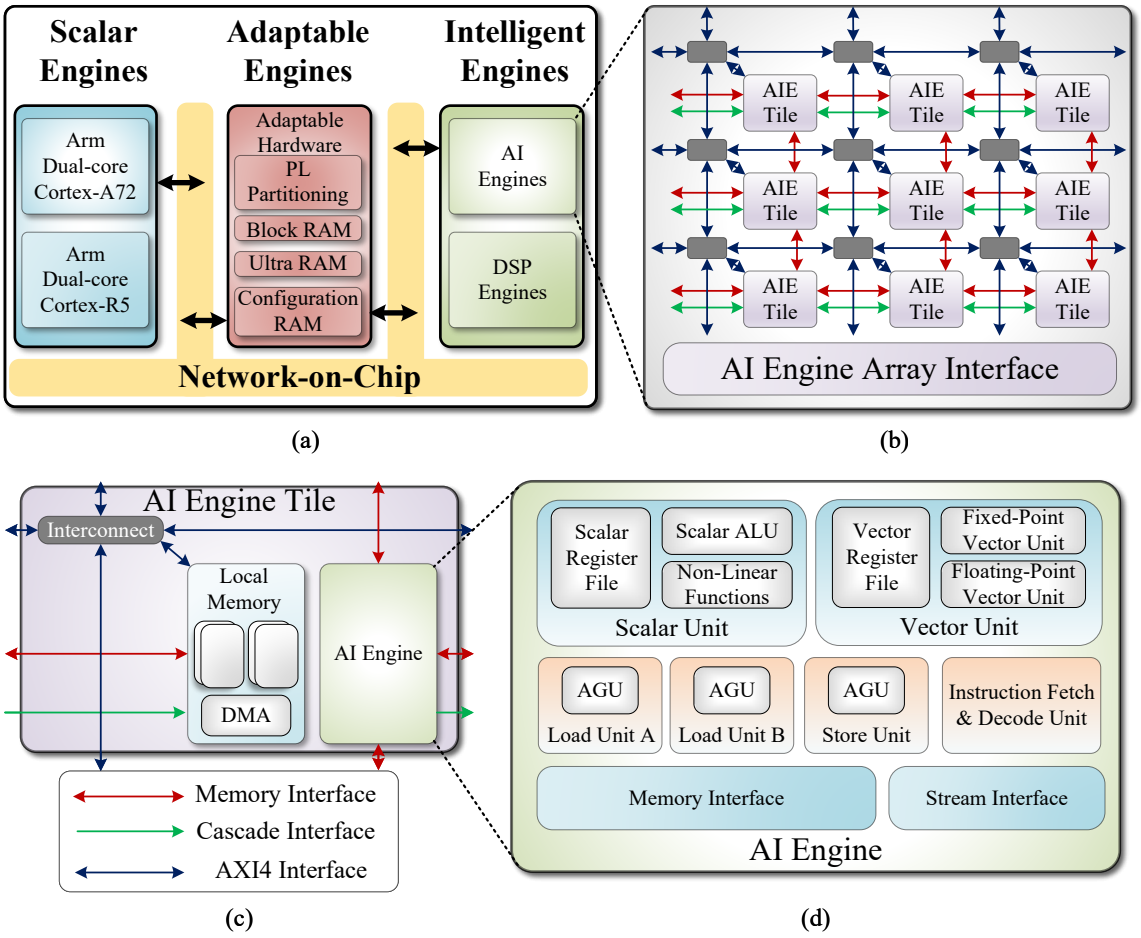



Figure 2.1: (a) Architecture of ACAP; (b) AI Engine array; (c) AI Engine tile; (d) AI Engine architecture.

- **AI Engine (AIE)**: As a key feature of the Xilinx® ACAP and a highly flexible and efficient programmable engine, AIE provides the ability to implement customized, high-performance machine learning and signal processing applications. As shown in Fig. 2.1 (b), the AIE array is the top-level hierarchy of the AIE architecture, which integrates a two-dimensional array of AIE tiles. The AIE array interface enables the AIE to communicate with the rest of the Versal™ device through the NoC or directly to the PL. The AIE tile architecture is shown in Fig. 2.1 (c), where each tile includes one tile interconnect module which handles AXI4 input/output, a memory module, and an engine, which can access up to 4 memory modules in four directions. The AIE, shown in Fig. 2.1 (d), is a highly-optimized processor that supports both fixed-point and floating-point precision and is organized as an array of AIE tiles, which can contain up to 400 tiles on the VC1902 device used in this work. This architecture

enables highly parallel, pipelined, and streaming computation, making it ideal for data-intensive applications.

- **Processing System (PS)**: The ACAP features a versatile processing system that includes both an Arm processing unit (APU) and a real-time processing unit (RPU). As shown in the scalar engine part of Fig. 2.1 (a), the APU is based on the ARM Cortex-A72 processor core to provide general-purpose computing in a standard programming environment [26], which offers higher capabilities and a high clock frequency of up to 1700MHz. The RPU based on a custom microarchitecture ARM Cortex-R5 processor, on the other hand, is a real-time processing unit that is optimized for deterministic, low-latency processing, which is designed to handle time-critical tasks. The OpenCL and the Xilinx® Runtime (XRT) methodology are adopted for software programming, which enables multiple kernels to be executed concurrently with initialized command queue and thus is highly efficient in performance.

- **Programmable Logic (PL)**: PL is an extensible structure that enables the creation of a wide range of conceivable functions. It consists of DSP engines, configurable logic blocks, Configuration RAM, and Block RAM (BRAM), which can be configured together to create numerous types of hardware functionalities including accelerators, processors, functional pipeline units, and peripherals [26]. As shown in the left part of Fig. 3.3, PL establishes connections between PS, NoC, AIE, high-density I/O buffers, and components instantiated within the PL. The GMIO port can be used to connect external memory mapped to or from the global memory, which accesses DDR memory directly with a bandwidth throughput of 3200 MB/s. The connections and configuration of the PL elements are captured in the Vivado® design suite and the Vitis® unified software platform toolchain using a programmable device image.

### 2.1.2 AI Engine Programming

The Xilinx® AI Engine is a highly optimized hardware architecture. One of its key programming features is the ability to define custom data flow graphs that map to the computation engines in the array. The programming paradigm for AI Engine based on the dataflow model only requires specifying data dependencies between computational blocks rather than explicitly defining the order of operations. Customized and optimized data flow graphs are typically created using the Xilinx® Vitis integrated design environment (IDE) and then automatically compiled and optimized for the AI Engine hardware such

as the VCK190. The Vitis IDE provides a unified platform for hardware and software development and includes a set of tools and libraries for creating, debugging, and deploying applications.

The AI Engine kernel is a basic building block for AI Engine programming. It is a function that implements a specific operation or calculation using the AI Engine architecture. AI Engine programming involves writing and optimizing kernels to perform specific tasks, and then orchestrating these kernels to form a complete application. The programming model is based on a dataflow graph, where kernels are connected to form the whole graph, and data flows through the graph to perform computations. The AI Engine kernel is a C/C++ program written in native C/C++ language with specialized intrinsic functions [27] for the Very Long Instruction Word (VLIW) scalar and vector processors. AI Engine kernel code is compiled using the AI Engine compiler (aiecompiler) included in the Vitis IDE. The aiecompiler compiles the kernel to generate the ELF files that run on the AI Engine processor [28].

One of the key features of the AI Engine kernel is its ability to execute a large number of independent, pipelined compute operations in parallel, which makes it particularly well-suited for accelerating complex machine learning and signal processing algorithms. The kernel supports a range of data types and precision levels, including floating-point and fixed-point arithmetic. AIE kernel programming is divided into two categories, scalars and vectors. Scalar programming operates on individual data elements, while vector programming operates on data in groups or vectors. Scalar programming is used when the data is not aligned or the operations are conditional or iterative. Vector programming, on the other hand, is used when the data is aligned and can be processed in parallel using vector instructions. By balancing the use of scalar and vector programming, the performance and efficiency of AI Engine applications can be optimized.

Additionally, the AI Engine kernel is highly versatile and supports various memory access patterns, including streaming, block, and gather/scatter access, which enables it to efficiently access data from a variety of sources. In terms of data communication in AI Engine, window mode represents a fixed-size block of data that is processed together, while stream mode represents a continuous stream of data that is processed in a sequential and non-blocking manner. The efficiency of window and stream processing modes can be compared based on factors such as latency and throughput. In window mode, data is processed in fixed-size chunks or windows, which can introduce a processing delay. Stream mode, on the other hand, processes data continuously, resulting in lower latency.

Throughput refers to the amount of data that can be processed in a given time frame, which is also a factor to be considered, with stream mode being able to support higher throughput due to processing data continuously.

Overall, AI Engine is a powerful and flexible compute engine that offers significant performance and power efficiency benefits for a wide range of data-intensive applications, the programming model for AI Engine is designed to provide a highly flexible and efficient platform, with a particular focus on performance, scalability, and ease of use.

## 2.2 High-Performance GPU

### 2.2.1 NVIDIA® Tesla V100 Hardware Architecture

The NVIDIA® Tesla V100 [29] is a high-performance computing (HPC) hardware platform as shown in Fig. 2.2.

The NVIDIA® V100 GPU consists of 84 Streaming Multiprocessors (SMs), which are responsible for executing the actual computational workloads. Each SM in the V100 is composed of 4 processing blocks, which are essentially smaller processing units that work together to execute instructions in parallel. Each processing block in the V100 consists of 16 FP32 cores, 16 INT32 cores, 8 FP64 cores and 2 Tensor cores. In addition to the processing blocks, each SM also includes shared memory, which can be accessed by all of the processing blocks within the SM. Global memory is used to store data that is shared across all of the SMs in the GPU.

The main hardware features of V100 include:

- Volta architecture: The V100 is built on NVIDIA's Volta architecture, which includes innovations such as Tensor Cores, and enhancements to the CUDA programming model for improved performance and ease of use.

- Large memory capacity: The V100 features a high-bandwidth memory (HBM2) capacity of up to 16GB, which delivers high performance for large-scale computing workloads.

- High-speed interconnect: The NVIDIA® NVLink™ interconnect enables quick and efficient communication between GPUs and CPUs, delivering up to 300 GB/s of bi-directional bandwidth.

- High memory bandwidth: The V100 is equipped with 900 GB/s of memory bandwidth, making it capable of handling data-intensive applications quickly and effi-

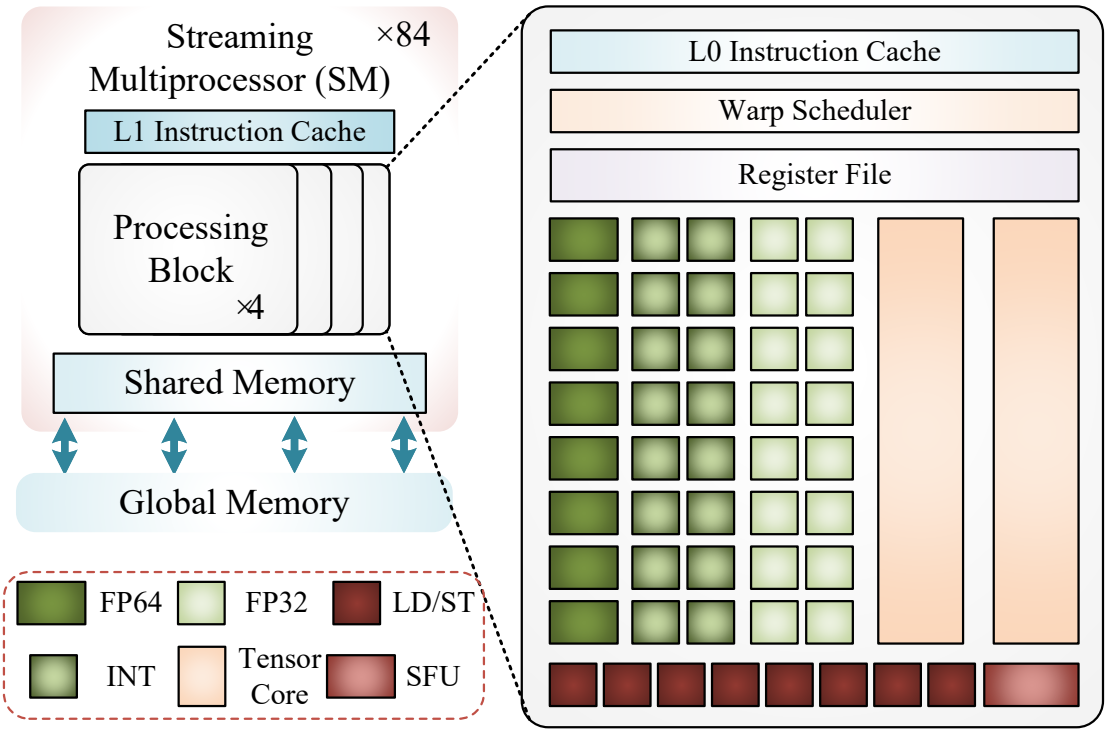



Figure 2.2: NVIDIA® Tesla V100 GPU streaming multiprocessor architecture

ciently.

### 2.2.2 CUDA Programming

CUDA programming is a high-level programming language that provides massive parallelism and efficient memory management, making it a popular choice for harnessing the computational power of GPU. CUDA C++ allows for the parallelization of code written in C++ syntax, which facilitates the leveraging of existing C++ code and libraries.

One of the key features of CUDA C++ is SIMT (Single Instruction Multi-Threading). Under this mode, multiple threads are grouped into blocks and scheduled to execute the same instruction simultaneously, with each thread operating on a different set of data. This allows the GPU to process a large number of power system simulation tasks in parallel, resulting in faster and more efficient simulations.

The memory hierarchy in the GPU is designed to optimize the performance of each memory type based on its characteristics. Global memory is accessible to all threads in the GPU, but has higher latency and is more suitable for tasks that require large amounts of data. Shared memory, on the other hand, is low-latency and high-bandwidth, and is only be accessed by threads within the same thread block. Shared memory is used to optimize

access to frequently accessed data within a block, while global memory is used to store large arrays or matrices of data.

In CUDA C++, kernel functions are the code that can be executed in parallel by the GPU's threads. They are customized to perform computations on specific data and can optimize performance by taking advantage of the GPU's parallel processing capabilities. Kernel functions are written in C++ and can take advantage of specialized syntax and functionality specific to CUDA. When a kernel is launched on the GPU, threads are organized into blocks and grids, with blocks being groups of threads that can cooperate through shared memory and grids being groups of blocks that can execute independently.

There are two types of kernel launches in CUDA: global and device. Global kernels can be invoked from the host and run on the device, while device kernel launches are invoked from within another kernel on the same device. Global kernels can access both global memory and shared memory, while device kernels can only access global memory. Global memory is shared by all threads in a grid, while shared memory is local to each block of threads. By specifying the size and layout of the grid and blocks of threads, the host CPU can control how the kernel is executed on the device, allowing for fine-grained control over the execution of the kernel.

For power system simulations that involve a large number of homogeneous modules, the parallel processing capabilities of GPUs can provide significant acceleration compared to traditional simulation tools. This is because power system simulation tasks can be efficiently executed on the massively parallel architecture of GPUs, thus benefiting greatly from the SIMT feature. Alternatively, the memory hierarchy of the GPU can be leveraged to optimize the performance of the simulation. The use of shared memory is particularly effective in power system simulations, where there are often large amounts of data that are frequently accessed by multiple threads such as system parameters or temporary results. By storing this data in shared memory, it can be accessed more efficiently by the threads in a block, reducing the overall simulation time. In contrast, large data arrays or input data and output data are stored in global memory.

## 2.3 Summary

Xilinx® ACAP and NVIDIA® GPU are two powerful hardware platforms that offer unique programming features for accelerating power electronics simulations. The ACAP AI Engine provides a scalable and versatile approach to simulation acceleration, with the sig-

nificantly extended functionality of programmable logic and an easier and more efficient programming flow compared to traditional and complex FPGA hardware flows. On the other hand, the CUDA C++ programming model used by GPUs offers a way to exploit massive parallelism, making it possible to accelerate complex simulations by several orders of magnitude compared to CPU-based solutions. With its high parallel processing power, the GPU can handle large-scale power system simulations that involve homogeneous modules more efficiently than traditional simulation tools. By leveraging the parallel processing power of GPUs, simulations can be performed in a more efficient and timely manner, which is crucial for the development of modern power systems.

# 3

# Real-Time Nonlinear Behavioral Electrothermal Device-Level Emulation of IGBT on Heterogeneous ACAP

## 3.1 Introduction

Power electronic converters have been playing a significant role in power supply systems in many domains, such as rail transportation [30], electric vehicles [31], and ship power systems [32]. The Insulated-gate bipolar transistor (IGBT) is now one of the most important and extensively used power semiconductor switches in the aforementioned applications for its advantages and characteristics, such as large capacity, simple driving, easy protection, and high switching frequency. There is a growing volume of literature that establishes the system-level simulation of these converter-based systems for their design and performance evaluations [33–35], where most of them are based on detailed modeling or average value modeling, which suffices for the testing and verification of system-level converter functions such as frequency regulation and voltage adjustment. When an in-depth study is required for a comprehensive electro-thermal transient analysis, the device-level modeling is compulsory [36], as it reveals the transient performance of the power semiconductor switch, so that the transient voltage, current, and thermal stresses can be monitored accurately for real converter design evaluation [37].

Various device-level IGBT models have been developed and widely used in the past for power converter simulation [38, 39], such as the analytical model, and the nonlinear

behavioral model (NBM). However, the modeling complexity due to the inclusion of device transients poses a significant challenge accompanied by a high chance of numerical divergence. This often results in a short simulation duration that is even insufficient for the system to reach its steady state, especially in commercial simulation tools such as PSpice®, Multisim™, and SaberRD®. Therefore, hardware acceleration using FPGA has been adopted for medium-scale power converters where a dramatic speedup over CPU was attained [40, 41]. In addition, [42] implements the device-level simulation of the IGBT model using the parallel algorithm on GPU, which also significantly improves the simulation efficiency. Real-time simulation [43] is playing an increasingly vital role in the development and testing stages of power electronics and requires the model to be updated strictly within the corresponding simulation time-step, but the nonlinear property of the device model determines that real-time execution can hardly be met due to a Newton-based iterative solution of a high-order matrix equation. As a result, both hardware acceleration and algorithm optimization are necessary to achieve that goal.

Machine learning (ML) has begun to be employed in power systems and power converters to reduce the computational burden of conventional models [44, 45], and various neural networks (NNs) including gate recurrent unit (GRU) [46] and recurrent neural networks (RNN) [47] are utilized to train models and obtain accurate results and improve the simulation efficiency. As a novel and time-saving approach, ML can also be applied to the study of circuit transients by learning a specific dataset and configuring the NN to create the design-compliant models [48]. However, this approach has yet to be explored for power electronics device simulations. In this chapter, the ML methodology is adopted for avoiding high-dimensional matrix equations that are challenging to solve by traditional methods.

Compared to the conventional FPGA, the Versal™ ACAP from Xilinx® has an innovative design in terms of hardware architecture, which combines Adaptable Engines, Scalar Engines, Intelligent Engines, and Network on Chip (NoC) to provide powerful heterogeneous acceleration for a wide range of applications [49]. As the most critical and innovative part of ACAP, the AI Engine (AIE) is a highly optimized processor with many features, such as the Single Instruction Multiple Data (SIMD) vector unit, and Very Long Instruction Word (VLIW) function that can be used in the field of real-time emulation to solve the data-intensive computing issues.

In this chapter, the IGBT electro-thermal NBM has been implemented and evaluated on the Versal™ ACAP's processing system (PS), programmable logic (PL), and AIE, sep-

arately. The ML-based model is proposed to accommodate the SIMD vector processing feature of the ACAP, specifically, the adoption of the NN enables faster matrix calculations to replace the complex iterative matrix inversion in the transient simulation process. The ML model is realized through learning from the dataset of IGBT NBM, and the AIE SIMD vector unit provides intrinsics functions to make the model emulation more efficient before being implemented on the ACAP. Finally, the simulation results of a multi-converter system are verified by MATLAB/Simulink®.

This chapter is organized as follows: Section 3.2 introduces the IGBT device-level nonlinear behavioral electro-thermal model. In Section 3.3, the Versal™ ACAP architecture including PS, PL, and AIE is introduced, and the implementation and performances of the NBM in these three domains are also presented. The machine learning model, training methodology, and vectorized implementation are described in Section 3.4. Section 3.5 shows the validation of the ML model and hardware simulation results, and Section 3.6 provides the conclusion.

## 3.2 Nonlinear Behavioral Electro-Thermal Device-Level Modeling of IGBT

### 3.2.1 IGBT Nonlinear Behavioral Model

The nonlinear behavioral model [50] of an IGBT with its inherent anti-parallel diode is shown in Fig. 3.1. According to the definition,

$$i(t) = C\frac{dv(t)}{dt}, \tag{3.1}$$

a capacitor can be discretized by Backward Euler as:

$$\int_{t-\Delta t}^{t} i(t)\,dt = C[v(t) - v(t-\Delta t)], \tag{3.2}$$

$$\begin{aligned} i(t) &= \frac{C}{\Delta t}v(t) - \frac{C}{\Delta t}v(t-\Delta t) \\ &= \frac{C}{\Delta t}v(t) + I_{c_{eq}}, \end{aligned} \tag{3.3}$$

where $\Delta t$ is the time-step.

The equivalent conductance is defined as:

$$G_{C_{eq}} = \frac{C}{\Delta t}, \tag{3.4}$$

and the equivalent current source:

$$I_{c_{eq}} = -\frac{C}{\Delta t}v(t-\Delta t). \tag{3.5}$$

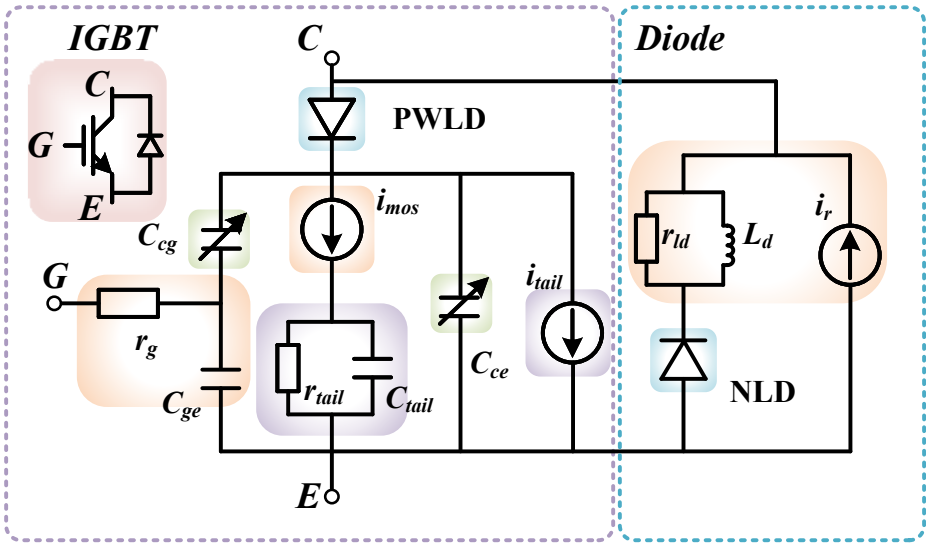



Figure 3.1: High-order nonlinear IGBT equivalent circuit.

Consequently, for capacitor $C_{ge}$, the conductance $G_{C_{ge}}$ and current source $i_{C_{geeq}}$ are given as:

$$G_{C_{ge}} = \frac{C_{ge}}{\Delta t}, \tag{3.6}$$

$$i_{C_{geeq}} = -G_{C_{ge}} \cdot v_{C_{ge}}(t - \Delta t). \tag{3.7}$$

The discretized forms of nonlinear capacitors $C_{cg}$ and $C_{ce}$ are identical, for example:

$$C_{cg} = \begin{cases} (C_{cgo} \cdot (1 + \frac{v_{C_{cg}}}{v_{C_{go}}})^{-m}), v_{C_{cg}} > 0 \\ C_{cgo}, \qquad\qquad v_{C_{cg}} \leq 0. \end{cases} \tag{3.8}$$

where $m$ is the Miller capacitance exponent coefficient, which is set to 0.5 by default, and $C_{cgo}$ is the fixed capacitance, given in Appendix.

Similar to $C_{ge}$, the conductance could be calculated as $G_{C_{cg}} = \frac{C_{cg}}{\Delta t}$, and the equivalent current source as:

$$i_{C_{cgeq}} = \frac{q_{C_{cg}}(t) - q_{C_{cg}}(t - \Delta t)}{\Delta t} - G_{C_{cg}} \cdot v_{C_{cg}}(t), \tag{3.9}$$

where $q_{C_{cg}}$ is the charge.

Since the IGBT has three operating states: OFF state, linear, and saturation regions, the metal-oxide-semiconductor field-effect transistor (MOSFET) is adopted for model description, and its equivalent current $i_{mos}$ can be formulated by three segments, namely

$$i_{mos} = \begin{cases} 0, (v_{C_{ge}} < V_{th}) \ \ \& \ \ (v_d \leq 0) \\ a_2 \cdot v_d^{(z+1)} - b_2 \cdot v_d^{(z+2)}, v_d < (y \cdot \Delta v_{C_{ge}})^{\frac{1}{x}} \\ \frac{\Delta v_{C_{ge}}^2}{(a_1 + b_1 \Delta v_{C_{ge}})}, others, \end{cases} \tag{3.10}$$

where $a_1$, $a_2$, $b_1$, $b_2$, $x$, $y$ and $z$ are coefficients, $v_{C_{ge}}$ and $v_d$ are the voltages over capacitor $C_{ge}$ and $i_{mos}$, respectively, $V_{th}$ is the IGBT channel threshold voltage, and $\Delta V_{C_{ge}}$ is defined as

$$\Delta v_{C_{ge}} = v_{C_{ge}} - V_{th}. \tag{3.11}$$

consequently, the conductance $G_{mosvd}$ and transconductance $G_{mosvcge}$ resulting from the discretization of the component can be derived by taking partial derivatives of $v_d$ and $v_{C_{ge}}$, respectively, and each operation state has a different form.

- **ON state**

  Under ON state, i.e. $v_d$ is less than the value of $(y \cdot \Delta v_{C_{ge}})^{\frac{1}{x}}$, the conductance and transconductance are expressed by the following equations

$$G_{mosvd} = \frac{\partial i_{mos}}{\partial v_d} = a_2(z+1) \cdot v_d^z - b_2(z+2) \cdot v_d^{(z+1)}, \tag{3.12}$$

$$G_{mosvcge} = \frac{\partial i_{mos}}{\partial v_{C_{ge}}} = \frac{\partial a_2}{\partial v_{C_{ge}}} \cdot v_d^{(z+1)} - \frac{\partial b_2}{\partial v_{C_{ge}}} \cdot v_d^{(z+2)}. \tag{3.13}$$

- **Transient state**

  Under the transient stage, the conductance $G_{mosvd}$ is zero, and the transconductance can be derived as

$$G_{mosvcge} = \frac{2\Delta v_{C_{ge}}}{(a_1 + b_1 \Delta v_{C_{ge}})} - \frac{b_1 \Delta v_{C_{ge}}^2}{(a_1 + b_1 \Delta v_{C_{ge}})^2}. \tag{3.14}$$

- **OFF state**

  When the IGBT is OFF, both $G_{mosvd}$ and $G_{mosvcge}$ are zero.

Taking the different forms of $G_{mosvd}$ into consideration, the companion current of $i_{mos}$ can be calculated by

$$I_{moseq} = i_{mos} - G_{mosvd} \cdot v_d - G_{mosvcge} \cdot V_{C_{ge}}. \tag{3.15}$$

The tail current $I_{tail}$ occurs when the IGBT is being turned off, and it can be estimated using the formula below

$$I_{tail} = \begin{cases} 0, \frac{V_{tail}}{R_{tail}} < i_{mos} \\ (\frac{V_{tail}}{R_{tail}} - i_{mos}) \cdot i_{rat}, others, \end{cases} \tag{3.16}$$

where $i_{rat}$ is a fixed current.

Finally, all subunits are combined and expressed as

$$\mathbf{G_{IGBT}} \cdot \mathbf{v_{IGBT}} = \mathbf{I_{IGBTeq}}, \tag{3.17}$$

where $\mathbf{G_{IGBT}}$ is the $5 \times 5$ admittance matrix, $\mathbf{v_{IGBT}}$ is the IGBT node voltage, and $\mathbf{I_{IGBTeq}}$ is the companion current.

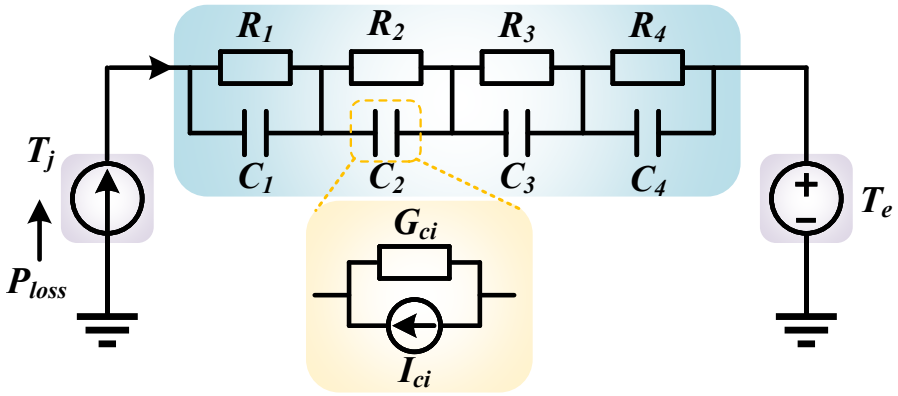



Figure 3.2: Equivalent thermal network.

### 3.2.2 Diode Nonlinear Behavioral Model

The nonlinear behavioral power diode model is demonstrated in the right part of Fig. 3.1. The relationship between diode static current $I_d$ and its junction voltage is expressed by

$$I_d = I_s \cdot [e^{(\frac{V_j}{V_b})} - 1], \tag{3.18}$$

where $I_s$ is the leakage current, $V_b$ is the junction barrier potential, and $V_j$ is the static junction voltage.

The nonlinear diode (NLD) conductance $G_j$ and the companion current $I_{jeq}$ are

$$G_j = \frac{\partial I_d}{\partial V_j} = \frac{I_s}{V_b} e^{\frac{V_j}{V_b}}, \tag{3.19}$$

$$I_{jeq} = I_d - G_j \cdot V_j. \tag{3.20}$$

### 3.2.3 Electro-Thermal Model

As given in Fig. 3.2, the process in which the power loss causes semiconductor junction temperature rise can be modeled by the $R$-$C$ pairs as an equivalent electro-thermal network [51] which is generally expressed as

$$Z_{th} = \sum_{i=1}^{N} R_{th(i)}(1 - e^{-\frac{t}{\tau_i}}), \tag{3.21}$$

$$C_{th(i)} = \frac{\tau_i}{R_{th(i)}}, \tag{3.22}$$

where $R_{th(i)}$ and $\tau_i$ are constants. The power loss of the IGBT $P_{loss}$ is numerically equal to the input current of the transient thermal impedance equivalent circuit. On the other hand,

the terminal voltage of the current source can be taken as the semiconductor's junction temperature $T_j$,

$$T_j(t) = \sum_{i=1}^{4} \frac{P_{loss}(t) + I_{ci}(t - \Delta t)}{G_{ci} + R_{th(i)}^{-1}} + T_e, \tag{3.23}$$

where $T_e$ stands for the ambient temperature, $G_{ci}$=$\Delta t/2C_{th(i)}$, and $I_{ci}$ is the capacitor history current.

## 3.3 IGBT NBM Implementation on ACAP

The ACAP consists of three distinct domains: processing system (PS), programmable logic (PL), and AI Engine (AIE). Each domain has its own unique design flow and optimization techniques for achieving high performance in simulation acceleration. The processing system provides a standard CPU and peripherals, while the programmable logic offers a flexible and customizable hardware fabric. The AI Engine is an innovative and specialized domain that enables the design of data processing. This section describes the implementation of each domain and compares their simulation times and resource consumption.

### 3.3.1 IGBT Designs on ACAP

#### 3.3.1.1 Processing System (PS)

The PS in ACAP is designed following a traditional software-based design flow, which typically involves developing software applications using a high-level language C/C++, and adopting development tools such as Xilinx® Vitis to compile, debug, and deploy. For programming the compute units and executing programs on the target device VCK190, OpenCL and Xilinx® Runtime (XRT) are utilized.

To design the IGBT NBM simulation, several steps are required. Firstly, the Arm Cortex-A72 is chosen as the high-performance computing unit for developing the IGBT NBM simulation algorithm, which includes calculating IGBT and diode parameters in different states, as well as the matrix-solving process required for simulation. Next, use C/C++ to write the OpenCL kernel, which enables the programs running on the ACAP's compute unit. The OpenCL application processing interface (API) can be used to create the necessary OpenCL context, command queues, and memory buffers for executing the kernel, and the OpenCL kernel is compiled to integrate the application in the device-specific binary code with the processing system of the VCK190. Finally, the emulation is run on the ACAP using an appropriate interface, such as Ethernet or USB. These steps allow for

the efficient execution of the OpenCL kernel on the target device, with XRT providing the necessary runtime infrastructure for the execution of the kernel.

#### 3.3.1.2 Programmable Logic (PL)

On the other hand, the PL domain in ACAP uses a hardware-based design flow, where the design could be described by using a hardware description language such as Verilog/VHDL, or converting C/C++ code to an FPGA-compatible format in Xilinx® Vitis HLS. The design flow for the programmable logic includes developing, simulating and verifying the design, as well as synthesizing it into a bitstream that can be loaded onto the programmable logic fabric of the ACAP.

For the IGBT NBM simulation in the PL domain of the ACAP, the design specification is first defined using C/C++, which includes the mathematical equations and matrix solver for the device-level model. The design specification is then synthesized into RTL (Register Transfer Level) code, a low-level hardware description language, and optimized for the target platform and can be further optimized using directives and pragmas. After the RTL code is implemented on the ACAP in Xilinx® Vitis, the implemented design could be verified by emulation to ensure that it meets the design specifications and functional requirements. Finally, the HLS tool generates reports with numerous metrics, such as resource utilization and latency, for analyzing performance.

#### 3.3.1.3 AI Engine (AIE)

The AIE programming flow is carried out in two phases with the Vitis IDE: kernel programming and graph programming. A kernel describes a specific computing process running on a single AIE tile where C/C++ code is used for programming, and a C++ framework is provided by Xilinx® to create graphs from kernels that contain declarations for the graph nodes and connections. A graph will instantiate and connect the kernels using buffers and streams, and also describe the data transfer between the AIE array and the rest of the ACAP device.

Fig. 3.3 shows the dataflow graph and kernels of the NBM implementation, which is achieved by 5 AIE kernels ($pre\_cal$, $diode$, $igbt\_on$, $igbt\_off$, and $igbt\_transient$), connections, and different types of buffer, where the data transfer between kernels is memory-to-memory and the transmission of data between kernels and PL is stream-to-memory or memory-to-stream. First, the node voltage of the IGBT is sent as input to the first kernel $pre\_cal$ for parameters precalculation, the second kernel $diode$ computes the parameters of

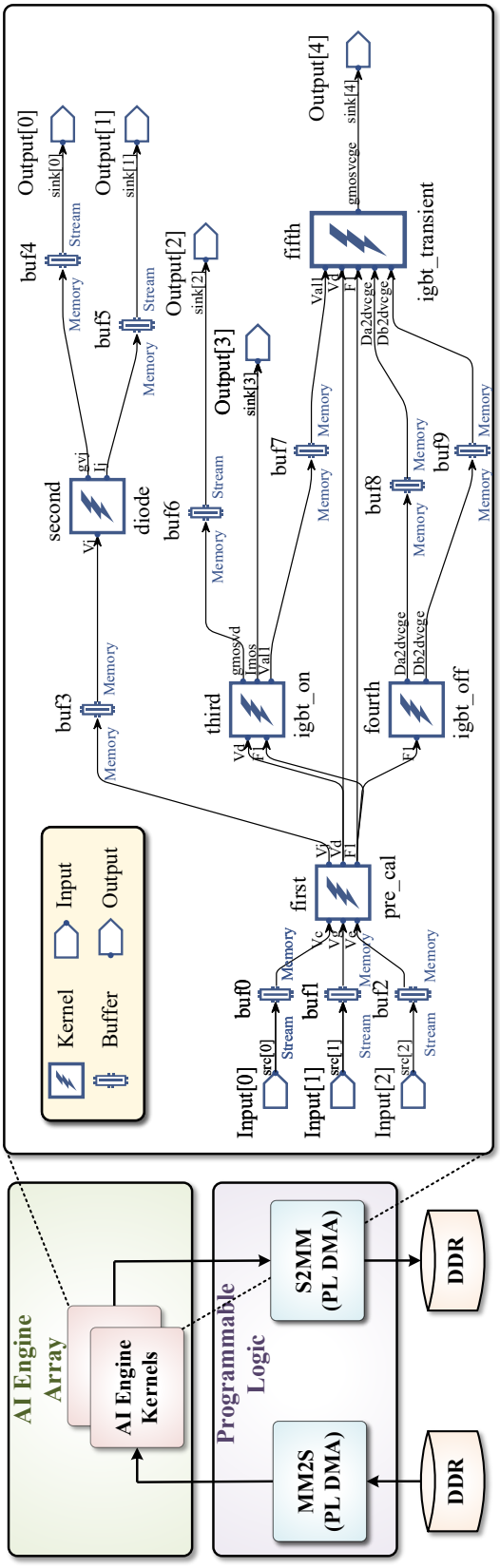



Figure 3.3: AI Engine data flow graph of IGBT NBM.

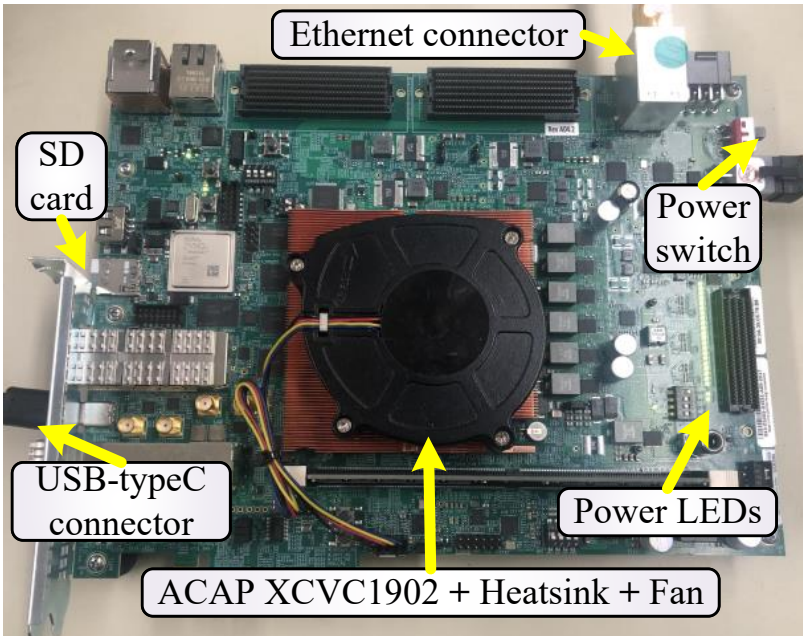



Figure 3.4: Xilinx® VCK190 board setup.

the diode, and the third to fifth kernels $igbt\_on$, $igbt\_off$, and $igbt\_transient$ are designed to perform IGBT nonlinear functions in the ON state, OFF state, and transient state, respectively, and finally, the outputs make up the admittance matrix in (3.17).

### 3.3.2 NBM Implementations Comparison on Three Domains

Fig. 3.4 shows the setup of the hardware platform Xilinx® Versal™ VCK190 board with the ACAP device XCVC1902. The IGBT NBM is implemented on the PS, PL, and AIE of the ACAP, respectively, for a comprehensive evaluation of different design schemes. When the simulation duration is 0.05s, the actual execution time for the simulation is 0.042s on the PS. Then the real-time ratio could be expressed as $\frac{0.05s}{0.042s} = 1.19$, which indicates that for a single IGBT, the simulation speed is slightly faster than real-time. However, the simulation of a power converter with many IGBTs slows down significantly due to the inadequate scalability of PS.

Table 3.1 lists the latency and resource utilization of NBM implementation on AIE and PL. While the PL has the advantages of numerous resources and customizability to support the simulation of systems with multiple IGBTs, a heavy data dependency of the NBM restricts parallelism and ultimately leads to high latency. The AIE has highly optimized processors and a data stream frequency of 1GHz for efficient parallel processing. The AIE scalar processor has an excellent performance on fixed-point data processing but is

Table 3.1: NBM implementation in AIE and PL

| Part | Latency | Resource Utilization | | |
|---|---|---|---|---|
| AIE Scalar Unit | 10.946$\mu$s | AIE Tiles | 5 | 1.25% |
| | | Kernels | 5 | - |
| PL | 3.37$\mu$s | BRAM | 28 | 1.45% |
| | | URAM | 0 | 0.00% |
| | | DSP | 252 | 12.80% |
| | | LUT | 52230 | 5.80% |
| | | FF | 21306 | 1.18% |

not ideal for floating-point data required by NBM, as shown in Table 3.1. To accelerate the computing process, the ML strategy and AIE Vector Unit are adopted, as the adapted vectorized data type and SIMD features enable the IGBT NN model to be processed simultaneously.

## 3.4 Machine Learning-Based Modeling and Realization of NBM

Based on the NBM performance evaluation in the previous section, it can be seen that the real-time performance is less than satisfactory. A machine learning-based co-simulation technique is proposed to streamline the computational procedure while maintaining simulation accuracy.

### 3.4.1 Selection of Neural Network Topology

Different neural networks such as convolutional neural networks (CNN), recurrent neural networks (RNN), and artificial neural networks (ANN) are novel trends in the realm of machine learning, providing impetus for various applications. Similarly, the NN methodology can be valuable in the field of real-time simulation, as one of its benefits is that it can take advantage of the numerical prediction property to derive the corresponding output model by training on specific data, thus avoiding the extensive computations caused by iterations during transient states.

In Fig. 3.5, an elementary version of the neural network is depicted, with a multilayer structure formed by certain neurons, notably the input layer, the hidden layer, and the output layer, each node in the upper layer is linked to all the nodes in the next layer. The mathematical expression is

$$\mathbf{Y} = \mathbf{f}(\mathbf{X} \cdot \mathbf{W} + \mathbf{b}) = \mathbf{f}(\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{X_i W_i} + \mathbf{b}), \tag{3.24}$$

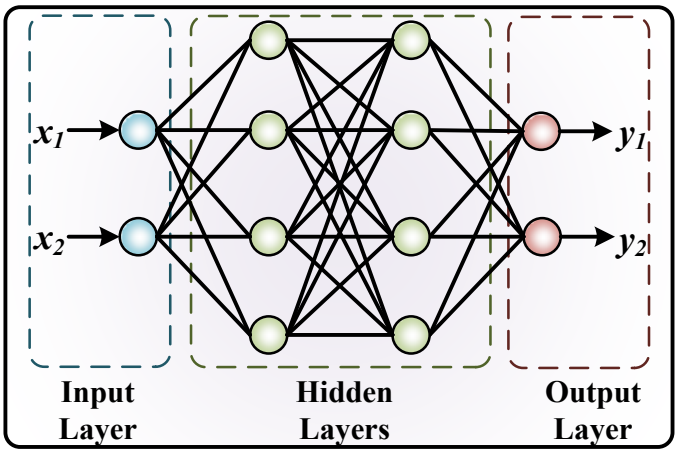



Figure 3.5: ANN basic structure with three layers

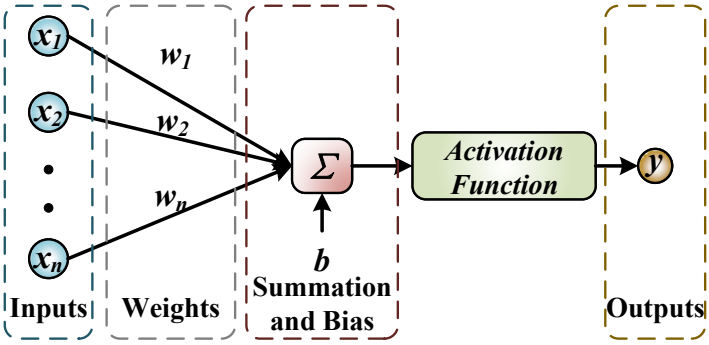



Figure 3.6: Neural network internal model.

where $\mathbf{X}$ is the input, $\mathbf{n}$ is the number of neurons, $\mathbf{Y}$ is the output, $\mathbf{W}$ is the weight, and $\mathbf{b}$ is the bias.

Fig. 3.6 represents the general mathematical model of NN, where the input variables from $x$ to $x_i$ are multiplied with the weight matrix $\mathbf{W}$ and summed with the bias value $\mathbf{b}$. Finally, the activation function serves as a nonlinear mapping, limiting the amplitude of the output to a specific range. Common activation functions include Sigmoid, Tanh, and rectified linear unit (ReLU) [52], of which ReLU is the most popular type in machine learning compared to the Sigmoid and Tanh functions since ReLU has only a linear relationship and its computation is faster than the other, which needs to perform exponential operations.

In this chapter, ANN is chosen as the IGBT NBM transient state machine learning model because it has the feature of fitting the intermediate data curve by the first and last data only, which avoids the problem of computational iterations in traditional EMT models, and its high parallelism and low execution delay can match the criteria of transient

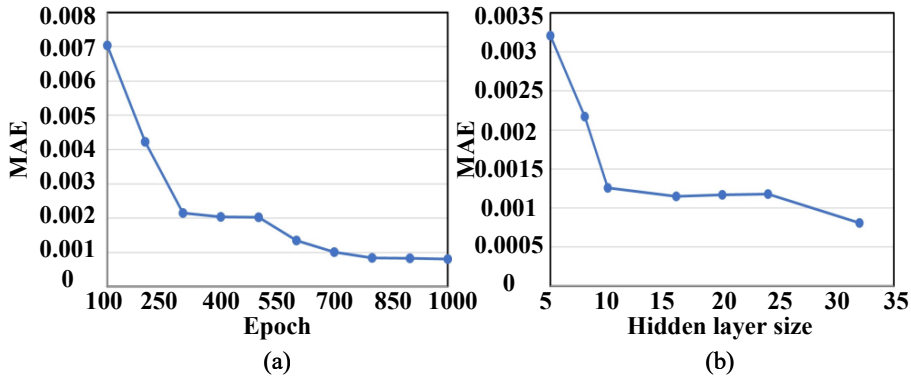



Figure 3.7: IGBT ANN model's error reduction process.

simulation.

### 3.4.2 Data Collection and Training Methodology

One crucial part of ML training of devices is the selection of the dataset since it will influence the accuracy of the training results and the generality of the model. For the IGBT Siemens BSM300GA160D, rated 1600V, 300A in this chapter, where the parameters are provided in Appendix, the dataset is extracted from the MATLAB simulation results of the IGBT NBM, and both the turn-on and turn-off data during the transient state should be of concern.

The corresponding IGBT NBM ANN model has 5 input variables including the initial and last status of the transient state voltage $V_{start}$, $V_{end}$, current $I_{start}$, $I_{end}$, and gate signal $V_g$. All these data are normalized to (-1,1) using min-max normalization, which allows for easier data processing and better training performance.

The mean absolute error (MAE) is used to measure the accuracy of the training model:

$$MAE = \sum_{i=1}^{n} \frac{|y_i^{pre} - y_i|}{n}, \tag{3.25}$$

where $n$ is the total number of the output, $y_i$ is $i^{th}$ originate value from the dataset, and the $y_i^{pre}$ is the corresponding output of the ANN model. The Adam optimization algorithm is adopted as the training methodology to minimize the error [53]. Fig. 3.7 shows the MAE of the IGBT ANN model, which presents the error reduction during the training process. The training epoch is selected as 1000 to reduce error, and the hidden layer size is set to 32 to improve the efficiency of the AIE vector code since the size of the accumulator is a multiple of 8-bit. Since the MAE of one hidden layer is not significantly distinct from that of two hidden layers, it is used to achieve optimal performance.

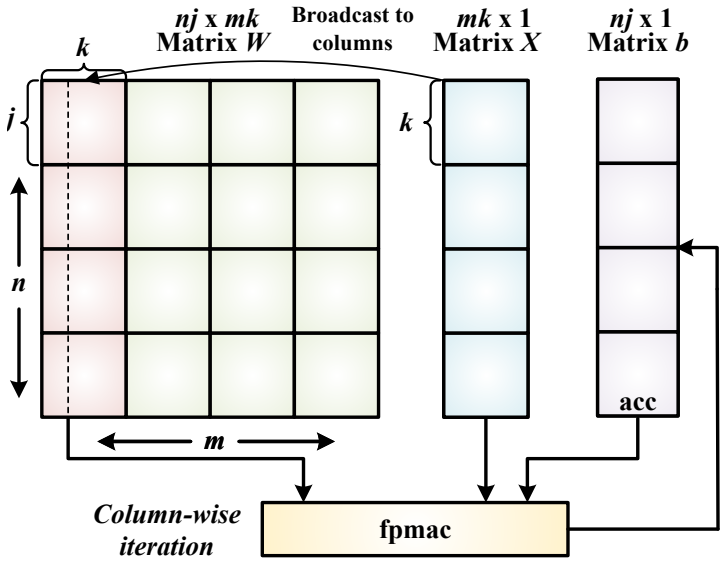



Figure 3.8: Vectorized matrix multiplication in column.

### 3.4.3 Matrix Multiplication Implementation with AIE

From the previous part of this section and the mathematical expression, the input variables need to be multiplied by the weight and summed by bias, which could be seen as the matrix multiplication and addition for the hidden layer and output layer. Some changes are performed to the matrix size that has no impact on the outcome to make the operations adaptable for the AIE vectorized code, for example, for the hidden layer, the size of the weight matrix $\mathbf{W}$ is $32 \times 8$, the input matrix $\mathbf{X}$ is $8 \times 1$, and the bias matrix $\mathbf{b}$ is $32 \times 1$.

The column-based matrix multiplication is implemented using vectorized AIE code, where the vector data types pack multiple scalar data elements into a wider vector. In this case, both the AIE API and intrinsics are employed to increase design productivity. The AIE API, which is implemented as a C++ header-only library and offers types and operations that are converted into effective low-level intrinsics, is a portable programming interface for accelerators. In the meantime, the vector data types and the MAC intrinsics [27] are deployed for application-level programming. There are two solutions based on AIE floating-point intrinsics to implement the matrix multiplication; the first strategy is to perform the multiplication with $fpmul$ and then add it with the bias matrix to the accumulator using $fpmac$. Another methodology, the more efficient way presented in this chapter, is to apply $fpmac$ intrinsic only as shown in Fig. 3.8. Firstly, the bias matrix $\mathbf{b}$ is loaded to the accumulator, then the weight matrix $\mathbf{W}$ is stored at several accumulators by

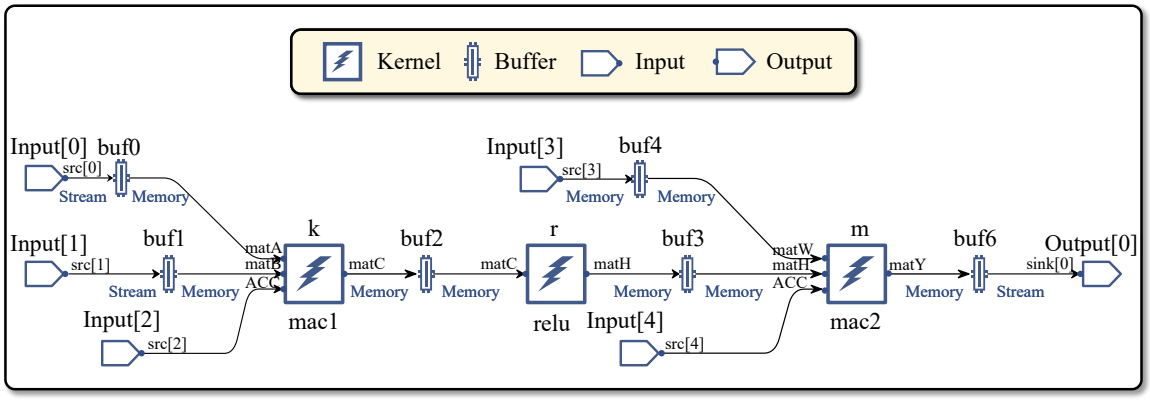



Figure 3.9: The IGBT ANN model AIE implementation.

Table 3.2: IGBT ANN model performance in AIE

| Part | Latency | Size | Resource |
|---|---|---|---|
| Hidden layer | 136 ns | $[32 \times 8] \times [8 \times 1] + [32 \times 1]$ | 0.5% |
| Output layer | 1706 ns | $[80 \times 32] \times [32 \times 1] + [80 \times 1]$ | 0.5% |
| ReLU | 68 ns | $[32 \times 1]$ | 0.25% |

column, and each column in the weight matrix is multiplied by the corresponding row of the input matrix $\mathbf{X}$, where the $fpmac$ intrinsic is applied to perform both the matrix multiplication and addition, the full IGBT ANN AIE vectorized matrix calculation is shown in Fig. 3.9.

## 3.5 Emulation Results and Discussion

### 3.5.1 IGBT ANN Model Validation and Performance

Fig. 3.10 gives the ANN model training results compared with the offline device-level (100 ns time-step) simulation tool SaberRD®, where Fig. 3.10 (a) is the IGBT transient current and voltage of the turn-on state and Fig. 3.10 (b) is the turn-off state. Fig. 3.10 (c) and (d) show the IGBT junction temperature at 200 A and 333 A, where the latter needs an additional cooling system. Table 3.2 shows the latency and resource consumption of different parts of the ANN model implemented in AIE. A comparison of matrix multiplication implementations on different hardware platforms is given in Table 3.3, for the same size matrix multiplication, AIE is 2.6 times faster than CPU and more than 28 times faster than FPGA.

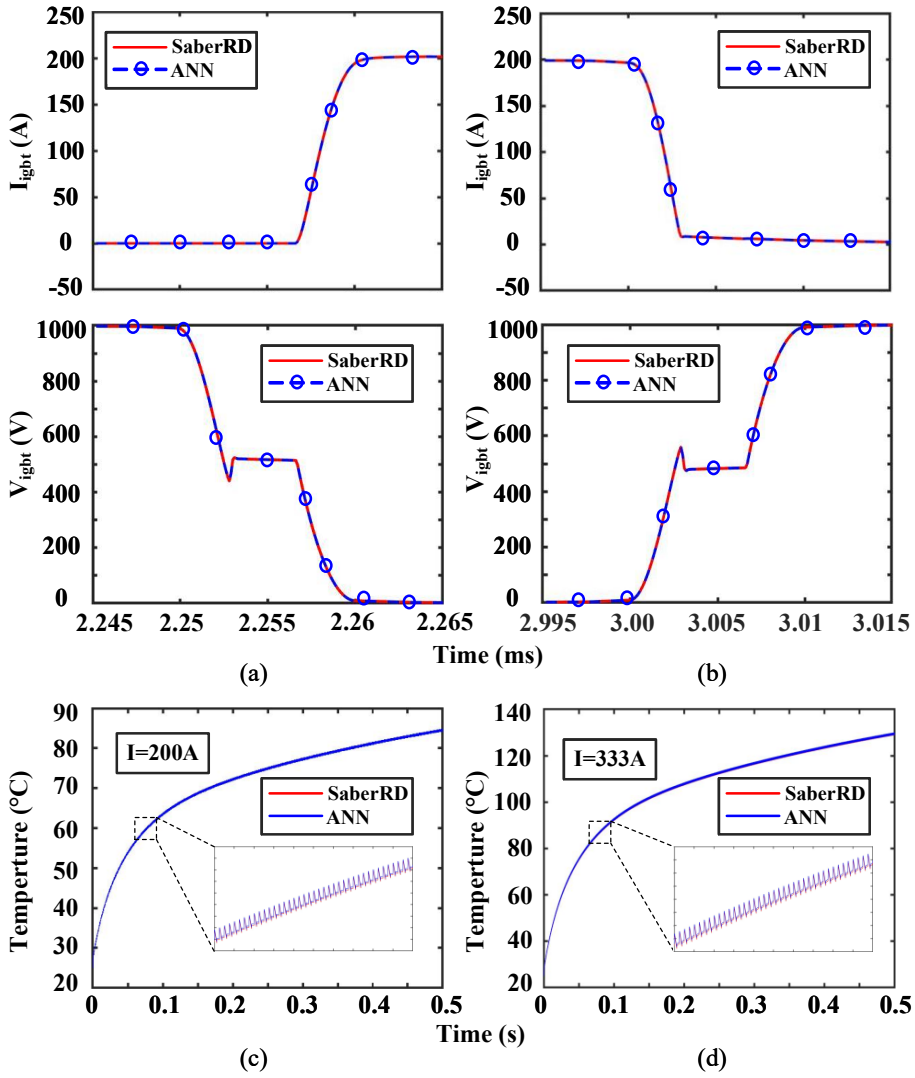



Figure 3.10: IGBT ANN model: (a)-(b) IGBT turn-on and turn-off state; (c)-(d) device junction temperature.

Table 3.3: Comparison of matrix multiplications on different hardware

| Hardware Type | Platform | Size | Latency |
|---|---|---|---|
| AI Engine | Versal™ VCK190 | $[32 \times 8] \times [8 \times 1] + [32 \times 1]$ | 136 ns |
| FPGA | Zynq® ZCU106 | $[32 \times 8] \times [8 \times 1] + [32 \times 1]$ | 3860 ns |
| CPU | Intel® Core™ i7 | $[32 \times 8] \times [8 \times 1] + [32 \times 1]$ | 360 ns |

### 3.5.2 Real-Time Emulation Results

The case study system is presented, where Fig. 3.11 shows the 2-level VSC converter. For the DC side, as shown in Fig. 3.12, there are 4 kinds of load circuits, namely half-bridge load, buck load, boost load, and full-bridge load, and Fig. 3.13 presents the control

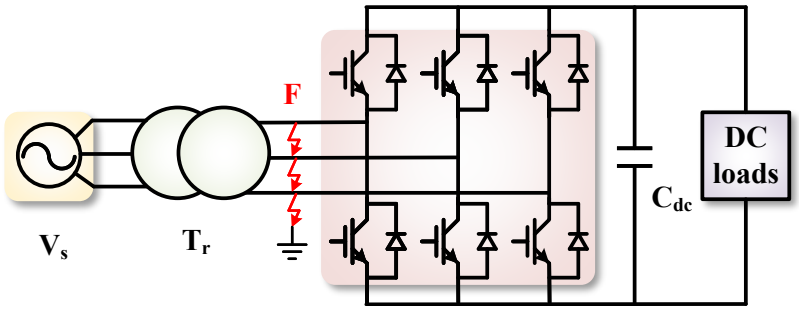



Figure 3.11: Case study of the full system: AC rectifier part.

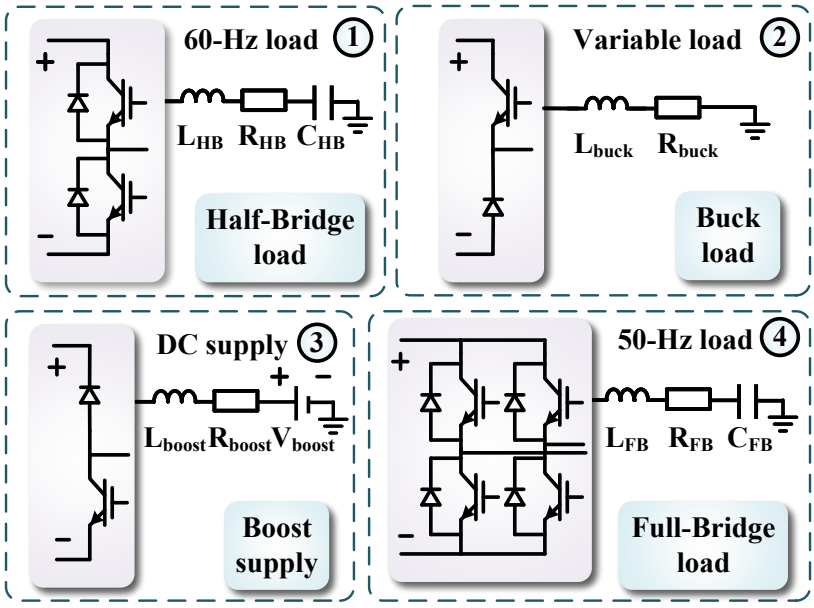



Figure 3.12: Case study of the full system: DC loads.

diagram. The system parameters are given in Appendix. The emulation of the system is implemented on the Xilinx® Versal™ ACAP XCVC1902, where the time-step is 5 $\mu$s. Table 3.4 provides the hardware resources consumption and the latency of the different parts of the system.

Table 3.4: Resources consumption of a VSC converter

| Part | Latency | BRAM | DSP | FF | LUT | URAM |
|---|---|---|---|---|---|---|
| Control | 4280 ns | 0.21% | 0.51% | 0.20% | 0.40% | 0 |
| Solver | 8900 ns | 0.10% | 0.20% | 0.28% | 0.62% | 0 |
| Converter | 1510 ns | 0.41% | 0.46% | 0.24% | 0.51% | 0 |

Fig. 3.14 demonstrates the simulation results of the case study system with the AC side fault $F$ at 0.4 s. In Fig. 3.14 (a), before the AC side fault, the power of the grid varied in the

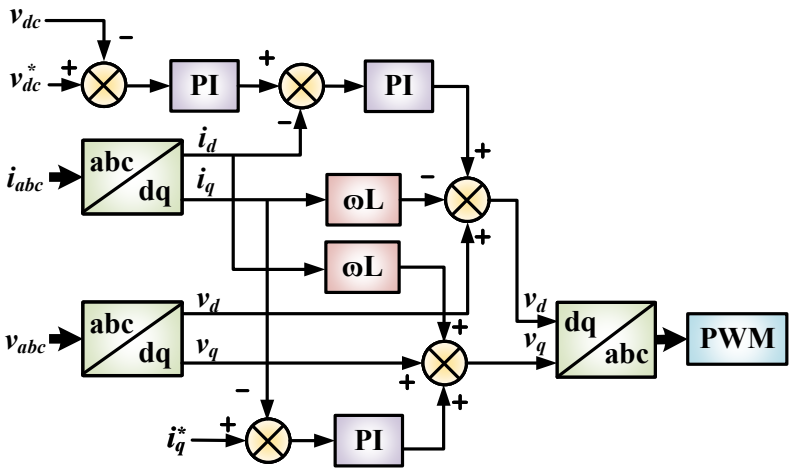



Figure 3.13: Case study of the full system: control diagram of 2-level VSC.

range of approximately 600 kW to 900 kW; and it quickly drops to about 50 kW when the fault occurs. Then after 0.1 s, the grid power is gradually restored. Fig. 3.14 (b) displays the power of the full-bridge and half-bridge load, which both decrease from their original power at fault, and increase to peak at 0.5 s, then reinstate at 0.6 s. Fig. 3.14 (c) is the power of the buck load and has the same trend as the previous figures while the value drops to 0 when the fault happens. Fig. 3.14 (d) is the boost load power and the power remains steady before the fault, and the value changes from about -124 kW to -110 kW between 0.4 s to 0.5 s, and recovery to the original value after 0.1 s. Fig. 3.14 (e) and (f) is the voltage on the DC side and AC side. Fig. 3.15 gives the junction temperature of an IGBT in the simulation of the whole system. In Fig. 3.15 (a), with Cooling System 1 which has the insufficient capacity as given in Appendix, the junction temperature reaches about 220° at the steady state. Fig. 3.15 (b) shows that with a decent capacity, such as Cooling System 2, the temperature remains below 70° even though the fault occurred.

In Fig. 3.16, the simulation results of the system are presented with the DC side half-bridge load circuit fault at 0.5 s and last for 2 seconds. Fig. 3.16 (a) shows the gird power between 0 and 3.0 s, and it can be seen that the power increases to about 95 kW at 0.5 s, and then returns to its original value at 2.5 s. Fig. 3.16 (b) is the power of the full-bridge and buck load, both of which do not change considerably after the fault occurs. In Fig. 3.16 (c), the power of the half-bridge load increases from its original value to 440 kW and becomes stable in the range of 390 kW to 420 kW, then restored after the fault ends at 2.5 s. Fig. 3.16 (d) shows the DC side voltage, which originally varied between approximately 950 V and 1040 V, and changed to between 940 V and 1050 V after the fault occurred.

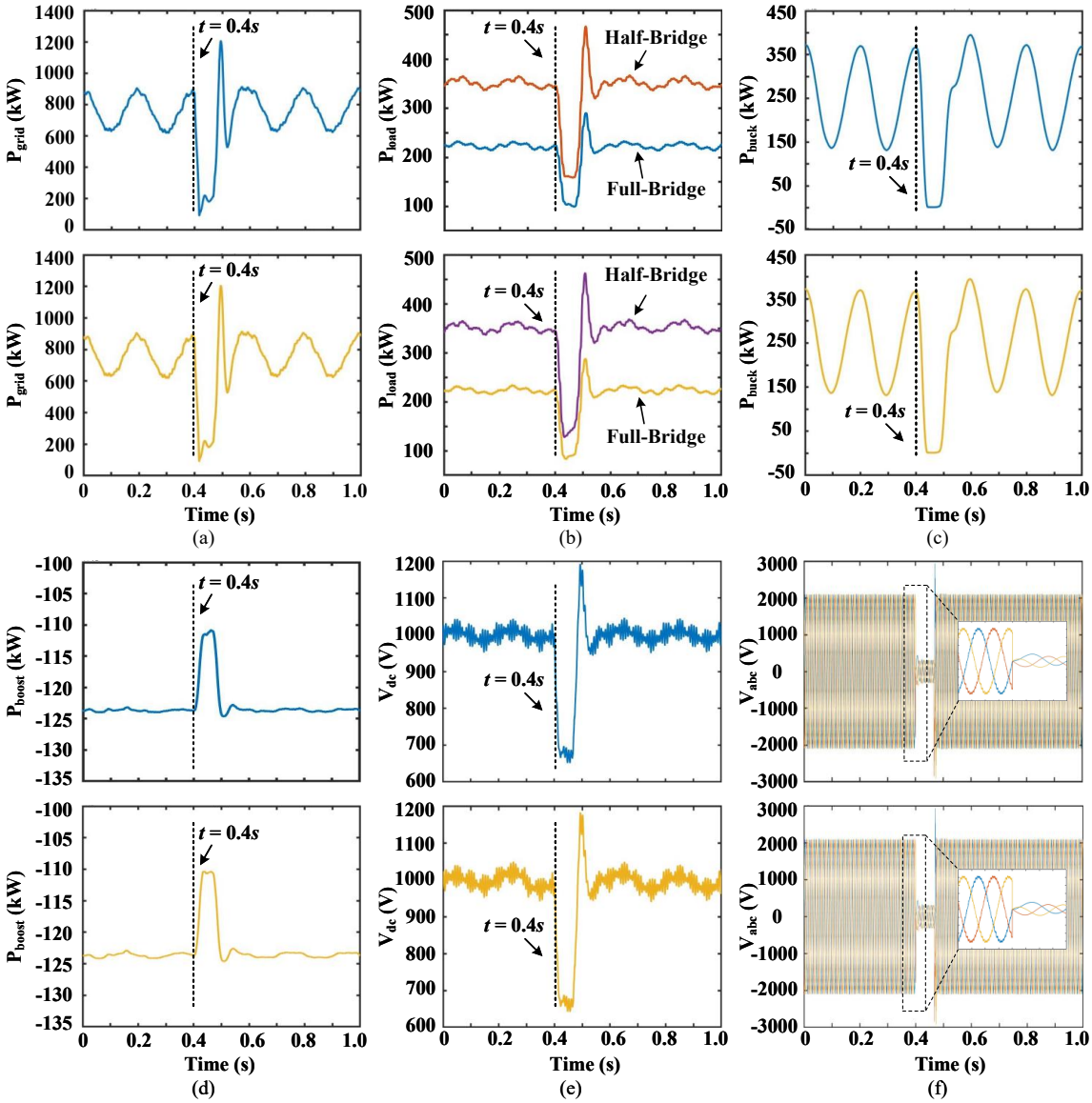



Figure 3.14: System-level results with AC fault from offline simulation (top), and ML model (bottom): (a)-(d) power of the grid, full-bridge and half-bridge loads, buck load, and boost load; (e)-(f) voltage of DC side and AC side.

## 3.6 Summary

Real-time emulation of a device-level nonlinear behavioral model of IGBT is a challenging task due to its high computation burden arising from the need for an iterative solution of device equations to obtain a convergent solution of every nanosecond scale time-step. In this chapter, a machine learning strategy is proposed to tackle the IGBT nonlinear behavioral electro-thermal model and demonstrated in a multi-converter supply-load system case study. The model is implemented on three main domains of a novel heterogeneous

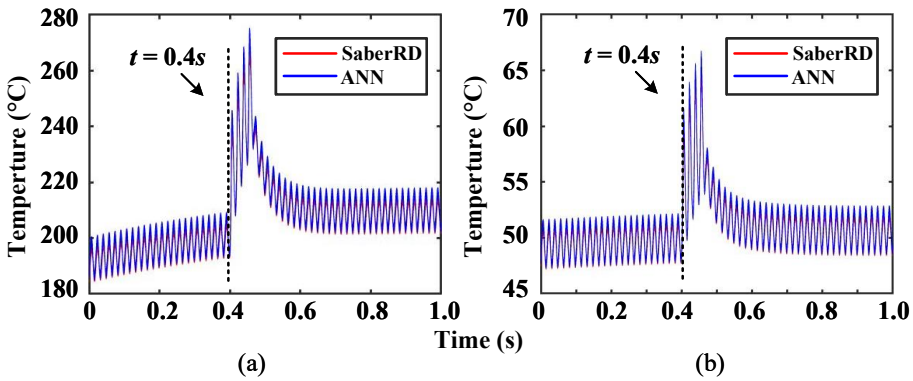



Figure 3.15: Device junction temperature with: (a) Cooling System 1; (b) Cooling System 2.

ACAP hardware: PS, PL, and AIE, which are introduced in detail in terms of functionality and features. The performance evaluation results, covering latency and hardware resource consumption, are provided separately. To make better utilization of the VCK190 hardware platform and AIE characteristics to achieve the requirements of real-time simulation, the IGBT ML-based model and NNs training methodology are proposed, where the ANN model is adopted to convert the complex computational iterative process of the transient state into the simpler matrix operations. From results comparisons with the conventional model in device-level emulation, the error of the IGBT ML model is within 1%, and the real-time requirement can be achieved with less resource consumption. The system-level simulation results are given for two different fault scenarios on both AC and DC sides and validated by MATLAB/Simulink®. The proposed modeling and implementation strategies can be applied in the future for real-time emulation of energy conversion systems in various practical applications.

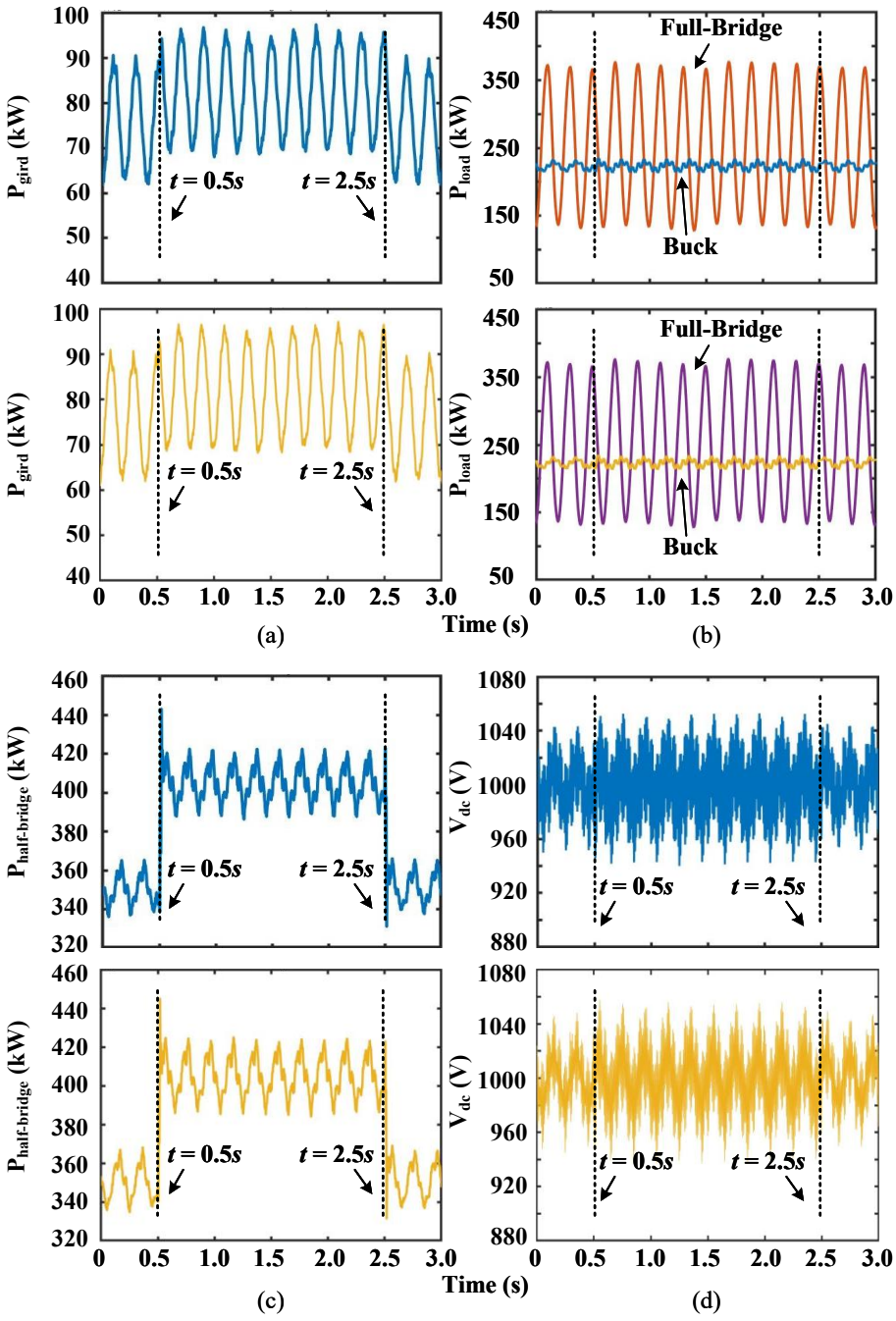



Figure 3.16: System-level results with half-bridge load circuit fault from offline simulation (top), and ML model (bottom): (a)-(c) power of the grid, full-bridge and buck load, and half-bridge load; (d) DC side voltage.

# 4

# EMT Modeling of Modular Multilevel Converter with Embedded Energy Storage for Wind Farm Grid Integration

## 4.1 Introduction

Wind has become a major resource for renewable energy generation because of ecological and environmental benefits. As the technology is mature, a wind farm can be constructed within a short cycle at a comparatively low cost [54]. However, wind power is unstable because the strength and direction of the natural wind are stochastic, bringing dramatic challenges to the stability of the power grid [55] when there is large-scale wind energy penetration. In such cases, power electronics-based energy storage systems as a solution can quickly provide a continuous and stable backup power supply to avoid economic losses [56].

Energy storage systems can be deployed in a centralized or distributed form. The high modularity and flexibility of the latter type make it more competitive than its centralized counterpart [57], and thus increasingly utilized [58, 59]. As an alternative to lithium batteries, energy storage based on supercapacitors has drawn attention in power apparatus, such as the wind turbine [60], and the motor-driven system [61] for advantages such as faster and safer charging, more eco-friendly raw materials, and longer lifetime. Supercapacitors can also be adopted as split energy storage elements to AC-DC converters as fault-resilient schemes [62]. In a simulated onboard network, they are also chosen to serve

as storage systems due to their fast dynamics and decent efficiency [63].

The modular multilevel converter (MMC) has grown rapidly in recent years and is widely adopted in high-voltage direct current (HVDC) for offshore wind farm integration [64,65]. An MMC with embedded energy storage (MMC-EES) allows the many batteries or supercapacitors to be distributed among its submodules and therefore enables more effective energy management. Electromagnetic transient (EMT) simulation plays an important role in the study of such kinds of power electronics apparatuses applied in power systems prior to in-situ commissioning. To expedite the simulation, the average-value models [66] and equivalent models [67] are adopted for MMC-EES. As a consequence, the simulation omits transient electromagnetic details of individual components which are crucial for a design evaluation.

The detailed model, on the other hand, can demonstrate the dynamics of each submodule (SM) accurately. For example, the detailed equivalent model of the MMC-EES yields results that agree with the experiments [68]. For the simulation of massive MMC-EES systems, practical challenges include a large time-varying admittance matrix brought by the converter, a small step-size owing to the high switching frequency required by the submodule with energy storage (SM-ES), and Newton-Raphson iterations demanded by the nonlinear component insulated gate bipolar transistor (IGBT). The central processing unit (CPU) will be easily overwhelmed by these factors if sequential processing is carried out. Therefore, the extremely slow simulation speed prompts the exploration of hardware parallelism, in which case the application of hardware computational acceleration such as FPGA [69, 70] has a distinct effect. However, for multi-terminal complicated systems, FPGA is deficient in terms of hardware resources to accommodate a practically large system.

Attributing to its massive numbers of cores and efficient parallelism, the graphics process unit (GPU) is promising in the high-performance computing of various electrical energy systems [71]. It provides satisfactory speedups over CPU and has been employed for EMT simulation acceleration [72–74]. The single-instruction multiple-thread (SIMT) feature of GPU is particularly suitable for power systems that exhibit a dominant homogeneity. Through CUDA C++ programming, a substantial number of threads can be executed in parallel, thus significantly reducing the simulation time. In this chapter, a detailed EMT model of MMC-EES is proposed and high-performance transient simulation using a fully iterative solution scheme is conducted on the GPU. The operation modes are analyzed and the performance of the controller is demonstrated by MMCs with a proper voltage

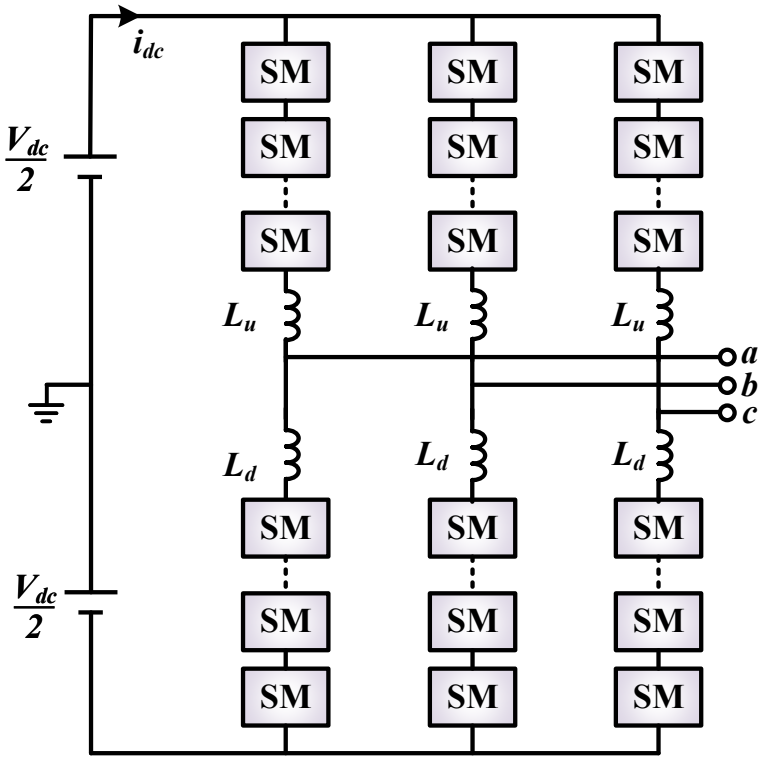



Figure 4.1: Topology of a three-phase modular multilevel converter.

level. As hundreds of submodules pose a dramatic computation burden, the nonlinearity caused by power semiconductor switches is excluded from the MMC main circuit so that both can be processed more efficiently. The parallelism is particularly enhanced for an extra speedup although explicit inhomogeneity exists in the MMC submodules.

This chapter is organized as follows. Section 4.2 introduces the topology and control strategy of the MMC with embedded energy storage. In Section 4.3, the EMT model of the MMC-EES is presented. The design of the GPU parallelism is provided in Section 4.4. Section 4.5 gives the implementation results and the validation, and conclusions are drawn in Section 4.6.

## 4.2 MMC with Embedded Energy Storage

### 4.2.1 Topology of MMC-EES

The configuration of a 3-phase MMC is shown in Fig. 4.1, where each phase consists of two bridge arms, both of which are composed of cascaded submodules in series with an inductor denoted as $L_u$ or $L_d$.

Fig. 4.2(a) shows the submodule structure of an MMC with embedded energy storage,

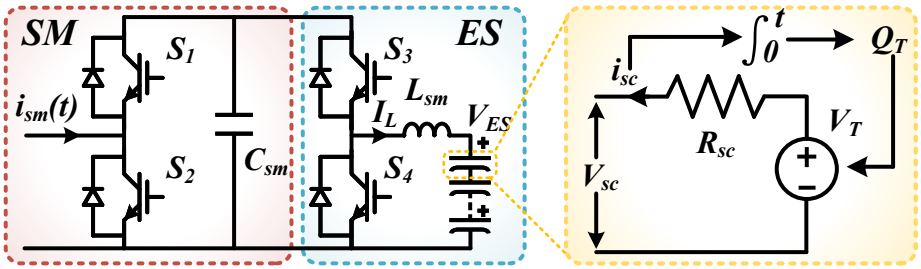



Figure 4.2: (a) SM-ES topology; (b) supercapacitor equivalent circuit.

which is a combination of the conventional half-bridge submodule (HBSM) and a DC-DC converter with an array of supercapacitors on the low voltage side. The HBSM consists of two complementary power switches $S_1$ and $S_2$ and a capacitor $C_{sm}$. The amount of energy that can be stored in $C_{sm}$ is relatively small and insufficient to serve as a grid energy supply.

In contrast, the SM-ES has a number of energy storage units and a DC-DC converter connected in parallel with the capacitor $C_{sm}$. The bi-directional DC converter allows the charge and discharge of supercapacitors. To be specific, the converter operates as a buck converter when the supercapacitors are storing energy, while it turns into a boost circuit to provide energy to the external system. Since all the energy storage units can be equally distributed, the power rating of each SM-ES is significantly lower compared with the entire MMC. This implies that a high switching frequency can be attained more easily to enhance power density, and to reduce the volume of inductor $L_{sm}$, a high switching frequency is particularly chosen for the two IGBTs $S_3$ and $S_4$.

The total amount of energy that three-phase MMC stores could be expressed as

$$W = \frac{1}{2} C_{ES} V_{ES}^2 \times 6N, \tag{4.1}$$

where $C_{ES}$ and $V_{ES}$ are the equivalent capacitance and voltage of the whole supercapacitor array, respectively, and $N$ is the number of SMs per arm. With a rated power $P_r$, the inertia of the MMC [75] can be described as

$$H = \frac{C_{ES} V_{ES}^2 \times 6N}{2P_r}. \tag{4.2}$$

To facilitate an energy management study in the high-fidelity EMT simulation, each supercapacitor is modeled. The Thévenin equivalent circuit for an individual supercapacitor

is shown in Fig. 4.2(b), and the overall voltage of the array could be calculated as

$$V_{ES} = \sum_{i=1}^{N_{sc}} V_{sc_i}, \tag{4.3}$$

where $N_{sc}$ is the number of supercapacitors in series, and $V_{sc}$ is the supercapacitor terminal voltage [76]:

$$V_{sc} = \frac{r}{\varepsilon} + \frac{2RT}{F} sinh^{-1}\left(\frac{Q_T}{\sqrt{8RT\varepsilon c}}\right) - R_{sc} i_{sc}, \tag{4.4}$$

$R_{sc}$ is the equivalent resistance, $i_{sc}$ is the supercapacitor current, $R$ is the ideal gas constant, $T$ is the operating temperature, $F$ is the Faraday constant, $r$ is the molecular radius, $c$ is the Molar concentration, $\varepsilon$ is the permittivity of the material, and $Q_T$ is the electric charge which is determined by the supercapacitor current

$$Q_T = \int_0^t i_{sc} dt. \tag{4.5}$$

The state of charge (SoC) of a supercapacitor is defined as the ratio between the remaining capacity and the rated capacity. A zero SoC means the supercapacitor is completely discharged while it is $100\%$ for a fully charged supercapacitor. It is formulated as follows

$$SoC = (SoC_{init} - \frac{Q_T}{Q}) \times 100\%, \tag{4.6}$$

where $SoC_{init}$ is the initial SoC, and $Q$ is the rated capacity of the supercapacitor.

Regardless of the energy flow direction, the continuous conduction mode (CCM) is always desired for energy storage units such as batteries and supercapacitors. Otherwise, they will be subject to frequent charge and discharge if under the discontinuous conduction mode (DCM), which not only affects the efficiency but the lifetime of the energy storage devices. To maintain the CCM, the critical value of the inductor needs to be determined. When the DC-DC converter operates under the boost mode, the current ripple of the inductor $\Delta i_L$ could be expressed as

$$\Delta i_L = \frac{V_{ES}}{L_{sm} f_{ES}} D \tag{4.7}$$

where $D$ is the duty cycle, and $f_{ES}$ is the switching frequency of $S_3$ and $S_4$. Substituting $D$ with variables that can be monitored leads to

$$\Delta i_L = \frac{V_{ES}(V_{C_{sm}} - V_{ES})}{L_{sm} f_{ES} V_{C_{sm}}} \tag{4.8}$$

where $V_{C_{sm}}$ is the instantaneous voltage of $C_{sm}$.

In the meantime, the DC component of the inductor $I_L$ can be determined from measurable quantities including the power provided or consumed by the submodule $P_{sm}$, i.e.,

$$I_L = \frac{P_{sm}}{V_{ES}} \tag{4.9}$$

To maintain CCM, the peak-to-peak current ripple should be smaller than 2 times of $I_L$, which consequently yields the critical inductance

$$L_{crit} = \frac{V_{ES}^2(V_{C_{sm}} - V_{ES})}{2P_{sm}f_{ES}V_{C_{sm}}}. \tag{4.10}$$

Similarly, analysis of the ripple current under the buck mode yields the same critical inductance. Then, the final value of $L_{crit}$ is chosen based on a number of factors, including the operational voltage range of the supercapacitor array, the maximum allowed charging or discharging power, as well as the submodule voltage scale. When $L_{sm}$>$L_{crit}$, the supercapacitors will be charged or discharged continuously; otherwise, they will encounter frequent interruptions that affect their capacity and lifetime.

### 4.2.2 MMC-EES Control

Fig. 4.3 shows the integration of two wind farms into a distribution network via the multi-terminal DC grid. The MMC-EES is located at the grid side as an inverter, and the conventional HBSM-based MMC operates as a grid-forming rectifier which provides a stable voltage at the point of common coupling (PCC) for offshore wind farms. The wind farm side and the AC grid side converters are linked by the transmission lines $TL_1$ and $TL_2$, and $TL_3$ connects the two grid-side MMCs to form a multi-terminal DC system.

For a lumped wind farm model that has $N_{wf}$ wind turbines, its total output power $P_{wf}$ can be calculated as

$$P_{wf} = \frac{1}{2}Av^3C_p\rho\eta \times N_{wf}. \tag{4.11}$$

where $A$ is the wind sweep area, $v$ is the wind speed, $C_p$ is the wind energy conversion rate value, $\rho$ is the air density, and $\eta$ is the coefficient. As the power delivered from the rectifier is not constant, the inverter MMC needs to supplement or absorb extra energy to keep the desired import power into the distribution grid, i.e., the balance between the power feeding into the grid and the power in the DC yard is maintained by the power of the MMC-EES.

The two control methods of MMC-EES as an inverter and the grid-forming MMC as a rectifier are shown in Fig. 4.4(a). The rectifier is expected to provide a stable voltage for wind turbines. The PCC voltages after $abc$-$dq$ transformation are compared with their

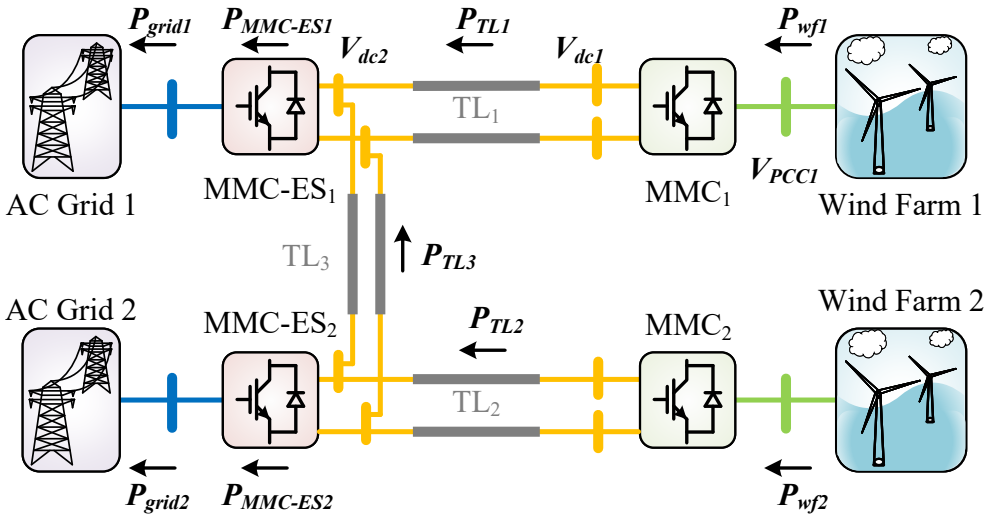



Figure 4.3: Multi-terminal DC grid with MMC-EES for wind farm integration.

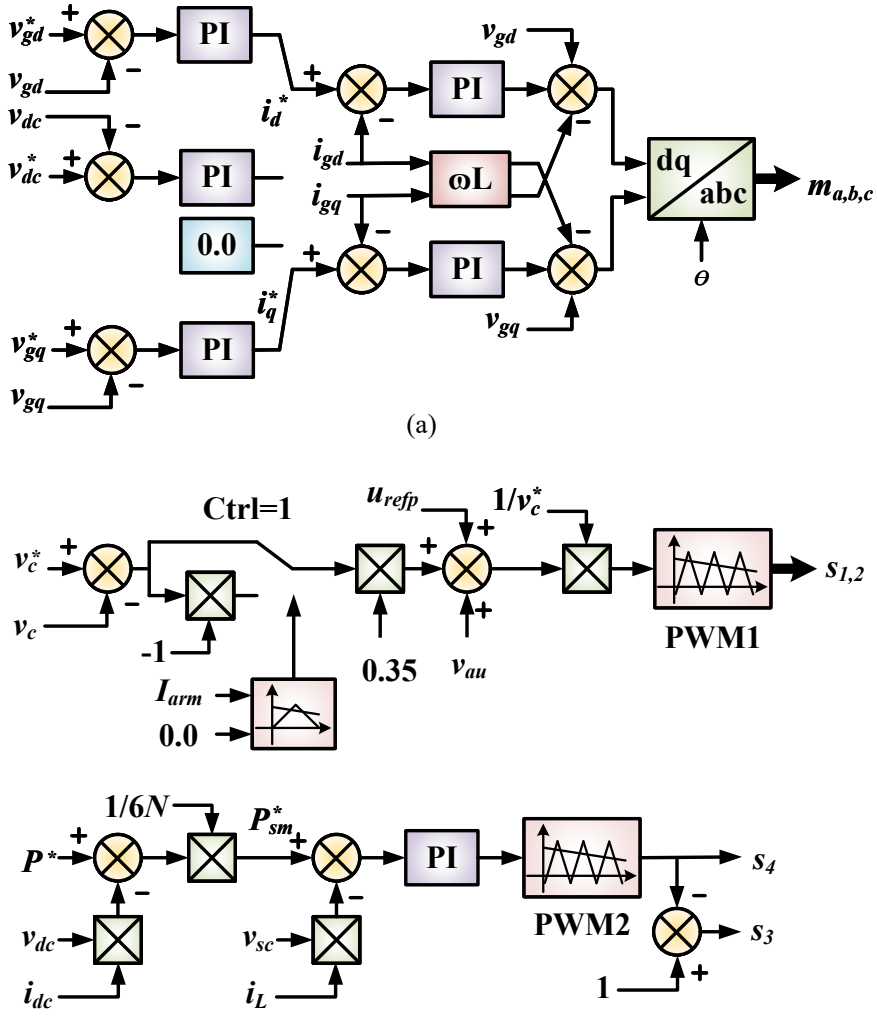
references $v_{gd}^*$ and $v_{gq}^*$ and then the errors are regulated by subsequent PI controllers to yield the current references $i_d^*$ and $i_q^*$. On the other hand, the inverters are in charge of establishing the DC voltage, the $d$-axis reference current $i_d^*$ comes from regulation of the DC voltage $v_{dc}$, and the $q$-axis reference current $i_q^*$ is related to reactive power or bus voltage control. The inner-loop current control remains identical regardless of converter types. The modulation signals $m$ for 3 phases are generated after the $dq$-$abc$ transformation. The angle reference $\theta$ is predefined without the usage of a phase-locked loop for the grid-forming MMC, whereas it is calculated based on the 3-phase grid voltage for an inverter.

Fig. 4.4(b) demonstrates the submodule internal controller, which includes regulations of capacitor voltage and supercapacitor power, respectively. The control of capacitor $C_{sm}$ voltage in an MMC SM is shown in the upper part of Fig. 4.4(b), which is divided into average control and balance control [77]. The actual capacitor voltage $v_c$ is compared with its reference $v_c^*$ in the balance control, and the result is added up with that of average control denoted by $v_{au}$, as well as the phase reference signal which links the internal controller to its outer-loop counterpart and takes the form of

$$u_{refp} = \frac{V_{dc}}{2N} - m, \tag{4.12}$$

where $V_{dc}$ is the converter DC voltage. As can be seen, the first two switches denoted as $S_1$ and $S_2$ are the objectives of the control scheme.

The second controller in Fig. 4.4(b) is designed for the DC-DC circuit with embedded energy storage for power compensation at the converter level. Since it independently controls the turn-on and turn-off of the remaining two switches $S_3$ and $S_4$, the switching




(a)

(b)

Figure 4.4: MMC control scheme: (a) Outer loop control; (b) coordinated submodule dc voltage and power control.

frequency, denoted as $f_{ES}$, can be much higher than that of the carrier in the first PWM scheme. In the meantime, the switch ON and OFF commands for upper and lower IGBTs are opposite to that of the MMC submodule. The set power reference of the entire MMC $P^*$ is distributed equally among the SMs with energy storage so that $P^*_{sm}$, the reference power for each SM, can be derived as

$$P^*_{sm} = \frac{P^* - (V_{dc} I_{dc})}{6N}, \tag{4.13}$$

where $I_{dc}$ is the converter-side current.

When there is a power shortage or surplus due to disturbance in the wind farm, the MMC-EES is able to provide stable power as a backup plant in this control strategy so

long as the complement is within its capacity.

## 4.3 MMC EMT Model Optimization

The detailed electromagnetic transient modeling of the MMC-EES is essential for a comprehensive design evaluation since it provides insight into the converter operation status. Tremendous computational resources are generally required when the simulation of a grid-connected high-level MMC modeled in its full scale is carried out. The consequent heavy computational burden is first tackled by circuit size reduction which results in the separation of submodules from the MMC main circuit, as depicted on the left side of Fig. 4.5.

Since the frequency of the arm current is much lower than the EMT simulation frequency, it can be considered as a constant for two adjacent time steps, and therefore the SM can be separated from the arm and forms a single subsystem by inserting one step latency between the voltage and current sources. The MMC main circuit becomes linear since after the exclusion of all SMs, the arm is comprised of voltage sources $v_{pn}$, where $n$ is from 1 to $N$, in addition to an inductor. On the nonlinear SM side, the current injected into it is equal to the arm current in the previous time step.

### 4.3.1 Nonlinear Submodule Splitting

The Norton equivalent circuit of the submodule with energy storage is shown in Fig. 4.5, where each nonlinear power switch is discretized and represented by a current source $f_i$ ($i$=1 - 4) in parallel with conductance $G_i$.

To reflect an accurate performance of the IGBT, the intrinsic diode is normally taken into consideration, which indicates the power semiconductor switch model is a combination of both. The gate signal $g$ determines the switching state of this combination. When the switch is turned on, the conductance is $1/r_{on}$ and the voltage drop $v_{on}$ which can be reliant on the collector current. When the diode is under conduction, i.e., the ideal diode $D_0$ is on, the total voltage drop is induced by the $p$-$n$ junction voltage $V_j$ and the resistor $r_{on}$. The internal voltage drop of the IGBTs and diode device results in the companion currents, i.e., $f_1$, $f_2$, $f_3$, and $f_4$.

Circuit partitioning of the MMC significantly reduces the number of nodes on both sides, as the submodule only has 5 nodes. The node voltage vector $\mathbf{v_{SM}}$ in SM-ES could

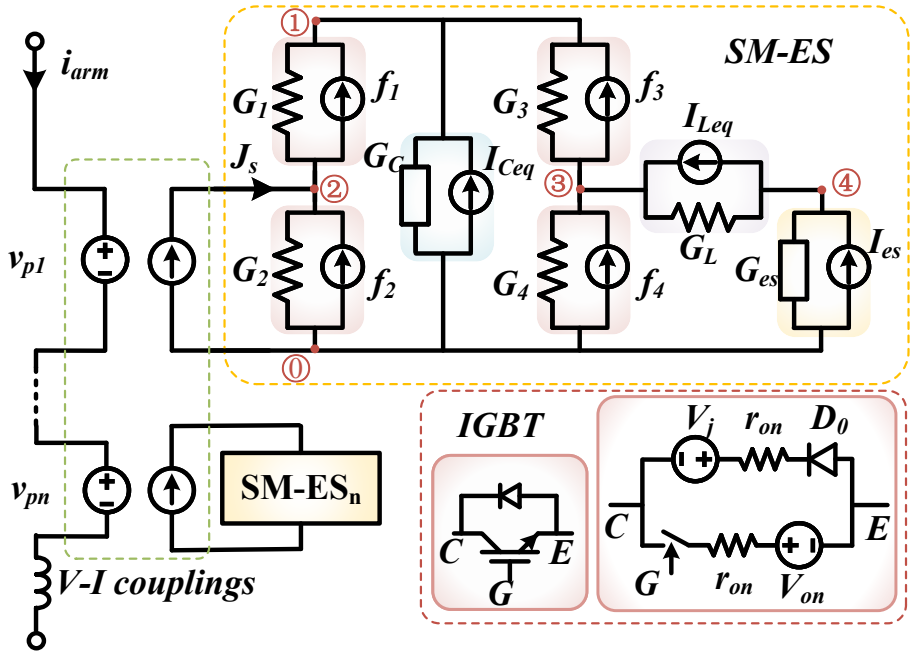



Figure 4.5: MMC partitioning by $V$-$I$ couplings and SM with embedded energy storage equivalent model.

be obtained by

$$\mathbf{v_{SM}} = \mathbf{G_{SM}^{-1}} \cdot \mathbf{J_{SM}}. \tag{4.14}$$

Since each submodule constitutes an independent circuit, an extra node can be omitted. By taking Node 0 as a virtual ground, the original 5th-order matrix is reduced to 4th-order. Then, the $4 \times 4$ admittance matrix $\mathbf{G_{SM}}$ can be organized as:

$$\begin{bmatrix} G_C + G_1 + G_3 & -G_1 & -G_3 & 0 \\ -G_1 & G_1 + G_2 & 0 & 0 \\ -G3 & 0 & G_L + G_3 + G_4 & -G_L \\ 0 & 0 & -G_L & G_L + G_{es} \end{bmatrix} \tag{4.15}$$

and the companion current vector is

$$\mathbf{J_{SM}} = \begin{bmatrix} I_{Ceq} + f_1 + f_3 \\ J_s - f_1 + f_2 \\ I_{Leq} - f_3 + f_4 \\ -I_{Leq} + I_{es} \end{bmatrix} \tag{4.16}$$

In the matrices $\mathbf{G_{SM}}$ and $\mathbf{J_{SM}}$, the transmission line model (TLM) technique [78] is deployed to model the reactive component capacitor and inductor as transmission line stubs, where the characteristic impedance are $Z_C = dt/(2C)$ and $Z_L = (2L)/dt$, respectively, and

equivalent current injection value in the SM-ES are

$$I_{Ceq} = \frac{2v_C^i}{Z_C}, I_{Leq} = \frac{2v_L^i}{Z_L}, \tag{4.17}$$

and $J_s$ is the arm current.

In order to improve the efficiency of circuit simulation, the supercapacitor units are placed in a separate function to calculate the equivalent conductance $G_{es}$ and equivalent current source $I_{es}$ before using them as elements of the matrix for the next step, where

$$G_{es} = \frac{1}{\sum_{i=1}^{N_{sc}} R_{sc}}, \tag{4.18}$$

and

$$I_{es} = G_{es} \sum_{i=1}^{N_{sc}} \left( \frac{r}{\varepsilon} + \frac{2RT}{F} sinh^{-1} \left( \frac{Q_T}{\sqrt{8RT\varepsilon c}} \right) \right). \tag{4.19}$$

The $\mathbf{G_{SM}}$ and $\mathbf{J_{SM}}$, along with the input current source $J_s$, are involved in the circuit solution. The solution of (4.14) is iterative because of the diode nonlinearity, with the history terms not updated until the solution converges.

### 4.3.2 MMC Constant Admittance Circuit

Following the splitting of submodules, each arm in the MMC main circuit only consists of cascaded voltage sources $v_{pi}$ ($i$=1 to $N$) along with an inductor, which takes the form of a Thévenin equivalent circuit and therefore, can be transformed into its Norton counterpart.

The arm voltage could be derived as

$$\begin{aligned} v_{arm}(t) = &\left( \sum_{i=1}^{N} v_{pi}(t - \Delta t) + 2v_{L_{u/d}}^i(t) \right) \\ &+ (Z_{L_{u/d}} + R_{arm}) i_{arm}(t), \end{aligned} \tag{4.20}$$

where $vi_{L_{u/d}}$ and $Z_{L_{u/d}}$ are the incident pulse and impedance of the inductor on the bridge arm as the TLM stub model, respectively, and $R_{arm}$ is the parasitic resistance of the inductor.

The equivalent conductance and companion current of an arm can be expressed as follows,

$$G_{eq} = \frac{1}{Z_{L_{u/d}} + R_{arm}}, \tag{4.21}$$

and

$$I_{eq} = (\sum_{i=1}^{N} v_{pi} + 2v_{L_{u/d}}^i) G_{eq}. \tag{4.22}$$

Depending on the role of the MMC, its AC side is connected to either a distribution grid that has a stiff voltage or a wind farm that is modeled as a current source. The AC side always accounts for 3 nodes irrespective of the converter function. Then, with the transmission line on its DC side, one converter station can be separated from another and a constant admittance matrix with a minimum dimension of 5 is formed. The arm current $I_{arm}$, i.e., the terminal current of a submodule $J_s$, is obtained after solving the corresponding matrix equation of the MMC main circuit and is used for calculating the SM voltages at the next time step.

## 4.4 GPU Parallel Design and Implementation

In this chapter, the NVIDIA® Tesla V100 GPU with 5120 CUDA cores and 16GB HBM2 memory [29] and 20-core Intel® Xeon E5-2698 v4 CPU are adopted for the high-performance computing of the DC grid integrated with wind farms, with a simulation time-step of 2 $\mu$s.

A general CUDA program architecture that contains several stages is shown in Fig. 4.6.

1. Perform data initialization on the CPU termed as the host where global variables are first defined and initialized.
2. Allocate memory for the GPU device to which data from the host are copied via PCI-Express (PCIe).
3. Invoke kernels to perform operations on the device where the time-domain simulation is conducted.
4. Copy the results to be analyzed from the device to the host.
5. Free the allocated memory.

Specifically, in the SM-ES kernel, the IGBT model is programmed as a device function that is called four times, and its outputs are involved in forming (4.14), which is solved after Newton-Raphson iterations to determine whether $\mathbf{v_{sm}}$ converges. If the result converges, the simulation proceeds to the next time step. Otherwise, the iterations will be repeated.

When a GPU kernel is invoked, it automatically launches a few blocks, with each having an identical number of threads that are specified in the CUDA C++ command. For example, as depicted in Fig. 4.6, the SC kernel invokes a total number of $x \times y$ threads, each corresponding to a physical component, i.e., a supercapacitor. The block number $x$

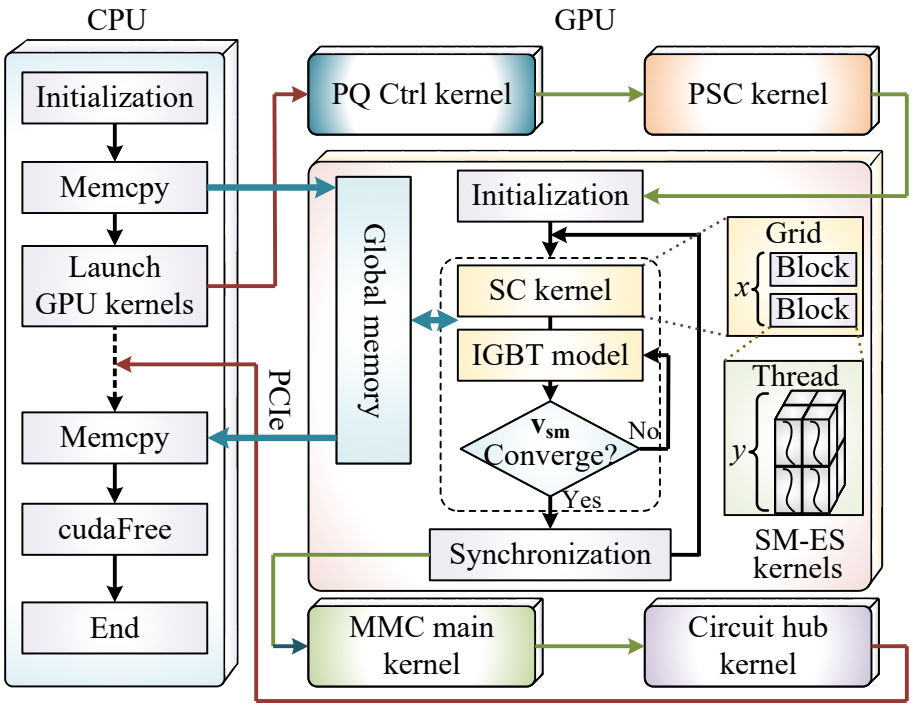



Figure 4.6: GPU simulation process flow chart.

and the thread quantity per block $y$ are determined based on the actual number of components in the entire DC system.

Fig. 4.7 shows massively parallel implementation of various kernels that compose the 4-terminal DC grid integrated with wind farms. All the variables exchanged between kernels are defined and stored in the global memory of the GPU device. Therefore, a global variable is accessible by an arbitrary thread and can also be exported conveniently to the host for further analysis and data processing.

The supercapacitor kernel $SC$ is responsible for calculating the impedance and output voltage of all supercapacitors. Then, the equivalent conductance and companion current composing the Norton circuit of a supercapacitor array in an SM are derived by another kernel $SCsum$. It is noticed that the former kernel has more threads than the latter and their exact numbers could be respectively expressed as

$$N_{SC}^{T} = \frac{N_{stn}}{2} \times 6N \times N_{sc}, \tag{4.23}$$

$$N_{SCsum}^{T} = \frac{N_{stn}}{2} \times 6N, \tag{4.24}$$

where $N_{stn}$ denotes the station number, which is 4 in this chapter. Once the $SC$ kernel completes the computation, its outputs $R_{sc}$ and $V_{sc}$, both of which are $N_{SC}^{T}$ dimensional, are assigned to $N_{SCsum}^{T}$ groups, in each of which $N_{sc}$ elements are summed up.

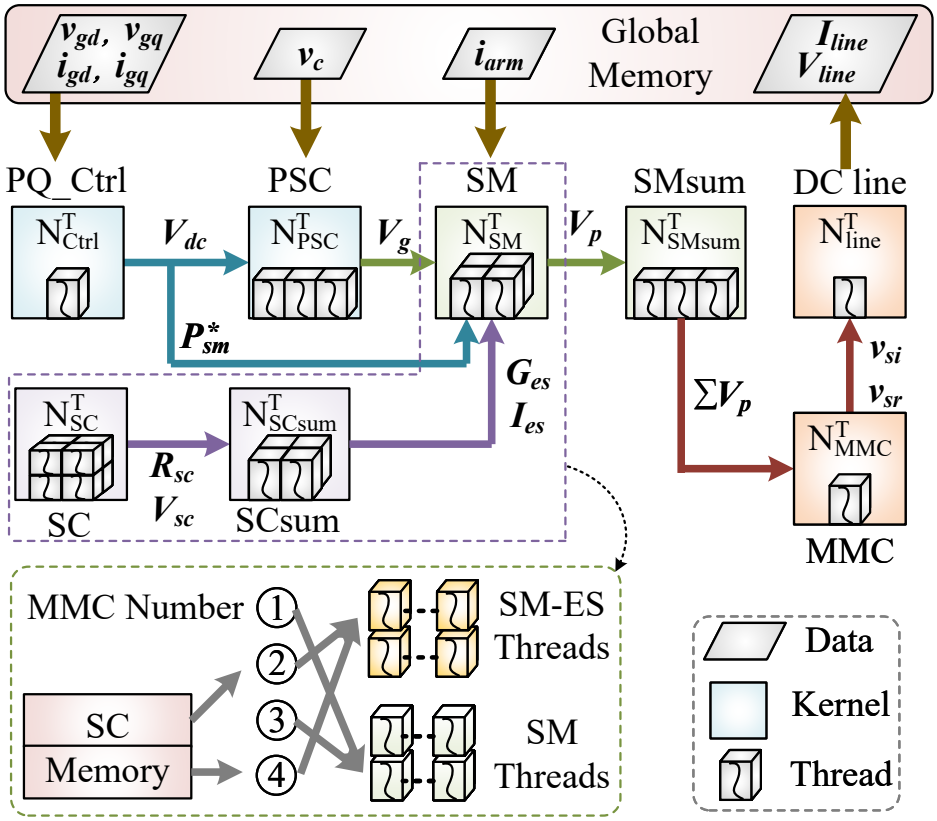



Figure 4.7: Overall GPU program architecture for the transient simulation of the MMC-EES based multi-terminal DC grid.

Although the multi-terminal DC grid shown in Fig. 4.3 comprises two types of MMC submodule structures, i.e., the HBSM and the SM-ES, they are written as one $SM$ kernel to improve the parallelism since under this circumstance, all submodules can be implemented concurrently despite the inhomogeneity. The two main differences are: the admittance matrix size of the HBSM is $2 \times 2$ while the SM-ES is $4 \times 4$, which can be solved by invoking the corresponding device function, and the power control strategy for the energy storage module is distinguished by the specific thread ID. Since not every $SM$ kernel in the GPU implementation needs the output from the $SC$ kernel, memory needs to be allocated reasonably during the GPU kernel programming. As shown at the bottom of Fig. 4.7, in the 4-terminal system, MMCs numbered 2 and 4 have embedded energy storage. Therefore, in the $SM$ kernel, the SC memory address needs to be biased accordingly by the thread ID. The number of threads $N_{SM}^{T}$ is the same as in (4.24).

In the meantime, the application of circuit partitioning results in identical MMC main circuits regardless of the roles these converters play in the DC grid because both the three-

phase voltage and current sources can be represented by Norton equivalent circuits and all the legs are structurally identical. The AC side can be appropriately differentiated between wind farms and grids by their types, and the computation can be implemented in a SIMT manner by the same kernel $MMC$ which has $N_{stn}$ threads.

Since a unified controller is available, the control process of the rectifier and inverter MMCs can also be programmed into the same kernels. The inner loop phase-shift control kernel $PSC$, which is in charge of SM capacitor voltages and does not need to branch off the scheme internally, has a thread quantity $N_{PSC}^{T}$ equal to $3N_{stn}$. In the outer-loop control kernel $PQ\_Ctrl$, the MMC-EES is distinguished from a conventional MMC by its type to maximize the efficiency of GPU implementation with a thread number of $N_{stn}$.

## 4.5 Results and Validation

The voltage level of a grid-connected MMC should be sufficiently high and therefore, various levels are simulated and the results of a 51-level MMC are provided. The simulation results are compared with the PSCAD™/EMTDC™ results in a number of cases to verify the accuracy and the GPU simulation speed is compared with the simulation on the CPU.

The GPU implementation results with the energy storage units being discharged are shown in Fig. 4.8 and Fig. 4.9. Fig. 4.8(a) shows the power of the wind farm, MMC-EES, grid, and transmission line, respectively. When $t$ is 1.0 s, the output power of the wind farm gradually decreases from 160 MW to approximately 100 MW to simulate the situation where the wind speed slows down. In this case, the power on the DC transmission line $TL_1$ changes in the same trend as the wind farm, and the power provided by MMC-EES increases from 140 MW to 200 MW so that the power on the grid side is able to remain stable at 300 MW. Even during the 0.4 s period when the wind speed is quickly reducing, the distribution grid is still provided a nearly constant 300 MW power attributing to a fast converter response.

The DC voltages of both the inverter and rectifier are presented in Fig. 4.8(b), where $V_{dc1}$ is the rectifier side voltage while $V_{dc2}$ is from the inverter side. $V_{dc1}$ drops from about 206 kV to 204 kV between 1.0 s and 1.4 s, while the grid side MMC DC voltage $V_{dc2}$ maintains at 200 kV because of its designed function. The voltage difference $\Delta V$ between the two sides is reduced from about 6 kV to 4 kV due to the power reduction of the wind farm. In Fig. 4.8(c), the 3-phase PCC voltage of the offshore wind farm is depicted, and the maximum value of the voltage is maintained exactly at the expected 110 kV.

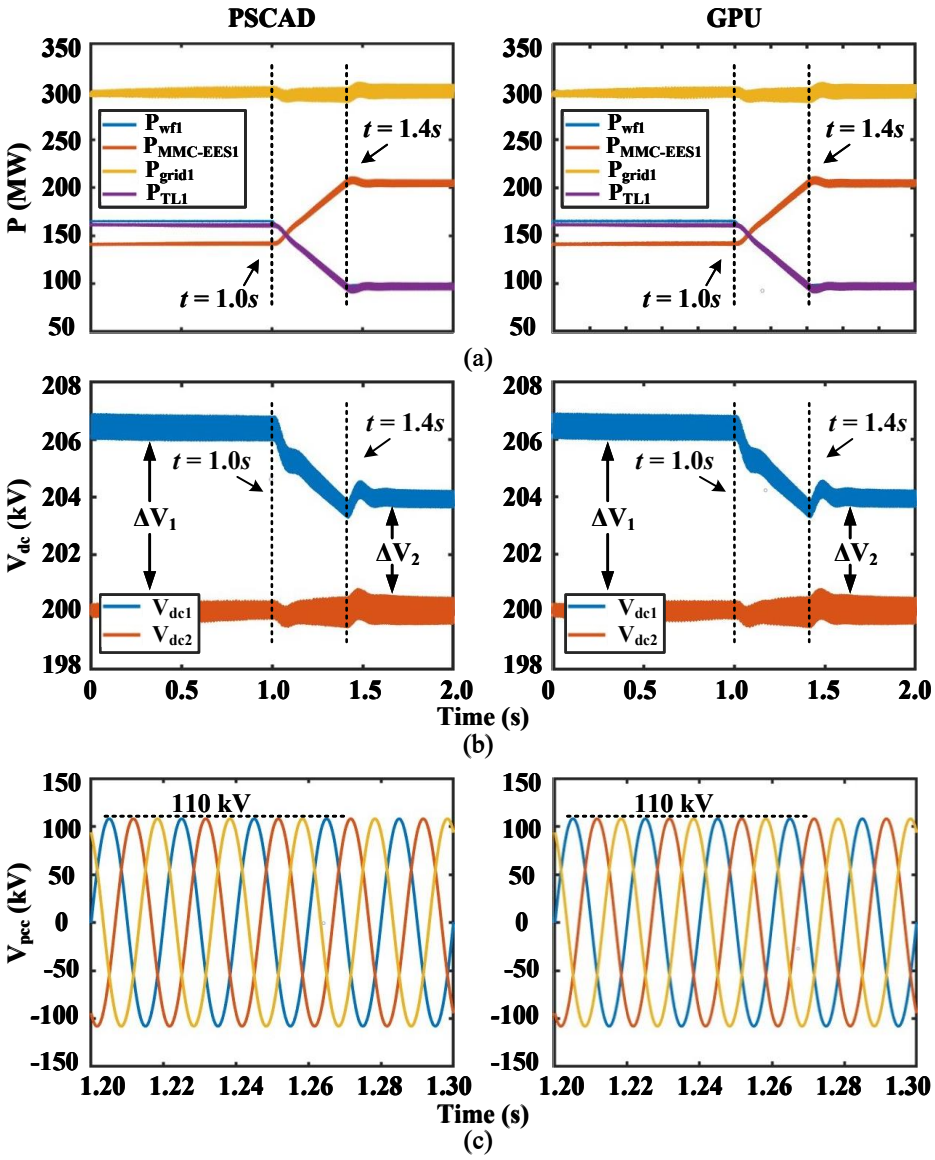



Figure 4.8: PSCAD/EMTDC and CPU simulation results of discharging mode: (a) Power of wind farm, MMC-EES, grid and transmission line; (b) DC voltages; (c) wind farm PCC voltage.

Fig. 4.9(a) depicts the voltage of the capacitor $V_{Csm}$ and the voltage of the supercapacitor array $V_{ES}$ in one of the SM-ES for 3 different arms. $V_{Csm}$ increases temporarily from around 3.9 kV to 4.0 kV during the dynamic period, while the voltage of the energy storage module $V_{ES}$ gradually decreases from about 3.18 kV to 3.16 kV as a result of discharge. As shown in Fig. 4.9(b), the current of the inductor in an arbitrary SM-ES maintains CCM. As the MMC-EES provides more energy to the distribution grid, the current $I_L$ increases from 0.148 kA to 0.218 kA between 1.0 s and 1.4 s, and the ripple current $\Delta I_L$ is about 0.007

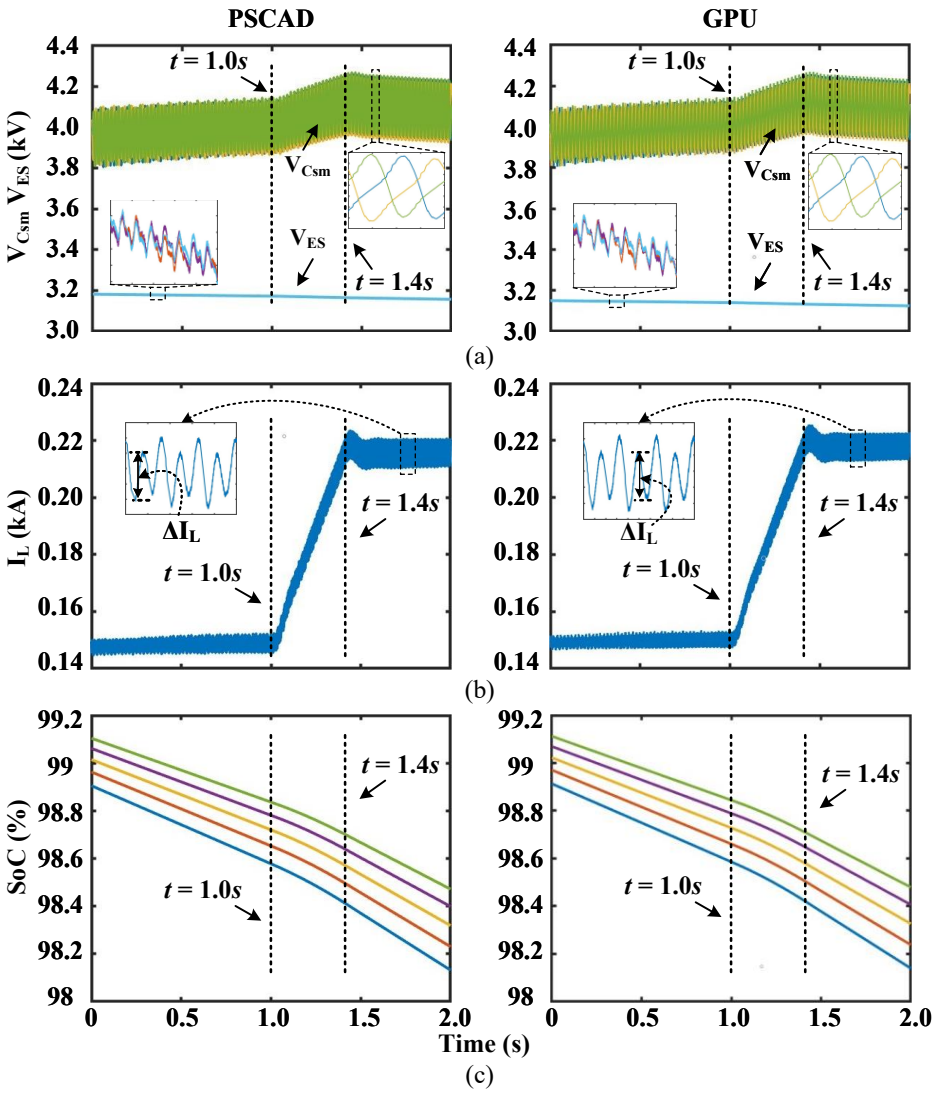



Figure 4.9: PSCAD/EMTDC and CPU simulation results of discharging mode: (a) Voltages of capacitor and supercapacitor in SM-ES; (b) DC-DC converter inductor current; (c) SoC of the supercapacitors.

kA. Fig. 4.9(c) illustrates the SoC of five supercapacitors with different initial voltages in an SM-ES. The high-fidelity modeling of each individual supercapacitor can provide more details of the behavior of all the supercapacitor components in the system, enabling improved monitoring and energy management. As can be observed, the supercapacitors discharge faster after $t$=1.0 s, so the slope becomes steeper and the overall SoC decreases from 99% to 98.4%.

As a common operation scenario of the system, Fig. 4.10 and Fig. 4.11 demonstrate the transition between the two states of charging and discharging. In Fig. 4.10(a), the initial power of the wind farm is still 160 MW, and at 0.4 s, $P_{wf1}$ starts to progressively increase,

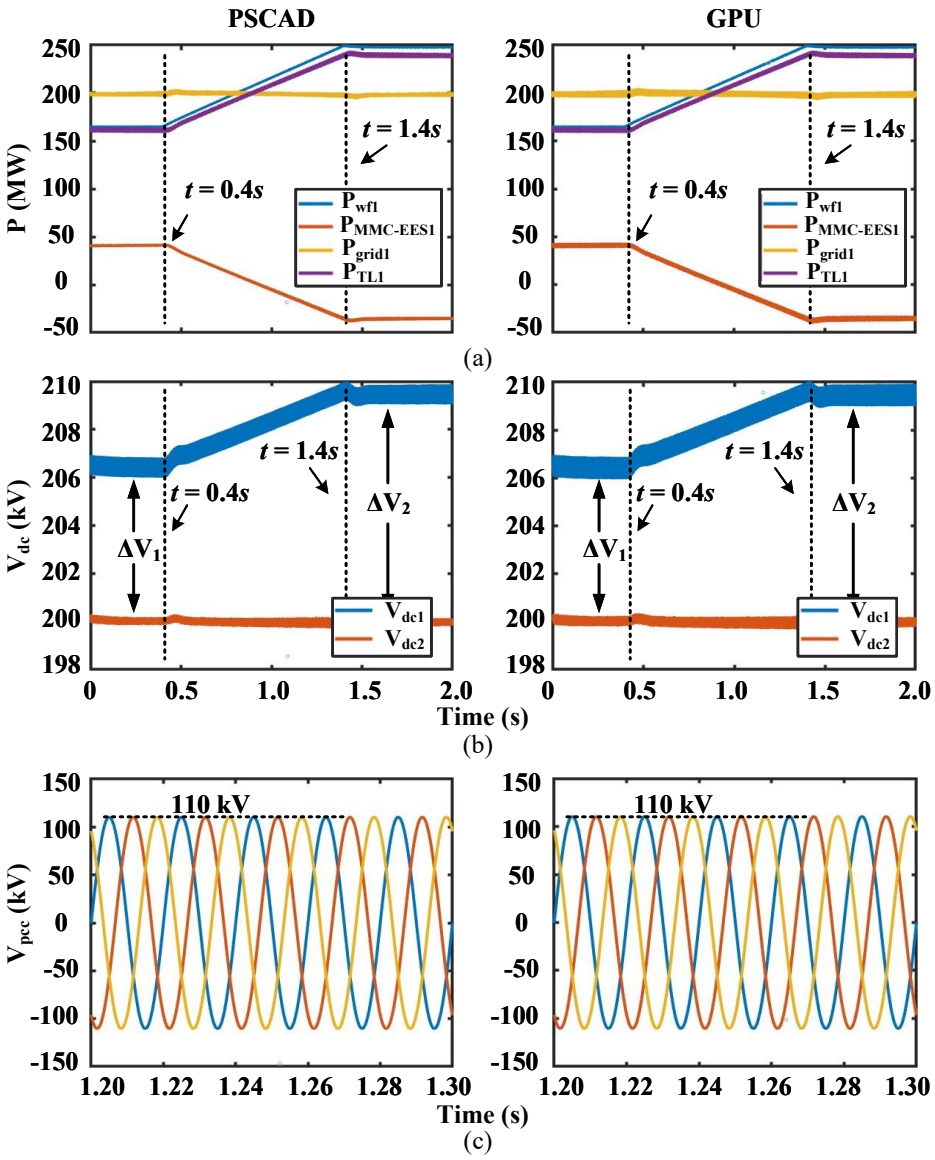



Figure 4.10: GPU and PSCAD/EMTDC simulation results of mode transition: (a) Power of wind farm, MMC-EES and grid; (b) DC voltages; (c) PCC voltage of wind farm.

reaching around 250 MW at 1.4 s. Since the grid-side reference power is set to 200 MW, it can be seen that the output power of MMC-EES1 $P_{MMC-EES1}$ decreased from its initial value of 40 MW to -50 MW at $t$=1.4 s due to the energy storage system transitioning from discharge mode to charge mode to store the additional amount of energy. Throughout the process, $P_{grid1}$ maintains a stable power level, proving the satisfactory performance of the system in both dynamic and steady-state conditions. It can be seen from Fig. 4.10(b) that $V_{dc1}$ keeps increasing from 206 kV at 0.4 s to about 209 kV at 1.4 s. The inverter side DC voltage $V_{dc2}$ stabilizes at 200 kV as a result of DC voltage control of MMC-ES. The DC

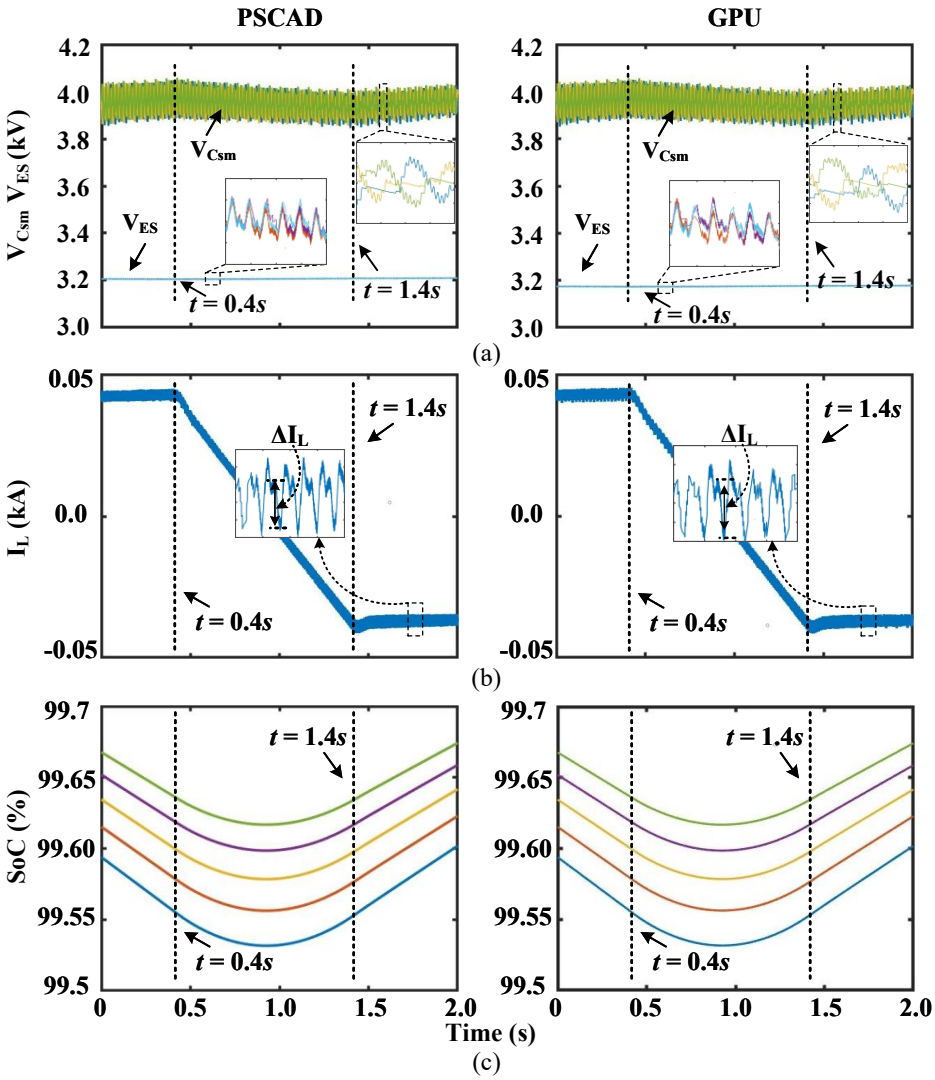



Figure 4.11: GPU and PSCAD/EMTDC simulation results of mode transition: (a) Voltages of capacitor and supercapacitor in SM-ES; (b) DC-DC converter inductor current; (c) SoC of the supercapacitors.

voltage on the rectifier side becomes larger at 1.4 seconds, i.e., $\Delta V_2$ is almost 4 kV greater than $\Delta V_1$. The wind farm PCC voltages are provided in Fig. 4.10(c) with the peak value remaining at 110 kV throughout.

The $V_{Csm}$ and $V_{ES}$ in three different SM-ESs are presented in Fig. 4.11(a). $V_{Csm}$ drops slightly until 1.4 s and then rises afterward, and its value keeps around 4.0 kV. $V_{ES}$ decreases briefly until 0.4 s and gradually increases between 0.4 and 1.4 s, with a slower rate of increase after 1.4 s. In Fig. 4.11(b), the variation of $I_L$ is shown. Before $t$= 0.4 s, it is greater than 0, at approximately 0.04 kA, because the supercapacitors are being discharged at this time. Later on, $I_L$ goes from positive to negative and remains at -0.04 kA at 1.4 s,

Table 4.1: Four-terminal DC system simulation speed comparison

| MMC | $N_{sc} = 2$ | | | Speedup | | $N_{sc} = 20$ | | | Speedup | |
|---|---|---|---|---|---|---|---|---|---|---|
| Level | $t_{CPU1}$(s) | $t_{CPU2}$(s) | $t_{GPU}$(s) | $S_1$ | $S_2$ | $t_{CPU1}$(s) | $t_{CPU2}$(s) | $t_{GPU}$(s) | $S_1$ | $S_2$ |
| 5 | 17.48 | 12.72 | 45.12 | 1.4 | 0.4 | 43.84 | 33.79 | 49.03 | 1.3 | 0.9 |
| 51 | 202.38 | 424.05 | 58.54 | 0.5 | 3.5 | 575.38 | 449.68 | 63.39 | 1.3 | 9.1 |
| 101 | 394.52 | 468.64 | 67.98 | 0.8 | 5.8 | 1126.73 | 517.63 | 75.83 | 2.2 | 14.9 |
| 201 | 743.16 | 567.49 | 89.23 | 1.3 | 8.3 | 2221.2 | 676.23 | 111.95 | 3.3 | 19.8 |
| 401 | 1522.66 | 705.96 | 143.3 | 2.2 | 10.6 | 4486.68 | 1052.73 | 196.82 | 4.3 | 22.8 |

indicating that the supercapacitor is under the charging state. The current ripple $\Delta I_L$ is about merely 0.004 kA which guarantees the CCM. Fig. 4.11(c) shows the trend of the SoC of supercapacitors with several various initial conditions. The SoC decreases steadily from the beginning to about 1.0 s and then starts to rise, indicating the transition of the energy storage system from a discharging state to a charging state.

In Table 4.1, the execution times of GPU, single-core CPU and multi-core CPU are compared for different levels of MMC systems. The speedups $S_1$ and $S_2$ are calculated as the ratios of single-core CPU simulation time to the multi-core CPU and GPU simulation times, respectively. Additionally, the speedup results are analyzed for two different amounts of supercapacitors. For 2 supercapacitors in each SM-ES of a 401-level MMC, $S_1$ is 2.2 and $S_2$ is 10.6, while for 20 supercapacitors in the same case, $S_2$ exceeds 20 and $S_1$ is 4.3. Due to the optimization algorithm, speedups are obtained by both multi-core CPU and GPU. It can be seen that the GPU as a parallel acceleration platform achieves an overall faster simulation speed than the CPU, and a higher speedup is gained with either a higher MMC level or more supercapacitor components in an array.

## 4.6 Summary

This chapter presented the parallel high-performance electromagnetic transient simulation of MMC with embedded energy storage system for wind energy grid integration. By absorbing or releasing an appropriate amount of power, the MMC which has embedded energy storage in its sub-modules reduces the risk of grid stability arising from stochastic wind power generation. Detailed modeling and control strategy design of the MMC-EES is carried out and applied to a four-terminal HVDC system. The modeling approach as detailed as individual supercapacitors allows their behavior to be monitored, thus providing more accurate information from system simulation for evaluation and energy man-

agement. As the high fidelity induced a remarkable computational burden to sequential processing on the CPU, the massively parallel computing advantage of GPU is exploited. Structures with homogeneity are designed and programmed into a single kernel, and manipulation of inhomogeneity is investigated to obtain a more significant acceleration. Different operation scenarios were performed to demonstrate the promising characteristics of the MMC-EES-based HVDC system from both dynamic and static perspectives. The accuracy of the implementation as well as the computational advantages are verified by comparing the results with off-line simulations on the CPU.

# 5 Conclusions and Future Work

With the growing complexity of modern power systems, there is an increasing demand for high-fidelity real-time power system simulations. To meet this demand, EMT simulation has emerged as a powerful tool for studying the dynamic behavior of power systems under various operating conditions. EMT simulation enables complex dynamic phenomena to be studied such as switching transients and faults in power systems. The ability to accurately simulate these transients is critical for designing and operating power systems that are reliable, efficient, and safe.

In power electronic converter simulation, there are different emphases between device-level and system-level simulations. Device-level models can provide a deep understanding of device behavior in transient states, but they come with a high computational cost due to the complexity of the system models. Machine learning techniques can be used to train the device-level models and implement them on the AI Engine of the ACAP platform, which can greatly improve simulation efficiency and meet real-time requirements. For system-level simulation, the high-level embedded energy storage MMC with multiple identical modules makes is well-suited for large-scale parallel implementation. Compared to single-core and multi-core CPUs, the multi-core GPU architecture brings unique computational capabilities for solving large-scale thread parallel problems while addressing resource limitations.

This thesis introduces an innovative real-time simulation of the IGBT nonlinear behavioral electro-thermal model and the corresponding ANN model combined with ma-

chine learning methodology on the ACAP platform's AI Engine. Furthermore, due to the resource-intensive nature of high-level MMC with embedded energy storage simulation, it is simulated on the GPU platform with massive parallelism, and the accuracy is validated. This chapter presents the contributions of this thesis and suggestions for future work.

## 5.1 Contributions

The main contributions of this thesis are summarized below:

- A comprehensive study of the cutting-edge ACAP, the high-performance GPU, as well as their architectures and programming methods are conducted. The application of ACAP and GPU as hardware platforms in the simulation of power electronic devices and systems expands the scope and complexity of power simulation, and provides a reference for exploring the potential of these platforms in power electronics and power systems research.

- Introduces a nonlinear behavioral electro-thermal model for IGBT, and describes its implementation in the processing system, programmable logic, and AI Engine domains on the innovative ACAP, respectively, as well as a comparison of simulation time and resource consumption. Machine learning techniques are adopted to train the IGBT ANN model and then implement the model in the vector unit of the AIE, which significantly improves the accuracy and efficiency of IGBT simulation. The two-level VSC converter was chosen as a case study, and the requirement of real-time simulation is met.

- The topology and control scheme of MMC with embedded energy storage for wind farm grid integration are introduced in this thesis. The corresponding EMT model is developed and the parallel implementation of the model on GPU is described. The simulation results and the effectiveness of energy storage modules are verified in a multi-terminal wind farm HVDC system. This modeling approach can diminish the stress caused by resource limitations and also enable more accurate and efficient simulations of MMC-based systems.

## 5.2 Future work

Several potential future research directions could be pursued:

- In this thesis, the real-time simulation of a 2-level VSC converter has been implemented. Future research could consider other complex power system structures as case studies, such as the MMC on the ACAP architecture.
- This thesis shows how machine learning techniques can be used in conjunction with the IGBT model to accelerate the simulation while maintaining accuracy. Future research could explore other ways in which machine learning could be integrated with power electronics, such as for control and fault detection.
- A modeling approach for MMC with embedded energy storage is presented and verified by simulating an offshore wind farm system on a GPU platform. Future research could focus on exploring its real-time simulation and hardware-in-the-loop testing to improve accuracy and reliability.
- As the energy storage devices in MMC-EES, supercapacitors can deliver and absorb charge quickly, making them suitable for high-power applications. However, they have lower energy density than batteries, so they are typically used in conjunction with batteries in hybrid energy storage systems. The combination of the two allows for higher energy and power densities than could be achieved with either technology alone.

# Bibliography


[1] L. Han, L. Liang, Y. Kang, and Y. Qiu, "A review of sic IGBT: Models, fabrications, characteristics, and applications," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 2080–2093, Feb. 2021.

[2] X. Li, D. Li, G. Chang, W. Gong, M. Packwood, D. Pottage, Y. Wang, H. Luo, and G. Liu, "High-voltage hybrid IGBT power modules for miniaturization of rolling stock traction inverters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1266–1275, Feb. 2022.

[3] P. Bakas, Y. Okazaki, A. Shukla, S. K. Patro, K. Ilves, F. Dijkhuizen, and A. Nami, "Review of hybrid multilevel converter topologies utilizing thyristors for HVDC applications," *IEEE Trans. Power Electron.*, vol. 36, no. 1, pp. 174–190, Jan. 2021.

[4] C. Dufour, J. Mahseredjian, and J. Bélanger, "A combined state-space nodal method for the simulation of power system transients," *IEEE Trans. Power Deliv.*, vol. 26, no. 2, pp. 928–935, Apr. 2011.

[5] K. Wang, J. Xu, G. Li, N. Tai, A. Tong, and J. Hou, "A generalized associated discrete circuit model of power converters in real-time simulation," *IEEE Trans. Power Electron.*, vol. 34, no. 3, pp. 2220–2233, Mar. 2019.

[6] H. J. Bahirat, H. K. Høidalen, and B. A. Mork, "Thévenin equivalent of voltage-source converters for dc fault studies," *IEEE Trans. Power Deliv.*, vol. 31, no. 2, pp. 503–512, Apr. 2016.

[7] S. Perez, R. M. Kotecha, A. U. Rashid, M. M. Hossain, T. Vrotsos, A. M. Francis, H. A. Mantooth, E. Santi, and J. L. Hudgins, "A datasheet driven unified Si/SiC compact IGBT model for N-channel and P-channel devices," *IEEE Trans. Power Electron.*, vol. 34, no. 9, pp. 8329–8341, Sep. 2019.

[8] W. Wang, Z. Shen, and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA," *IEEE Trans. Industr. Inform.*, vol. 10, no. 4, pp. 2166–2179, Nov. 2014.

[9] P. Liu, N. Lin, and V. Dinavahi, "Integrated massively parallel simulation of thermo-electromagnetic fields and transients of converter transformer interacting with mmc in multi-terminal dc grid," *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 3, pp. 725–735, Jun. 2020.

[10] N. Lin and V. Dinavahi, "Behavioral device-level modeling of modular multilevel converters in real time for variable-speed drive applications," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 5, no. 3, pp. 1177–1191, Sep. 2017.

[11] Z. Li, Y. Gao, X. Zhang, B. Wang, and H. Ma, "A model-data-hybrid-driven diagnosis method for open-switch faults in power converters," *IEEE Trans. Power Electron.*, vol. 36, no. 5, pp. 4965–4970, May. 2021.

[12] D. Wang, Z. J. Shen, X. Yin, S. Tang, X. Liu, C. Zhang, J. Wang, J. Rodriguez, and M. Norambuena, "Model predictive control using artificial neural network for power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3689–3699, Apr. 2022.

[13] M. Ali, M. Tariq, K. A. Lodi, R. K. Chakrabortty, M. J. Ryan, B. Alamri, and C. Bharatiraja, "Robust ANN-based control of modified PUC-5 inverter for solar PV applications," *IEEE Trans. Ind. Appl.*, vol. 57, no. 4, pp. 3863–3876, July-Aug. 2021.

[14] S. Wang, T. Dragicevic, G. F. Gontijo, S. K. Chaudhary, and R. Teodorescu, "Machine learning emulation of model predictive control for modular multilevel converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11 628–11 634, Nov. 2021.

[15] S. Cao, N. Lin, and V. Dinavahi, "Mitigation of subsynchronous interactions in hybrid ac/dc grid with renewable energy using faster-than-real-time dynamic simulation," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 670–679, Jan. 2021.

[16] S. Debnath, J. Qin, B. Bahrani, M. Saeedifard, and P. Barbosa, "Operation, control, and applications of the modular multilevel converter: A review," *IEEE Trans. Power Electron.*, vol. 30, no. 1, pp. 37–53, Jan. 2015.

[17] C. Wang, J. Xu, X. Pan, W. Gong, Z. Zhu, and S. Xu, "Impedance modeling and analysis of series-connected modular multilevel converter (mmc) and its compara-

tive study with conventional MMC for HVDC applications," *IEEE Trans. Power Deliv.*, vol. 37, no. 4, pp. 3270–3281, Aug. 2022.

[18] M. Asoodar, M. Nahalparvari, C. Danielsson, R. Söderström, and H.-P. Nee, "Online health monitoring of DC-link capacitors in modular multilevel converters for FACTS and HVDC applications," *IEEE Trans. Power Electron.*, vol. 36, no. 12, pp. 13 489–13 503, Dec. 2021.

[19] S. Wu, X. Zhang, W. Jia, Y. Zhu, L. Qi, X. Guo, and X. Pan, "A modular multilevel converter with integrated energy dissipation equipment for offshore wind VSC-HVDC system," *IEEE Trans. Sustain. Energy.*, vol. 13, no. 1, pp. 353–362, Jan. 2022.

[20] X. Pan, S. Pan, P. Jin, L. Yao, J. Gong, F. Liu, and X. Zha, "Decoupling capacitor minimization of an MMC-based photovoltaic system with three-winding power channel," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 1012–1026, Jan. 2022.

[21] A. Christe, A. Faulstich, M. Vasiladiotis, and P. Steinmann, "World's first fully rated direct ac/ac MMC for variable-speed pumped-storage hydropower plants," *IEEE Trans. Ind. Electron.*, vol. 70, no. 7, pp. 6898–6907, Jul. 2023.

[22] Z. Blatsi, P. D. Judge, S. J. Finney, and M. M. C. Merlin, "Blackstart capability of modular multilevel converters from partially-rated integrated energy storage," *IEEE Trans. Power Deliv.*, vol. 38, no. 1, pp. 268–276, Feb. 2023.

[23] W. Zeng, R. Li, L. Huang, C. Liu, and X. Cai, "Approach to inertial compensation of hvdc offshore wind farms by mmc with ultracapacitor energy storage integration," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 12 988–12 998, Dec. 2022.

[24] N. Lin and V. Dinavahi, "Parallel high-fidelity electromagnetic transient simulation of large-scale multi-terminal dc grids," *IEEE Power and Energy Technology Systems Journal*, vol. 6, no. 1, pp. 59–70, Mar. 2019.

[25] N. Lin and V. Dinavahi, "Variable time-stepping modular multilevel converter model for fast and parallel transient simulation of multiterminal dc grid," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 6661–6670, Sep. 2019.

[26] Xilinx. Inc. *Versal ACAP technical reference manual.* (2022, Apr.). [Online]. Available: `https://docs.xilinx.com/r/en-US/am011-versal-acap-trm`

[27] Xilinx. Inc. *AI engine intrinsics.* (2021). [Online]. Available: https://www.xilinx.com/htmldocs/xilinx2021_2/aiengine_intrinsics/intrinsics/index.html

[28] Xilinx. Inc. *AI Engine Kernel Coding.* (2021, July.). [Online]. Available: https://docs.xilinx.com/r/2020.2-English/ug1079-ai-engine-kernel-coding/Single-Kernel-Programming

[29] NVIDIA Corp., "Whitepaper NVIDIA Tesla V100 GPU architecture," *Santa Clara, CA, USA*, Aug. 2017.

[30] D. Ronanki and S. S. Williamson, "Modular multilevel converters for transportation electrification: Challenges and opportunities," *IEEE Trans. Transport. Electrific.*, vol. 4, no. 2, pp. 399–407, Jan. 2018.

[31] Q. Sun, J. Wu, C. Gan, J. Si, J. Guo, and Y. Hu, "Cascaded multiport converter for SRM-Based hybrid electrical vehicle applications," *IEEE Trans. Power Electron.*, vol. 34, no. 12, pp. 11 940–11 951, Apr. 2019.

[32] A. Francés-Roger, A. Anvari-Moghaddam, E. Rodríguez-Díaz, J. C. Vasquez, J. M. Guerrero, and J. Uceda, "Dynamic assessment of cots converters-based dc integrated power systems in electric ships," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5518–5529, Feb. 2018.

[33] S. Horiuchi, K. Sano, and T. Noda, "An inverter model simulating accurate harmonics with low computational burden for electromagnetic transient simulations," *IEEE Trans. Power Electron.*, vol. 36, no. 5, pp. 5389–5397, May. 2021.

[34] A. Hadizadeh, M. Hashemi, M. Labbaf, and M. Parniani, "A matrix-inversion technique for FPGA-Based real-time EMT simulation of power converters," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1224–1234, Feb. 2019.

[35] X. Meng, J. Han, J. Pfannschmidt, L. Wang, W. Li, F. Zhang, and J. Belanger, "Combining detailed equivalent model with switching-function-based average value model for fast and accurate simulation of MMCs," *IEEE Trans. Energy Convers.*, vol. 35, no. 1, pp. 484–496, Mar. 2020.

[36] N. Lin and V. Dinavahi, "Detailed device-level electrothermal modeling of the proactive hybrid hvdc breaker for real-time hardware-in-the-loop simulation of dc grids," *IEEE Trans. Power Electron.*, vol. 33, no. 2, pp. 1118–1134, Mar. 2018.

[37] H. Bai, C. Liu, E. Breaz, K. Al-Haddad, and F. Gao, "A review on the device-level real-time simulation of power electronic converters: Motivations for improving performance," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 12–27, Mar. 2021.

[38] L. Han, L. Liang, Y. Kang, and Y. Qiu, "A review of SiC IGBT: Models, fabrications, characteristics, and applications," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 2080–2093, Feb. 2021.

[39] K. Sheng, B. Williams, and S. Finney, "A review of IGBT models," *IEEE Trans. Power Electron.*, vol. 15, no. 6, pp. 1250–1266, Nov. 2000.

[40] N. Lin, B. Shi, and V. Dinavahi, "Non-linear behavioural modelling of device-level transients for complex power electronic converter circuit hardware realisation on FPGA," *IET Power Electron.*, vol. 11, no. 9, pp. 1566–1574, Jun. 2018.

[41] H. Bai, C. Liu, R. Ma, D. Paire, and F. Gao, "Device-level modelling and FPGA-based real-time simulation of the power electronic system in fuel cell electric vehicle," *IET Power Electronics*, vol. 12, no. 13, pp. 3479–3487, Nov. 2019.

[42] C. Lyu, N. Lin, and V. Dinavahi, "Device-level parallel-in-time simulation of mmc-based energy system for electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5669–5678, Jun. 2021.

[43] V. Dinavahi and N. Lin, *Real-Time Electromagnetic Transient Simulation of AC-DC Networks*. Wiley-IEEE Press, Jun. 2021.

[44] O. A. Alimi, K. Ouahada, and A. M. Abu-Mahfouz, "A review of machine learning approaches to power system security and stability," *IEEE Access*, vol. 8, pp. 113 512–113 531, Jun. 2020.

[45] G. Rojas-Dueñas, J.-R. Riba, and M. Moreno-Eguilaz, "A deep learning-based modeling of a 270 V-to-28 V DC-DC converter used in more electric aircrafts," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 509–518, Jan. 2022.

[46] F. Zhang, Q. Liu, Y. Liu, N. Tong, S. Chen, and C. Zhang, "Novel fault location method for power systems based on attention mechanism and double structure GRU neural network," *IEEE Access*, vol. 8, pp. 75 237–75 248, 2020.

[47] X. Fu, S. Li, and I. Jaithwa, "Implement optimal vector control for LCL-Filter-Based Grid-Connected converters by using recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4443–4454, Jul. 2015.

[48] S. Zhang, T. Liang, and V. Dinavahi, "Machine learning building blocks for real-time emulation of advanced transport power systems," *IEEE Open J. Power Electron.*, vol. 1, pp. 488–498, Nov. 2020.

[49] Xilinx. Inc. *Versal: The first adaptive compute acceleration platform (ACAP).* (2020, Sep.). [Online]. Available: `https://docs.xilinx.com/v/u/en-US/wp505-versal-acap`

[50] A. Courtay, "MAST power diode and thyristor models including automatic parameter extraction," *SABER User Group Meeting. Brighton, U.K.*, pp. 1–10, Sep. 1995.

[51] R. Wu, H. Wang, K. B. Pedersen, K. Ma, P. Ghimire, F. Iannuzzo, and F. Blaabjerg, "A temperature-dependent thermal model of IGBT modules suitable for circuit-level simulations," *IEEE Trans. Ind. Appl.*, vol. 52, no. 4, pp. 3306–3314, July-Aug 2016.

[52] J. Pomerat, A. Segev, and R. Datta, "On neural network activation functions and optimizers in relation to polynomial regression," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 6183–6185, Dec. 2019.

[53] R. Zaheer and H. Shaziya, "A study of the optimization algorithms in deep learning," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 536–539, Jan. 2019.

[54] F. Blaabjerg and K. Ma, "Future on power electronics for wind turbine systems," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 1, no. 3, pp. 139–152, Sep. 2013.

[55] X. Han, Y. Qu, P. Wang, and J. Yang, "Four-dimensional wind speed model for adequacy assessment of power systems with wind farms," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2978–2985, Aug. 2013.

[56] S. Vazquez, S. M. Lukic, E. Galvan, L. G. Franquelo, and J. M. Carrasco, "Energy storage systems for transport and grid applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 3881–3895, Dec. 2010.

[57] T. Kovaltchouk, A. Blavette, J. Aubry, H. B. Ahmed, and B. Multon, "Comparison between centralized and decentralized storage energy management for direct wave

energy converter farm," *IEEE Trans. Energy Convers.*, vol. 31, no. 3, pp. 1051–1058, Sep. 2016.

[58] P. D. Judge and T. C. Green, "Modular multilevel converter with partially rated integrated energy storage suitable for frequency support and ancillary service provision," *IEEE Trans. Power Deliv.*, vol. 34, no. 1, pp. 208–219, Feb. 2019.

[59] J. M. L. Fonseca, S. R. P. Reddy, K. Rajashekara, and K. R. R. Potti, "Reduced capacitor energy requirements in battery energy storage systems based on modular multilevel converters," *IEEE Trans. Ind. Appl.*, vol. 58, no. 6, Nov.-Dec. 2022.

[60] L. Qu and W. Qiao, "Constant power control of DFIG wind turbines with supercapacitor energy storage," *IEEE Trans. Ind. Appl.*, vol. 47, no. 1, pp. 359–367, Jan.-Feb. 2011.

[61] H. Zhang, F. Zhang, L. Yang, Y. Gao, and B. Jin, "Multi-parameter collaborative power prediction to improve the efficiency of supercapacitor-based regenerative braking system," *IEEE Trans. Energy Convers.*, vol. 36, no. 4, pp. 2612–2622, Dec. 2021.

[62] S. Wang, D. Vozikis, K. H. Ahmed, D. Holliday, and B. W. Williams, "Comprehensive assessment of fault-resilient schemes based on energy storage integrated modular converters for ac-dc conversion systems," *IEEE Trans. Power Deliv.*, vol. 37, no. 3, pp. 1764–1774, Jun. 2022.

[63] H. Zhang, F. Mollet, C. Saudemont, and B. Robyns, "Experimental validation of energy storage system management strategies for a local dc distribution system of more electric aircraft," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 3905–3916, Dec. 2010.

[64] R. Vidal-Albalate, H. Beltran, A. Rolán, E. Belenguer, R. Peña, and R. Blasco-Gimenez, "Analysis of the performance of mmc under fault conditions in hvdc-based offshore wind farms," *IEEE Trans. Power Deliv.*, vol. 31, no. 2, pp. 839–847, Apr. 2016.

[65] G. Guo, H. Wang, Q. Song, J. Zhang, T. Wang, B. Ren, and Z. Wang, "HB and FB mmc based onshore converter in series-connected offshore wind farm," *IEEE Trans. Power Electron.*, vol. 35, no. 3, pp. 2646–2658, Mar. 2020.

[66] N. Herath and S. Filizadeh, "Average-value model for a modular multilevel converter with embedded storage," *IEEE Trans. Energy Convers.*, vol. 36, no. 2, pp. 789–799, Jun. 2021.

[67] L. Zhang, Y. Tang, S. Yang, and F. Gao, "Decoupled power control for a modular-multilevel-converter-based hybrid ac-dc grid integrated with hybrid energy storage," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 2926–2934, Apr. 2019.

[68] N. Herath, S. Filizadeh, and M. S. Toulabi, "Modeling of a modular multilevel converter with embedded energy storage for electromagnetic transient simulations," *IEEE Trans. Energy Convers.*, vol. 34, no. 4, pp. 2096–2105, Dec. 2019.

[69] N. Lin and V. Dinavahi, "Dynamic electro-magnetic-thermal modeling of mmc-based dc-dc converter for real-time simulation of MTDC grid," *IEEE Trans. Power Deliv.*, vol. 33, no. 3, pp. 1337–1347, Jun. 2018.

[70] W. Li, L.-A. Grégoire, and J. Bélanger, "A modular multilevel converter pulse generation and capacitor voltage balance method optimized for FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2859–2867, May. 2015.

[71] V. Dinavahi and N. Lin, *Parallel Dynamic and Transient Simulation of Large-Scale Power Systems*. Springer, Jan. 2022.

[72] N. Lin and V. Dinavahi, "Exact nonlinear micromodeling for fine-grained parallel EMT simulation of MTDC grid interaction with wind farm," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6427–6436, Aug. 2019.

[73] J. Sun, S. Debnath, M. Saeedifard, and P. R. Marthi, "Real-time electromagnetic transient simulation of multi-terminal HVDC-AC grids based on GPU," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7002–7011, Aug. 2021.

[74] J. K. Debnath, A. M. Gole, and W.-K. Fung, "Graphics-processing-unit-based acceleration of electromagnetic transients simulation," *IEEE Trans. Power Deliv.*, vol. 31, no. 5, pp. 2036–2044, Oct. 2016.

[75] H. Akagi, "Multilevel converters: Fundamental circuits and systems," *Proc. IEEE*, vol. 105, no. 11, pp. 2048–2065, Nov. 2017.

[76] K. B. Oldham, "A gouy-chapman-stern model of the double layer at a (metal)/(ionic liquid) interface," *J. Electroanal. Chem.*, vol. 613, no. 2, pp. 131–138, Feb. 2008.

[77] M. Hagiwara and H. Akagi, "Control and experiment of pulsewidth-modulated modular multilevel converters," *IEEE Trans. Power Electron.*, vol. 24, no. 7, pp. 1737–1746, Jul. 2009.

[78] H. Selhi, C. Christopoulos, A. Howe, and S. Hui, "The application of transmission-line modelling to the simulation of an induction motor drive," *IEEE Trans. Energy Convers.*, vol. 11, no. 2, pp. 287–297, Jun. 1996.

# A

# Parameters for Case Studies

## A.1 Parameters of the IGBT in Chapter 3

The parameters of the IGBT Siemens BSM300GA160D, rated 1600V, 300A behavioral model: $V_t$ = 6.3 V, $x$ = 0.974, $y$ = 1.429, $z$ = 0.369, $a_1$ = 0.022, $b_1$ = 0.004, $a_2$ = 92.5129, $b_2$ = 4.0188, $r_{tail}$ = 1 $\mu\Omega$ , $C_{tail}$ = 10 F, $i_{rat}$ = 0.05, $C_{geo}$ = 40 nF, $C_{cgo}$ = 110 nF.

Cooling System 1:

$R_1$ = 2.1 K/kW, $R_2$ = 9.2 K/kW, $R_3$ = 42.6 K/kW, $R_4$ = 6.3 K/kW, $\tau_1$ = 0.0008 s, $\tau_2$ = 0.013 s, $\tau_3$ = 0.05 s, $\tau_4$ = 0.063 s.

Cooling System 2:

$R_1$ = 1.33 K/kW, $R_2$ = 7.05 K/kW, $R_3$ = 5.23 K/kW, $R_4$ = 2.8 K/kW, $\tau_1$ = 0.00147 s, $\tau_2$ = 0.034 s, $\tau_3$ = 0.168 s, $\tau_4$ = 1.11 s.

## A.2 Parameters for Case Study in Chapter 3

The parameters of the case study system:

The grid voltage $V_s$ = 490 $V$ (L-L), 60 $Hz$; the transformer 1MVA, 25 $kV/490$ $V$; $C_{dc}$ = 0.0333 $F$; the half-bridge load 400+j50 $kVA$; the buck load 250 $kW$, duty D = 0.55; the boost supply $V_{boost}$ = 500 $V$, duty D = 0.8; the full-bridge load 200+j50 $kVA$.

## A.3 Parameters for Case Study in Chapter 4

The MMC-EES submodule parameters:
$N_{sc}$ = 2-20, $R_{sc}$ = 2.1 m$\Omega$, $C_{sm}$ = 10mF, $L_{sm}$ = 0.03 H, $V_{dcref}$ = 200 kV, $f_{sm}$ = 1 kHz, $f_{ES}$ = 5 kHz.

The parameters of the 51 to 401-level MMC: $L_{u,d}$ = 50 mH, $L_{dc}$ = 200 mH.