On Computation and Implementation Techniques for Fast and Parallel Electromagnetic Transient Power System Simulation

by

Tong Duan

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

Doctor of Philosophy
in
Energy Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Electromagnetic transient (EMT) simulation is a paramount tool to study the electrical system's behavior and reproduce the transient waveforms prior to manufacturing and deployment. However, the simulation process slows down significantly when the circuit scale expands, and thus the fast and parallel circuit simulation techniques are required to be studied and applied, especially for modern large-scale AC/DC grids where the MMC converters composed of hundreds of submodules generate a large matrix. Besides, the traditional power system is evolving into a complex cyber-physical system (CPS), which also proposes a new challenge to simulate the entire behavior of the system for quickly and adequately evaluating the interplay between digital world and physical appliance.

To conduct fast EMT simulation for large and complex power systems, in this thesis, the existing computation and implementation techniques are investigated and improved, including the the multi-rate (MR) scheme, variable time-stepping (VTS) scheme, domain decomposition (DD) scheme, and hardware based acceleration. 1) For the MR scheme, an extended multi-rate mixed-solver (MRMS) hardware architecture is proposed for real-time EMT emulation of hybrid AC/DC networks, which is an implementation-level work taking advantages of the hybrid FPGA-MPSoC platform to emulate AC/DC systems in real-time while guaranteeing the accuracy and low resource cost. 2) For the VTS scheme, the new mathematical computational processes for the universal line model (ULM) and universal machine (UM) model are proposed, which greatly improve the stability of the models when the time-step changes compared to the traditional ULM and UM model. The faster-than-real-time emulation architecture on FPGA and 4-level parallelism architecture on GPU are also proposed to conduct the VTS-based EMT simulation in parallel. 3) For the DD scheme, a novel linking-domain extraction (LDE) decomposition method is proposed, which is a matrix-based decomposition method and can obtain the general formulation of the inverse of the circuit conductance matrix based on the mathematical analysis. Using the LDE method, a circuit can be simulated in parallel for the decomposed subsystems. To fully exploit the potential of the LDE method, the hierarchical LDE decomposition method is also proposed for further applications. 4) In addition, by leveraging the fast and parallel computing capabilities of FPGA/MPSoC/GPU hardware platforms, the real-time co-emulation hardware architectures of EMT-based power system and communication network are proposed on both the FPGA-MPSoC and Jetson®-FPGA platforms to accelerate the co-simulation process for AC/DC cyber-physical power systems and study the communication-enabled global control schemes.

Although the proposed methods belong to different computation and implementation techniques, the essential goal of those works is the same: conducting fast and parallel EMT simulation to deal with the complexity of large-scale power systems and to significantly accelerate the simulation process. The proposed mathematical models, computational approaches, and implementation architectures contribute to the exiting EMT simulation techniques and have potential to be applied in the future EMT simulation research.

Preface

The material presented in this thesis is based on original work by Tong Duan. As detailed in the following, material from some chapters of this thesis has been published as journal articles under the supervision of Dr. Venkata Dinavahi in concept formation and by providing comments and corrections to the article manuscript.

Chapter 2 includes the results from the following paper:

- T. Duan, Z. Shen, and V. Dinavahi, "Multi-rate mixed-solver for real-time nonlinear electromagnetic transient emulation of AC/DC networks on FPGA-MPSoC architecture," *IEEE Power Energy Technol. Syst. J.*, vol. 6, no. 4, pp. 183-194, Dec. 2019.
- Z. Shen, T. Duan and V. Dinavahi, "Design and implementation of real-time MPSoC-FPGA based electromagnetic transient emulator of CIGRÉ DC grid for HIL application," *IEEE Power Energy Technol. Syst. J.*, vol. 5, no. 3, pp. 104-116, Sept. 2018.

Chapter 3 includes the results from the following paper:

- T. Duan and V. Dinavahi, "Adaptive time-stepping universal line and machine models for real-time and faster-than-real-time hardware emulation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6173-6182, Aug. 2020.
- T. Duan and V. Dinavahi, "Variable time-stepping parallel electromagnetic transient simulation of hybrid AC/DC grids," *IEEE J. Emerg. Sel. Topics. Ind. Electron.*, vol. 2, no. 1, pp. 90-98, Jan. 2021.

Chapter 4 includes the results from the following paper:

- T. Duan and V. Dinavahi, "A novel linking-domain extraction decomposition method for parallel electromagnetic transient simulation of large-scale AC/DC networks," *IEEE Trans. Power Del.*, early-access, pp. 1-9, May 2020.

Chapter 5 includes the results from the following papers:

- T. Duan and V. Dinavahi, “Hierarchical linking-domain extraction decomposition method for fast and parallel power system electromagnetic transient simulation,” *Submitted to IEEE Open J. Ind. Appl.*, pp.1-9, 2020.

Chapter 6 includes the results from the following paper:

- T. Duan, Z. Huang, and V. Dinavahi, “RTCE: real-time co-emulation framework for EMT-based power system and communication network on FPGA-MPSoC hardware architecture,” *IEEE Trans. Smart Grid,* early-access, pp. 1-10, 2020.

Chapter 7 includes the results from the following paper:

- T. Duan, T. Cheng, and V. Dinavahi, “Heterogeneous real-time co-emulation for communication-enabled global control of AC/DC grid integrated with renewable energy,” *IEEE Open J. Ind. Electron. Soc.,* vol. 1, pp. 261-270, Sept. 2020.

To my parents and my wife,
for their unconditional support and love.

Acknowledgements

I would like to express my genuine gratitude to *Dr. Venkata Dinavahi*, who is the supervisor of my Ph.D. program at the University of Alberta. Without his excellent academic proposals and patient thesis supervision, it is hard to imagine that I could overcome so many challenges on the path of this research. His passion, patience and positive attitude also helped me a lot during the research and daily life.

I have spent a fantastic time in the RTX-Lab, which provided wonderful hybrid hardware emulation environment for my research, including the latest FPGA, MPSoC and GPU devices. I would like to thank all the colleagues in our Lab for their kind help and suggestions.

I am sincerely grateful for the unconditional support from my family: my father, Xin Duan, my mother Guoping An, and my wife, Junqi Wang. During the study at the University of Alberta, they gave me a lot of psychological help that made me always feel warm and peaceful.

The research of this thesis was supported by the China Scholarship Council, the University of Alberta, and NSERC. I greatly appreciate their financial support.

Table of Contents

List of Tables

List of Figures

List of Acronyms

APU	Application Processing Unit
ATP	Alternative Transients Program
AVM	Average Value Model
AXI	Advanced eXtensible Interface
BE	Backward Euler
BLM	Bergeron Line Model
BRAM	Block Random Access Memory
CFM	Curve-Fitting Model
CPS	Cyber-Physical System
CPU	Central Processing Unit
DBM	Diagonal Block Matrix
DD	Domain Decomposition
DDR4	Double Data Rate Fourth-generation
DIDT	di/dt
DMA	Direct Memory Access
DP	Dynamic Parallelism
DSP	Digital Signal Processing (Processor)
DVDT	dv/dt
EMT	Electro-Magnetic Transient
FDLM	Frequency-Dependent Line Model
FE	Forward Euler
FF	Flip-Flop
FIFO	First-In First-Out
FPGA	Field-Programmable Gate Array
FTRT	Faster-Than-Real-Time
FTS	Fixed Time-Step
GJ	Gauss-Jordan
GPU	Graphics Processing Unit
GT	Gigabit Transceiver
H-LDE	Hierarchical Linking-Domain Extraction
HBSM	Half-Bridge Sub-Module
HIL	Hardware-In-the-Loop
HLS	High-Level Synthesis
HVDC	High Voltage Direct Current
I/O	Input/Output
ICT	Information and Communication Technique
IGBT	Insulated Gate Bipolar Transistor
IP	Internet Protocol

KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
LDE	Linking-Domain Extraction
LDM	Linking-Domain Matrix
LIM	Latency Insertion Method
LSS	Linear Subsystem Solver
LTE	Local Truncation Error
LUT	Look-Up Table
lwIP	light weight Internet Protocol
MAC	Media Access Controller
MMC	Modular Multi-Level Converter
MPSoC	Multi-Processor System-on-Chip
MR	Multi-Rate
MRMS	Multi-Rate Mixed-Solver
MTDC	Multi-Terminal DC
N-R	Newton-Raphson
NSS	Nonlinear Subsystem Solver
PCIe	Peripheral Component Interconnect express
PCS	Physical Coding Sublayer
PD	Phase-Disposition
PDC	Phasor Data Concentrator
PL	Programmable Logic, Piecewise Linearization
PMU	Phasor Measurement Unit
PS	Processing System
PWM	Pulse Width Modulation
QSFP	Quad Small Form-Factor Pluggable
RAM	Random Access Memory
RTCE	Real-Time Co-Emulation
SC	Schur Complement
SDK	Software Development Kit
SFP	Small Form-Factor Pluggable
SG	Smart Grid
SM	Sub-Module
SPWM	Sinusoidal Pulse Width Modulation
SS	SubSystem
STDC	Super Transient Data Concentrator
TCP	Transmission Control Protocol
TDC	Transient Data Concentrator
TLM	Transmission Line Model
TLN	Transmission-Level Networking
TMU	Transient Measurement Unit
TR	Trapezoidal Rule
TSA	Time-Step Area
TSSM	Two-State Switching Model
ULM	Universal Line Model
UM	Universal Machine
VTS	Variable Time-Stepping

1 Introduction

Electromagnetic Transient (EMT) simulation is a paramount tool in planning, operation, design and commissioning of power systems [1–4]. In EMT simulation, the power system can be described using a set of differential equations based on Kirchhoff's current law (KCL) and voltage law (KVL) analysis, where the unknown variables of the equations are to be solved using numerical integration schemes within each discrete time slot (so-called time-step, typically at microsecond level); if the modeled system contains nonlinear elements, iterative solution is often necessary to obtain accurate results. However, the simulation process slows down significantly when the circuit scale expands: the direct matrix inversion or other algorithms such as the Gauss-Jordan elimination and LU factorization method have large computational complexities, which result in a high computational latency for large-scale networks [5]. Therefore, the fast simulation technique is becoming one of the most important areas in the EMT simulation research, and is increasingly required to be studied and applied. Besides, the traditional power system is evolving into a communication-enabled cyber-physical system (CPS), which also proposes new challenges to conduct fast simulation for the entire system while considering the interplay between communication infrastructure and power system appliance [6].

To deal with the complexity of the EMT simulation for large-scale power systems, three existing fast simulation techniques are commonly used: the multi-rate (MR) scheme that uses different time-step sizes for different decomposed subsystems to balance the accuracy and computational cost; the variable time-stepping (VTS) scheme that changes the time-step size during the simulation to accelerate the process under normal conditions while guaranteeing accuracy; and the domain decomposition (DD) scheme that decomposes a large-scale system into small subsystems and then handles the small subsystems in parallel. However, there is still a lot of room to improve those methods and to study how to implement those methods on practical hardware platforms such as field-programmable gate array (FPGA), multi-processor system-on-chip (MPSoC) and graphics processing unit (GPU) for acceleration.

Based on the above observations, the goal of this research is to conduct fast and parallel EMT simulation for complex power systems to significantly accelerate the simulation process by proposing new computation methods and implementation architectures:

1. In the computation level, new power equipment models with VTS and new mathematical methods for DD are proposed in this thesis: the new mathematical computational processes of the universal line model (ULM) and universal machine (UM) model are proposed, which greatly improve the stability of the models when the time-step changes compared to the traditional ULM and UM model; a novel linking-domain extraction (LDE) method is proposed, which is a new non-overlapping decomposition method that can compute the matrix inversion in parallel based on the found general formula of the inverse of circuit conductance matrix. To fully exploit the potential of LDE method, the hierarchical LDE decomposition method is further proposed.

2. In the implementation level, the new simulation architectures on hardware platforms are proposed: for the MR scheme, an extended multi-rate mixed-solver (MRMS) hardware architecture is proposed for real-time EMT emulation of hybrid AC/DC networks, which takes advantages of the hybrid FPGA-MPSoC platform to emulate AC/DC systems in real-time; for the VTS scheme, the faster-than-real-time architecture on FPGA and 4-level parallel architecture on GPU are proposed to conduct massively parallel simulation with variable time-steps. In addition, the novel real-time co-emulation architectures for the power system and communication network are also proposed and conducted on the FPGA-MPSoC platform and FPGA-Jetson® platform respectively to accelerate the co-simulation process of cyber-physical power systems.

The proposed mathematical models, computational approaches, and implementation architectures contribute to the exiting EMT simulation research and have potential to be applied in the future EMT simulation area.

1.1 Research Definition and Literature Review

The principal objective of the proposed research is to perform fast and parallel EMT simulation for large-scale power systems on GPU/FPGA/MPSoC hardware architectures based on both the existing and the proposed simulation acceleration techniques. The key aspects of the proposed research are presented in this section.

1.1.1 Multi-Rate Simulation Method

In modern power systems, the high voltage AC and DC transmission networks co-exist, wherein both may contain nonlinear elements [7]. The hardware emulation of solving nonlinear elements has been evaluated in [8], which provides the nonlinear solver for this work. Since iterative solution of the large-scale system may involve extremely intensive computation, the complete system can be decomposed into multiple smaller subsystems by leveraging the traveling wave latencies of the widely distributed long-distance transmission lines. The location and the contained nonlinear elements can vary for different subsystems. In fact, there is no need to apply the same step-size for all subsystems [9] since the size of the simulation time-step is dependent on the changing rate during the transient in a certain subsystem and the accuracy requirement. For example, a small time-step (of the order of tens of nanoseconds) that is chosen to capture the device-level switching transients of AC/DC converters results in excessive execution run time, while a relatively large time-step chosen for modeling only the system-level transients would be obviously ineffectual in reproducing the device transients.

Therefore, the multi-rate (MR) simulation is usually adopted to accelerate the simulation process and reduce computational resource consumption. In multi-rate simulation, different subsystems may apply different time-steps, and the selected time-step sizes are determined by the changing rate of the concerned waveforms and the accuracy requirements [10]. In multi-rate simulation, both the iterative solver for nonlinear elements and the conventional non-iterative solver can be applied for different subsystems, which enables applying iterative schemes locally to reduce the computational effort. Different from the variable time-stepping simulation [11] that changes the time-step over the simulation time, in multi-rate simulation the time-steps of different subsystems are fixed, which should be evaluated and configured by users properly in advance to the simulation begin.

1.1.2 Variable Time-Stepping Simulation

Most real-time simulators as well as off-line EMT simulators such as the PSCAD/EMTDC® [12], ATP [13], EMTP-RV [14], PSpice [15] and HSPICE [16] use a fixed time-step (FTS) to proceed the simulation; however, it may be not an efficient approach when the time constants of the power equipment in a system are widely varying and do not change very frequently. For example, a large time-step is usually enough to show the waveforms under normal steady-state conditions, but a small time-step is required when the fast transients happen. Although the variable time-stepping (VTS) method that changes the time-step during simulation according to accuracy requirements has been adopted in the Saber [17] simulator, it is purely targeting on the device-level simulation of power electronics. To accelerate the power simulation process without losing accuracy, the VTS method has been studied and applied in power system simulation over the past years [11, 18–24].

In modern power systems the AC and DC grids are interconnected, wherein linear and nonlinear elements co-exist. In such a system, measuring the system perturbation is the prerequisite to determine the time-step change and control scheme. In this work, different methods are applied to estimate the accuracy:

(1) *Linear Equipment*: It is easy to find the solution for linear elements even with variable time-steps because the network conductance matrix only depends on the history items at t_{n-1}. The local truncation error (LTE) is usually used to estimate the accuracy of the solved variable $\boldsymbol{x}$, given by [25]:

$$LTE(t_n) \approx C_{p+1} \Delta t_n^{p+1} (p+1)! \, \boldsymbol{g}[t_n, ..., t_{n-1-p}] \tag{1.1}$$

where C_{p+1} is the error constant of a specific discretization method, p is the order, and $\boldsymbol{g}[t_{n-1}, ..., t_{n-1-k}]$ can be calculated step-by-step:

$$\boldsymbol{g}(t_{n-1}) = \boldsymbol{x}_{n-1} \tag{1.2}$$

$$\boldsymbol{g}[t_{n-1}, ..., t_{n-k}] = \frac{\boldsymbol{g}[t_{n-1}, ..., t_{n-k+1}] - \boldsymbol{g}[t_{n-2}, ..., t_{n-k}]}{t_{n-1} - t_{n-k}} \tag{1.3}$$

(2) *Nonlinear Equipment*: Finding $\boldsymbol{x}_n$ for nonlinear equipment requires solving the nonlinear system using an iterative approach. The standard method is to first use an explicit method or interpolation polynomial (called the predictor) to get a candidate value of $\boldsymbol{x}_n$, and then use it as the initial solution to apply Newton's iterative method for the implicit integrator (called the corrector) until convergence is achieved. For the predictor, the interpolation polynomial is commonly used:

$$\boldsymbol{x}_n^{(0)} = \boldsymbol{x}_{n-1} + \sum_{k=1}^{p} [\prod_{j=1}^{k} (t_n - t_{n-j})] \boldsymbol{g}[t_{n-1}, ..., t_{n-1-k}] \tag{1.4}$$

Then the LTE can be estimated by comparing the initial solution $\boldsymbol{x}_n^0$ and final solution $\boldsymbol{x}_n$ [25]:

$$LTE(t_n) \approx \frac{C_{p+1}}{1 - C_{p+1}} (\boldsymbol{x}_n - \boldsymbol{x}_n^0) \tag{1.5}$$

(3) *AC/DC Converter*. Although a modular multilevel converter (MMC) is also made up of linear and nonlinear equipment such as the IGBT switches, capacitors and inductors, the LTE method may not be suitable for MMC because it is hard to find which state variable is most representative among the thousands of switches and capacitors and six arm inductors. Thus for the system-level simulation, the differential value $\mathrm{d}v/\mathrm{d}t$ (DVDT) or $\mathrm{d}i/\mathrm{d}t$ (DIDT) of DC voltage or current is computed to measure the system disturbance and determine the time-step change [23]; for the device-level simulation, the switching operation is used to trigger the time-step change.

Since the universal line model (ULM) [26] and universal machine (UM) [27] model serve a wide range of transmission lines and rotating machines, they are required to be properly modeled for variable time-stepping (VTS) EMT simulation. Although the variable time-stepping model for the traveling-wave line model has already been studied in [18, 20], to the best of our knowledge, the VTS models for ULM and UM have not been derived. The work [21, 22] applied the variable time-stepping simulation in frequency-domain, but the frequency-domain line model was simplified without involving the convolution process and both works simulated the system in software but not in hardware for real-time. The work [24] implemented the variable time-stepping simulation in real-time with nonlinear systems such as the power electronic converters and surge arresters; however, all of the power equipment models use the same time-step size, which is not suitable for the VTS simulation of large-scale systems. Although the system decomposition can be applied to use different time-steps for different subsystems [10], the time-step of each subsystem also changes in the VTS simulation; thus how to deal with data exchange and synchronization between subsystems with variable time-steps remains to be discussed.

1.1.3 Network Domain Decomposition

To deal with the complexity of simulating large-scale systems, network domain decomposition [28] is a commonly-used method that splits a large network into small subsystems and simulates them in parallel. One main challenge of using domain decomposition is how to uncouple the inter-connected subsystems, which leads to two representative categories of decomposition methods [29]: overlapping domain decomposition and non-overlapping domain decomposition. In overlapping domain decomposition, the basic logic is to allocate the overlapping domain between two connected subsystems (multiple subsystems have the same procedure) into both subsystems, so that each subsystem can compute the values of the overlapping domain simultaneously. However, to obtain the correct values of the overlapping domain, data exchange and iteration are required to guarantee the difference of results between the two subsystems are smaller than a predetermined threshold. In addition, when the number of decomposed subsystems increases, the convergence time will become much longer, and thus the overlapping domain decomposition method is not the scope of this work.

In non-overlapping domain decomposition, the decomposed subsystems have no overlapping domains thus they could be simulated in parallel while not requiring iteration to synchronize the connected subsystems. For the non-overlapping domain decomposition, the most widely-used methods are the transmission line modeling (TLM) [30, 31], latency insertion method (LIM) [32, 33] and Schur Complement (SC) method [29]. The TLM and LIM methods are latency-based decomposition methods, which leverage the transmission latency between two ends of a line or the latency produced by the LC circuits to decompose the network. Both methods need to consider the simulation time-step size. For example, if the transmission latency of a line is smaller than the time-step size, then the two ends of the line could not be calculated simultaneously. For networks where transmission lines do not exist, the SC method is most commonly used [34, 35], which is a matrix-based decomposition method. It moves all the overlapping area in the network conductance matrix to the bottom right; then the remaining parts are diagonal block matrices that can be handled in parallel. However, the SC method could not obtain the network matrix inversion directly, thus the corresponding procedures need to be executed in each time-step. In addition, the efficiency reduces quickly when the overlapping domain expands.

1.1.4 Co-Simulation between Communication and Power Systems

With the development of cyber physical power systems, the co-simulation between power systems and communication networks is becoming a hot topic in EMT simulation. Various co-simulation frameworks for interacting communication and power domains have been proposed in the recent past since the first interface for the EMTDC/PSCAD® simulator [36] to integrate an agent-based distributed application into the simulation was developed. Most of these works are not to design a complete simulator that could finish simulation in one package but are targeting interfacing two existing simulators in each domain [37–42], because there are already various mature power system and network simulators to use. For example, EMTDC/PSCAD®, DigSilent®, PowerWorld®, and OpenDSS® etc are widely used in power system simulation; while NS-2/NS-3, OMNeT++, OPNET and NeSSi have been successfully used in network development and evaluation. Unfortunately, there does not have existing interfaces for data exchange between simulators of the two domains due to the different working principles. Thus the main concern of existing co-simulator frameworks is to properly handle data exchange and synchronization for related events in both domains at run-time [6, 43]. However, the performance of software-based simulators is relatively low compared with actual power and network devices even without taking the data exchange and synchronization time between two simulators into account. It is therefore difficult to simulate and test the adequacy of manufactured protection and control equipment responding to damage and upset by transient in real-time. To the best of our knowledge, the real-time co-simulator implemented on FPGA/MPSoC board has not been studied. Instead of interfacing the software based simulators, FPGA enables flexible programmability and highly paralleled computing, which is able to capture and response to the system change quickly in both power system and communication network domains.

1.1.5 GPU, FPGA and SoC

For large-scale power systems, the parallel simulation is often required to accelerate the simulation process, which is usually achieved on the parallel computing architectures: the graphics processing unit (GPU), field-programmable gate array (FPGA) and multi-processor system-on-chip (MPSoC). The GPU device is composed of a huge amount of processing cores, which enables the generation of numerous grids, blocks, and threads and parallel simulation of large-scale power systems [44–47]. FPGA provides numerous hardware and rich I/O resources, which has been used in both industry and academia for

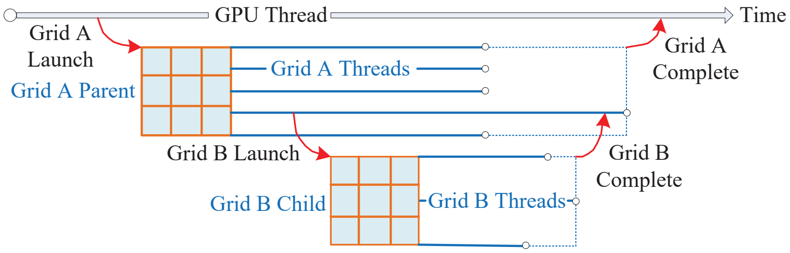


Figure 1.1: Illustration of GPU dynamic parallelism.

the emulation of power electronics and large-scale power systems [8,9,24,48]. MPSoC integrates both FPGA board and multi-processors, which makes a complete emulation system with both parallel and sequential algorithms [49,50].

(1) *GPU Architecture*. The GPU-based programming involves two parts of the hardware resources: *host*, on which the CPU programs run serially, and *device*, on which the GPU programs run in parallel. The GPU programming model is based on primitives of threads, blocks, and grids: a grid is a collection of threads, and the threads in a grid are divided into blocks that is a group of threads which execute on the same multiprocessor and have access to the same shared memory. Typically, the kernel function defining the program executed by individual threads within a block and grid can only be called by the host, which involves sophisticated execution control and frequent data transfer between host and device. As an extended capability to the GPU programming model, the dynamic parallelism feature [51] enables the kernel function to create and synchronize with new kernel functions on the GPU device dynamically at whichever point in a program. The grid that has launched new grid(s) is called a "parent" grid, and the one is launched by a parent grid is called "child" grid. Grids launched with dynamic parallelism are fully nested, which means the parent is not considered completed until all of its launched child grids have also completed, as shown in Fig. 1.1.

Despite the advantages of dynamic parallelism, it also introduces a cost in launching kernels, which is considerable compared with the execution time of child kernels. If the child kernels do not extract much parallelism and there is not much benefit against their non-parallel counterparts, then the little benefit may be canceled out by the child kernel launching overheads. Thus when applying the dynamic parallelism, the massive parallelism of child kernel functions is preferred to guarantee the performance gain in global scope.

(2) *FPGA Architecture*. FPGAs are integrated circuits designed to be reconfigured to meet different application requirements, composed of an array of programmable logic blocks and a hierarchy of reconfigurable interconnections that make the blocks be wired together. Taking advantage of hardware parallelism and fast inputs and outputs (I/O) at the hardware level, FPGA provides significant processing performance and flexibility, and thus is extensively used for EMT simulation. The Virtex UltraScale+ FPGA VCU118

board [52] used in this research contains both highly programmable UltraScale XCVU9P device and rich external resources such as block RAMs, transceivers, DSP slices and I/O pins, which enables the usage of iterative method and the detailed models applied in this work. High-level synthesis (HLS), provided by Xilinx®, translates C/C++ language to HDL with highly paralleled hardware structure [53]. HLS supports the arbitrary data precision and provides abundant directives for optimization, such as loop unroll, array partition, pipelining, etc. Since C/C++ language has much higher abstraction than HDL, the coding effort is substantially reduced.

(3) *MPSoC Architecture*. The MPSoC itself integrates the programmable logic (PL) resource and the ARM® multi-core processor system (PS) on the same chip. Compared with the solution of using discrete CPU and FPGA on different boards, the single-chip solution provides substantially higher communication bandwidth and coherence between the PS and the PL. The improved overall performance of both sequential and parallel computing by using FPGA-MPSoC platform enables the usage of the iterative method and the detailed models applied in this work. The Zynq ZCU102 board [54] used in this research is featured with a quad-core ARM® Cortex-A53, dual-core Cortex-R5 real-time processors, and a Mail-400 MP2 graphics processing unit (GPU) based on programmable logic fabric. The PS communicate with the PL using high-bandwidth Advanced eXtensible Interface (AXI) channels, enabling low-latency data exchange. Using such an architecture, sequential computing and configurations can be moved into PS that can achieve high clock frequency, while parallel computing can be processed in PL that can achieve high parallelism.

1.2 Research Objectives

The motivation of this thesis is consistent with one of the most concerned aspects in the general research of EMT simulation area, which is to accelerate the simulation process for complex power systems while guaranteeing reasonable accuracies. The new technologies of hardware platforms and emerging cyber-physical power systems also require the corresponding developments of the simulation architectures. Such challenges come from the higher demand of simulation efficiencies for the modern grid, which requires the solutions from both the computation methods and hardware implementation architectures.

The major tasks and specific research objectives for this work are listed as following:

- **Multi-Rate Mixed-Solver Architecture for AC/DC Network Emulation**
  In modern AC/DC systems, linear and nonlinear elements co-exist, while different power equipment has widely different time constants. To simulate such a power system, the multi-rate scheme requires to be applied. The task of this work is to design a multi-rate real-time emulation architecture with linear and nonlinear solvers deployed, called the "multi-rate mixed-solver" architecture, to emulate the AC/DC network in real-time. The emulation is expected to be conducted on the hybrid

FPGA-MPSoC hardware platform to fully utilize the advantages in both parallel and serial computing.

- **Variable Time-Stepping Universal Line and Machine Models and Implementation**
  In the research of variable time-stepping power equipment modeling, the universal line model (ULM) and universal machine (UM) models with variable time-steps have not been investigated. Due to the convolution computation in ULM and exciter of UM, the instability problem when the time-step changes should be addressed. The goal of this work is to propose a stable ULM and UM model no matter how the time-step changes. Besides, the parallel VTS simulation architecture is quite different from the FTS architecture, which should also be proposed and implemented on both the FPGA and GPU platform.

- **Linking-Domain Extraction Based Domain Decomposition Method**
  Different from the latency-based decomposition method, the matrix-based non-overlapping domain decomposition methods still have a lot of room to be studied. The traditional Schur complement decomposition method is not efficient when the number of decomposed subsystems increases. The task of this work is to study the special features of the conductance matrix of the traditional power systems, and to find the general formulation of the matrix inversion and solve the matrix equations in parallel. The simulation and verification is expected to be conducted on FPGA, GPU and CPU for different test systems and application contexts.

- **Co-Emulation Hardware Architecture for Cyber-Physical Systems**
  The emerging cyber-physical power system combines the physical layer with the information and communication techniques (ICT), which propose a new challenge to the fast co-simulation of power system and communication networks. The influence of communication behaviours between the smart meters, phasor measurement unit, controller and controllable devices should be evaluated practically and precisely. The goal of this work is to implement the real-time co-emulator (RTCE) on hybrid FPGA-MPSoC hardware platform and FPGA-Jetson® platform instead of interfacing the software-based simulators, which aims to leverage the hardware based simulator that is able to response to the system change quickly in both power system and communication network domains.

1.3 Summary of Contributions

The major contributions of this work can be summarized into two aspects: computational method and implementation architecture, as shown in Fig. 1.2. The proposed computational methods are based on the mathematical analysis and discoveries; while the pro-

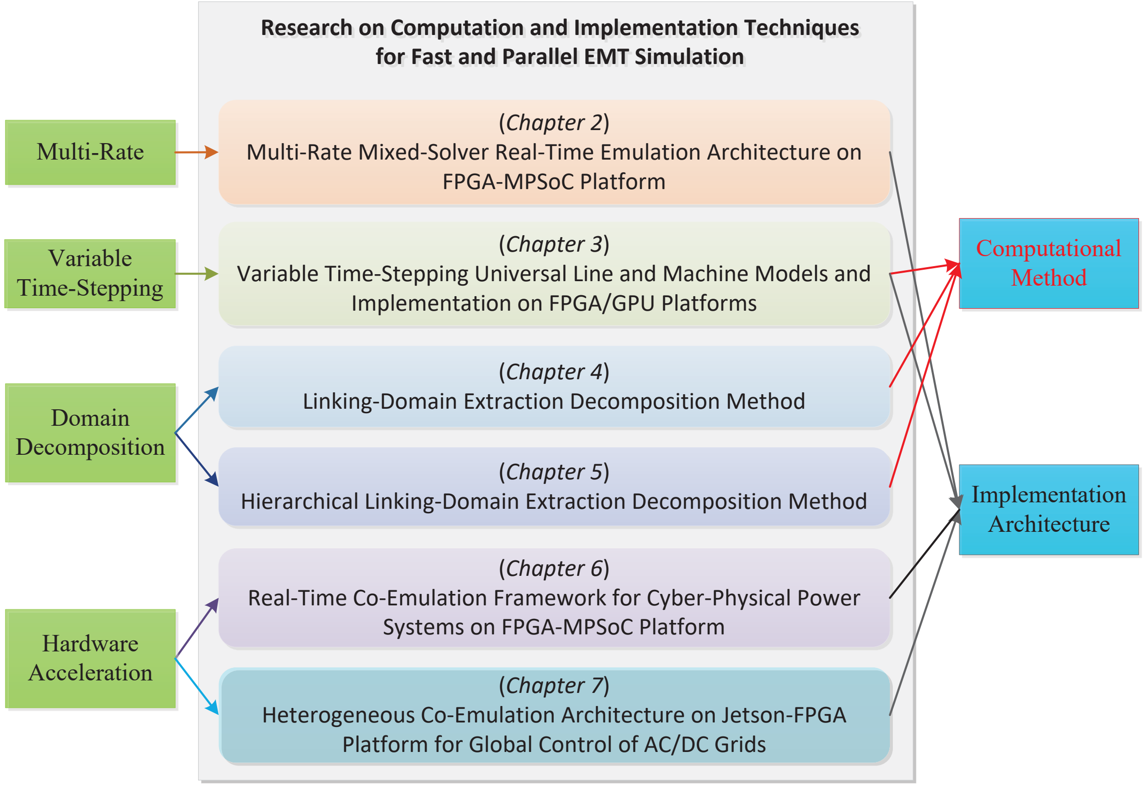

Research on Computation and Implementation Techniques for Fast and Parallel EMT Simulation
Multi-Rate
Variable Time-Stepping
Domain Decomposition
Hardware Acceleration
(Chapter 2) Multi-Rate Mixed-Solver Real-Time Emulation Architecture on FPGA-MPSoC Platform
(Chapter 3) Variable Time-Stepping Universal Line and Machine Models and Implementation on FPGA/GPU Platforms
(Chapter 4) Linking-Domain Extraction Decomposition Method
(Chapter 5) Hierarchical Linking-Domain Extraction Decomposition Method
(Chapter 6) Real-Time Co-Emulation Framework for Cyber-Physical Power Systems on FPGA-MPSoC Platform
(Chapter 7) Heterogeneous Co-Emulation Architecture on Jetson-FPGA Platform for Global Control of AC/DC Grids
Computational Method
Implementation Architecture

Figure 1.2: Contributions of the proposed research and structure of this thesis.

posed implementation architectures focus on achieving fast EMT simulation using the latest FPGA/MPSoC/GPU parallel platforms.

- **Contributions to Computational Method**

  (1) *Variable Time-Stepping Universal Line and Machine Models*

  The key point of computing the ULM and exciter of UM is the convolution part. Using the traditional computational procedure, the result of the convolution is not continuous when the time-step changes. The proposed "process-reverse" computational procedure and equivalent circuit model can perform a stable computation no matter how the time-step changes, which greatly improve the stability of the ULM and exciter of the UM model for VTS simulation.

  (2) *Linking-Domain Extraction (LDE) Based Domain Decomposition Method*

  Based on the mathematical analysis over the conductance matrix generated by traditional power systems, in this work, the conductance matrix is decomposed into a diagonal block matrix and a linking-domain matrix; then the general formulation of the matrix inversion is found, which is a strong mathematical basis for the parallel computation of matrix inversion. The LDE method can not only be used in computing matrix inversion in parallel, but can also be used in solving matrix equations with several advantages over the traditional Schur complement method.

(3) *Hierarchical LDE Decomposition Method*

The original LDE method is inefficient in both the computational procedure and storage cost. To proposed hierarchical LDE (H-LDE) method is an all-round improvement over the original LDE method, which eliminates the necessity of computing the entire conductance matrix inversion and uses a multi-level decomposition to accelerate the process of inverting the decomposed block matrices. The H-LDE method can even achieve lower computation latencies than the sparse LU based solvers within a certain power system scale.

- **Contributions to Implementation Architecture**

  (1) *Real-Time Multi-Rate Mixed-Solver Emulation Architecture on FPGA-MPSoC Platform*

  To simulate AC/DC power systems in real-time, the multi-rate mixed-solver emulation architecture is proposed. By moving the MMC control tasks into the ARM® based processor system of MPSoC board, the MMC model can be computed in real-time; by re-using the linear solvers when the nonlinear solvers are working, the hardware resource costs can be reduced a lot; by allocating the large system into multiple FPGA boards, the multi-board solution is exploited and the fast data exchange between different boards is achieved via the Xilinx® Aurora core.

  (2) *Faster-than-Real-Time Emulation Architecture on FPGA and 4-level Parallel Simulation Architecture on GPU for VTS Simulation*

  The parallel VTS simulation architecture is quite different from the FTS architecture due to the synchronization between decomposed subsystems. In this work, the FPGA-based and GPU-based parallel simulation architectures are proposed for VTS simulation. Through elaborate configuration to the time-step sizes of different subsystems, the "faster-than-real-time" mode is achieved on FPGA; using the dynamic parallelism features and hierarchical decomposition, the massively parallel VTS simulation is achieved on GPU.

  (3) *Co-Emulation Hardware Architecture for Cyber-Physical Systems*

  The existing software-based co-simulation platforms are facing the difficulties of accelerating the simulation process due to the large overhead of data exchange and synchronization. In the proposed real-time co-emulation (RTCE) framework on FPGA-MPSoC based hardware architecture, the real-time *discrete-time* based power system EMT emulation and the *discrete-event* based communication network emulation can be achieved. The data exchange between two domains is handled within each board with an extremely low latency, which is sufficiently fast for real-time interaction. In the proposed heterogeneous Jetson®-FPGA based co-emulation architecture, the communication-enabled global control schemes are studied for AC/DC cyber physical power systems.

1.4 Thesis Outline

This thesis consists of eight chapters. The subsequent chapters shown in Fig. 1.2 are outlined as follows:

- **Chapter 2**
  This chapter proposes a novel multi-rate mixed-solver architecture for AC/DC system emulation to fully utilize the time space and optimize hardware computation resources without loss of accuracy, wherein both iterative and non-iterative solvers with different time-steps are applied to the decomposed subsystems, and the linear solvers are reused within each time-step. The proposed solver and the complete real-time emulation system are implemented on FPGA-MPSoC platform. The real-time results are captured by the oscilloscope and verified with PSCAD/EMTDC® and SaberRD® for system-level and device-level performance evaluation.

- **Chapter 3**
  This chapter derives the VTS models for ULM and UM, and the proposed ULM model is more stable than the traditional model. Both VTS models are emulated on the parallel and pipelined architecture of the FPGA. The proposed subsystem-based VTS scheme and the local truncation error (LTE) based time-step control enable the large-scale systems to be simulated in real-time. The "faster-than-real-time" modes on FPGA boards, and 4-level dynamic parallelism architecture on GPU are also proposed for variable time-stepping EMT simulation. The transient waveforms and execution time speed-ups indicate that the proposed method can extremely accelerate the simulation process while guaranteeing reasonable accuracy compared to the fixed time-step based simulation.

- **Chapter 4**
  In this chapter, a novel linking-domain extraction (LDE) based decomposition method is proposed, in which the network matrix is expressed as the sum of a linking-domain matrix (LDM) and a diagonal block matrix (DBM) composed of multiple block matrices in diagonal. Through mathematical analysis over LDM, one *lemma* about the nature of LDM and its proof are proposed. Based on this *lemma*, the general formulation of the inverse matrix of the sum of LDM and DBM can be found using the Woodbury matrix identity, and based on the formulation the network matrix inversion can be directly computed in parallel to accelerate the matrix inversion process. Test systems were implemented on both the FPGA and GPU parallel architectures, and the simulation results and speed-ups over the Schur complement method and Gauss-Jordan elimination demonstrate the validity and efficiency of the proposed LDE method.

- **Chapter 5**

In this chapter, a novel hierarchical LDE (H-LDE) method is proposed to further improve the LDE method, which leverages all the hidden features of LDE that are not exploited in the original work to perform a multi-level decomposition of power systems. The LDE-based matrix equation solution computation procedure is first proposed to eliminate the necessity of computing the entire matrix inversion, and then the multi-level computation structure is proposed for fast matrix inversion of the decomposed sub-matrices. The 4-level LDE decomposition is applied on the IEEE 118-bus test power system and implemented in both sequential and parallel, which is used to verify the validity and efficiency of the proposed H-LDE decomposition method. The simulation results of various benchmark test power systems show that the proposed H-LDE method can achieve lower computation latency than the classical LU factorization and sparse KLU method within a certain system scale.

- **Chapter 6**
  This chapter proposes a novel real-time co-emulation (RTCE) framework on FPGA-MPSoC based hardware architecture for a more practical emulation of real-world cyber-physical systems. The *discrete-time* based power system electromagnetic transient (EMT) emulation is executed in programmable hardware units so that the transient level behaviour can be captured in real-time, while the *discrete-event* based communication network emulation is modeled in abstraction-level or directly executed on the hardware PHY and network ports of the FPGA-MPSoC platform, which can perform the communication networking in real-time. The data exchange between two domains is handled within each platform with an extremely low latency, which is sufficiently fast for real-time interaction; and the multi-board scheme is deployed to practically emulate the communication between different power system areas. The hardware resource cost and emulation latency for the test system and case studies are evaluated to demonstrate the validity and effectiveness of the proposed RTCE framework.

- **Chapter 7**
  In this chapter, a heterogeneous hardware real-time co-emulator composed of FPGAs, many-core GPU, and multi-core CPU devices is proposed to study the communication enabled global control schemes of hybrid AC/DC networks. The electromagnetic transient (EMT) power system emulation is conducted on the Xilinx® FPGA boards to provide nearly continuous instantaneous waveforms for cyber layer sampling; the communication layer is simulated on the ARM® CPU cores of the embedded NVIDIA® Jetson platform for flexible computing and programming; and the control functions for modular multi-level converters are executed on GPU cores of the Jetson® platform for parallel calculation. The data exchange between FPGAs and Jetson® is achieved via the PCI express interface, which simulates the sampling operation of the AC phasor measurement unit (PMU) and DC merging unit (DC-MU).

The power overflow and DC fault cases are investigated to demonstrate the validity and effectiveness of the proposed co-emulation hardware architecture and global control schemes.

- **Chapter 8**
  This chapter summarizes the contributions of this research and discusses the future work for the EMT simulation study.

2

Multi-Rate Mixed-Solver for Real-Time Nonlinear Electromagnetic Transient Emulation on FPGA-MPSoC Architecture

Nonlinear phenomena widely exist in AC/DC power systems, which should be accounted for accurately in real-time EMT simulation for obtaining precise results for hardware-in-the-loop applications. However, iterative solutions such as the Newton-Raphason method that can precisely obtain the results for highly nonlinear elements, are time consuming and computationally onerous. To fully utilize the time space and optimize hardware computation resources without loss of accuracy, this chapter proposes a novel multi-rate mixed-solver hardware architecture for real-time emulation of AC/DC systems, wherein both iterative and non-iterative solvers with different time-steps are applied to the decomposed subsystems, and the linear solvers are reused within each time-step. The proposed solver and the complete real-time emulation system are implemented on FPGA-MPSoC platform. The real-time results are captured by the oscilloscope and verified with PSCAD/EMTDC® for system-level performance evaluation.

2.1 Proposed Multi-Rate Mixed-Solver for EMT Simulation

In the real-time EMT simulation, the size of simulation time-step is an essential variable that directly determines the time-step dependent parameters and influences the element model selection and computational resource costs. Since the time-step requirements can vary between different subsystems, the multi-rate mixed-solver for real-time EMT simulation is proposed to reduce the hardware resource costs and improve the overall accuracy.

Typically, by applying the KVL and KCL to the network to be solved, the network

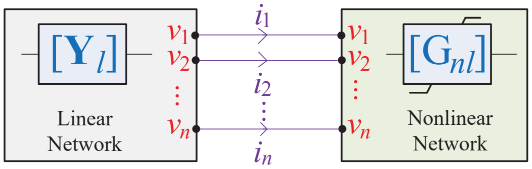


Figure 2.1: Decomposing a network into separated pure linear and nonlinear network.

equation can be derived for time-discretized EMT simulation, which is expressed as follows:

$$\mathbf{Y}\boldsymbol{v} = \boldsymbol{i}_{eq} \tag{2.1}$$

where $\mathbf{Y}$ is the network conductance matrix, $\boldsymbol{i}_{eq}$ is the equivalent injected current source vector that changes at every time-step, and $\boldsymbol{v}$ is the unknown nodal voltage vector to be solved. For networks that only contain linear elements, $\mathbf{Y}$ is constant over simulation time. However, if the networks contain nonlinear elements, $\mathbf{Y}$ may change during the simulation process. In such a case, the network can be decomposed into linear and nonlinear networks, in which the linear network only contains linear elements and leave the nonlinear elements as open-circuits, while the nonlinear network only contains nonlinear elements and leave the linear elements as open-circuits, as shown in Fig. 2.1. The current $\boldsymbol{i}_c = [i_1, i_2, ..., i_n]^T$ flows from the linear network to the nonlinear network.

The linear network can be solved as:

$$\mathbf{Y}_l\boldsymbol{v} = \boldsymbol{i}_{eq,l} - \boldsymbol{i}_c \tag{2.2}$$

where $\mathbf{Y}_l$ and $\boldsymbol{i}_{eq,l}$ are the linear network conductance matrix and equivalent injected current source vector only considering linear elements.

Nonlinear elements in the nonlinear network can be represented by piecewise linearization [55], Newton-Raphson (N-R), or compensating current source methods [56]. The piecewise linear method uses piecewise linear segments to approximate nonlinear $i-v$ functions, wherein the segment of next time-step is determined by the voltage of previous time-step, which may induce the overshoot problem. The N-R method can provide more accurate results by iteratively calculating the conductance matrix within each time-step, which is essential to sensitively respond to system changes. In this work, the N-R method is applied:

$$\mathbf{G}_{nl}(\boldsymbol{v}^k)\boldsymbol{v}^{k+1} = \boldsymbol{i}_{eq,nl}(\boldsymbol{v}^k) + \boldsymbol{i}_c, \tag{2.3}$$

where k is the iteration number, $\boldsymbol{v}^k = [v_1^k, v_2^k, ..., v_n^k]^T$ is the results of k^{th} iteration, $\mathbf{G}_{nl}$ and $\boldsymbol{i}_{eq,nl}$ are the Jacobian matrix representing conductance and equivalent injected current source vector only considering nonlinear elements, given by:

$$\mathbf{G}_{nl}(\boldsymbol{v}^k) = \begin{bmatrix} \frac{\partial f_1(v_1)}{\partial v_1}\big|_{\boldsymbol{v}^k} & \frac{\partial f_1(v_1)}{\partial v_2}\big|_{\boldsymbol{v}^k} & \cdots & \frac{\partial f_1(v_1)}{\partial v_n}\big|_{\boldsymbol{v}^k} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(v_n)}{\partial v_1}\big|_{\boldsymbol{v}^k} & \frac{\partial f_n(v_n)}{\partial v_2}\big|_{\boldsymbol{v}^k} & \cdots & \frac{\partial f_n(v_n)}{\partial v_n}\big|_{\boldsymbol{v}^k} \end{bmatrix}. \tag{2.4}$$

where the function $f_i(v_i)$ represents the nonlinear $i - v$ characteristics for node voltage v_i. Then the iterative matrix equation for solving the nonlinear network can be derived from (2.2) and (2.3) by eliminating $\boldsymbol{i}_c$:

$$[\mathbf{Y}_l + \mathbf{G}_{nl}(\boldsymbol{v}^k)]\boldsymbol{v}^{k+1} = \boldsymbol{i}_{eq,l} + \boldsymbol{i}_{eq,nl}(\boldsymbol{v}^k) \tag{2.5}$$

Since the iteration times are uncertain and the conductance matrix could be re-factorized, the N-R method could consume more time and resource than piecewise linear method. Thus, it is extremely hard to apply N-R method in large AC/DC systems where the matrix size is large. However, since transmission lines widely exist in AC/DC systems and the line length is usually sufficiently long to guarantee the traveling time is longer than the simulation time-step, the large AC/DC network can be decomposed into m subsystems using the traveling-wave line model or frequency-dependent line model (FDLM), as shown below:

$$\begin{bmatrix} \mathbf{Y}_{11} & 0 & \cdots & 0 \\ 0 & \mathbf{Y}_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \mathbf{Y}_{mm} \end{bmatrix} \begin{pmatrix} \boldsymbol{v}_{S1} \\ \boldsymbol{v}_{S2} \\ \vdots \\ \boldsymbol{v}_{Sm} \end{pmatrix} = \begin{pmatrix} \boldsymbol{i}_{eq,S1} \\ \boldsymbol{i}_{eq,S2} \\ \vdots \\ \boldsymbol{i}_{eq,Sm} \end{pmatrix} \tag{2.6}$$

where $\mathbf{Y}_{ii}$ is the conductance matrix of subsystem $S_i, 1 \leqslant i \leqslant m$. Assume the first m_l subsystems are linear networks, and the last m_{nl} subsystems are nonlinear. These subsystems can be solved concurrently within each time-step. The linear solver only involves one process of solving the matrix equations, whereas the nonlinear solver may need several iterations of such process, which could take several times the latency of processing than that of the linear solver. Therefore, during the processing of nonlinear solver iterations in each time-step, there will be much idle time for linear solver, and as a result, there will be a lot of hardware resource wasted. On the other hand, the transient behaviors of subsystems where transients such as lightning and switching occur need to be adequately modeled and precisely revealed, while the subsystems distant from the transients are only slightly affected by them and thus they do not need very small time-step to capture the system behavior.

Based on the above observations, the multi-rate mixed-solver is proposed: to ensure high accuracy, both the iterative solver for nonlinear elements and the conventional non-iterative linear solver are applied for different subsystems; and to reduce computation resource consumption, the multiple time-step scheme is used and carefully designed for different subsystems. The proposed multi-rate mixed-solver can be formulated as follows:

$$\mathbf{Y}_{ii}\boldsymbol{v}_{Si}^{\Delta t_l(i)} = \boldsymbol{i}_{eq,Si}^{\Delta t_l(i)}, 1 \leqslant i \leqslant m_l \tag{2.7}$$

$$\mathbf{Y}_{ii}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)})\boldsymbol{v}_{Si}^{k+1,\Delta t_{nl}(i)} = \boldsymbol{i}_{eq,Si}^{\Delta t_{nl}(i)}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)}),\ m_l + 1 \leqslant i \leqslant m \tag{2.8}$$

where

$$\mathbf{Y}_{ii}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)}) = \mathbf{Y}_{l,i} + \mathbf{G}_{nl,i}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)}) \tag{2.9}$$

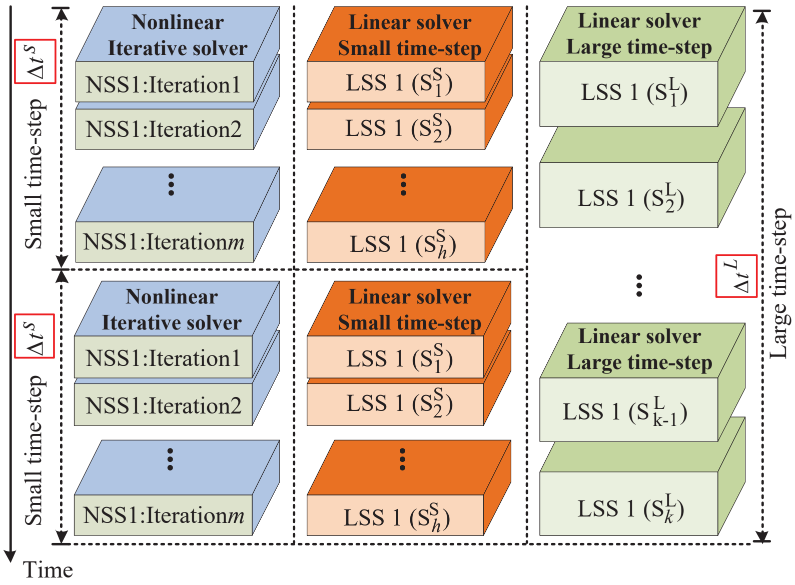


Figure 2.2: Illustration of the multi-rate mixed-solver simulation.

$$\boldsymbol{i}_{eq,Si}^{\Delta t_{nl}(i)}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)}) = \boldsymbol{i}_{eq,l,Si}^{\Delta t_{nl}(i)} + \boldsymbol{i}_{eq,nl,Si}^{\Delta t_{nl}(i)}(\boldsymbol{v}_{Si}^{k,\Delta t_{nl}(i)}) \tag{2.10}$$

$$\Delta t_l(i), \Delta t_{nl}(i) \in \{\Delta t_j | 1 \leqslant j \leqslant p\} \tag{2.11}$$

Equation (2.11) denotes that there are p different time-steps (Δt_1, ..., Δt_p) applied, and subsystem S_i is assigned time-step $\Delta t_l(i)$ or $\Delta t_{nl}(i)$ depending on linear or nonlinear systems. Equations (2.9) and (2.10) have the same form as the derived iterative matrix equation (2.5). After each time-step, the results may need to be exchanged between connected subsystems, thus interpolation is required if the two subsystems use different time-steps. For example, if subsystem S_i needs the results $\boldsymbol{v}_{Sj}^{\Delta t_l(j)}$ at simulation time t (t is exactly integer multiple of $\Delta t_l(i)$) from subsystem S_j, then S_i should interpolate the results received from S_j into the data for its own use. In this work, linear interpolation is used:

$$\boldsymbol{v}_{Sj}^{\Delta t_l(j)} |_t = \boldsymbol{v}_{Sj}^{\Delta t_l(j)} |_{t_1} + \frac{t - t_1}{\Delta t_l(j)} (\boldsymbol{v}_{Sj}^{\Delta t_l(j)} |_{t_2} - \boldsymbol{v}_{Sj}^{\Delta t_l(j)} |_{t_1}), \tag{2.12}$$

$$t_1 = rounddown(\frac{t}{\Delta t_l(j)}) \times \Delta t_l(j) \tag{2.13}$$

$$t_2 = roundup(\frac{t}{\Delta t_l(j)}) \times \Delta t_l(j) \tag{2.14}$$

For the case shown in Fig. 2.2 as an example, there are two time-steps applied (small time-step Δt^S and large time-step Δt^L). Within one small time-step, nonlinear subsystem solvers (NSS) perform iterative calculations, while the linear subsystem solver (LSS) with small time-step is reused by subsystems $S_1^S - S_h^S$ to fully use the time space; and within one large time-step, linear solvers with large time-step are reused by subsystems $S_1^L - S_k^L$ and the results at the end of small time-step can be obtained by interpolation between two large time-step results. After each time-step, results of the NSS and LSS with small time-step and the LSS with large time-step are outputted for display respectively and history items are exchanged between adjacent subsystems.

In the proposed multi-rate mixed-solver, the selection of time-step sizes and solver types for different subsystems should be carefully analyzed. Assume there are m subsystems **S** $(S_1, S_2, ..., S_m)$, and p different rates with time-step sizes of $\Delta\mathbf{T}$ $(\Delta t_1, \Delta t_2, ..., \Delta t_p)$ to be selected. Other than the time-step size, reuse of the linear solver for multiple subsystems should also be evaluated. Let $\mathbf{K} = (K_1, K_2, ..., K_q)$ denotes the used solvers including linear and nonlinear solvers, then the selection can be seen as a mapping $g : \mathbf{S} \mapsto (\Delta\mathbf{T}, \mathbf{K})$. The principle of time-step selection is to minimize the total cost including the accuracy and hardware resource consumption while guaranteeing the accuracy requirements, which can be formulated as follows:

$$min\, C(g) = \sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{k=1}^{q}[\alpha E(i,j,k) + \beta R(i,j,k)] \cdot g(i,j,k) \tag{2.15}$$

$$s.t. \qquad E(i,j,k) \cdot g(i,j,k) \leqslant E_{th,i} \tag{2.16}$$

$$\sum_{i=1}^{m} g(i,j,k) \cdot t_k \leqslant \Delta t_j \tag{2.17}$$

where $g(i,j,k) = 1$ if S_i uses Δt_j as time-step, and is calculated by the solver K_k; and otherwise $g(i,j,k) = 0$. $E(i,j,k)$ and $R(i,j,k)$ represent the simulation error and the corresponding resource cost respectively of subsystem S_i with time-step size of Δt_j using solver K_k, and they are both nonlinear functions of mapping g. Besides, as indicated in (2.16), $E(i,j,k)g(i,j,k)$ should not be bigger than the pre-determined threshold error $E_{th,i}$ of subsystem S_i, which means if $E(i,j)$ is larger than $E_{th,i}$ then $g(i,j,k)$ should be equal to zero. Equation (2.17) indicates that the total calculating time of each solver selected by subsystem S_i (denoted as t_i) should not exceed the selected time-step size, the summation sign means the reuse of solver is taken into consideration. α and β are scaling factors that unify the two parts of cost. It also should be noted that the number of used solvers q is not a pre-determined constant but a variable of which the optimal value can be solved by (2.15). However, the equations above are just the principle for time-step selection, because the precise function of $E(i,j,k)$ and $R(i,j,k)$ can only be obtained by experiment and can vary between different systems and different implementation platforms.

2.2 Comprehensive Real-Time Emulator Implementation

The data-flow of the MRMS simulation is illustrated in Fig. 2.3. In the example, there are two rates with time-step of Δt_1 and Δt_2, and three solvers named NSS1, LSS1 and LSS2. NSS1 is a nonlinear subsystem solver performing several iterations, after each iteration the voltage $\boldsymbol{v}$ and conductance matrix **G** is updated until $|(\boldsymbol{v}^k - \boldsymbol{v}^{k-1})/\boldsymbol{v}^k|$ is smaller than the

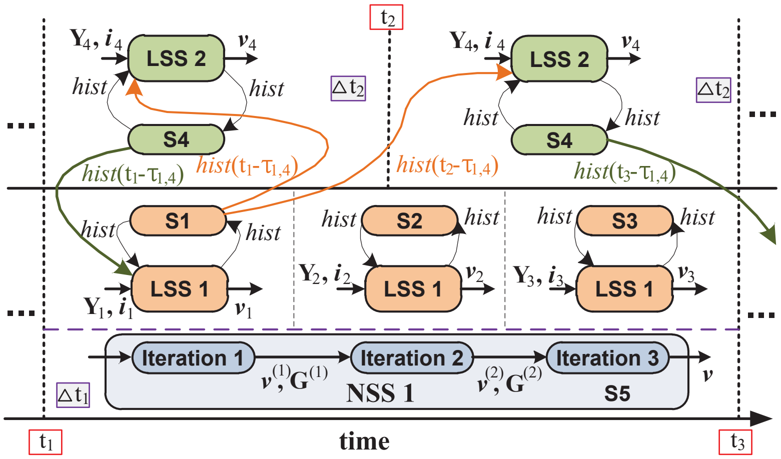


Figure 2.3: Data-flow in the proposed multi-rate mixed-solver.

pre-determined threshold. LSS1∼2 are linear solvers, and LSS1 is reused by subsystems S_1, S_2, and S_3. For simplicity, Fig. 2.3 only illustrates the data exchange between subsystem S_1 and S_4, and the other connections are omitted.

For thorough analysis of the proposed MRMS solver on real-time FPGA-MPSoC emulator, an AC/DC grid composed of two IEEE 39-bus systems [57] and a three-terminal HVDC system was chosen as the circuit topology, as shown in Fig. 2.4. In each IEEE 39-bus system, 10 generators, 12 transformers, 19 loads and 34 transmission lines are deployed, and the two IEEE 39-bus systems are connected by three AC/DC MMC converter stations that are connected via two DC transmission lines. The control of converter C1 is used for DC voltage regulation, while in the converters C2/C3 the active power flow is chosen as the controlled variable. To protect generators, transformers, cables and other devices from overvoltages caused by lightning, short circuit, switching, etc, 6 surge arresters are also installed in the system.

The MPSoC ZCU102 board [54] used in this chapter is featured with a quad-core ARM® Cortex-A53, dual-core Cortex-R5 real-time processors, and a Mail-400 MP2 graphics processing unit (GPU) based on programmable logic fabric. These processors (PS) communicate with the programmable logic (PL) using high-bandwidth Advanced eXtensible Interface (AXI) channels, enabling low-latency data exchange. The hybrid Virtex UltraScale+ FPGA VCU118 board [52] and MPSoC ZCU102 board platform enable the usage of the iterative method and the detailed models applied.

To extend the resource capacity for simulating the large system, the multi-board solution is applied in this work, as shown in Fig. 2.5, there are totally three FPGA/MPSoC boards (two VCU118 boards and one ZCU102 board) used to run the study case. ZCU102 board is the master board connecting with two VCU118 boards and sending instructions to control the behavior of the other two boards. The two VCU118 boards are slave boards, which start or stop to perform simulation under the instruction of the master board. The

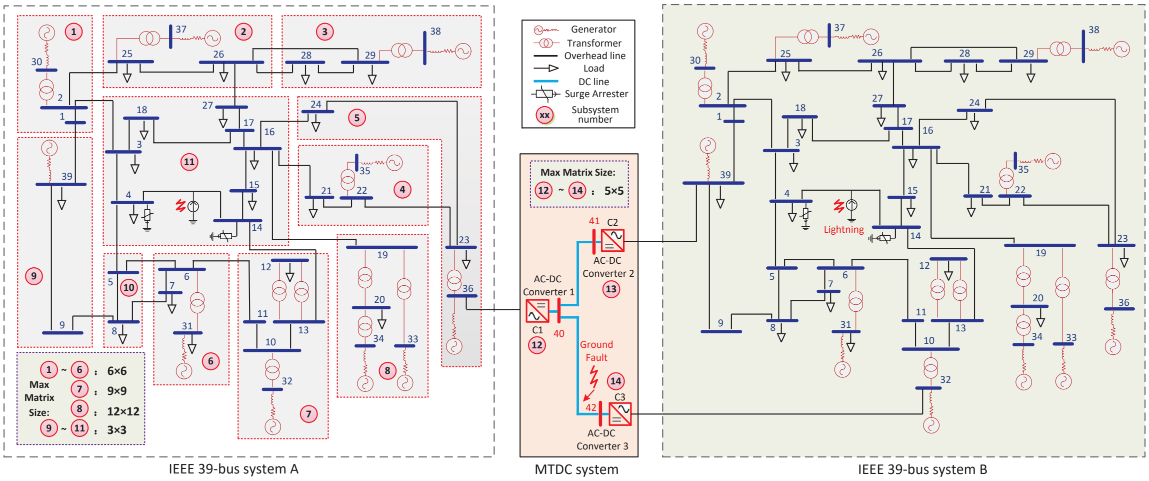


Figure 2.4: Topology of the AC/DC grid test case.

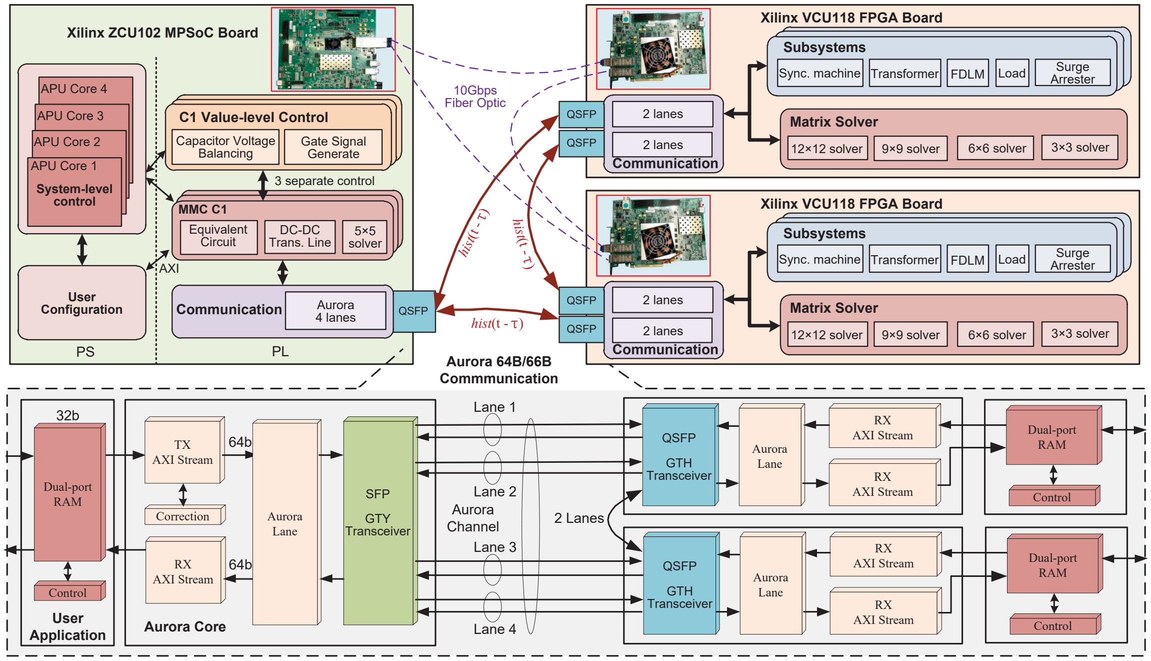


Figure 2.5: Hardware emulation of the case study on two FPGA boards and one MPSoC board.

master MPSoC board has multiple processors which can be used to run sequential computing and state control, whereas the two slave boards have more hardware resources and more communication transceivers which enable larger subsystems to be simulated and faster data exchange between each other.

Subsystem Allocation. There are two IEEE 39-bus systems to be simulated, and each of them are allocated at one single VCU118 FPGA board for simplicity, which also reduces the amount of data to be exchanged between boards. The three MMC converters are simulated in the ZCU102 MPSoC board to make full use of the APU resources for complex system-level control algorithms. The subsystem allocation for each board is shown in Fig. 2.4, which is determined by the specific circuit. For example, since almost every generator is connected with a transformer, it is beneficial for solver reuse if every generator-transformer pair is allocated to different individual subsystems, as marked as subsystem $S_1 \sim S_6$.

Due to the transmission lines between converters, the three MMC modules are also divided into three subsystems, each of which is composed of equivalent circuit calculation, value-level control and system-level control. Since the system-level control of MMC converter is sequentially calculated and may consume many hardware resources to execute, it is more efficient to implement the control logic including the inner loop and outer loop control in the PS part of MPSoC board. The computation of value-level control is more intensive than system-level control but the tasks can be well paralleled due to the independence of each SM, and thus are performed in the PL part.

Multi-Rate Mixed-Solver. To perform the EMT simulation on FPGA board, the main complexity is contributed by solving the matrix equation. Firstly, the reuse of matrix solver is discussed. The conductance matrix of most subsystems can be divided into smaller matrices, for example, subsystem S_{11} contains eight buses, which will generate a 24×24 matrix to be solved. However, considering the uncoupling function of the transmission line model, the 24×24 matrix can be divided into eight separated 3×3 matrices and they can be solved by reuse of a 3×3 linear solver (except for buses with surge arresters that require iterative solvers). The subsystems $S_1 \sim S_6$ composed of a generator and transformer can not be divided, because the two sides of transformer are coupled and thus at least a 6×6 matrix is generated. Therefore, the 6×6 linear solver can be also reused among these subsystems. Subsystem S_7 and S_8 contain the largest matrix (9×9 and 12×12 respectively) and cannot share the solver with other subsystems, thus consume the longest time to finish calculation within each time-step.

Secondly, multi-rate with four time-step sizes of 0.2μs, 5μs, 10μs, and 20μs is applied among subsystems. Based on the principles of time-step selection, the time-step of the subsystems where transients happen should firstly conform to (2.17), and then should be as small as possible to meet the accuracy requirements (2.16). Therefore, the time-step of 10μs is widely used in subsystems of the IEEE 39-bus, because the processing time of most subsystems is just less than 10μs. Subsystem $S_1 \sim S_6$ can reuse the 6×6 solver between two subsystems to fully occupy the time space. The reuse of 3×3 solver is also adopted in subsystem S_{11} to make full use of the time space when the iterative solver is dealing with buses containing surge arresters. The time-step of subsystem S_8 and subsystem $S_{12} \sim S_{14}$ (MMC converters) is set at 20μs, by considering the large processing delay of the 12×12 matrix solver, the complex control of MMC and the communication latency between boards. The time-step of 5μs is applied only in subsystem S_{10} just for a demonstration of multi-rate, because subsystem S_{10} has the smallest scale and matrix size. The time-step of 0.2μs is adopted for device-level simulation, and considering the limited hardware resources in the MPSoC board only the first SM of MMC C2 is simulated in device-level to produce the device-level transient. The simple curve-fitting model is applied so that the device-level behaviours can be emulated in real-time.

Data Exchange. The history values of the two ends of a transmission line are exchanged after each time-step and stored in a FIFO, and the FIFO shift based on the common time-step size, which is the minimum step-size for the system (5μs). For example, if the time-step of a subsystem is 10μs, then the shifted-register will shift 2 to store the data, which makes the same location of the memory in different subsystems store the data generated simultaneously.

If the two ends of a line are computed in different boards, the communication between interfaced boards should be designed. To enable high-speed communication between the three boards, lightweight communication protocol should be used instead of the common TCP/IP protocol that involves too much time cost during connection establishment. Xilinx® provides a scalable link-layer communication protocol, Aurora [58], which is open and supported by different type of transceivers such as GTY and GTH transceivers. The Xilinx® aurora core can automatically initialize and maintain the channel, and the AXI-4 user interface enables users to conveniently generate and receive data without considering the transmission details and handling transmission states. The communication part of the implementation is shown in Fig. 2.5, the three boards are connected with each other via two lanes. After channel establishment, the aurora core reads data from the RAM and a 64b AXI-4 stream based data is generated by combining the 32b user data and 32b address together. The 32b address is used to identify the user data and put it into the right address of the RAM after receiving.

2.3 Results and Verification

The example test case described above is emulated on the three FPGA/MPSoC boards and the results are compared with PSCAD/EMTDC® and SaberRD® to show the effectiveness of the proposed multi-rate mixed-solver. The APU cores of MPSoC board run at 1.2GHz, while the clock frequency of FPGA boards is set at 100 MHz.

2.3.1 Hardware Resource Utilization and Latency

According to the hardware implementation details and subsystem allocation described above, the system-level hardware resource consumption and time-step size are presented in Table 2.1, in which VCU118-1 represents the version that does not reuse the mixed-solver, and VCU118-2 represents the optimized cost by reusing the linear solvers. Since the two VCU118 boards have nearly the same cost by simulating the same size of circuit topol-

Table 2.1: Hardware Resource Utilization of the Case Study

Resource	VCU118-1	VCU118-2	ZCU102
LUT	867,134 (73.3%)	769,148 (65.1%)	253,798 (92.6%)
FF	1,027,012 (43.4%)	914,041 (38.7%)	452,780 (82.6%)
BRAM	640 (29.6%)	624 (28.9%)	416 (45.6%)
DSP	6,748 (98.6%)	5,864 (85.7%)	2,490 (98.8%)
Time-step	5/10/20μs	5/10/20μs	0.2/10/20μs

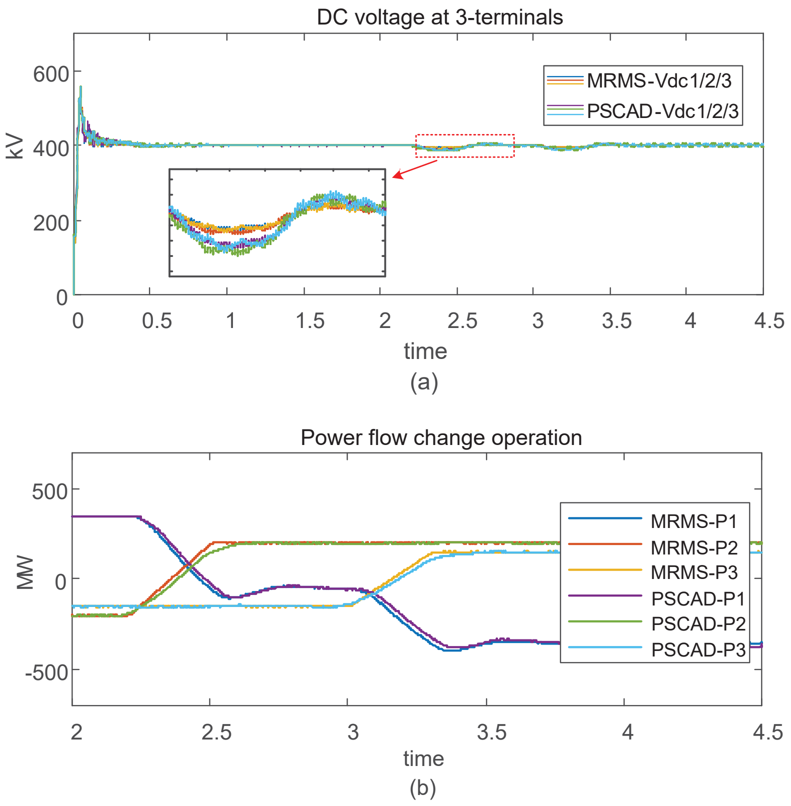
Table 2.2: Processing Latency of Communication and Subsystems

Subsystem/Element	Latency	Subsystem/Element	Latency
Subsystem 1∼6	5.1∼5.6 μs	Subsystem 7	7.6 μs
Subsystem 8	11.2 μs	Subsystem 9	4.2 μs
Subsystem 10	3.2 μs	Subsystem 11	9.8 μs
Subsystem 12∼14	14.2μs	Aurora	0.95 μs
Trans. Line	2.35 μs	Transformer	2.15 μs
Generator	1.05 μs	Surge Arrester	4.65 μs
3×3 iterative solver	4.1 μs	3×3 linear solver	0.71 μs
5×5 linear solver	1.67 μs	6×6 linear solver	1.81 μs
9×9 linear solver	4.47 μs	12×12 linear solver	7.75 μs

ogy (IEEE 39-bus system), Table 2.1 only shows one of them. Four representative types of resources are recorded: lookup table (LUT), flip-flops (FF), block RAM (BRAM), and digital signal processing unit (DSP). Through reuse of solver, the logic resource (mainly refers to LUT) cost can be reduced by about 11.3%, and the computing resource (mainly refers to DSP) cost can be reduced by about 13.1%.

The processing latency of different solvers and functions are recorded in Table 2.2, which indicates that processing latency varies between different subsystems and different solvers. For example, subsystem S_{11} contains nonlinear surge arresters and subsystem S_{10} only contains linear elements, although the matrix equations they need to solve are both 3×3, the average latency has a big difference because the iterative matrix solver consumes about five times latency of the non-iterative matrix solver averagely. Subsystem S_8 consumes the most simulation time because it has the largest matrix (12×12) to solve. It should be noticed that since the hardware-based calculation is running in parallel, the latency is not just the simple addition of processing latency of each element.

The latency of Aurora communication is 0.95 μs for data transmission of fifteen 32bit single floating-point data, which includes the transmission latency and the latency of writing and reading data to/from the RAM. Since the three boards use the same clock frequency, the communication latency is almost the same although they use different types of transceivers. The transmission time through fiber is also estimated by end-to-end transmission latency test, which is less than three clocks thus is negligible.



(a)

(b)

Figure 2.6: Steady-state operation of converters. (a) DC voltage at 3-terminals; (b) Power flow change operation of multi-converters.

2.3.2 Real-Time Emulation Results

To simulate the non-linear behavior of the AC/DC system, the lightning surge at Phase C of AC transmission line 4-14 (between bus 4 and bus 14) in both 39-bus systems is chosen as the fault condition. The results are evaluated by the proposed emulator and PSCAD/EMTDC®, in which PSCAD/EMTDC® uses constant time-step of 10μs and 20μs respectively while the proposed emulator uses multiple time-step of 0.2/5/10/20μs.

Firstly, the steady state operation results are recorded. As representatives, the DC voltage and power flow of the three converters are used to show the power flow between the two 39-bus systems. As shown in Fig. 2.6(a), it takes about 0.2s for capacitor charging before the DC voltages reach at steady-state of 400kV. The results of the proposed emulator marked as MRMS match well with PSCAD/EMTDC® results with 20μs, and the difference is less than 3%, which is relatively small considering the large scale of topology and number of nonlinear elements. The power flow change operation is shown in Fig. 2.6(b), in which the power flow from C1 to C2 changes at simulation time of 2.2s, and the power flow from C1 to C3 changes at 3.0s. The simulation results of MRMS are almost the same as PSCAD/EMTDC® at steady-state, but there are some differences during power flow changing operation, because the values outputted by outer and inner loop control will

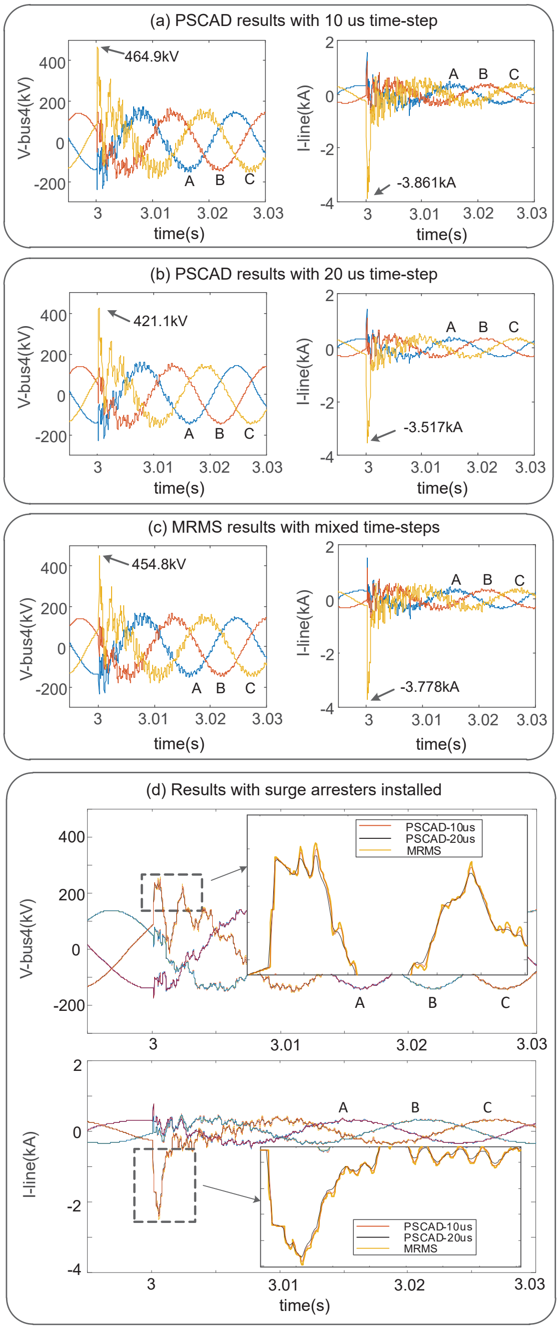


Figure 2.7: Lightning transient results. (a)∼(c) PSCAD/EMTDC® results with 10, 20μs time-step and MRMS results without surge arresters deployed; (d) Results with surge arresters installed.

change a lot during power flow change and thus will generate a bigger difference.

Secondly, the transient of lightning surge current is simulated to show the nonlinear behavior of surge arresters and transformers. The standard $10/350\mu s$ lightning surge current [59] is applied in this work, given as:

$$I_{LS}(t) = CI_m(t/\tau_1)^k e^{\frac{-t}{\tau_2}}/[1 + (t/\tau_1)^k] \tag{2.18}$$

where the coefficient $C = 1.075$, $k = 10$; the time constant $\tau_1 = 19\mu$s, $\tau_2 = 485\mu$s; the maximum value of the surge current $I_m = 10$kA. In this simulation, the lightning surge current is applied at exactly $3s$ of the simulation, and the results without and with surge arresters are shown in Fig. 2.7(a)∼(c) and (d) respectively. The peak value and transient details of surge voltage and current without surge arresters installed indicate that changing the time-step value will significantly impact the accuracy. MRMS uses mixed time-step of 0.2/5/10/20μs, and the results are more reasonable than the PSCAD/EMTDC® results with 20μs time-step and are close to that of 10μs. When the surge arresters are installed, MRMS uses the iterative solver to solve the nonlinear elements for accuracy, although the nonlinear function is simplified to piecewise linear segments. But in PSCAD/EMTDC®, the piecewise linear method is used to deal with the nonlinearity of surge arresters. As shown in Fig. 2.7(d), the MRMS results are close to those of the PSCAD/EMTDC® with 10μs time-step, and can even show more details although there is not a judgement which one is more correct.

2.4 Summary

To optimize the accuracy as well as the resource cost, a novel multi-rate mixed-solver hardware emulation architecture is proposed. In the proposed solver, the power system is decomposed into several subsystems, in which multiple time-steps are applied for different subsystems according to the accuracy requirements; by applying the iterative schemes locally and reusing the linear solver among subsystems, the computational resource consumption is reduced. The AC/DC network composed of two IEEE 39-bus systems and three MMC converters is emulated in real-time on the hybrid FPGA-MPSoC platform. The processing delay of different subsystems and solvers is evaluated, which shows the practicality of multi-rate with 0.2/5/10/20μs time-steps applied. The multi-rate mixed-solver can be used for large AC/DC system simulations that consist of various types of elements with requirements of high accuracy and optimum computation resource consumption. In the future work, the emulation system can be further enlarged with more complicated nonlinear models [60–63] for conventional power equipment as well as power electronic apparatus.

3

Variable Time-Stepping Universal Line and Machine Models and Implementation on FPGA and GPU Platforms

In the conventional EMT simulator the time-step is fixed, which may lead to inefficiencies when the time constants of the system change. The variable time-stepping (VTS) method can efficaciously solve this problem; however, the VTS schemes for the universal transmission line model (ULM) and universal machine (UM) model remain to be investigated. This chapter derives the VTS models for ULM and UM, and the proposed ULM model is more stable than the traditional model. Both VTS models combined with other equipment are emulated on the parallel architecture of the FPGA and GPU platform. The proposed hierarchical VTS scheme and the local truncation error (LTE) based time-step control enable the large-scale systems to be simulated in real-time and "faster-than-real-time" modes on FPGA. The 4-level massively parallel VTS simulation architecture is also proposed for GPU implementation. The IEEE 39-bus and 118-bus test power systems with VTS models were emulated on FPGA and GPU boards respectively, and the emulation results compared with PSCAD/EMTDC® and fixed time-stepping (FTS) hardware emulator verified the effectiveness of the proposed models and implementation architectures.

3.1 Universal Transmission Line Model Computation

The ULM and UM model computation for VTS is not the same as that for the fixed time-step because the model parameters may change when the time-step changes. The equivalent circuit for ULM is shown in Fig. 3.1, in which the two ends (k and m) of the line are abstracted into two disconnected parts, and each part combines an equivalent conductance

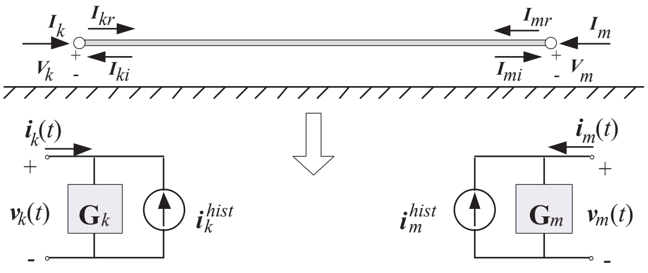


Figure 3.1: Equivalent circuit of the ULM.

matrix in parallel with a compensating current source ($\boldsymbol{i}_k^{hist}$ and $\boldsymbol{i}_m^{hist}$) through which the two ends are interacting.

The electromagnetic behavior of a transmission line in frequency domain can be characterized by two matrix transfer functions: $\mathbf{Y}_c(\omega)$, the characteristic admittance matrix, and $\mathbf{H}(\omega)$, the propagation matrix. The frequency-domain relationship between currents and voltages at the two ends can be expressed using these two matrices:

$$\boldsymbol{I}_k(\omega) = \mathbf{Y}_c(\omega)\boldsymbol{V}_k(\omega) - 2\boldsymbol{I}_{ki}(\omega) \tag{3.1}$$

$$\boldsymbol{I}_m(\omega) = \mathbf{Y}_c(\omega)\boldsymbol{V}_m(\omega) - 2\boldsymbol{I}_{mi}(\omega) \tag{3.2}$$

where

$$\boldsymbol{I}_{ki}(\omega) = \mathbf{H}(\omega) \cdot \boldsymbol{I}_{mr}(\omega),\ \boldsymbol{I}_{mi}(\omega) = \mathbf{H}(\omega) \cdot \boldsymbol{I}_{kr}(\omega) \tag{3.3}$$

and $\boldsymbol{I}_{kr}$ and $\boldsymbol{I}_{ki}$ are the incoming current wave and reflected current wave at k point respectively. Then the time-domain relationship between currents and voltages at the two ends can be expressed using these two matrices [26, 64] by transforming the frequency-domain equations into time-domain, the equivalent current source can be obtained ($\boldsymbol{i}_k(t) = \mathbf{G}\boldsymbol{v}_k(t) - \boldsymbol{i}_k^{hist}$):

$$\boldsymbol{i}_k^{hist} = \mathbf{G}_k\boldsymbol{v}_k(t) - \left[\mathbf{y}_c * \boldsymbol{v}_k(t) - 2\mathbf{h} * \boldsymbol{i}_{mr}(t)\right] \tag{3.4}$$

$$\boldsymbol{i}_m^{hist} = \mathbf{G}_m\boldsymbol{v}_m(t) - \left[\mathbf{y}_c * \boldsymbol{v}_m(t) - 2\mathbf{h} * \boldsymbol{i}_{kr}(t)\right] \tag{3.5}$$

where $\mathbf{y}_c$ and $\mathbf{h}$ are obtained via an inverse Fourier transform for $\mathbf{Y}_c(\omega)$ and $\mathbf{H}(\omega)$, $\boldsymbol{i}_{kr}$ and $\boldsymbol{i}_{ki}$ are the incoming current wave and reflected current wave at k point respectively. In practice, the convolution operation represented by the symbol "$*$" is usually not easy to carry out because $\mathbf{Y}_c(\omega)$ and $\mathbf{H}(\omega)$ may be too complex to have a simple formula in time domain.

By applying proper fitting techniques [65], the time-domain elements of $\mathbf{Y}_c$ and $\mathbf{H}$ can be simplified as:

$$\mathbf{y}_c^{(i,j)}(t) = \sum_{k=1}^{N_p} \mathbf{r}_{Y_c}^{(i,j,k)} e^{\mathbf{p}_{Y_c}^{(k)} t} + \mathbf{d}^{(i,j)}\delta(t) \tag{3.6}$$

$$\mathbf{h}^{(i,j)}(t) = \sum_{k=1}^{N_m} \sum_{n=1}^{N_{p,k}} \mathbf{r}_H^{(i,j,k,n)} e^{\mathbf{p}_H^{(k,n)}(t-\tau_k)} \tag{3.7}$$

where N_p, $N_{p,k}$ are the number of poles, N_m is the number of modes, the superscript of each symbol indicates the elements of a vector or matrix. Note that the italic symbols represent vectors and non-italic symbols denote matrices. The residue matrix **r** is a three-dimension matrix (for $\mathbf{Y}_c$), because it contains a 3×3 matrix (three conductors for example) for each pole. The pole parameter $\boldsymbol{p}$ is a vector for $\mathbf{Y}_c$ but is a matrix for **H** because it has multiple modes. **d** is the proportional terms, and τ_k is the time delay for the k^{th} mode.

In EMT simulation, the $\boldsymbol{g}(t) = \mathbf{y}_c * \boldsymbol{v}_k(t)$ convolution calculation can be discretized to be performed step-by-step by using a state variable $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N_p)}]$ defined below [64]:

$$\dot{\boldsymbol{x}}^{(i)}(t) = \boldsymbol{p}_{Y_c}^{(i)} \boldsymbol{x}^{(i)}(t) + \boldsymbol{v}_k(t), 1 \leqslant i \leqslant N_p \tag{3.8}$$

$$\boldsymbol{g}(t) = \sum_{i=1}^{N_p} \mathbf{r}_{Y_c}^{(:,:,i)} \boldsymbol{x}^{(i)}(t) + \mathbf{d}\boldsymbol{v}_k(t) \tag{3.9}$$

where the superscript $(:, :, i)$ means the i-th 3×3 matrix of $\mathbf{r}_{Y_c}$. By applying the TR discretization, (3.8) could be calculated as:

$$\boldsymbol{x}^{(i)}(n) = \boldsymbol{\alpha}_{Y_c}^{(i)}(n)\boldsymbol{x}^{(i)}(n-1) + \boldsymbol{\lambda}_{Y_c}^{(i)}(n)\boldsymbol{v}_k(n) + \boldsymbol{\mu}_{Y_c}^{(i)}(n)\boldsymbol{v}_k(n-1) \tag{3.10}$$

$$\boldsymbol{g}(n) = \sum_{i=1}^{N_p} \mathbf{r}_{Y_c}^{(:,:,i)} \boldsymbol{x}^{(i)}(n) + \mathbf{d}\boldsymbol{v}_k(n) \tag{3.11}$$

where

$$\boldsymbol{\alpha}_{Y_c}^{(i)}(n) = (1 + \boldsymbol{p}_{Y_c}^{(i)}\frac{\Delta t_n}{2})/(1 - \boldsymbol{p}_{Y_c}^{(i)}\frac{\Delta t_n}{2}) \tag{3.12}$$

$$\boldsymbol{\lambda}_{Y_c}^{(i)}(n) = \boldsymbol{\mu}_{Y_c}^{(i)}(n) = (\frac{\Delta t_n}{2})/(1 - \boldsymbol{p}_{Y_c}^{(i)}\frac{\Delta t_n}{2}) \tag{3.13}$$

From (3.10) it can be observed that $\boldsymbol{v}_k(n)$ should be known to compute the state variable $\boldsymbol{x}^{(i)}(n)$, but the value of $\boldsymbol{v}_k(n)$ is unknown before the the state variable $\boldsymbol{x}^{(i)}(n)$ is obtained. The applied methods to deal with this problem is the main difference between the traditional ULM and proposed ULM representation

Traditional Model: The traditional method [64] that is widely used in fixed time-stepping ULM model is to use a new state variable $\boldsymbol{x}^{*(i)}(n) = \boldsymbol{x}^{(i)}(n) - \boldsymbol{\lambda}_{Y_c}^{(i)}(n)\boldsymbol{v}_k(n)$, $1 \leqslant i \leqslant N_p$ instead to eliminate the $\boldsymbol{v}_k(n)$ item:

$$\boldsymbol{x}^{*(i)}(n) = \boldsymbol{\alpha}_{Y_c}^{(i)}(n)\boldsymbol{x}^{*(i)}(n-1) + (\boldsymbol{\alpha}_{Y_c}^{(i)}(n)\boldsymbol{\lambda}_{Y_c}^{(i)}(n-1) + \boldsymbol{\mu}_{Y_c}^{(i)}(n))\boldsymbol{v}_k(n-1), \tag{3.14}$$

$$\boldsymbol{g}(n) = \sum_{i=1}^{N_p} \mathbf{r}_{Y_c}^{(:,:,i)} \boldsymbol{x}^{*(i)}(n) + \mathbf{G}_k(n)\boldsymbol{v}_k(n). \tag{3.15}$$

And $\mathbf{G}_k(n)$ is the equivalent conductance matrix:

$$\mathbf{G}_k(n) = \mathbf{G}_m(n) = \mathbf{d} + \sum_{k=1}^{N_p} \boldsymbol{\lambda}_{Y_c}^{(k)}(n)\mathbf{r}_{Y_c}^{(:,:,k)} \tag{3.16}$$

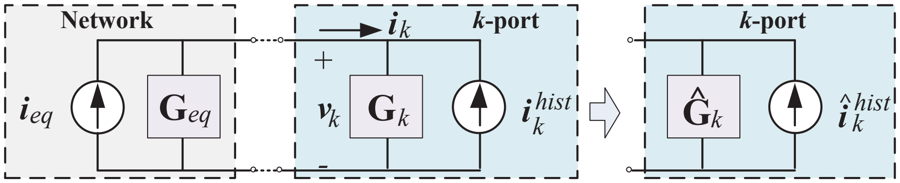


Figure 3.2: Illustration for the process-reversed model for the ULM.

Note that $\mathbf{G}(n)\boldsymbol{v}_k(n)$ in (3.15) can be eliminated by the current items in (3.4) when calculating the equivalent current source $\boldsymbol{i}_k^{hist}$; thus $\boldsymbol{v}_k(n)$ is not required, which makes the equivalent current source to be computed only depending on the results of previous steps. The $\mathbf{h} * \boldsymbol{i}_{mr}(t)$ convolution has the same computation flow, but only the history items $\boldsymbol{i}_{mr}(t-\tau_k), k = 1, .., N_m$ are needed for the calculation at time t because there is a time-delay τ_k in the $\mathbf{h}$ matrix function.

Limitations of the traditional ULM: The traditional method for fixed time-step may not be applicable with variable time-steps. Because when the time-step changes, the constant $\boldsymbol{\lambda}_{Y_c}$ will change to a new value, which makes the state variable $\boldsymbol{x}^*$ actually change to a different new state variable from the one before time-step changes. The problem is, the new state-variable needs to have a correct initial value that is consistent with the former state-variable when the time-step changes, but the initial value that the former state-variable provides is not correct for the new state-variable. That means, the new state-variable may need several steps to make itself stable and consistent with the former state-variable, which will cause instability when the time-step changes. This phenomenon will be verified later in the simulation section.

Process-Reversed Model: To solve the instability problem of the traditional method, this work proposes a novel current-source based method that has a reverse logic of the traditional method. The rest of the system in each end of the transmission line can be equivalenced to a current source $\boldsymbol{i}_{eq}$ in parallel with a conductance $\mathbf{G}_{eq}$, as shown in Fig. 3.2. At the port k, the system equation can be written as:

$$[\mathbf{G}_{eq} + \mathbf{G}_k]\boldsymbol{v}_k(t) = \boldsymbol{i}_{eq}(t) + \boldsymbol{i}_k^{hist} \tag{3.17}$$

Substituting $\boldsymbol{i}_k^{hist}$ by (3.4) and expressing as discrete-time:

$$[\mathbf{G}_{eq} + \mathbf{G}_k]\boldsymbol{v}_k(n) = \boldsymbol{i}_{eq}(n) + \mathbf{G}_k\boldsymbol{v}_k(n) - \left[\mathbf{y}_c * \boldsymbol{v}_k(n) - 2\mathbf{h} * \boldsymbol{i}_{mr}\right] \tag{3.18}$$

From (3.10) (3.11) we know:

$$\mathbf{y}_c * \boldsymbol{v}_k(n) = \sum_{i=1}^{N_p} \mathbf{r}_{Y_c}^{(:,:,i)} \boldsymbol{x}^{(i)}(n) + \mathbf{d}\boldsymbol{v}_k(n) \tag{3.19}$$

Here, $\mathbf{G}_k(n) = \mathbf{d}$. Combining (3.10)(3.18)(3.19) together, we get:

$$[\mathbf{G}_{eq} + \mathbf{G}_k + \mathbf{G}_r]\boldsymbol{v}_k(n) = \boldsymbol{i}_{eq}(n) + 2\mathbf{h} * \boldsymbol{i}_{mr} - \boldsymbol{x}^{hist} \tag{3.20}$$

where:

$$\mathbf{G}_r = \sum_{k=1}^{N_p} \boldsymbol{\lambda}_{Y_c}^{(k)}(n)\mathbf{r}_{Y_c}^{(:,:,k)} \tag{3.21}$$

$$\boldsymbol{x}^{hist} = \sum_{i=1}^{N_p} \mathbf{r}_{Y_c}^{(:,:,i)}[\boldsymbol{\alpha}_{Y_c}^{(i)}(n)\boldsymbol{x}^{(i)}(n-1) + \boldsymbol{\mu}_{Y_c}^{(i)}(n)\boldsymbol{v}_k(n-1)] \tag{3.22}$$

Since the convolution $\mathbf{h} * \boldsymbol{i}_{mr}$ at time t_n actually only needs the history items $\boldsymbol{i}_{mr}(t_n - \tau_k), k = 1, .., N_m$, the only unknown variable to be solved at t_n in (3.20) is the node voltage $\boldsymbol{v}_k(n)$. Equations (3.20) (3.22) actually generate a new equivalent admittance matrix $\hat{\mathbf{G}}_k$ and $\hat{\boldsymbol{i}}_k^{hist}$, as shown in Fig. 3.2, where:

$$\hat{\mathbf{G}}_k = \mathbf{G}_k + \mathbf{G}_r, \quad \hat{\boldsymbol{i}}_k^{hist} = 2\mathbf{h} * \boldsymbol{i}_{mr} - \boldsymbol{x}^{hist} \tag{3.23}$$

And the $\hat{\mathbf{G}}_k$ will change if the time-step changes. After the node voltages are solved, the other variables ($\boldsymbol{x}, \boldsymbol{i}_{kr}(n), \boldsymbol{i}_{mr}(n)$, etc.) at t_n can be solved based on the node voltages.

The logic of this method is totally different from the traditional method. In the traditional method, the state variables $\boldsymbol{x}^{(i)}(n)$ are calculated first to obtain $\boldsymbol{i}_k^{hist}$, and then the node voltages are solved; however, this method will cause instability when the time-step changes. In the process-reversed method, the calculation sequence is reversed: firstly the node voltages are solved and then the state variables $\boldsymbol{x}(n)$ are updated. In the fixed time-stepping simulation, these two methods essentially have the same presentation, however, in the VTS simulation, using the process-reversed method the state variables $\boldsymbol{x}(n)$ will remain the same no matter how the time-step changes, which greatly improves the stability.

3.2 Universal Machine Model Computation

For the UM model, there are three stator windings $\{d, q, 0\}$, several damper windings $\{kd, kq\}$ in the direct and quadrature (d and q) rotor axis, and one field winding $\{f\}$. In this work, the machine with one kd winding and two kq windings is considered as a general model to represent the synchronous generators. The fixed time-stepping model and hardware implementation can be found in [66], in which the relationship between voltages and currents can be expressed as follows:

$$\boldsymbol{v}_{um}(t) = -\mathbf{R}_{um}\boldsymbol{i}_{um}(t) - \frac{d}{dt}\boldsymbol{\psi}_{um}(t) + \boldsymbol{u}(t) \tag{3.24}$$

$$\boldsymbol{\psi}_{um}(t) = \mathbf{L}_{um} \cdot \boldsymbol{i}_{um}(t) \tag{3.25}$$

where $\boldsymbol{v}_{um}$ = $[v_d, v_q, v_0, v_f, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{i}_{um}$ = $[i_d, i_q, i_0, i_f, i_{kd}, i_{kq1}, i_{kq2}]^{\mathrm{T}}$, $\boldsymbol{\psi}_{um}$ = $[\psi_d, \psi_q, \psi_0, \psi_f, \psi_{kd}, \psi_{kq1}, \psi_{kq2}]^{\mathrm{T}}$, $\boldsymbol{u}$ = $[-\omega\psi_q, \omega\psi_d, 0, 0, 0, 0, 0]^{\mathrm{T}}$, $\mathbf{R}_{um} = diag(R_d, R_q, R_0, R_f, R_{kd}, R_{kq1}, R_{kq2})$ and $\mathbf{L}_{um}$ is the leakage inductance matrix.

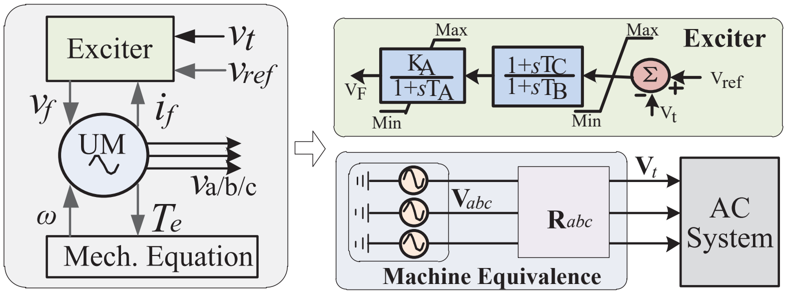


Figure 3.3: Equivalent circuit for the universal machine model.

To solve the machine equations above in discrete time, (3.24) is discretized using the trapezoidal rule (TR), which leads to a Thévenin voltage source representation [67] shown in Fig 3.3:

$$\boldsymbol{v}_{um}(n) = -[\mathbf{R}_{um} + \frac{2}{\Delta t_n}\mathbf{L_{um}} - \omega \mathbf{L_u}]\boldsymbol{i}_{um}(n) + \boldsymbol{v}^{hist} \tag{3.26}$$

where:

$$\boldsymbol{v}^{hist} = \boldsymbol{u}(n-1) - \boldsymbol{v}_{um}(n-1) - [\mathbf{R}_{um} - \frac{2}{\Delta t_n}\mathbf{L}_{um}]\boldsymbol{i}_{um}(n-1) \tag{3.27}$$

and $\boldsymbol{u}(n) = \omega \mathbf{L_u}\boldsymbol{i}_{um}(n)$, $\mathbf{L_u} = [-\mathbf{L_{um}}(2); \mathbf{L_{um}}(1); \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0}]$. Note that the time-step Δt_n is not a constant but may change during the simulation process, which is different from the FTS model.

Let $\mathbf{R}_{um,eq} = [\mathbf{R}_{um} + \frac{2}{\Delta t}\mathbf{L_{um}} - \omega\mathbf{L_u}]$, and $\mathbf{R}_{um,eq}$ = $[\mathbf{R}_{ss}\ \mathbf{R}_{sr}; \mathbf{R}_{rs}\ \mathbf{R}_{rr}]$. Then the dq0 frame can be extracted from the vector of (3.26):

$$\boldsymbol{v}_{dq0}(n) = -\mathbf{R}_{ss}\boldsymbol{i}_{dq0}(n) - \mathbf{R}_{sr}\boldsymbol{i}_r(n) + \boldsymbol{v}_{dq0}^{hist} \tag{3.28}$$

$$\boldsymbol{v}_r(n) = -\mathbf{R}_{rs}\boldsymbol{i}_{dq0}(n) - \mathbf{R}_{rr}\boldsymbol{i}_r(n) + \boldsymbol{v}_r^{hist} \tag{3.29}$$

where $\boldsymbol{i}_r = [i_f, i_{kd}, i_{kq1}, i_{kq2}]^T$, and $\boldsymbol{v}_r = [v_f, 0, 0, 0]^T$ only contains the field voltage of which the value at time t_n is known from the exciter module. Thus from (3.28) and (3.29), the relationship between $\boldsymbol{v}_{dq0}(n)$ and $\boldsymbol{i}_{dq0}(n)$ is derived:

$$\boldsymbol{v}_{dq0}(n) = -\mathbf{R}_{dq0}\boldsymbol{i}_{dq0}(n) + \boldsymbol{v}_{dq0,eq}^{hist} \tag{3.30}$$

where:

$$\mathbf{R}_{dq0} = \mathbf{R}_{ss} - \mathbf{R}_{sr}\mathbf{R}_{rr}^{-1}\mathbf{R}_{rs} \tag{3.31}$$

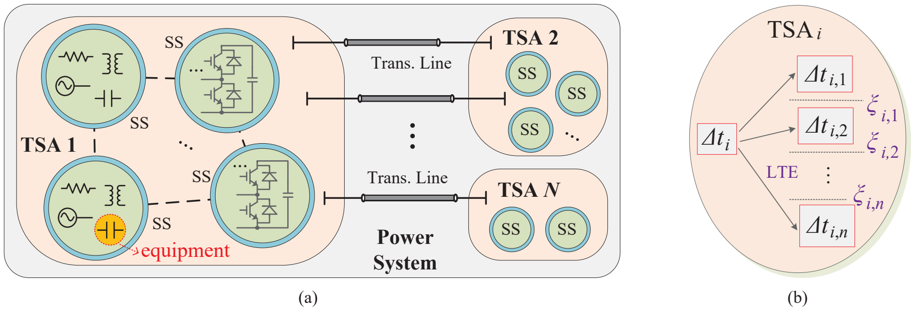
$$\boldsymbol{v}_{dq0,eq}^{hist} = -\mathbf{R}_{sr}\mathbf{R}_{rr}^{-1}\{-\boldsymbol{v}_r(n) + \boldsymbol{v}_r^{hist}\} + \boldsymbol{v}_{dq0}^{hist} \tag{3.32}$$

The equivalent voltage source $\boldsymbol{v}_{abc,eq}(n)$ and resistance $\mathbf{R}_{abc,eq}(n)$ can be obtained by transforming (3.30) into abc frame using the Park's transformation matrix $\mathbf{P}_n$:

$$\mathbf{R}_{abc,eq}(n) = \mathbf{P}_n^{-1}\mathbf{R}_{dq0}\mathbf{P}_n, \quad \boldsymbol{v}_{abc,eq}(n) = \mathbf{P}_n^{-1}\boldsymbol{v}_{dq0,eq}^{hist} \tag{3.33}$$

The change of ω is handled by the mechanical equation, and since ω at time t_n is unknown while solving the machine equations, it is predicted first and then updated iteratively by solving the network equations until convergence.

The AC4A type exciter model in PSCAD/EMTDC [12] is adopted to make the machine work stable. As can be seen in the diagram, it also involves the convolution operation when passing the transfer function that has the same format with transmission lines. Thus it has the same step-by-step calculation flow of convolution as described in the ULM model.



(a) (b)

Figure 3.4: TSA-based variable time-stepping control scheme: (a) global scheme, (b) local scheme.

3.3 Time-Step Configuration and Control Scheme

Generally, using a same time-step for the whole system is not practical for simulation of large-scale hybrid AC/DC grids because the time constants of various equipment are quite different. Although the whole system can be divided into subsystems based on the traveling wave line model or frequency-dependent line model and each subsystem may run in different time-steps [10], their time-step sizes also cannot be arbitrarily assigned in hardware due to the complexity of synchronization and necessity of storing the circuit parameters related to a specific time-step size. For example, if a subsystem uses a 3μs time-step and its connected subsystem uses a 7μs time-step, then it will be extremely complicated to synchronize the two subsystems because they could not reach at the same synchronization point after each time-step. Another reason for specific consideration on system decomposition and time-step size is that the transmission delay of the transmission lines between decomposed subsystems should be larger than the time-step size, which is prerequisite to decouple the connected systems.

Thus in general, there should be several pre-determined candidate time-steps for the hardware-based VTS simulation, and the time-step adaption is determined by the LTE of the previous step and the threshold. Considering that the state-variables and time-constants of different equipment (such as ULMs, UMs, and transformers) are also different, this work proposes a hierarchical VTS scheme shown in Fig. 3.4: in the first level, the power system is decomposed into several time-step areas (TSAs), and all the equipment models within the same TSA utilize the same time-step size; in the second level, each TSA is then decomposed into various subsystems (SSs) for parallel processing. After each time-step, each TSA ($A_1, ..., A_N$) compares the LTEs or DVDTs of the contained representative equipment to increase or decrease the time-step based on the time-step adaptation threshold. There are several LTE thresholds corresponding to different time-step sizes. For example, if there are n candidate time-step sizes ($\Delta t_1, ..., \Delta t_n$) then there will have ($n$-1) LTE thresholds ($\xi_1, ..., \xi_{n-1}$); then if the LTE is bigger than ξ_i and smaller than ξ_{i+1}, the time-step size will change to Δt_{i+1}.

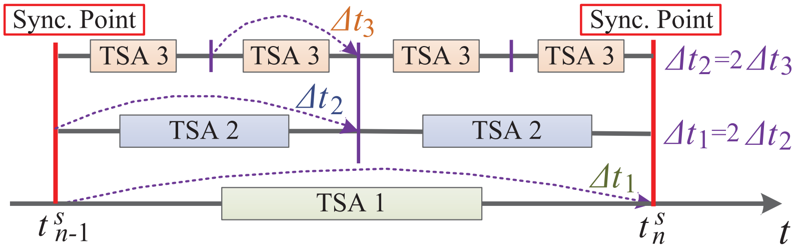
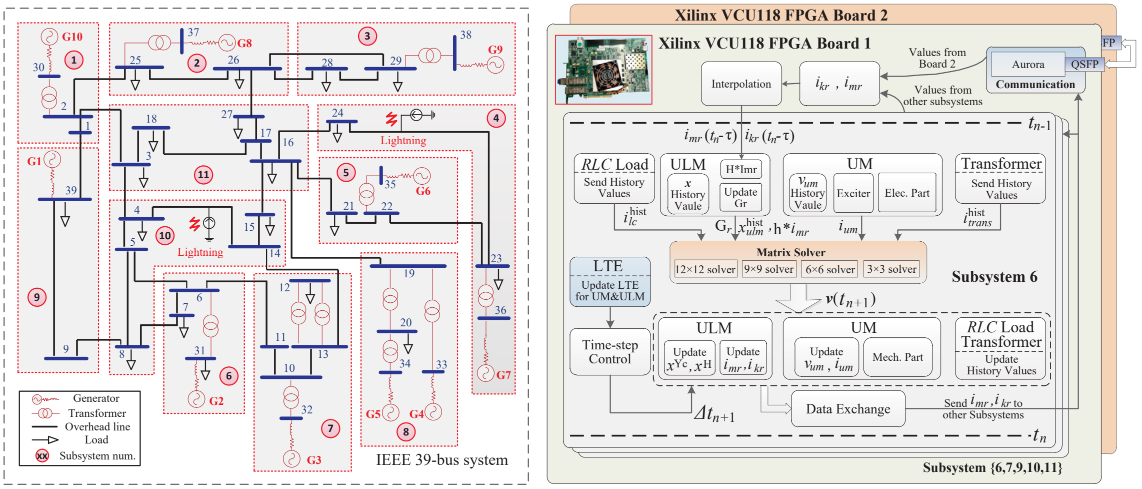


Figure 3.5: Example of synchronization between TSAs.

The interactions between the decomposed TSAs mainly include two aspects: 1) data exchange between TSAs, referring to the history items of the transmission line model; 2) time-step coordination and synchronization, indicating how to determine the synchronization point based on the dynamically changed time-step sizes. To handle these two aspects, the time-step set of each TSA should be first configured properly. Typically, the time-step size and LTE threshold are determined by user experience, experimental results, as well as the specific accuracy requirements. In this work, the time-step sizes for different TSAs always belong to the same time-step set ΔT=$\{\Delta t_{min} \times (2^0, 2^1, ..., 2^n)\}$, where Δt_{min} is minimum time-step size for the system. Then the large time-step size of difference TSAs are always multiple times of the smaller time-step size, which makes the synchronization between TSAs easy. As for the LTE thresholds determination, the LTE thresholds of linear elements are calculated and assigned based on the LTE equation (1.1) given the desired values of state variables; the thresholds of nonlinear elements or DVDT of MMCs are determined using the pre-simulation results to just demonstrate the work principle of the proposed VTS scheme. The above configurations are mostly performed manually in this work, however, how to determine the time-step size set and time-step changing threshold automatically based on precise mathematical analysis still remains a topic to be studied, and is left for future research.

The synchronization process refers to the concept that each TSA proceeds to the same synchronization point with different time-steps to exchange data with connected TSAs. The time-space between synchronization points is a variable, which is the maximum time-step of these TSAs after the last synchronization point. Since their large time-steps are always multiple times of the smaller one, the other TSAs with smaller time-steps are only required to execute several times to reach that synchronization point. An example is shown in Fig. 3.5, after time-step change at the $(n-1)^{th}$ synchronization point, the time-step of TSA-1 (Δt_1) becomes the largest one and thus determines the time-space to the next synchronization point. Then TSA-2 and TSA-3 execute several steps using their own time-step size to the next synchronization point so that all the TSAs could exchange their data and proceed the simulation.



(a) (b)

Figure 3.6: Test system and the hardware emulation on two interfaced FPGA boards.

3.4 Real-Time FPGA-Based Implementation

To test and verify the proposed VTS transmission line and machine models, the IEEE 39-bus power system [57] was first selected as the case study and implemented on FPGA. As shown in Fig. 3.6(a), the power system consists of 34 transmission lines, 10 generators, 12 transformers and 19 loads. Since the power transformer and RLC loads are not the main focus of this work, the lumped parameter transformer model based on admittance matrix representation [68] without saturation was utilized. When the time-step changes, the equivalent resistance of L and C will change, which only causes the change of the equivalent admittance matrix of the system.

3.4.1 Hardware Implementation

The Virtex UltraScale+ FPGA VCU118 board [52] used in this work contains both highly programmable UltraScale XCVU9P device and rich external resources. Considering the heavy computation task within each time-step and the large system scale, the multi-board solution is adopted, in which two VCU118 boards were interfaced through SFP transceivers, and each board only models a part of the system.

Subsystem Allocation. As shown in Fig. 3.6(a), the power system is divided into 11 TSAs taking advantage of the distributed transmission lines, and the connected TSAs exchange data (i_{mr} and i_{kr}) after calculation of each time-step. Since almost every generator is connected with a transformer, it is convenient to implement if every generator-transformer pair is allocated to an individual TSA, as marked as TSA $S_1 \sim S_6$ in Fig. 3.6(a). In each TSA, the transmission lines can again divide the TSA into subsystems, which effectively reduces the size of admittance matrix to be solved. For example, the maximum matrix size of $S_9 \sim S_{11}$ is 3×3, $S_1 \sim S_6$ is 6×6. S_7 and S_8 have the largest matrix to solve due the coupled transformers, which are 9×9 and 12×12 respectively. Since the transients happen in TSA S_{10} and S_4, to reduce the latency of data exchange between connected TSAs, the TSAs (S_6, S_7, S_9, S_{10}, S_{11}) that are connected with S_{10} are all allocated at the same FPGA board, and the other six TSAs are allocated to another FPGA board. The board-to-board communication is handled by the lightweight communication protocol Aurora [58], which is supported by different type of transceivers such as GTX, GTY and GTH, was used. The two boards are interfaced via two Aurora lanes, with 32b floating point data transferred in each lane.

Adaptive Time-Step Control. In this work, four time-step sizes are applied: 5μs, 10μs, 20μs and 40μs. Considering the matrix to be solved in different TSAs may be different and some TSAs contain generators that will occupy more time-space to compute, the time-step set of different TSAs are different. After the minimum computing latency of each TSA is obtained through hardware implementation, the proper time-step set can be determined: TSA $S_1 \sim S_6$ has the same time-step set {10μs, 20μs, 40μs}, TSA $S_7 \sim S_8$ has the same time-step set {20μs, 40μs}, and TSA $S_9 \sim S_{11}$ has the same time-step set {5μs, 10μs, 20 μs}. The time-step increase or decrease is determined by the LTE of the previous step and the threshold, wherein the threshold of various TSAs are also different. For the TSAs where transients happen, the LTE threshold is relatively smaller compared with that of the other TSAs.

Table 3.1: Demonstration of FTRT

Mode	Subsystem	Δt	t_p	Subsystem	Δt	t_p
RT	S_{10}	5μs	5μs	S_1	10μs	10μs
		10μs	10μs		10μs	10μs
FTRT2	S_{10}	10μs	5μs	S_1	20μs	10μs
		20μs	10μs		40μs	20μs
FTRT4	S_{10}	20μs	5μs	S_1	40μs	10μs

Faster-Than-Real-Time (FTRT). Although the system with fixed time-step can also be implemented in real-time, the main advantage of variable time-step is to accelerate the simulation progress without loss of accuracy. Thus the proposed VTS simulation could be even faster-than-real-time, which means, the actual processing time of each step is smaller than the time-step applied for the simulation. FTRT can be achieved by proper coordination between different TSAs. First, the maximum processing latency of a TSA limits the minimum time-step of the TSA, that implies that the large time-step is actually achieved by adding a time space between two steps. If the time space size can be narrowed synergistically, the simulation time will be reduced when the time-step changes to be larger than the minimum time-step. Secondly, the narrowed time space should be consistent among different TSAs, because the time-step of various TSAs may be different and it will easily cause inconsistency if the actual latency is not reduced proportionally.

Table 3.2: Processing Latency of Different Subsystems

Subsystem/Element	Latency	Subsystem/Element	Latency
Subsystem 1∼6	9.7μs	Subsystem 7	14.6μs
Subsystem 8	19.2μs	Subsystem 9	4.8μs
Subsystem 10∼11	3.3μs	Trans. Line	1.05μs

Table 3.3: Hardware Resource Utilization of the Case Study

Resource	VCU118-Board1	VCU118-Board2
Subsystems	S_6, S_7, S_9, S_{10}, S_{11}	S_1, S_2, S_3, S_4, S_5, S_8
LUT	767,174 (64.9%)	819,148 (69.3%)
FF	927,012 (39.2%)	1,014,011 (42.8%)
DSP	6,448 (94.3%)	6,714 (98.1%)
BRAM	570 (26.4%)	624 (28.9%)

Take TSAs S1 and S10 as an example, their time-step Δt and actual processing time t_p is illustrated in Table 3.1. The minimum time-step of S1 and S10 is near 10μs and 5μs respectively, thus the time-step of 10μs and 5μs can not be reduced. As shown in the second row, if S1 runs at 10μs, the latency of S10 with time-step size of 10μs also could not be reduced because the processing time of S1 cannot be smaller than 10μs. This indicates that no matter which TSA was running at the minimum time-step (typically is under transient condition), the latency of other TSAs cannot be reduced. Under normal steady-state conditions the time-step of all TSAs is usually changed to larger values, in this case, the actual latency can be reduced to make the simulation faster, as shown in the last three rows in Table 3.1. Note that FTRT2 means two times faster, and FTRT4 means four times faster.

3.4.2 Latency and Hardware Resource Utilization

The latency of different TSAs on hardware are recorded in Table 3.2, which indicates the minimum time-step that can be applied for different TSAs. For example, Subsystem S_8 consumes the most latency because it involves the iteration for UM and has the largest matrix (12×12) to solve. The hardware resource consumption on the two VCU118 boards is presented in Table 3.3.

3.5 4-Level Parallel GPU-Based Implementation

The dynamic parallelism feature of GPUs enables nested kernel functions execution, which is suitable for the hierarchical VTS processing architecture. Through proper system decomposition and GPU run-time configurations, the massively parallel VTS simulation can be achieved.

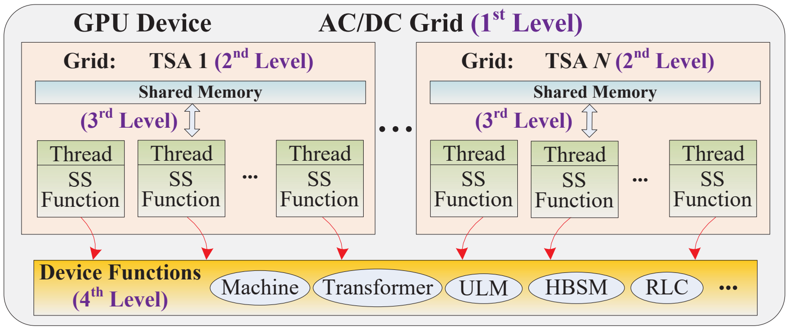


Figure 3.7: GPU-based VTS simulation: dynamic parallelism based simulation on GPUs.

3.5.1 GPU-Based VTS Simulation Architecture

The programming architecture and memory model of dynamic parallelism (introduced in Chapter 1) form the basics of parallel EMT simulation, which enables the proposed hierarchical VTS scheme to be executed in a massively parallel way. Considering the nested "parent-child" grids have almost the same hierarchical structure as the hierarchical VTS method, it is a natural idea to map the TSAs and subsystems into specific virtual processing units. The hybrid AC/DC system is divided into N TSAs using the equivalent circuit of transmission lines, and the hierarchy of the simulation is listed as follows and illustrated in Fig. 3.7:

1. **First-level Function**, the top kernel function used to simulate the whole system, which is called by the CPU program directly;
2. **Second-Level Function**, a TSA function used to simulate one specific TSA, containing SSs that have the same time-step sizes and changing rate during the simulation;
3. **Third-Level Function**, an SS function used to simulate a small circuit containing various power equipment or power electronic devices;
4. **Fourth-Level Function**, an equipment function used to calculate a specific device model such as machines, transformers, loads and power converters.

Generally, the first-level function must be a kernel function to run the simulation program on the GPU device, but the necessity of applying dynamic parallelism in the 2∼4 level function should be evaluated before the simulation since the overhead of launching kernels is not negligible. Assume the time of launching child kernels is t_c, and processing time of K SSs in a TSA is $t_{ss}^i, i = 1, .., K$, then the second-level function should be a kernel function running as the "child" grid of the first-level kernel function when:

$$t_c + max(t_{ss}^i) < \sum_{i=1}^{K} t_{ss}^i \tag{3.34}$$

The consideration of applying dynamic parallelism in each SS function and device function is the same as (3.34). Generally, once the scale of a TSA is determined, the granularity of system decomposition will significantly impact the application of dynamic parallelism. For example, if the system decomposition is fine-grained, which means an SS contains very limited equipment and a TSA is composed of large number of SSs, then the second-level function is usually running as a kernel function to improve the parallelism but the third-level kernel function is not required to be a kernel function. On the contrary, if there are not many SSs in each TSA and each SS contains numerous devices then the third-level kernel function should be generated.

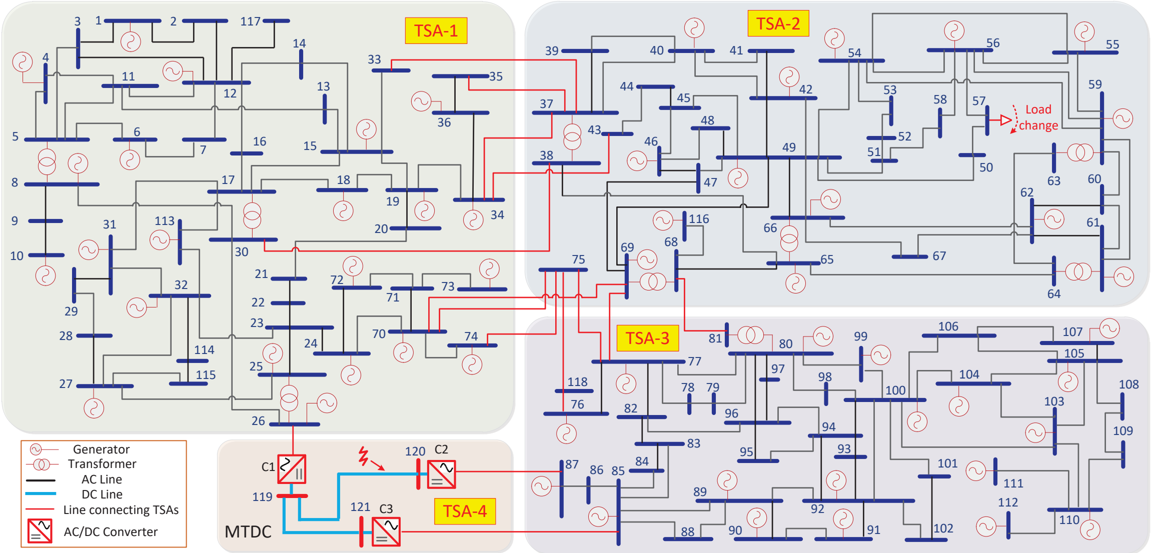


Figure 3.8: Topology of the AC/DC grid test case with time-step areas (TSAs).

Note that the equipment models can also run in parallel in the SS function even though the fourth-level kernel function is not used because if the SS function is a kernel function then it can be divided into blocks and threads to run the equipment models in parallel. In fact, the main application of dynamic parallelism for the fourth-level function is the frequency-dependent transmission line equipment model (FDLM), because it involves many convolution process that can run in massively parallel. If an SS is connected with many other SSs via transmission lines, then these transmission line equipment models should be executed in kernel functions to achieve fully parallel calculation and improve the overall performance.

3.5.2 GPU-Based Parallel Implementation

To test and verify the advantages of the proposed GPU-based VTS parallel simulation architecture, the integrated AC/DC grid composed of one IEEE 118-bus system [69] and three MMC converter stations is selected as the case study. As shown in Fig. 3.8, the AC power system consists of 118 buses, 54 generators, 177 lines, 9 transformers, and 91 loads; the DC power system consists of three AC/DC converter stations connected via DC transmission lines.

The GPU device used in this chapter is the NVIDIA Tesla V100 GPU featured with 5120 cores, 16GB HBM2 memory and a memory path with bandwidth of 900GB/s. The V100 GPU's 7.0 computation capability enables the application of dynamic parallelism, and the large number of cores allows the utilization of detailed equipment models and massively parallel calculation of large-scale EMT simulation. Using such an architecture, the hybrid AC/DC test system can be mapped and computed efficiently.

System Partition. To decompose the system into TSAs, two problems are involved: 1) determining the number of TSAs; 2) partitioning the topology given the number of TSAs. The number of TSAs is configured by users, for example, in the test system in Fig. 3.8, the 118-bus is partitioned into three TSAs to demonstrate the hierarchical time-step control scheme. But it can also be configured as just one TSA, if only distinguishing the AC system from DC system. Once the number of TSAs is determined, how to partition the topology is an optimization problem when taking the synchronization latency into account. Although the partitioning for the case study in Fig. 3 is performed manually for simplicity, minimizing the connection links between different TSAs may be a good optimization goal if automatic partition algorithm is exploited. Based on the above discussion, the hybrid AC/DC grid is firstly decomposed into four TSAs to apply the proposed hierarchical VTS method on the GPU cores, where the 118-bus system is decomposed into three TSAs and the DC system is separated as one TSA. Every TSA is only connected with two adjacent TSAs to reduce the data exchange. Then each TSA is decomposed into small SSs based on the connected transmission lines since the two ports of the transmission line can be calculated separately. The principle of SS decomposition is decomposing the system as fine-grained as possible according to the abundant GPU cores. Therefore, if there are transmission lines connecting SSs, then these SSs could be decomposed for parallel computing.

After decomposition, TSA-1 contains 42 SSs (45 buses), TSA-2 contains 30 SSs (35 buses), TSA-3 contains 37 SSs (38 buses), and TSA-4 contains 3 SSs. The SSs in AC side have simple equipment and small matrix (6×6 in maximum) to solve; however, in TSA-4, each SS refers to an MMC converter with DC transmission line connections, which involves heavy computational task of the equivalent circuit calculation, value-level switch control and system-level power flow control.

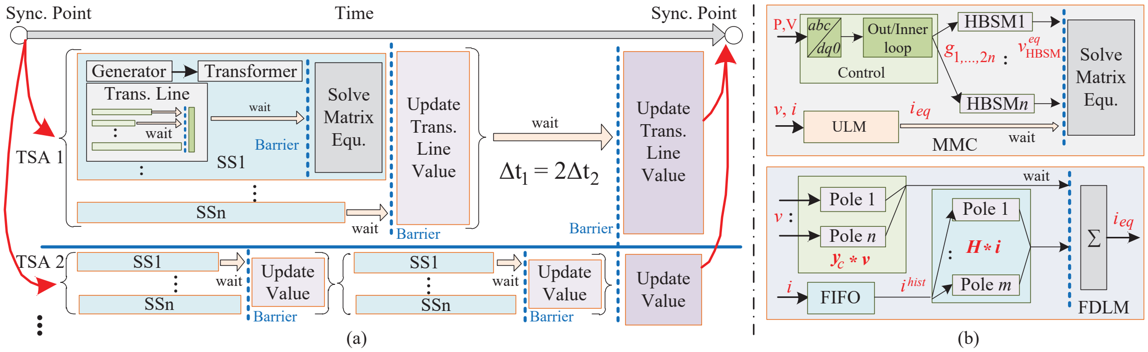


Figure 3.9: GPU implementation: (a) detailed parallel processing on GPU; (b) parallel computing for MMC and ULM.

Processing Hierarchy. The processing hierarchy is illustrated in Fig. 3.9 (a). The 1^{st}-level kernel function run the TSA functions in parallel, and at the synchronization point the TSAs exchange the transmission line data for consistence. The application of dynamic parallelism (DP) in 2∼4 level function is decided by evaluating the processing time over equation (3.34), and is shown in Table 3.4. The number of used cores N_c in each level is also listed, note that the four grids are generated in the first level parallelism as there are four TSAs divided. For the 2^{nd}-level TSA-1∼3 function, K (number of SSs) is so large that the parallel processing will benefit more and thus dynamic parallelism is necessary; for the TSA-4 function although K (=3) is small, the long processing time of each SS makes dynamic parallelism also necessary to improve the overall performance. Note that all of the parallelism in each level function ends with a barrier, at which point all the threads must reach to synchronize the data with each other.

Table 3.4: Application of Dynamic Parallelism and Cores Used in Each Level

DP	TSA-1 (N_c)	TSA-2	TSA-3	TSA-4
2^{nd}-DP/TSA	√ (42)	√ (30)	√ (37)	√ (3)
3^{rd}-DP/SS	optional	optional	optional	√ (97)
4^{th}-DP/Device	optional	optional	optional	optional

Within each SS, parallel processing is required if containing many equipment. For example, the SS composed of bus 68 and bus 69 has a coupled transformer and generator, and seven transmission line connections. The generator-transformer pair must be executed in serial, but the transmission lines can run in parallel. However, the SS only composed of Bus 2 does not require parallel processing of equipment model because it only contain two transmission lines. Thus the application of dynamic parallelism for 3^{rd}-level SS functions is optional in TSA-1∼3, but is necessary in TSA-4 due to the huge amount of HBSMs in MMC converters. As shown in Fig. 3.9(b), although the control logic can not expose enough parallelism, the HBSMs should run in parallel to obtain the equivalent voltage source of each HBSM. The 4^{th}-level device function refers to the computation of detailed equipment models such as the synchronous machine, transformer, ULM and HBSM, and the parallel processing is also optional because only the ULM model requires to apply dynamic parallelism due to the benefits by applying parallel calculation for convolution process while the other equipment models can not run in parallel sufficiently.

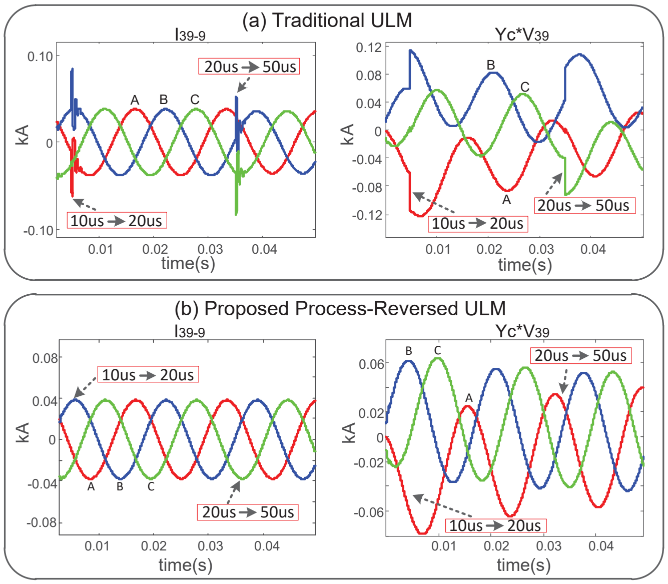


Figure 3.10: Waveforms under time-step change operation: (a) traditional ULM model; (b) proposed process-reversed ULM model.

Time-Step Size and Control. As analyzed in Section 3.3, the time-step sizes of different TSAs always belong to the same time-step set ΔT^{sys}=$\{\Delta t_{min} \times (2^0, 2^1, ..., 2^n)\}$. In this work, $n = 4$, and Δt_{min} is set at 10μs for the system-level simulation. The time-step set for device-level simulation is ΔT^{dev}=$\{$0.05μs, 0.1μs, 0.2μs, 0.5μs$\}$. Note that in the device-level simulation, ΔT^{dev} is only applied for MMC converters, the 118-bus system is still simulated with ΔT^{sys}. The time-step increase or decrease is determined by comparing the LTE or DVDT of the previous step and the predetermined threshold, wherein the threshold of various subsystems are also different. For the TSAs where transients happen, the threshold is relatively smaller compared with that of the other TSAs. Since the unknown variables and discretization methods applied vary between equipment models, there are different thresholds for different equipment and the time-step of a TSA will change no matter which equipment exceeds its own LTE threshold.

3.6 Results and Verification

The two test cases described above were simulated on the FPGA and GPU platform respectively and the results are compared with PSCAD/EMTDC® to show the effectiveness of the proposed VTS models. Note that PSCAD/EMTDC® is only a representative of fixed time-stepping EMT simulation tools, and it does not matter if other software are used for comparison because the ULM and UM model with fixed time-steps are both commonly-used models. The clock frequency of FPGA boards is set at 100 MHz.

3.6.1 Verification of the ULM Model

Before the ULM model is integrated into the IEEE 39-bus system, it is necessary to verify the stability of the propose process-reversed method. Note that the UM model in VTS is stable because the state variable remains the same when the time-step changes. To solely validate the proposed ULM model, the subsystem S_9 in Fig. 3.6(a) with ideal voltage source is simulated while keeping the load and connection with other subsystems as open circuit. At time 0.005s, the time-step changes from 10μs to 20μs; and at time 0.035s, the time-step changes from 20μs to 50μs.

The results of i_{39-9} (from Bus39 to Bus9) and $\mathbf{Y}_c * v_{39}$ using the traditional model are shown in Fig. 3.10(a), when the time-step changes, there will have a large oscillation. Since the actual state variable changes when the time-step changes, the initial value of the new state variable will be incorrect, which causes an abrupt change of the convolution results. However, when using the proposed process-reversed model, it can be observed that the convolution results will remain continuous and stable, which results in a stable current i_{39-9} when the time-step changes.

3.6.2 Real-Time Emulation Results of IEEE 39-Bus System on FPGA

To simulate the dynamic behavior of the system, the lightning surge at phase C of transmission line L$_{4-14}$ (between bus 4 and bus 14) and L$_{23-24}$ is chosen as the transient test. The results are evaluated by the proposed emulator and PSCAD/EMTDC® that used a fixed time-step of 10μs while the proposed emulator used adaptive time-steps described in Section 3.4. The standard $10/350\mu s$ lightning surge current [59] is applied in this work, given as:

$$I_{LS}(t) = CI_m(t/\tau_1)^k e^{\frac{-t}{\tau_2}}/[1+(t/\tau_1)^k] \tag{3.35}$$

where the coefficient $C = 1.075$, $k = 10$; the time constant $\tau_1 = 19\mu$s, $\tau_2 = 485\mu$s; the maximum value of the surge current $I_m = 3$kA. The lightning surge current at L$_{4-14}$ and L$_{23-24}$ is applied at exactly 2s and 2.5s of the simulation to demonstrate the transient behavior of ULM and UM respectively.

First, the simulation is executed in real-time and compared with PSCAD and the hardware emulator with fixed time-step (FTS), and the results are recorded in Fig. 3.11. The

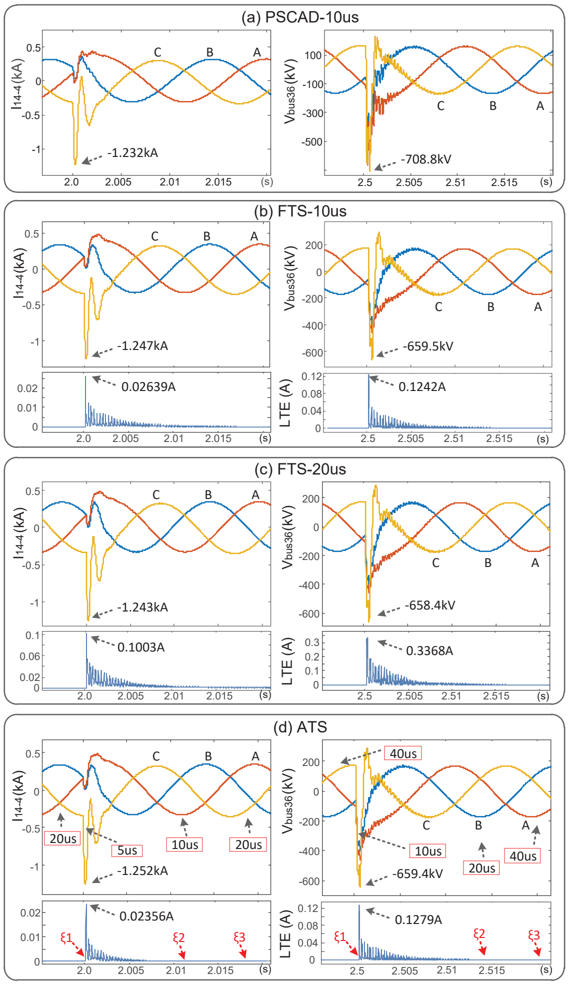


Figure 3.11: Lightning transient results of i_{14-4} and v_{bus36}. (a) PSCAD results with 10us fixed time-step; (b)(c) FPGA-based emulator with 10us and 20us time-steps; (d) FPGA-based emulator with VTS.

FTS emulator applies the same time-step for the entire system. Since the maximum processing latency of the decomposed subsystems is near 20μs, FTS-20μs is actually the FTS emulator with minimum time-step size that can be achieved in real-time. But through the proposed subsystem-based VTS scheme, the entire system can be simulated in real-time with smaller time-step sizes by applying different variable time-steps for different subsystems. From Fig. 3.11(a)(b) with the same time-step size it can be observed that, the peak values of the machine terminal voltage under transients are -708.8kV and -659.5kV respectively, which indicates that the UM model implemented in this work is more stable than PSCAD.

In Fig. 3.11(b)(c), the emulator with 10μs time-step size produces a smaller LTE, and the peak value of LTEs of ULM and UM reduces 63.1% and 73.7% respectively compared with that using a 20μs time-step. That means reducing the time-step size will generate a more accurate result. The results using VTS are shown in Fig. 3.11(d), from which we can see

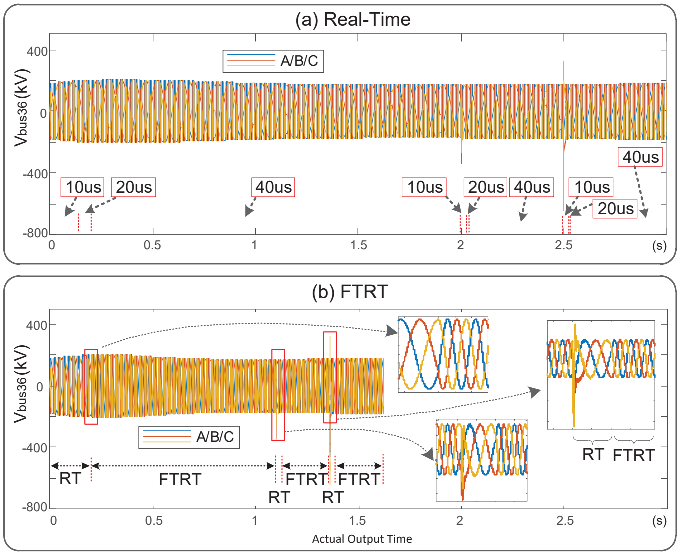


Figure 3.12: Demonstration of FTRT results. (a) Real-time results in VTS; (b) Results of FTRT2 mode in VTS.

the large time-step is applied under normal operation, and when the transient happens, the time-step is reduced automatically according to the LTE. For example, the time-step is 20μs for L$_{14-4}$ before the lightning, and when the transient happens, the LTE directly rises to an extremely large value that exceed the max threshold ξ_1 immediately, then the time-step reduces to the minimum one (5μs) directly. When the LTE is below ξ_2, the time-step increases into a larger one (10μs), and as the LTE is reduced below the minimum threshold ξ_3, the time-step regains to the maximum one (20μs). Since the time-step size of ULM reduces to 5μs under transients, the maximum LTE reduces 76.5% compared with that of FTS-20μs. The UM has the same process although the time-step set is not the same since L$_{14-4}$ and L$_{23-24}$ belongs to the different subsystems.

The UM model cannot be executed in a time-step less than 10μs, thus the minimum time-step size of UM is 10μs, resulting in a LTE 62.0% smaller than that of FTS-20μs. The LTE of VTS is a little different with FTS-10μs because the time-step sizes in VTS also vary between subsystems while FTS-10μs applies the same time-step for the whole system.

Secondly, to show the acceleration of VTS simulation scheme, the test case is emulated in FTRT2 mode (shown in Table 3.1) and compared with the results of real-time simulation. Figure. 3.12 demonstrates the Bus 36 voltage of subsystem S_4, it can be observed the two versions have the same numerical results because the applied time-step sizes and parameters are the same. But due to the different actual processing time of each step, the output rate of the results are different. The FTRT2 version can produce waveforms at a faster rate than the real-time simulation under normal operations, and when the lightning transient happens, the two versions have the same output rate. In this work, only when the simulation time-step increases to 40μs under normal conditions, the results output rate is accelerated by two times. The duration of the period when the time-step size increases to 20μs is quite short, thus it is not necessary to accelerate it due to the extra complexity of simulation time control logic. Besides, the maximum time-step is only four times of the minimum time-step in this work, it can be expected that if the maximum time-step is larger, the output rate will be even faster.

Table 3.5: Execution Time and Speed-up of Different Methods for 10s Simulation

Version	FTS-CPU (Base)	VTS-CPU/Sp-CPU	VTS-DP0/Sp-0	VTS-DP1/Sp-1
System-Level	641.4 s	82.2s/7.8	356.3s/1.8	85.5s/7.5
Device-Level	4142.7s	881.4s/4.7	3452.3s/1.2	1183.6s/3.5
Version	FTS-CPU (Base)	VTS-DP2/Sp-2	VTS-DP3/Sp-3	VTS-DP4/Sp-4
System-Level	641.4 s	30.7s/20.9	1.56s/411.2	1.51s/424.8
Device-Level	4142.7s	505.2s/8.2	20.4s/203.3	20.3s/204.1

3.6.3 Latency and Speed-Up of AC/DC Grid on GPU

The hybrid AC/DC test system for the GPU-based parallel VTS simulation architecture is simulated and the latencies are recorded and compared. Generally, there are 2-stage speed-ups of the GPU-based VTS simulation that should be evaluated compared to the CPU-based FTS simulation: the speed-up by applying VTS scheme on CPU, and the speed-up by conducting the 4-level parallel simulation on GPU. The 2-stage speed-ups of system-level and device-level simulations are all recorded in Table 3.5, and the duration of simulation is 10s. Note that the CPU simulation time is measured on the developed C-code program on Visual Studio® 2018 but not on existing EMT simulation software because the C-program is more dedicated to the case study and can achieve lower latencies. The time-step of FTS simulation is set at 20μs for system-level simulation; in the device-level simulation, the time-step is set at 0.1μs for MMC converters and 20μs for 118-bus system. The VTS simulation uses the variable time-steps assigned in Chapter 3.5.2.

Since the VTS simulation uses large time-steps to proceed under the steady-state conditions, the speed-up of the VTS simulation on CPU is nearly 8 and 5 for system-level and device-level simulation respectively, which is fairly considerable. Parallel processing on GPU will accelerate the simulation process, however, the performance slows down significantly if only using GPU but not applying the parallel mechanism, denoted as Sp-0. The reason is that the frequency of GPU is not as high as that of the CPU. For the system-level simulation, if the 1^{st}-level DP is utilized, the speed-up increases to 7.5, but the acceleration effect is still not obvious compared to CPU because only four TSAs run in parallel but each TSA still runs in serial. The 2^{nd}-level should introduce massive parallelism due to the numerous SSs in AC system, but the speed-up is not very high because TSA-4 can only be decomposed into three SSs and the lowest speed of TSAs determine the overall speed-up due to the requirement of synchronization. Once the 3^{rd}-level DP is applied, the speed-up increases dramatically because of the parallel computation of HBSMs in MMC and device functions in TSA-1$\sim$3. The speed-up by adding the 4^{th}-level DP is not as obvious as the 2^{nd}-level and 3^{rd}-level DP, however, it is also necessary to fully exploit the abundant GPU cores.

For device-level simulation, the speed-up is not as large as that of system-level simulation because TSA-4 consumes much more latency than TSA-1$\sim$3 then the speed-ups of TSA-1$\sim$3 can not be revealed in the overall performance. Compared to the FTS simulation on CPU, the overall speed-ups of GPU-based VTS simulation are 424.8 and 204.1 respectively for system-level and device-level simulation.

3.7 Summary

In this section, the variable time-stepping universal transmission line model and universal machine model are proposed, and the real-time and faster-than-real-time emulation architectures are proposed for FPGA based implementation, while the 4-level parallel simulation architecture is proposed for GPU based implementation. In the proposed ULM model, through a novel process-reversal of the traditional model, the stability is significantly improved during the time-step change operation. By using LTE as the time-step change criteria, the time-step can be adjusted properly when the transient happens. The hardware emulator for large-scale power systems is presented, in which the system is divided into small TSAs and each TSA maintains its own time-step set and LTE threshold. By elaborate coordination between TSAs, the *faster-than-real-time* mode and 4-level parallel VTS simulation can be achieved on FPGA and GPU respectively. The hardware resource cost, processing delay and speed-ups of test power systems are evaluated, which shows the practicality of variable time-stepping scheme for EMT simulation.

4

Linking-Domain Extraction Decomposition Method for Parallel Electromagnetic Transient Simulation of AC/DC Networks

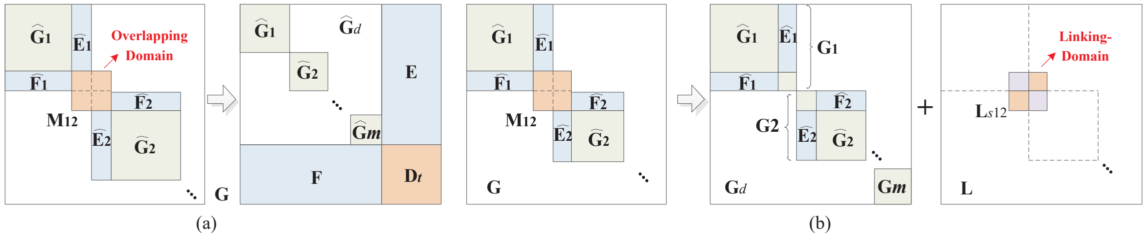
Domain decomposition of the network conductance matrix is one of the efficient approaches to solve large-scale networks in parallel. In this work, a novel Linking-Domain Extraction (LDE) based decomposition method is proposed, in which the network matrix is expressed as the sum of a linking-domain matrix (LDM) and a diagonal block matrix (DBM) composed of multiple block matrices in diagonal. Through mathematical analysis over LDM, one *lemma* about the nature of LDM and its proof are proposed. Based on this *lemma*, the general formulation of the inverse matrix of the sum of LDM and DBM can be found using the Woodbury matrix identity, and based on the formulation the network matrix inversion can be directly computed in parallel to significantly accelerate the matrix inversion process. Test systems were implemented on both the FPGA and GPU parallel architectures, and the simulation results and speed-ups over the SC method and Gauss-Jordan elimination demonstrate the validity and efficiency of the proposed LDE method.

4.1 Schur Complement Method

The Schur complement (SC) method is one of the representative non-overlapping domain decomposition methods. Once the network equations have been gathered, the solution of a linear network reduces to a solution of the linear algebraic system:

$$\mathbf{G}\,\boldsymbol{v} = \boldsymbol{i}^{eq} \tag{4.1}$$

The SC method is a non-overlapping method, which means the network conductance



(a)

(b)

Figure 4.1: Example of matrix decomposition: (a) Schur decomposition method; (b) Proposed LDE method.

matrix $\mathbf{G}$ is decomposed into m uncoupled small block matrices ($\hat{\mathbf{G}}_1, \hat{\mathbf{G}}_2, ..., \hat{\mathbf{G}}_m$). The overlapping domains between block matrices are moved to the overlapping domain matrix $\mathbf{D}_t$, as shown in Fig. 4.1(a). The nodes located in $\mathbf{D}_t$ actually represent the interface nodes used to connect decoupled subsystems. Applying the block matrix multiplication, the following equations can be obtained:

$$\begin{bmatrix} \hat{\mathbf{G}}_1 & 0 & \cdots & 0 & \mathbf{E}_1 \\ 0 & \hat{\mathbf{G}}_2 & \cdots & 0 & \mathbf{E}_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & \hat{\mathbf{G}}_m & \mathbf{E}_m \\ \mathbf{F}_1 & \mathbf{F}_2 & \cdots & \mathbf{F}_m & \mathbf{D}_t \end{bmatrix} \begin{pmatrix} \boldsymbol{v}_{g1} \\ \boldsymbol{v}_{g2} \\ \vdots \\ \boldsymbol{v}_{gm} \\ \boldsymbol{v}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{i}_{g1} \\ \boldsymbol{i}_{g2} \\ \vdots \\ \boldsymbol{i}_{gm} \\ \boldsymbol{i}_t \end{pmatrix} \tag{4.2}$$

$$(\mathbf{D}_t - \mathbf{F}\hat{\mathbf{G}}_d^{-1}\mathbf{E})\,\boldsymbol{v}_t = \boldsymbol{i}_t - \mathbf{F}\hat{\mathbf{G}}_d^{-1}\boldsymbol{i}_g \tag{4.3}$$

$$\hat{\mathbf{G}}_d\,\boldsymbol{v}_g = \boldsymbol{i}_g - \mathbf{E}\,\boldsymbol{v}_t \tag{4.4}$$

where $\hat{\mathbf{G}}_d$, $\mathbf{F}$ and $\mathbf{E}$ are the combination of the corresponding block matrices, and $\boldsymbol{v}_g, \boldsymbol{i}_g$ are the combination of the corresponding node voltages and current injections.

The computation process is executed as follows:

1. compute $\hat{\mathbf{G}}_d^{-1}$ and $\hat{\mathbf{G}}_d^{-1}\boldsymbol{i}_g$;
2. compute $\mathbf{F}\hat{\mathbf{G}}_d^{-1}\boldsymbol{i}_g$ and $\mathbf{F}\hat{\mathbf{G}}_d^{-1}\mathbf{E}$;
3. solve equation (4.3) to get $\boldsymbol{v}_t$;
4. solve $\boldsymbol{v}_g = \hat{\mathbf{G}}_d^{-1}\boldsymbol{i}_g - \hat{\mathbf{G}}_d^{-1}\mathbf{E}\,\boldsymbol{v}_t$.

Parallel computing can be exploited based on the computation that involves the diagonal block matrix $\hat{\mathbf{G}}_d$, such as computing $\hat{\mathbf{G}}_d^{-1}$, $\mathbf{F}\hat{\mathbf{G}}_d^{-1}\mathbf{E}$ and $\boldsymbol{v}_g$ $(= \hat{\mathbf{G}}_d^{-1}\boldsymbol{i}_g - \hat{\mathbf{G}}_d^{-1}\mathbf{E}\,\boldsymbol{v}_t)$. However, it can be observed that the SC method could not obtain $\mathbf{G}^{-1}$ directly due to the involvement of the current injections even though $\hat{\mathbf{G}}_d^{-1}$ and the other matrices could be computed in advance for linear circuits, and thus in each time-step the four processing steps could not be avoided. In addition, when the number of decomposed matrices increases, the amount of interface nodes will increase quickly, which significantly influences the overall performance due to the large computational effort in solving $\boldsymbol{v}_t$.

4.2 Proposed Linking-Domain Extraction based Decomposition Method

Focusing on the overlapping domain between subsystems, the linking-domain matrix is defined and extracted from the original network matrix. Through mathematical analysis over the linking-domain matrix, an important *lemma* is put forward, based on which the inverse matrix of the network matrix can be computed in parallel.

4.2.1 LDE Matrix Decomposition

Different from the logic of the SC method that operates on the decomposition of the original conductance matrix, the proposed LDE method decomposes the original matrix into two separate matrices: the diagonal block matrix $\mathbf{G}_d$ and the linking-domain matrix **L**.

$$\mathbf{G} = \mathbf{G}_d + \mathbf{L} \tag{4.5}$$

As shown in Fig. 4.1(b), the sizes of the decomposed block matrices $(\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_m)$ are larger than using the SC method, because in the SC method all the overlapping areas and their corresponding rows and columns are removed from the block matrices, while in the LDE method only the linking-domain matrix is extracted from the original matrix and the size of each block matrix does not decrease. Note that the linking-domain has a different meaning from the overlapping domain, because the elements in the overlapping domain actually are the sum of the corresponding values in the linking-domain matrix and the block matrices, as marked in Fig. 4.1.

The construction of **L** matrix is as follows: let the linking-domain matrix contain a small matrix with non-zero diagonal elements ($\mathbf{L}_s$) and the other all-zero matrices, as illustrated in Fig. 4.1(b). The location of $\mathbf{L}_s$ is the same as the overlapping domain in **G** and the size of $\mathbf{L}_s$ is $(n_1 + n_2) \times (n_1 + n_2)$, where n_1 is the number of interface nodes belonging to the first subsystem and n_2 is the number of interface nodes belonging to the second subsystem (taking two decomposed subsystems as an example). $\mathbf{L}_s$ can be regarded as the combination of four block matrices: the top left block matrix and the bottom right block matrix are both diagonal matrices, and the top right and bottom left block matrices are transpose of each other. The top right $(n_1 \times n_2)$ and bottom left $(n_2 \times n_1)$ block matrices are the same as those of the overlapping domain matrix, and after the two parts are assigned, the diagonal elements in the top left $(n_1 \times n_1)$ and bottom right $(n_2 \times n_2)$ block matrix are determined as follows:

$$\mathbf{L}_s = \begin{bmatrix} \Sigma_{1,1} & 0 & \cdots & 0 & \sigma_{1,1} & \cdots & \sigma_{1,n_2} \\ 0 & \Sigma_{1,2} & \cdots & 0 & \sigma_{2,1} & \cdots & \sigma_{2,n_2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \Sigma_{1,n_1} & \sigma_{n_1,1} & \cdots & \sigma_{n_1,n_2} \\ \sigma_{1,1} & \sigma_{2,1} & \cdots & \sigma_{n_1,1} & \Sigma_{2,1} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \sigma_{1,n_2} & \sigma_{2,n_2} & \cdots & \sigma_{n_1,n_2} & 0 & \cdots & \Sigma_{2,n_2} \end{bmatrix} \tag{4.6}$$

$$\Sigma_{1,i} = -\sum_{j=1}^{n_2} \sigma_{i,j}, \quad \forall\, 1 \leqslant i \leqslant n_1 \tag{4.7}$$

$$\Sigma_{2,i} = -\sum_{j=1}^{n_1} \sigma_{j,i}, \quad \forall\, 1 \leqslant i \leqslant n_2 \tag{4.8}$$

Since the inverse of the diagonal block matrix $\mathbf{G}_d$ can be calculated in parallel, if the inverse matrix of $\mathbf{G} = \mathbf{G}_d + \mathbf{L}$ can also be calculated in parallel, then the simulation process can be accelerated significantly. Therefore, the core task of LDE method is to find a general formulation of the inverse of $\mathbf{G}$. Fortunately, taking advantage of the special features of linking-domain matrix, the relationship between $\mathbf{G}_d^{-1}$ and $\mathbf{G}^{-1}$ can be found.

4.2.2 Mathematical Analysis over LDM

For a common linear network, the linking-domain matrix $\mathbf{L}$ ($N \times N$) has some specific characteristics. For example, $\mathbf{L}$ is a symmetric matrix, and the rank of $\mathbf{L}$ is usually smaller than N (except that all of the nodes are interface nodes). But more importantly, it can be observed that the sum of each row (or column) of $\mathbf{L}$ is always equal to 0, that is:

$$\sum_{j=1}^{N} \mathbf{L}_{i,j} = 0, \quad \forall\, 1 \leqslant i \leqslant N \tag{4.9}$$

$$\sum_{i=1}^{N} \mathbf{L}_{i,j} = 0, \quad \forall\, 1 \leqslant j \leqslant N \tag{4.10}$$

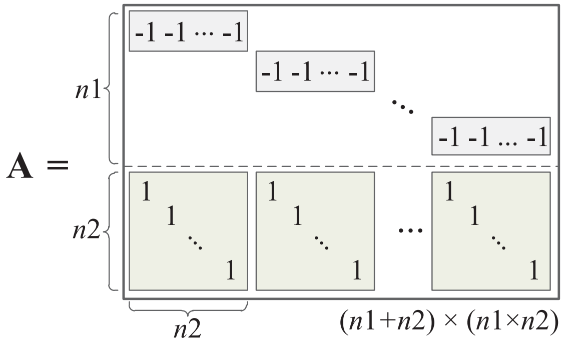
Based on this observation, *Lemma 1* that decomposes the linking-domain matrix into the multiplication of three matrices is proposed.

Lemma 1: The linking-domain matrix $\mathbf{L}$ can be expressed as $\mathbf{L} = \mathbf{C}\Lambda\mathbf{C}^T$, where Λ is a diagonal matrix with non-negative real numbers on the diagonal, and the transform matrix $\mathbf{C}$ is a rectangular matrix of which the element values are only equal to 1, -1, or 0.

Proof: To simplify the proving process, we first start with the two connected subsystems and then extend it to multiple subsystems.

Step 1: prove that $\mathbf{L}_s = \mathbf{A}\Delta\mathbf{A}^T$, where the size of Δ is $(n_1 \times n_2) \times (n_1 \times n_2)$ and the size of transform matrix $\mathbf{A}$ is $(n_1 + n_2) \times (n_1 \times n_2)$. More specifically, Δ and $\mathbf{A}$ can be expressed as:

$$\Delta = \begin{bmatrix} -\sigma_{1,1} & 0 & \cdots & 0 & 0 \\ 0 & -\sigma_{1,2} & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & -\sigma_{n_1,n_2-1} & 0 \\ 0 & 0 & 0 & \cdots & -\sigma_{n_1,n_2} \end{bmatrix} \tag{4.11}$$



(4.12)

The elements of the diagonal matrix Δ are exactly the negative values of the elements in the top right block matrix of $\mathbf{L}_s$. Note that the blank area in transform matrix $\mathbf{A}$ is filled with zero. The transform matrix $\mathbf{A}$ shown in (4.12) has some important features:

$$\mathbf{A}^{(i,r)} = -1, \forall\, 1 \leqslant i \leqslant n_1, \quad (i-1)n_2 < r \leqslant i \times n_2 \tag{4.13}$$

$$\mathbf{A}^{(i,r)} = 1, \forall\, n_1 < i \leqslant n_1 + n_2, \quad r = n_2 \times \{0, 1, ..., n_1 - 1\} + (i - n_1) \tag{4.14}$$

$$\mathbf{A}^{(i,r)}\mathbf{A}^{(j,r)} = 0, \forall 1 \leqslant r \leqslant (n_1 \times n_2), \quad \forall 1 \leqslant i \neq j \leqslant n_1 \,\text{or}\, n_1 < i \neq j \leqslant n_1 + n_2 \tag{4.15}$$

$$\mathbf{A}^{(i,r)}\mathbf{A}^{(j,r)} = -1,\ \forall 1 \leqslant i \leqslant n_1, \quad (i-1)n_2 < r \leqslant i \times n_2;\ j = r - n_2 \times (i-1) + n_1 \tag{4.16}$$

Then the elements in $\mathbf{L}_s$ can be expressed as:

$$\mathbf{L}_s^{(i,j)} = -\sum_{r=1}^{n_1 \times n_2} \mathbf{A}^{(i,r)}\mathbf{A}^{(j,r)}\sigma_{p,q},\ \forall 1 \leqslant i, j \leqslant (n_1 + n_2) \tag{4.17}$$

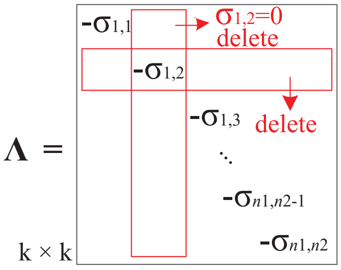
$$p = ceil(r/n_2),\ \ q = r - n_2 \times (p-1) \tag{4.18}$$

where the function $ceil(r, n_2)$ rounds up the division of r by n_2 to an integer, which means the maximum values of p and q are n_1 and n_2 respectively. For the diagonal elements, the correctness of the matrix elements values can be verified based on (4.9)(4.10)(4.13)(4.14). For example, if $i = 2$, then $\mathbf{L}_s^{(2,2)} = \Sigma_{1,2} = -(\sigma_{2,1} + \sigma_{2,2} + ... + \sigma_{2,n_2})$, which matches with (4.17). For the other non-diagonal elements in $\mathbf{L}_s$, the correctness can also be verified based on (4.15)(4.16).

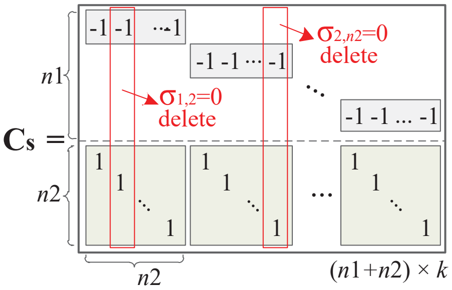
Step 2: delete the diagonal elements with zero values in Δ to generate Λ, and delete their corresponding columns in $\mathbf{A}$ to generate $\mathbf{C}_s$, then:

$$\mathbf{L}_s = \mathbf{C}_s \Lambda \mathbf{C}_s^T \tag{4.19}$$

where the sizes of Λ and $\mathbf{C}_s$ are $k \times k$ and $(n_1 + n_2) \times k$ respectively:



(4.20)



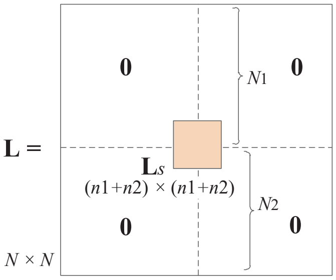
(4.21)

Typically, the new transform matrix size k is much smaller than $n_1 \times n_2$, because most elements in the $(\sigma_{1,1}, ..., \sigma_{n_1,n_2})$ are zero if the connection is sparse. In fact, $k = n_1 \times n_2$ is only valid for the extreme case that all the interface nodes are connected with each other via a conductance.

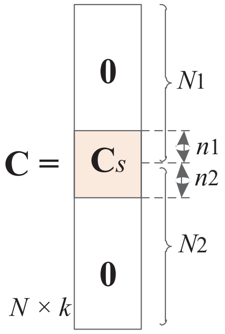
The equation (4.19) can be verified in the same way as in Step 1, because all the items in (4.17) corresponding to the deleted elements are equal to zero, and thus deleting these elements does not influence the result of $\mathbf{L}_s$.

Step 3: extend the small matrices $\mathbf{L}_s$ and $\mathbf{C}_s$ into bigger matrices $\mathbf{L}$ $(N \times N)$ and $\mathbf{C}$ $(N \times k)$ with zero elements, then:

$$\mathbf{L} = \mathbf{C}\Lambda\mathbf{C}^T \tag{4.22}$$



(4.23)



(4.24)

This step just adds some block matrices with all-zero elements to extend $\mathbf{L}_s$ and $\mathbf{C}_s$, which actually increases the size of the resulting matrix $\mathbf{C}\Lambda\mathbf{C}^T$ while maintaining the correct values at the non-zero area in $\mathbf{L}$.

The above three steps actually provides the proof of *Lemma 1* in the situation of two decomposed matrices. For the case of multiple decomposed matrices, the proof just has the same logical steps: the linking-domain matrix $\mathbf{L}$ is the sum of more than one small block matrix ($\mathbf{L}_{s,(i,j)}$), where the subscript (i, j) refers to the linking-domain between subsystem S_i and subsystem S_j. Accordingly, the transform matrix $\mathbf{C}$ is also composed of more than one non-zero block matrix ($\mathbf{C}_{s,(i,j)}$) at the corresponding area. The matrix Λ has the same format because its diagonal elements are all non-zero values.

The general extension of (4.22) with multiple linking-domains is illustrated in Fig. 4.2, there are two cases for multiple linking-domain matrices ($\mathbf{L}_{s,(i,j)}$): adjacent case and non-adjacent case.

1. In the adjacent case, as illustrated in Fig. 4.2(a), subsystem S_1 and subsystem S_2 are adjacent in the network conductance matrix, thus their linking-domain is extracted as $\mathbf{L}_1$ containing the small matrix $\mathbf{L}_{s,(1,2)}$, and that is just the case proved in the above three steps.

2. In the non-adjacent case, for example, subsystem S_1 and subsystem S_3 are not adjacent in the network conductance matrix, and their connection results in a split linking-domain matrix containing the split $\mathbf{L}_{s,(1,3)}$. In this case, the $\mathbf{C}_{s,(1,3)}$ matrix just needs to add a all-zero block matrix between the split part, as shown in Fig. 4.2(b).

For the two cases, the corresponding linking-domain matrix $\mathbf{L}$ is the sum of the two sub-LDM $\mathbf{L}_{1,2}$, while the transform matrix $\mathbf{C}$ is not the sum but the combination of the two small transform matrices $\mathbf{C}_1$ and $\mathbf{C}_2$, as shown in Fig. 4.2(b). The Λ matrix is also the combination of the σ values of the two small $\Lambda_{1,2}$ matrix. The correctness of this alignment can be verified through block matrix multiplication, which just has the same procedure as Step 2. To conclude, the above proof can not only be used to prove the correctness of *Lemma 1*, but also to construct the transform matrix $\mathbf{C}$ and Λ.

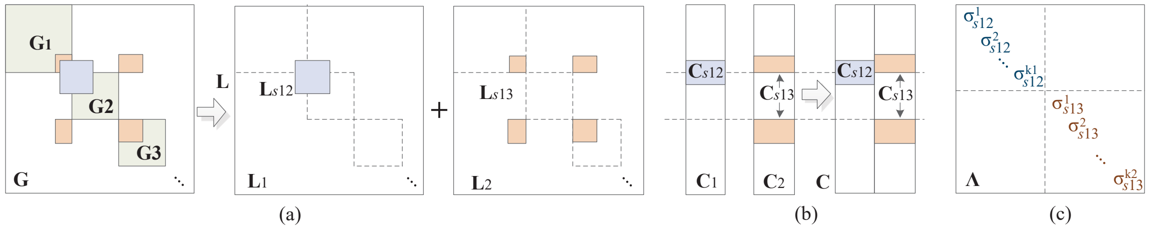


Figure 4.2: Example of multiple linking-domain matrices: (a) linking-domain matrix decomposition; (b)(c) **C** and Λ matrix construction.

4.2.3 Inverse Matrix of the Sum of LDM and DBM

Generally, the inverse matrix of the sum of two matrices is not equal to the sum of the two inverse matrices, i.e., $(A+B)^{-1} \neq A^{-1} + B^{-1}$. Therefore, it is difficult (or impossible) to find a general formulation of the inverse matrix of the sum of two matrices. However, since the **L** matrix in the network matrix $\mathbf{G} = \mathbf{G}_d + \mathbf{L}$ can be expressed as $\mathbf{L} = \mathbf{C}\Lambda\mathbf{C}^T$, the general formulation of $\mathbf{G}^{-1}$ can be found using the Woodbury matrix identity [70]:

$$\mathbf{G}^{-1} = (\mathbf{G}_d + \mathbf{L})^{-1} = \mathbf{G}_d^{-1} - \mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1} \tag{4.25}$$

where:

$$\mathbf{P} = \mathbf{C}\mathbf{Q}\mathbf{C}^T \tag{4.26}$$

$$\mathbf{Q} = (\Lambda^{-1} + \mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1} \tag{4.27}$$

The correctness of (4.25)(4.26)(4.27) can be verified by re-constructing the linking-domain matrix **L** using $\mathbf{G}_d$ and **P**:

$$\begin{aligned}
\mathbf{L} &= \mathbf{C}\Lambda\mathbf{C}^T \\
&= \mathbf{C}[\Lambda\mathbf{Q}^{-1}\mathbf{Q}]\mathbf{C}^T \\
&= \mathbf{C}[(\mathbf{I} + \Lambda\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})(\Lambda^{-1} + \mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}]\mathbf{C}^T \\
&= \mathbf{C}\mathbf{Q}\mathbf{C}^T + \mathbf{C}\Lambda\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C}\mathbf{Q}\mathbf{C}^T \\
&= \mathbf{P} + \mathbf{C}\Lambda\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{P} \\
&= (\mathbf{I} + \mathbf{L}\mathbf{G}_d^{-1})\mathbf{P}
\end{aligned} \tag{4.28}$$

After (4.28) is obtained, (4.25) can be verified directly:

$$\begin{aligned}
&(\mathbf{G}_d + \mathbf{L})(\mathbf{G}_d^{-1} - \mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1}) \\
&= \mathbf{I} + \mathbf{L}\mathbf{G}_d^{-1} - \mathbf{P}\mathbf{G}_d^{-1} - \mathbf{L}\mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1} \\
&= \mathbf{I} + (\mathbf{I} + \mathbf{L}\mathbf{G}_d^{-1})\mathbf{P}\mathbf{G}_d^{-1} - \mathbf{P}\mathbf{G}_d^{-1} - \mathbf{L}\mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1} \\
&= \mathbf{I}
\end{aligned} \tag{4.29}$$

which means the inverse matrix of $(\mathbf{G}_d + \mathbf{L})$ is exactly $\mathbf{G}_d^{-1} - \mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1}$. This important feature could be used to accelerate the computation of inverse matrix because the diagonal block matrix $\mathbf{G}_d^{-1}$ is easier to compute.

4.2.4 Parallel Computation Using LDE

Based on (4.25), computing the inverse matrix of **G** actually only needs to know the value of $\mathbf{G}_d^{-1}$, because the diagonal elements of Λ^{-1} are just reciprocals of those of Λ, and **C** is constant once the network circuit topology is determined. Since $\mathbf{G}_d$ is a diagonal block matrix, the inverse matrix of its block sub-matrices on the diagonal can be computed in parallel. As suggested by (4.25)(4.26)(4.27), $\mathbf{P} = \mathbf{CQC}^T$ should be calculated first and then $\mathbf{G}_d^{-1}\mathbf{P}\mathbf{G}_d^{-1}$ is computed; however, this procedure may be not efficient because the size of **P** is $N \times N$, which increases the size of **Q** and improves the complexity of matrix multiplication. It is suggested that $\mathbf{T} = \mathbf{G}_d^{-1}\mathbf{C}$ is computed first and then computing $\mathbf{TQT}^T$. Because generally the block matrices in $\mathbf{G}_d$ is symmetric, which means $\mathbf{G}_d^{-1}$ is symmetric, then:

$$(\mathbf{G}_d^{-1}\mathbf{C})^T = \mathbf{C}^T(\mathbf{G}_d^{-1})^T = \mathbf{C}^T\mathbf{G}_d^{-1} = \mathbf{T}^T \tag{4.30}$$

Besides, based on the special feature of transformation matrix **C**, $\mathbf{T} = \mathbf{G}_d^{-1}\mathbf{C}$ could be obtained directly even without multiplication operations. Therefore, the parallel matrix inversion procedure can be performed as follows:

1. compute $\mathbf{G}_d^{-1}$ and Λ^{-1};
2. compute $\mathbf{Q} = (\Lambda^{-1} + \mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}$;
3. compute $\mathbf{T} = \mathbf{G}_d^{-1}\mathbf{C}$ and $\mathbf{TQT}^T$;
4. compute $\mathbf{G}^{-1} = \mathbf{G}_d^{-1} - \mathbf{TQT}^T$.

Complexity analysis: For the first step, the computation of the inverse matrix can be executed in parallel with a computational complexity of $O(N_j^3)$, where N_j is the maximum size of the diagonal block matrices. For the second step, since the transform matrix **C** is already known and only contains +1, -1, or 0, the elements in $\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C}$ actually can be obtained from $\mathbf{G}_d^{-1}$ with only plus or minus operations. This means, $\mathbf{Q}^{-1}$ can be obtained without multiplication operations after the first step. Then the computational complexity of computing **Q** is $O(k^3)$. For the third step, the complexity is determined by the computation of $\mathbf{TQT}^T$, which depends on the parallel technique applied to calculate the matrix multiplication. If the matrix multiplications run in parallel for each block matrix, the complexity will be $O(N_j^3)$; if the matrix multiplication run in massively parallel fashion for each row and each column, then the complexity is $O(N_j^2)$. Thus the total complexity of the LDE method is $O(N_j^3 + k^3)$ if block-based parallel processing is exploited for the third step.

4.2.5 Advantages and Limitations of LDE

Compared to the SC method that requires the information of the equivalent current injections, the biggest advantage of the LDE method is that it does not need to know the right hand side of the network matrix equation, because it could directly compute the inverse

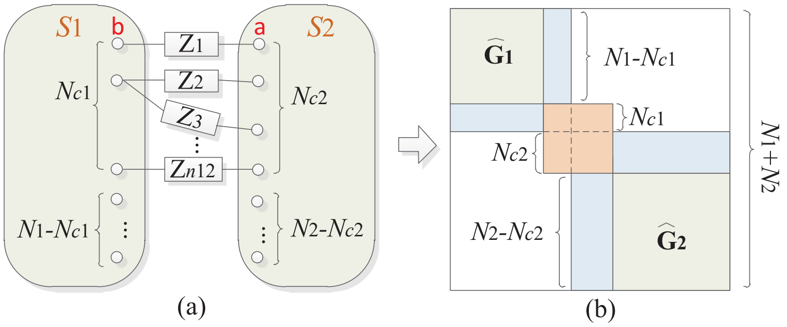


Figure 4.3: Example of two decomposed subsystems: (a) subsystem connection; (b) matrix decomposition.

matrix of the network conductance matrix. Therefore, the LDE method is essentially a matrix inversion method rather than a circuit solution method. If the network conductance matrix does not change during the simulation, the LDE method will accelerate the simulation process dramatically.

In addition, if the network conductance matrix changes over the simulation duration, the LDE method may also possibly run faster than SC method in many cases. According to the complexity analysis, the LDE method has higher complexity in solving the block matrix inversion because the sizes of the decomposed block matrices (N_j) in LDE are usually larger than (or equal, when the decomposed matrices have no connections) that of SC method. However, the matrix inversion of the Λ matrix in LDE and $\mathbf{D}_t$ matrix in SC method also contribute a considerable part in complexity, ignoring the matrix multiplication operations since they can be calculated in massively parallel fashion. Thus if the Λ matrix is smaller than the $\mathbf{D}_t$ matrix in SC method and the benefits introduced by a smaller Λ matrix are larger than the extra computational cost caused by a larger N_j, then LDE will be more suitable over SC method under a variable network conductance matrix.

To demonstrate this, the simple two decomposed block matrices case is illustrated in Fig. 4.3. There are N_1 and N_2 nodes in the two decomposed subsystems (S1 and S2), and N_{c1} nodes in S1 and N_{c2} nodes in S2 are connected through n_{12} conductors. Then the corresponding matrix decomposition is shown in Fig. 4.3(b). The basic precondition to make LDE better is that the size of Λ matrix is smaller than the size of overlapping domain, i.e., the connections between interface nodes are not dense:

$$k = n_{12} < N_{c1} + N_{c2} \tag{4.31}$$

In contrast to the above advantages, there are also limitations for the application of LDE method and they are summarized as follows:

1. The connections between interface nodes are not dense, otherwise the Λ matrix will be too large to achieve desired performance;

2. There is no trans-conductance between the interface nodes and the other nodes, which means the non-zero elements in the linking-domain matrix only refer to the relationship between the interface nodes;

3. The diagonal block matrix $\mathbf{G}_d$ should be invertible, because all the computation is based on $\mathbf{G}_d^{-1}$.

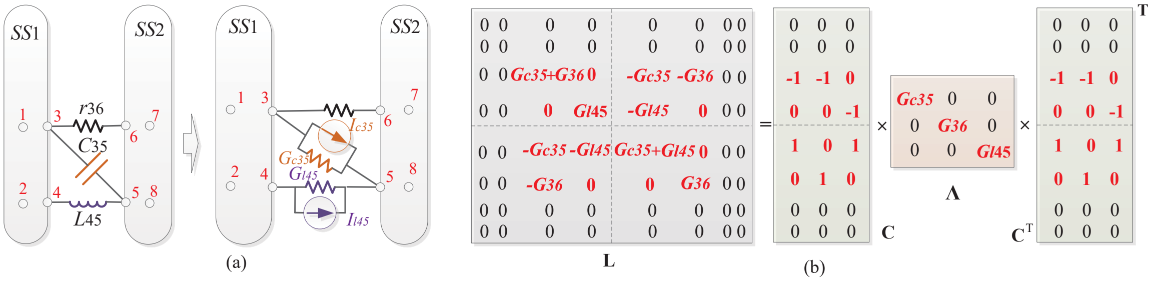
In fact, if the network is decomposed properly, an invertible $\mathbf{G}_d$ can usually be guaranteed. The case where $\mathbf{G}_d$ is not invertible is that there exist elements with 0 values in the diagonal location of $\mathbf{G}_d$. That is, the resulting matrix after the overlapping domain matrix subtracts linking-domain matrix has 0 elements in diagonal locations, which means that the corresponding interface nodes only connect with other nodes via the linking conductors. For an example of this case: in Fig. 4.3(a) the interface node *a* only connects with the other interface node *b* via a conductor, but the other side of node *a* connects with a voltage source. Then the resulting $\mathbf{G}_d$ will not be invertible because the diagonal location of *a* in $\mathbf{G}_d$ will be equal to 0. In practical AC/DC networks, this type of node usually exists at the "edge", and assigning this type of node as interface nodes can neither achieve a good acceleration nor result in an invertible $\mathbf{G}_d$. Avoiding this type of nodes as interface nodes when decomposing the network may lead to an invertible $\mathbf{G}_d$.

4.2.6 Optimal Decomposition based on LDE

LDE is a matrix-based decomposition method, which means, the overall speed-up is actually determined by the number of decomposed subsystems and matrix size of these subsystems given a specific network topology. However, how to decompose the network and how to allocate the power system equipment into properly decomposed subsystems to achieve optimal speed-up remains a problem to be solved. This problem is similar to the *minimum k-section* problem in graph theory, which aims to minimize the links between decomposed sub-graphs. However, the optimal decomposition problem for LDE decomposition is a little different from the *minimum k-section* problem, because both the number of links and the size of decomposed subsystems will affect the overall performance. For large-scale systems with large number of nodes, this problem is NP-hard and is difficult to solve. Therefore, there is a trade-off between the achieved speed-up and latency to perform the decomposition algorithm. Some heuristic methods could be exploited to solve this problem, and they are left for future research.

4.3 Simulation Results and Speed-Up

In this section, two test systems are simulated on the FPGA and GPU board respectively. The speed-up of computing matrix inversion is also evaluated in comparison to the SC method and Gauss-Jordan (GJ) method.



$$\underbrace{\left[\begin{array}{cccc:cccc} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&Gc35{+}G36&0&-Gc35&-G36&0&0\\ 0&0&0&Gl45&-Gl45&0&0&0\\ \hdashline 0&0&-Gc35&-Gl45&Gc35{+}Gl45&0&0&0\\ 0&0&-G36&0&0&G36&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{array}\right]}_{\mathbf{L}} = \underbrace{\left[\begin{array}{ccc} 0&0&0\\ 0&0&0\\ -1&-1&0\\ 0&0&-1\\ \hdashline 1&0&1\\ 0&1&0\\ 0&0&0\\ 0&0&0 \end{array}\right]}_{\mathbf{C}} \times \underbrace{\left[\begin{array}{ccc} Gc35&0&0\\ 0&G36&0\\ 0&0&Gl45 \end{array}\right]}_{\mathbf{\Lambda}} \times \underbrace{\left[\begin{array}{ccc} 0&0&0\\ 0&0&0\\ -1&-1&0\\ 0&0&-1\\ \hdashline 1&0&1\\ 0&1&0\\ 0&0&0\\ 0&0&0 \end{array}\right]^{\mathrm{T}}}_{\mathbf{C}^{\mathrm{T}}}$$

Figure 4.4: Simple demonstration case: (a) two connected subsystems (SS1 and SS2) and their equivalent circuits; (b) corresponding linking-domain matrix, **C** and Λ matrix.

4.3.1 Simple Demonstration Case

The simple single-phase test case composed of eight nodes and its equivalent circuit are illustrated in Fig. 4.4(a), where the L and C elements are equivalenced to the parallel combination of a current source and a resistor from the trapezoidal rule based dircretization. Then the 8×8 network conductance can be decomposed into two small 4×4 matrices for example, where nodes 1$\sim$4 belong to the first matrix and the other four nodes belong to the second matrix. Then the corresponding linking-domain matrix can be extracted from the original matrix, and the transform matrix and Λ matrix can be obtained as shown in Fig. 4.4(b). The size of Λ matrix is 3×3, which indicates that there are three conductance linking the two domains; the transform matrix **C** is a 8×3 matrix, which is generated by adding zero-value elements to the small 4×3 matrix $\mathbf{C}_s$. After the transform matrix is obtained, the rest diagonal block matrix $\mathbf{G}_d$ can be divided into two 4×4 block matrices $\mathbf{G}_{d1}$ and $\mathbf{G}_{d2}$, and thus can be computed in parallel.

$\mathbf{G}_d^{-1}$ can be expressed as:

$$\mathbf{G}_d^{-1} = \begin{bmatrix} \mathbf{G}_{d1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{d2} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{G}_{d1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{d2}^{-1} \end{bmatrix} \tag{4.32}$$

Then the matrix inversion of the whole network conductance matrix is:

$$\mathbf{G}^{-1} = \mathbf{G}_d^{-1} - \mathbf{G}_d^{-1}\mathbf{C}(\Lambda^{-1} + \mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{G}_d^{-1} \tag{4.33}$$

4.3.2 Speed-Up of Matrix Equation Solution on FPGA

To verify the validity and effectiveness of the LDE method in solving large matrix equations with fully exploited parallelism, the IEEE 39-bus system [57] shown in Fig. 4.5(a) is simulated on the Xilinx VCU-118 board with the XCVU9P FPGA [52] at 100MHz frequency. The system is not split through transmission lines; thus a 39×39 matrix is generated. The execution time of the LDE method with varying number of subsystems (N_{ss}) is compared with that of the GJ method and SC method, as shown in Fig. 4.6. Although in this case the LDE method can (but the SC method can not) pre-compute the matrix inversion before the

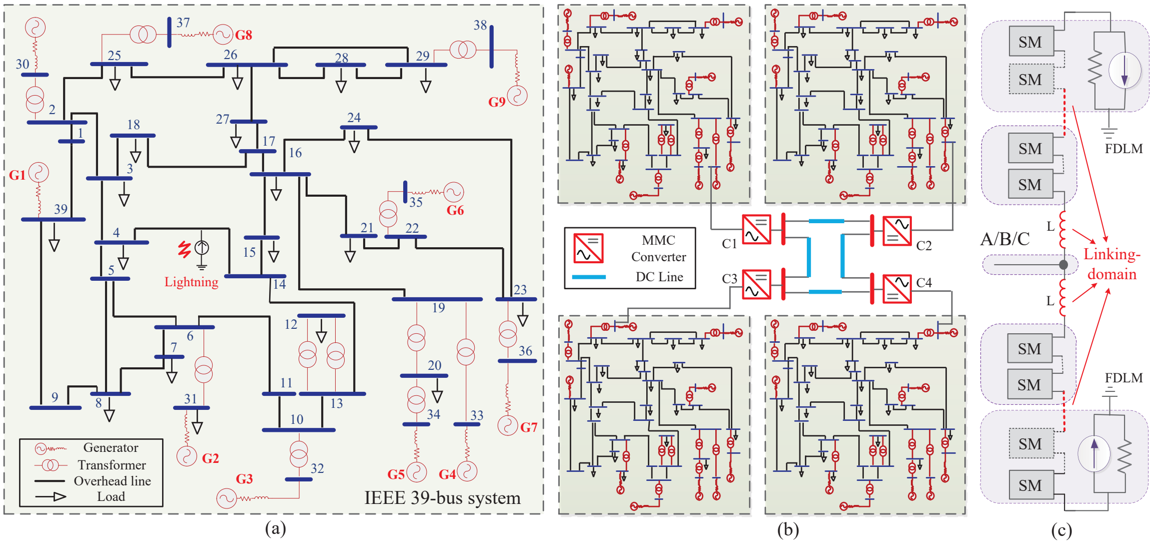


Figure 4.5: Test circuits: (a) IEEE 39-bus test system; (b) hybrid AC/DC grid; (c) decomposition of each phase circuit of MMC.

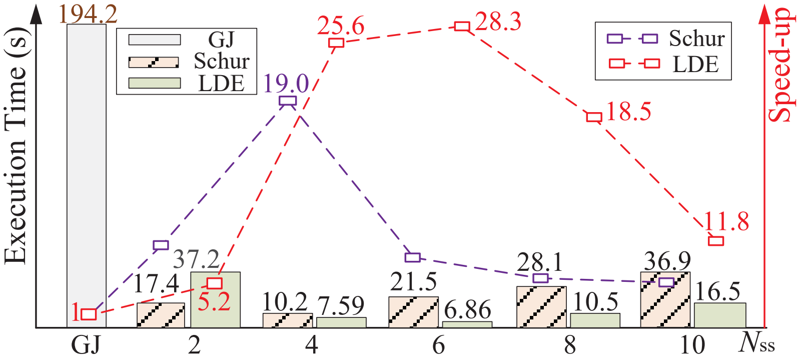


Figure 4.6: IEEE 39-bus execution time comparison between the GJ, SC, and LDE method under varying number of decomposed subsystems on the FPGA.

simulation due to the constant network matrix, the latency of LDE method in Fig. 4.6 only shows the version that computes the matrix inversion in each time-step for pure comparison of solving linear matrix equations. In this case the conductance matrix is constant, then using LU factorization can greatly reduce the latency compared to GJ method since it can pre-compute the **L** and **U** matrices, just as the LDE method pre-computes the matrix inversion. However, since the LU factorization has nearly the same complexity as GJ method (or even larger than GJ due to the extra time to solve $\mathbf{L}(\mathbf{U}v) = \boldsymbol{i}^{eq}$) and it cannot be paralleled well unlike the GJ method, it is better to use GJ as the base method in the parallel platform. The duration of the simulation is 1s, and the time-step is 10μs. Note that the latency depends on the specific network decomposition, and Fig. 4.6 only shows one of the possible cases.

The GJ-based matrix inversion can not be decomposed [71,72], thus the 39×39 matrix inversion consumes the most latency. It can be observed that if the network is decomposed into two subsystems, the SC method has the smaller latency. Because in this case, the number of interface nodes is 8 and the number of links between interface nodes is 4, then the maximum matrix sizes of $\hat{\mathbf{G}}_i$ and $\mathbf{D}_t$ in the SC method are 16×16 and 8×8, while the maximum matrix sizes of $\mathbf{G}_i$ and Λ in LDE method are 20×20 and 4×4. Since in FPGA the matrix multiplication operation can be parallelized efficiently, the matrix inversion consumes the majority of the latency. Therefore, the inversion of a 20×20 matrix consumes much more latency than 16×16, which can not be compensated by a smaller inversion of a 4×4 matrix compared to a 8×8 matrix. When the number of decomposed subsystems increases, the number of interface nodes increases quickly, which makes the size of $\mathbf{D}_t$ in SC method to increase. Therefore, when N_{ss} increases to 6 or larger, the latency of matrix inversion of $\mathbf{D}_t$ in SC method boosts rapidly resulting in a larger overall latency. However, the Λ matrix size is only determined by the number of connecting links between subsystems but not the number of interface nodes, thus the latency of matrix inversion increases not so fast compared to that of the SC method. This result also confirms the discussion in Section 4.2.5. The maximum speed-up of 28.3 can be achieved when the system is decomposed into 6 subsystems.

In terms of the hardware resource consumption on FPGA, the GJ method consumes the most resources (LUT, near 73%) due to the largest matrix inversion; the LDE method consumes the fewest (LUT, near 63%) when the speed-up is larger than that of SC method.

4.3.3 Large-Scale AC/DC Network Simulation on GPU

The large-scale hybrid AC/DC network composed of four IEEE 39-bus systems and four MMC converters is simulated on the NVIDIA Tesla V100 GPU with 5012 cores, as shown in Fig. 4.5(b). In this simulation, detailed models for all the equipment are applied. The frequency-dependent model [65] (FDLM) is utilized for the AC and DC transmission lines; the voltage-behind-reactance model [67] is utilized for the synchronous machines; the conductance matrix model [68] is utilized for the transformers. Since FDLM can be used to decompose the network between two line ends, the four 39-bus systems and four MMC converters connected through long DC lines can be decomposed to simulate in parallel; however, within each 39-bus system, the system decomposition through FDLM can be customized by users: for example, if the line between buses are too short to apply latency-based decomposition, then the two ends are aligned; or if the user want to combine some buses together to avoid large storage consumption, then FDLM can be computed without decomposition. Then within the subsystems decomposed via FDLM, the LDE method can be applied.

Inside each MMC, the system-level two-state switching model (TSSM) [73] for converters C1/C2/C3 and device-level curve-fitting model (CFM) [74] for converter C4 are applied. The TSSM models each half-bridge submodule (HBSM) as a serial connection

Table 4.1: Execution Time and Speed-up of Different Decompositions for One Cycle (16.67ms) Simulation on GPU

m_{arm}	m_{MMC}	n_L	N_{max}	Latency (s)	Speed-up
1	5	6	301×301	213331s	1
2	11	12	76×76	11241s	19
3	17	18	44×44	4293s	50
4	23	24	31×31	2919s	73
5	29	30	30×30	2703s	79
6	35	36	36×36	3232s	66

of an equivalent resistor and a voltage source, thus the HBSM equivalent circuits in each arm can be merged as a arm resistor and voltage source. However, the CFM equivalence for each HBSM cannot be merged because the switching transient details will be lost after merging. To avoid solving the resulting extremely large matrix (usually composed of hundreds of HBSMs in each arm), a traditional method is to use the current source of the last time-step on the HBSM side to calculate the voltage source on the main circuit side, for which the one-step delay will generate an error [75]. In this work, the large matrix is inverted using the LDE method. Since the HBSMs are connected with wires (or short lines), to generate the linking-domain matrix, the wires between decomposed HBSMs are represented by the π-section circuit [76], although the values are very small. Figure 4.5(c) illustrates the decomposition between HBSMs in one phase leg using the LDE method.

The four 39-bus systems and three TSSM-based MMC converters (101-level) are simulated at system-level with the time-step of 10μs, while the device-level CFM-based MMC converter is simulated at 0.1μs. To synchronize these decomposed systems, the system-level simulations need to wait for the completion of device-level simulation every 10μs, thus the overall speed-up is actually produced by the acceleration of CFM-based MMC module [45,46]. The overall speed-ups for the hybrid AC/DC network with varying number of decomposed HBSM systems in each MMC arm are recorded in Table 4.1. Note that if there are m_{arm} HBSM systems decomposed in each arm, then the total MMC circuit is actually decomposed into $m_{MMC} = (6m_{arm} - 1)$ subsystems, where the HBSM systems connecting the DC line equivalence consist of three phases while the others only contain one phase. The case $m_{arm} = 2$ is shown in Fig. 4.5(c). In Table 4.1, n_L is the number of linking-domains, and N_{max} denotes the size of the largest matrix to be inverted. Since inverting the entire matrix (605 × 605) involves too huge computational effort that exceeds the parallel capability, the MMC circuit is first decomposed to five subsystems ($m_{arm} = 1$) as the basic decomposition. Although the GPU-based implementation consumes much more latency than FPGA due to the limitations of parallelism, the maximum speed-up of matrix inversion process can be even larger because of the large system scale and the sparse connection links between HBSMs in each arm. As show in the Table, the maximum speed-up over the basic decomposition is 79, which indicates the latency can be reduced by much more than 79 times compared to directly solving the whole MMC circuit. Besides, when the number of decomposed subsystems exceeds 29, the speed-up starts to decrease due to the increased size of Λ matrix.

4.4 Summary

In this chapter, a new non-overlapping domain decomposition method is proposed: the linking-domain extraction (LDE) based decomposition method. In LDE, the linking-domain matrix (LDM) is first extracted from the original network conductance matrix, which makes the remaining part to be a diagonal block matrix (DBM) that is easy to calculate in parallel. An important *lemma* describing the feature of LDM and its proof are presented, which indicates that the LDM can be transformed from a diagonal matrix. Based on this *lemma*, the general formulation of the inverse matrix of the sum of LDM and DBM can be found using the Woodbury matrix identity, and then the network matrix inversion can be computed for each block in parallel. Compared to the Schur complement (SC) method, LDE method can compute the matrix inversion directly; and when the network matrix changes, the LDE method can also run faster than the SC method in many cases. The simulation results for the IEEE 39-bus system and speed-ups over the SC method and Gauss-Jordan method demonstrate the validity and efficiency of the proposed LDE method.

5

Hierarchical Linking-Domain Extraction Decomposition Method for Fast and Parallel Power System Electromagnetic Transient Simulation

The linking-domain extraction (LDE) decomposition method is a new non-overlapping domain decomposition method for parallel circuit simulation. However, the original LDE method is inefficient in both the computational procedure and storage cost. In this chapter, a hierarchical LDE (H-LDE) method is proposed to further improve the LDE method, which leverages all the hidden features of LDE that are not exploited in the original work to perform a multi-level decomposition of power systems. The improved LDE computation procedure is first proposed to eliminate the necessity of computing the entire matrix inversion, and then the multi-level computation structure is proposed for fast matrix inversion of the decomposed sub-matrices. The mathematical complexity of the H-LDE method is analyzed, which is used to derive the two principles for decomposing a power system. These principles can be applied on both parallel and sequential compute architecture. The 4-level LDE decomposition is applied on the IEEE 118-bus test power system and implemented in both sequential and parallel, which is used to verify the validity and efficiency of the proposed H-LDE decomposition method. The simulation results of various benchmark test power systems show that the proposed H-LDE method can achieve lower computation latencies than the classical LU factorization and sparse KLU method within a certain system scale.

5.1 Introduction

The new linking-domain extraction (LDE) decomposition method [78] is similar to the Schur complement method, but it can find the general matrix inversion formulation of the circuit conductance matrix. The efficiency of the LDE method is achieved by computing the inversion of the small decomposed block matrices (in serial or parallel, which occupies much smaller computational latencies compared to computing the whole matrix equation) and then assembling the inverted block matrices via a correction matrix to obtain the inversion of the entire conductance matrix. However, this process is not scalable due to several reasons: 1) For large-scale power systems, the conductance matrix is large and sparse, but the inverted matrix is usually dense, which costs much more storage than simply solving the matrix equations and makes the LDE method inapplicable. That is why the LU factorization [5] and fast sparse solutions such as the KLU [79] and NICSLU [80] methods are widely applied in the commercial and open-source power system simulators such as EMTDC/PSCAD® [12] and SPICE [81]. 2) The original LDE method decomposes the conductance matrix once, which may also result in a large size of the decomposed block matrices and computing the inversion of the block matrices is also costly. Therefore, although the LDE method has a strong mathematical base, it seems that it can only achieve lower latencies than the classical solvers in simulating a very small-scale power system.

To improve the original LDE method, in this chapter, the hierarchical LDE (H-LDE) decomposition method is proposed, which utilizes all the hidden features of the LDE method to achieve an all-around improvement. First, the matrix equation solver computation procedure based on LDE is proposed, which avoids computing the entire matrix inversion so that the storage cost is reduced significantly; second, a multi-level decomposition structure is proposed to reduce the computational cost of inverting the decomposed block matrices. The approximate complexity of H-LDE decomposition is analyzed, based on which the two decomposition principles are presented to instruct the detailed decomposition configuration for a specific number of decomposition levels: before the last level, the decomposition does not need to find a balance between the sizes of the DBMs and LDM; and in the last level, the decomposition should take the balance between DBMs and LDM into consideration. The detailed decomposition logic depends on the power system topology at hand, the parallelism capabilities of the parallel platform used and the number of decomposition levels.

The IEEE 118-bus test system is used to verify the validity and efficiency of the proposed H-LDE decomposition performance, both the sequential and GPU-based parallel implementation are discussed. The performance of H-LDE is also compared with the LU factorization and KLU method in several standard benchmarks, which shows the H-LDE method is much more scalable than the original LDE method and could achieve lower latencies than the pure LU factorization. The computational latencies also shows that the H-LDE cost lower latencies than the sparse KLU solver within a certain system scale.

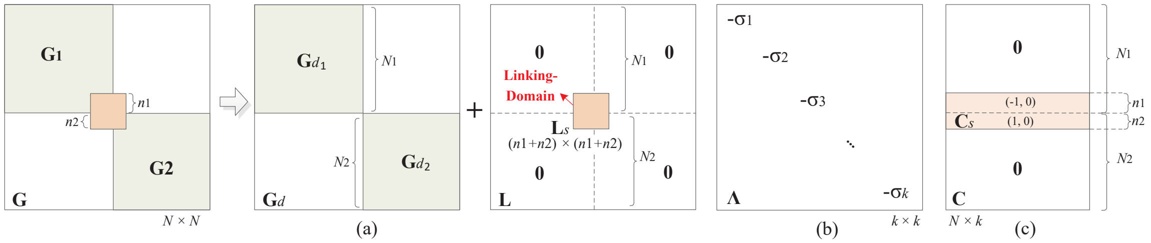


Figure 5.1: Example of LDE decomposition of two subsystems: (a) decomposition of G; (b) Λ matrix; (c) transformation matrix C.

5.2 Improved Linking-Domain Extraction based Decomposition Method

LDE is a matrix-based decomposition method, which is able to obtain the general formulation of the matrix inversion and compute the inverse in parallel [78]. However, computing the entire matrix inversion may be costly and meaningless. In this section, the mathematical formulation of the LDE method is introduced, and the improved LDE calculation procedure is proposed to optimize the matrix equation solution.

5.2.1 LDE Matrix Decomposition

Given a power system containing N nodes, a $N \times N$ conductance matrix **G** will be generated. Then **G** could be decomposed into two separate matrices: the diagonal block matrix $\mathbf{G}_d$ and the linking-domain matrix **L**:

$$\mathbf{G} = \mathbf{G}_d + \mathbf{L} \tag{5.1}$$

The number of block matrices in $\mathbf{G}_d$ depends on the number of subsystems decomposed. Figure 5.1 illustrates the case of two decomposed subsystems: there are n_1 nodes in subsystem S1 (matrix $\mathbf{G}_1$) connecting with n_2 nodes in subsystem S2 (matrix $\mathbf{G}_2$) via a conductance or voltage source (note that if two interface nodes are connected via a current source then it will not be revealed in the conductance matrix). All of the information of connection is recorded in the linking-domain matrix **L** ($N \times N$), which is composed of several matrices with all-zero elements and a small linking-domain $\mathbf{L}_s$ with size of $(n_1+n_2)\times(n_1+n_2)$. If the number of decomposed subsystems is larger than two, the linking-domain matrix will contain several small linking-domains ($\mathbf{L}_s^1, \mathbf{L}_s^2, ..., \mathbf{L}_s^m$), and these small linking-domains may not have a integral matrix format if the interface nodes are not continuous in node indexes.

For a common network, the linking-domain matrix **L** can be expressed as a transformation from a diagonal matrix Λ ($k \times k$), and the transformation matrix **C** is a rectangular matrix ($N \times k$) of which the element values are only equal to 1, -1, or 0.

$$\mathbf{L} = \mathbf{C}\Lambda\mathbf{C}^T \tag{5.2}$$

Note that k is the number of links connecting the decomposed matrices. Take the case of two decomposed subsystems as an example, as shown in Fig. 5.1(b)(c), Λ has k negative elements in diagonal, and **C** is composed of all-zero matrices and a small transformation matrix $\mathbf{C}_s$ with size of $(n_1+n_2)\times k$. $\mathbf{C}_s$ can be regarded as two parts: the upper n_1 rows $\mathbf{C}_s^{(1)}$ and the lower n_2 rows $\mathbf{C}_s^{(2)}$; every column of $\mathbf{C}_s^{(1)}$ only has one element equal to -1, and the other elements are equal to 0; while every column of $\mathbf{C}_s^{(2)}$ only has one element equal to 1, and the other elements are equal to 0.

Then the general formulation of the inverse matrix of $\mathbf{G}=\mathbf{G}_d+\mathbf{L}$ could be found based on the Woodbury matrix identity [70]:

$$\mathbf{G}^{-1}=(\mathbf{G}_d+\mathbf{L})^{-1}=\mathbf{G}_d^{-1}-\mathbf{G}_d^{-1}\mathbf{C}\mathbf{Q}\mathbf{C}^T\mathbf{G}_d^{-1} \tag{5.3}$$

where:

$$\mathbf{Q}=(\Lambda^{-1}+\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1} \tag{5.4}$$

5.2.2 Improved LDE Computation Procedure

Although computing the matrix inversion $\mathbf{G}^{-1}$ directly could accelerate the simulation process significantly when the the conductance matrix **G** does not change over the simulation duration, the storage and I/O cost increase dramatically when the power system scale expands. Therefore, in this chapter, the LDE computation procedure is improved accordingly.

The improved computational procedure is used to solve the matrix equations without storing the inverted conductance matrix. The goal is to solve for the node voltages v:

$$\mathbf{G}v=i_{eq} \tag{5.5}$$

Applying the LDE matrix inversion:

$$\begin{aligned} v&=\mathbf{G}^{-1}i_{eq}=[\mathbf{G}_d^{-1}-\mathbf{G}_d^{-1}\mathbf{C}\mathbf{Q}\mathbf{C}^T\mathbf{G}_d^{-1}]i_{eq}\\ &=\mathbf{G}_d^{-1}i_{eq}-\mathbf{G}_d^{-1}\mathbf{C}\mathbf{Q}\mathbf{C}^T\mathbf{G}_d^{-1}i_{eq}\\ &=v_{DBM}-\mathbf{G}_d^{-1}\mathbf{C}(\Lambda^{-1}+\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}\mathbf{C}^Tv_{DBM}\end{aligned} \tag{5.6}$$

where $v_{DBM}=\mathbf{G}_d^{-1}i_{eq}$ is the solution of each decomposed subsystems. The matrix inversion process of $(\Lambda^{-1}+\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})$ can also be avoided to reduce the computational and storage cost:

$$\begin{aligned} v&=v_{DBM}-\mathbf{G}_d^{-1}\mathbf{C}(\Lambda^{-1}+\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}\mathbf{C}^Tv_{DBM}\\ &=v_{DBM}-\mathbf{G}_d^{-1}\mathbf{C}v_{LDM}\end{aligned} \tag{5.7}$$

where v_{LDM} is the solution of the matrix equation below:

$$(\Lambda^{-1}+\mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})v_{LDM}=\mathbf{C}^Tv_{DBM} \tag{5.8}$$

As stated in the matrix inversion procedure, $\mathbf{C}$ and $\mathbf{C}^T$ matrix have very special features, which make the computation of $\mathbf{G}_d^{-1}\mathbf{C}$ and $\mathbf{C}^T v_{DBM}$ simple without multiplication operations. Therefore, the improved LDE solver computation procedure can be executed as follows:

1. compute $\mathbf{G}_d^{-1}$ and $v_{DBM} = \mathbf{G}_d^{-1} i_{eq}$;
2. compute $\mathbf{T} = \mathbf{G}_d^{-1}\mathbf{C}$;
3. compute $\Lambda^{-1} + \mathbf{C}^T\mathbf{T}$ and solve v_{LDM};
4. compute the final solution $v = v_{DBM} - \mathbf{T}v_{LDM}$;

Using the above improved LDE computation procedure, the storage cost can be dramatically reduced. For a $N \times N$ conductance matrix that is decomposed into m submatrices with equal sizes, the storage cost of the improved LDE method is $O[m(N/m)^2 + k^2] = O(N^2/m + k^2)$, which is reduced nearly m times over the original LDE method. Although the storage is still large compared to the sparse LU factorization based solvers, the benefits of the fast solution process can be reflected within a certain power system scale.

5.3 Hierarchical LDE Method

From the above procedure it can be observed that the computation of $\mathbf{G}_d^{-1}$ cannot be avoided although the computation of $\mathbf{G}^{-1}$ is eliminated. Since only the diagonal block matrices in $\mathbf{G}_d^{-1}$ are non-zeros, the storage cost can be reduced significantly. However, for large-scale power systems, the conductance matrix could not be decomposed in a fine-grained fashion because such decomposition will result in a large Λ matrix that is not beneficial to the overall performance. Therefore, the LDE decomposition will generate large block matrices in $\mathbf{G}_d$ even after decomposition, which makes the computation of $\mathbf{G}_d^{-1}$ also costly.

Fortunately, the LDE method is essentially a matrix inversion method, which is able to accelerate the matrix inversion process of the block matrices in $\mathbf{G}_d$. This means, although the LDE matrix inversion is not used in the improved computation procedure, it can also be used to compute $\mathbf{G}_d^{-1}$, and this application of LDE method is just to decompose the subsystems into further sub-subsystems. Based on this, the multi-level LDE decomposition is proposed, which is called "hierarchical LDE (H-LDE) method".

5.3.1 Multi-Level LDE Decomposition

When the decomposed subsystems also have relatively large scales, computing the inversion of block matrices in $\mathbf{G}_d$ still requires a lot of compute effort. In the H-LDE decomposition, the computation of $\mathbf{G}_d^{-1}$ could be executed based on the second or even higher level LDE decomposition to reduce the computational latency. Thus the application of LDE method could be extended to simulate power systems in a hierarchical manner:

1. decompose the whole system into subsystems for the first-level LDE decomposition, and solve the unknown state-variables using the improved LDE computation procedure;

2. if the sizes of decomposed subsystems are still large, decompose the subsystems into small sub-subsystems for the second-level LDE decomposition; else, calculate the inversion of the decomposed block matrices directly;

3. if the second-level decomposition still has room for deeper decomposition, then the third or even higher level decompositions could be exploited to increase the parallelism and accelerate the overall execution.

The improved LDE computation procedure is applied for the first level LDE decomposition to reduce the storage and computation cost, because there is no need to compute the inversion of the whole matrix; however, the second or higher level LDE decomposition should be computed using the original LDE matrix inversion procedure, because the goal of multi-level LDE decomposition is to obtain $\mathbf{G}_d^{-1}$ quickly, which could not be avoided.

5.3.2 Computational Complexity Analysis of Hierarchical LDE

Assume the complexity of inverting a $N \times N$ matrix is $O(N^3)$ based on the Gauss-Jordan method. Then the original LDE method has a complexity of $O[N_j^3 + k^3]$, where N_j is the maximum size of the decomposed block matrices, and k is the number of links connecting these decomposed subsystems (SSs). However, this is not applicable for the H-LDE method, because complexity of computing $\mathbf{G}_d^{-1}$ should be re-evaluated. Assume that there are r levels of LDE decomposition in total, and the decomposed subsystems nearly have the same size while the number of links between the decomposed sub-subsystems are also the same for the decomposition of different subsystems. This assumption may be not rigorous, but considering that the power system topology is not dense and the connections are relatively distributed on average, this assumption is just an approximation and makes sense in the complexity analysis. As shown in Fig. 5.2, after the i^{th}-level decomposition, there are $m_{(i)}$ subsystems decomposed from each subsystem located on the upper level in Fig. 5.2, and each decomposed subsystem contains $N_{(i)}$ nodes with total $k_{(i)}$ links connecting these subsystems. The relationship between $N_{(i)}$ and $m_{(i)}$ are:

$$N_{(i-1)} = m_{(i)} N_{(i)} \tag{5.9}$$

$$N = m_{(1)} N_{(1)} = ... = (\prod_1^i m_{(i)}) N_{(i)} \tag{5.10}$$

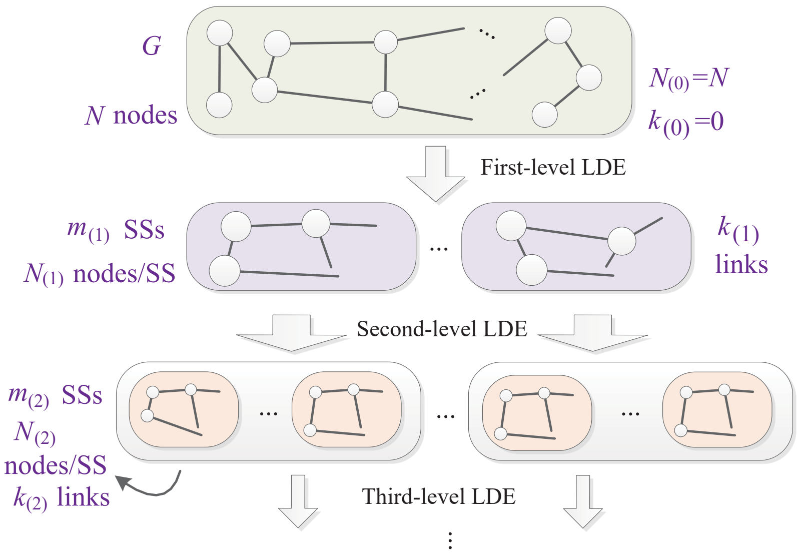


Figure 5.2: Demonstration of hierarchical LDE decomposition.

The actual parallelism applied for different level depends on the parallel capabilities of the hardware platform, which greatly impacts the computational complexity. Therefore, the complexity analysis should be performed in two cases for each level: parallel case and sequential case.

Parallel Case: In this case, the matrix inversion processes for each block matrix are computed in parallel. If the $m_{(i)}$ decomposed block matrices after the $i^{(th)}$-level LDE decomposition can be computed in parallel, then the computational time for each subsystem in $(i-1)^{(th)}$-level that are decomposed into $m_{(i)}$ subsystems has the computational complexity of:

$$O[f_p(N_{(i-1)}, k_{(i-1)})] = O[f(N_{(i)}, k_{(i)}) + {k_{(i)}}^3 + t_{p(i)}], i < r \tag{5.11}$$

$$O[f_p(N_{(r-1)}, k_{(r-1)})] = O[N_{(r)}^3 + {k_{(r)}}^3 + t_{p(r)}], i = r \tag{5.12}$$

where $f(N_{(i)}, k_{(i)})$ is the computational complexity for the $(i)^{th}$-level decomposed subsystems: $f = f_p$ if the $(i)^{th}$-level can also be computed in parallel, else $f = f_s$. $t_{p(i)}$ denotes the overhead of launching the i^{th} level threads for parallel computation, which cannot be neglected and can even be the dominant part of the overall cost when i is large. Because generally the performance will slow down for the higher level parallel computation in the nested parallelism of the compute platforms such as the GPU.

Sequential Case: In this case, the the matrix inversion processes can only be computed in sequential, which makes the computational time for each subsystem in $(i-1)^{(th)}$-level after the $i^{(th)}$-level LDE decomposition has the computational complexity of:

$$O[f_s(N_{(i-1)}, k_{(i-1)})] = O[m_{(i)} f_s(N_{(i)}, k_{(i)}) + {k_{(i)}}^3] \tag{5.13}$$

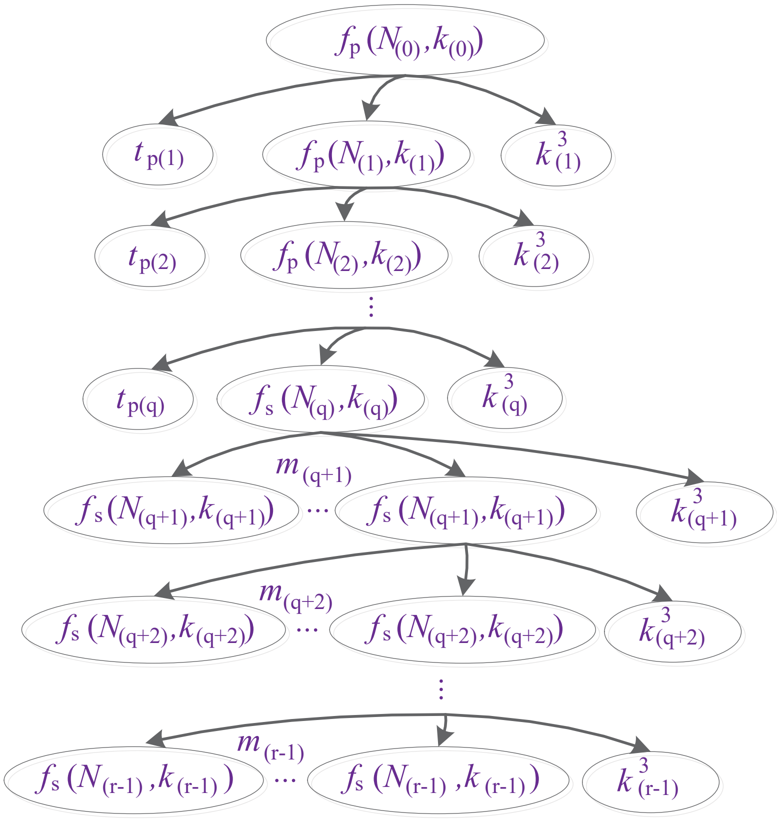


Figure 5.3: Recursive complexity analysis of the hierarchical LDE decomposition.

$$O[f_s(N_{(r-1)},k_{(r-1)})] = O[m_{(r)}N^3_{(r)} + {k_{(r)}}^3], i = r \tag{5.14}$$

Here, the inversion of $m_{(i)}$ decomposed block matrices is computed in sequential, and in this chapter, if the $(i-1)^{(th)}$-level could not be parallelized, then the $i^{(th)}$-level could only be computed in sequential. Then the total complexity $O[f(N_{(0)},k_{(0)})]$ actually can be obtained in a recursive way, as illustrated in Fig 5.3, assuming the $(1 \sim q-1)^{th}$ level computation is parallelized and $(q \sim r)^{th}$ level computation is sequential.

The complexity of computing the q^{th} level block matrix inversion is given as:

$$O[f_s(N_{(q)},k_{(q)})] = O[N^3_{(r)} \prod_{i=q+1}^{r} m_{(i)} + \sum_{j=q+1}^{r} (\prod_{p=q+1}^{j-1} m_{(p)}) k^3_{(j)}] \tag{5.15}$$

When $j = q+1$, let $\prod_{p=q+1}^{j-1} m_{(p)} = 1$. Then the total complexity of the hierarchical LDE method is given as:

$$O[f(N_{(0)},k_{(0)})] = O[f_s(N_{(q)},k_{(q)}) + \sum_{i=1}^{q} (k^3_{(i)} + t_{p(i)})] \tag{5.16}$$

For pure parallel H-LDE, $q = r$, then the overall computational complexity becomes:

$$O[f_p(N_{(0)},k_{(0)})] = O[N^3_{(r)} + \sum_{i=1}^{r} (k^3_{(i)} + t_{p(i)})] \tag{5.17}$$

For pure sequential H-LDE, $q = 0$, then the overall computational complexity becomes:

$$O[f_s(N_{(0)}, k_{(0)})] = O[N^3_{(r)} \prod_{i=1}^{r} m_{(i)} + \sum_{j=1}^{r} (\prod_{p=1}^{j-1} m_{(p)}) k^3_{(j)}] \tag{5.18}$$

where $\prod_{p=1}^{0} m_{(p)} = 1$. Note that (5.16)-(5.18) are only the approximation and ideal results of complexity analysis, but they can still be used to guide the specific decomposition configuration given a power system case at hand.

5.3.3 Specific Decomposition Principles

Based on the complexity analysis, the number of decomposition levels and the number of decomposed subsystems in each level can be evaluated given a specific power system topology and parallel platform. Generally, two decomposition principles are proposed to improve the decomposition performance.

Principle 1: In the $(i = 1 \sim r-1)^{th}$ level, make the number of connecting links $k_{(i)}$ to be smaller than the size of the decomposed subsystems $N_{(i)}$; and in the r^{th} level, make a balance between $k_{(r)}$ and $N_{(r)}$.

This principle is inspired by (5.11)-(5.14), as can be seen that if $k_{(i)} > N_{(i)}$ in the $(i = 1 \sim r-1)^{th}$ level, then $O[k^3_{(i)}]$ will be the dominant part of the complexity no matter in the parallel case or sequential case, which means, there is no need to perform a higher level decomposition. And in the r^{th} level, the balance should be made between $k_{(r)}$ and $N_{(r)}$ because it is the last level and should make sure that $O[f(N_{(r-1)}, k_{(r-1)})]$ is minimized, just like the one-level traditional LDE method.

Principle 2: The launching of a higher level should achieve a lower computation time but not result in a larger latency, that is, for parallel computing:

$$O[f(N_{(i)}, k_{(i)}) + {k_{(i)}}^3 + t_{p(i)}] < O[N^3_{(i-1)}] \tag{5.19}$$

where $t_{p(i)}$ contains the overhead of launching child kernels for parallel computation as well as the latency of synchronization between the kernels. And for sequential computing:

$$O[m_{(i)} f_s(N_{(i)}, k_{(i)}) + {k_{(i)}}^3] < O[N^3_{(i-1)}] \tag{5.20}$$

This principle is used to judge whether a deeper decomposition is required, because the common parallel platforms such as GPU have limited capabilities of parallelism and the overhead of launching child kernels is significant. For sequential computation, the decomposition should also make sure the sum of latencies for each subsystem computation is smaller than the latency without decomposition. For example, when decomposing a 60×60 block matrix into three 20×20 block matrices for sequential computing, it should be guaranteed that $3 \times t_{20} + t_{k20} < t_{60}$, where t_{20} denotes the latency of computing the inversion of a 20×20 block matrix, and t_{k20} denotes the latency of computing the **Q** matrix

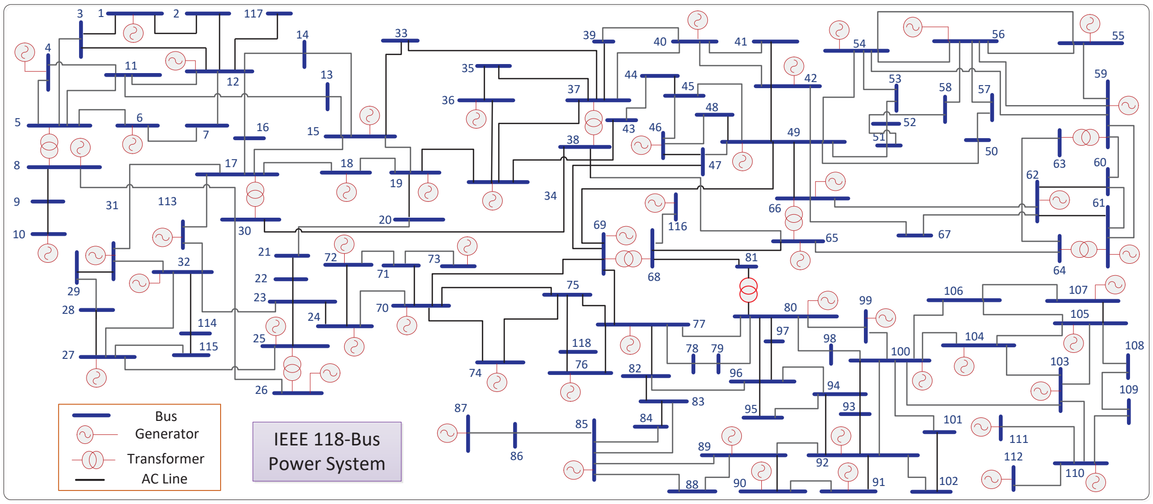


Figure 5.4: Diagram of the IEEE 118-Bus test power system.

with size of $k \times k$ and correcting the block matrices with the **Q** matrix. Practically, the latency of computing a matrix inversion given a specific size can be evaluated in advance, then the decision can be made on whether a deeper decomposition is necessary or not.

5.4 CPU-Based Sequential and GPU-Based Parallel Implementation

The dynamic parallelism feature [51] of GPUs enables nested kernel function execution, which is suitable for the H-LDE decomposition architecture. In this section, both the CPU-based sequential and GPU-based parallel implementation of the IEEE 118-bus [69] test system is described for demonstration.

5.4.1 Sequential and Parallel Configuration

The IEEE 118-bus power system [69] shown in Fig. 5.4 is chosen as the test system to show the application of the proposed H-LDE method, which contains 118 buses, 54 generators, 177 lines, 9 transformers, and 91 loads. The equivalent network topology is illustrated in Fig. 5.5, where the bus number is shown on each node. The power equipment models are the same as those in PSCAD/EMTDC® [12]. For sequential H-LDE computation, let $r = 4, q = 0$, which means there are 4 level of LDE decomposition applied. The reason of choosing $r = 4$ is explained in the following part, and when the system scale increases, the level of LDE decomposition can increase to achieve the optimal performance.

For parallel implementation, the dynamic parallelism feature [51] of GPUs is utilized. The dynamic parallelism enables the kernel function to create new kernel functions on the GPU device dynamically. For example, the grid A that is a collection of several parallel threads is the first-level parallelism, in which every thread launches a new grid B. Grid A is called a “parent” grid, and the one launched by it is called “child” grid. Launching a set of new “child” grids also introduces a considerable cost including the latency of launching kernels and synchronizing these kernels. Therefore, if the child kernels do not extract much parallelism and there is not much benefit against their non-parallel counterparts, then the little benefit may be canceled out by the child kernel launching overheads. As an example, we use $r = 4, q = 2$ for parallel H-LDE computation of the 118-bus power system, which means there are 4 levels of LDE decomposition in total and the first two levels are computed using the GPU dynamic parallelism, while the 3^{rd} and 4^{th} level are computed sequentially.

5.4.2 Test System Decomposition

Following the two principles proposed in Section 5.3.3, the 4-level H-LDE decomposition is shown in Fig. 5.5. The partition lines are highlighted in different line types for different levels, and the number shown besides a partition line denotes the number of links connecting the decomposed subsystems, that is, $k_{(i)}$. Note that this chapter only shows a specific partition, which may not be the optimal solution.

First-Level: As shown in Fig. 5.5, in the 1^{st} level decomposition, the 118-bus is decomposed into four subsystems: SS-1, SS-2, SS-3 and SS-4 with sizes of 30, 30, 28 and 30 respectively. The number of links connecting the decomposed subsystems is 15, which means the size of $\mathbf{Q}$ matrix is $k_{(1)} = 15$. This decomposition follows the *Principle 1*, making $k_{(1)} < N_{(1)}$, then after the inversion of the 30×30 and 28×28 matrices is computed in sequential or parallel, $\mathbf{G}_d^{-1}$ is obtained, and finally the improved LDE solver procedure (5.7) can be applied to solve the unknown state variables, which is extremely simple and can achieve a good speed-up.

Second-Level: The $2^{nd} \sim 4^{th}$-level decomposition is applied to compute the inversion of the SS-1, SS-2, SS-3 and SS-4 conductance matrices, therefore, the actual computation sequence is performed as a bottom-up pattern, from the 4^{th}-level to the 2^{nd}-level. Taking SS-1 as an example, the 30 nodes are decomposed into two subsystems with the block matrix size of 15×15. This is also decomposed following *Principle 1*, making $k_{(2)} < N_{(2)}$. For parallel computing, as instructed by *Principle 2*, the overhead of launching the second-level parallelism should be smaller than the benefit from parallel computation of the block matrices inversions. We can see that $k_{(2)} = 6$ and $N_{(2)} = 15$, which could meet the requirement (5.19) through experimental results; that means the second-level decomposition could benefit from the parallel computation using a small linking-domain matrix. For sequential computing, the second-level decomposition could obviously achieve larger speed-ups since computing the 30×30 matrix inversion involves much more computational efforts than computing two 15×15 matrices inversions. The second-level decomposition for SS-2, SS-3 and SS-4 has the same logic and procedure.

Third-Level: The third level block matrix inversion is computed in sequential as config-

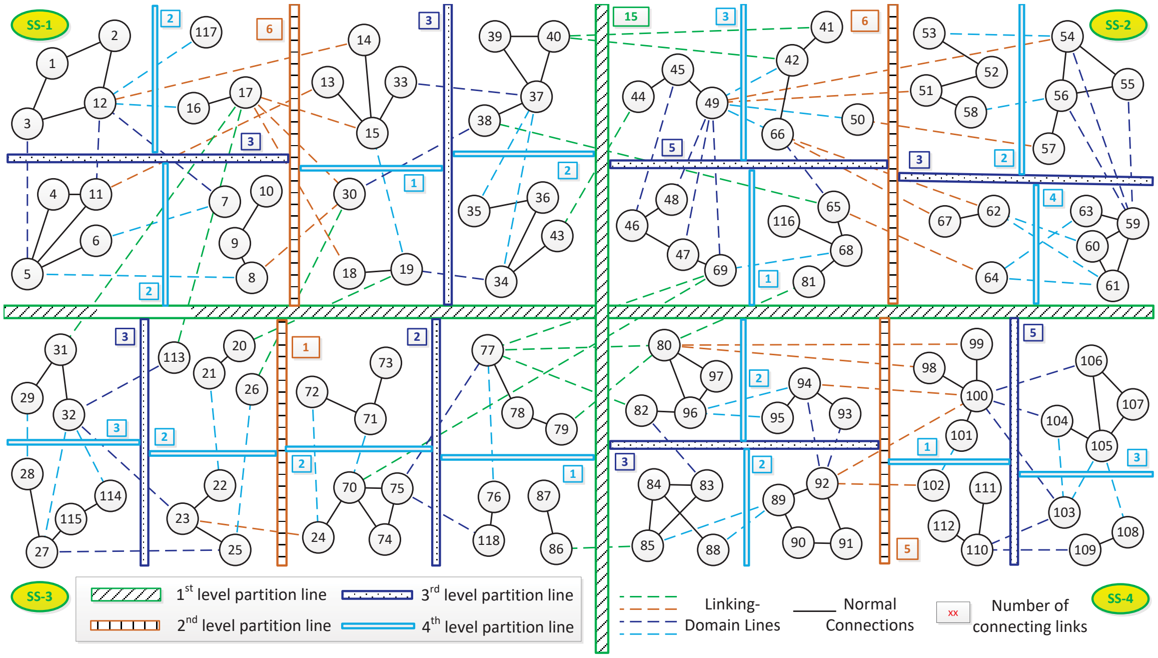


Figure 5.5: Topology partitioning of the IEEE 118-Bus test power system using the 4-level LDE decomposition.

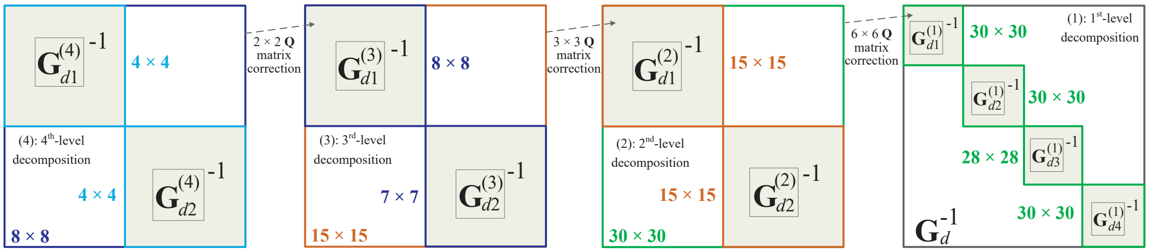

$\mathbf{G}_{d1}^{(4)\ -1}$
4 × 4
(4): 4th-level decomposition
4 × 4
$\mathbf{G}_{d2}^{(4)\ -1}$
8 × 8
2 × 2 Q matrix correction
$\mathbf{G}_{d1}^{(3)\ -1}$
8 × 8
(3): 3rd-level decomposition
7 × 7
$\mathbf{G}_{d2}^{(3)\ -1}$
15 × 15
3 × 3 Q matrix correction
$\mathbf{G}_{d1}^{(2)\ -1}$
15 × 15
(2): 2nd-level decomposition
15 × 15
$\mathbf{G}_{d2}^{(2)\ -1}$
30 × 30
6 × 6 Q matrix correction
$\mathbf{G}_{d1}^{(1)\ -1}$
30 × 30
(1): 1st-level decomposition
$\mathbf{G}_{d2}^{(1)\ -1}$
30 × 30
28 × 28
$\mathbf{G}_{d3}^{(1)\ -1}$
30 × 30
$\mathbf{G}_{d4}^{(1)\ -1}$
$\mathbf{G}_{d}^{-1}$

Figure 5.6: Assembling process for inverting the block matrices of the first-level decomposition with a 4-level H-LDE decomposition.

ured in Section 5.3.2, which means that the decomposition should take the actual computing latency of the matrix inversion with a specific size into consideration. For example, the 15×15 block matrix is decomposed into two block matrices with sizes of 8×8 and 7×7, and based on *Principle 2*, the decomposition should satisfy $t_8 + t_7 + t_{k3} < t_{15}$, where t_x denotes the latency of computing the inversion of a $x \times x$ block matrix, and t_{k3} denotes the latency of computing the inversion of the generated $\mathbf{Q}$ matrix and correct the block matrix with the $\mathbf{Q}$ matrix. Since in this partition $k_{(3)} = 5$ for the worst case, it can be verified on the implemented computing platform (both CPU and GPU) that the requirements can be satisfied.

Fourth-Level: The final level decomposition follows the same logic as the third level decomposition, but as indicated in *Principle 1*, it should also make the balance between $k_{(4)}$ and $N_{(4)}$. In fact, this principle is not very rigorous, and in this case, $N_{(4)} = 4$ or 3, $k_{(4)}$

is also equal to or smaller than 3.

Fifth or Higher Level: From Fig. 5.5 it can be seen that for $N_{(4)} = 4$ or 3, making a higher level LDE decomposition is not necessary. Because computing the inverse of a 4×4 matrix does not involve much overhead, and if it is decomposed into smaller matrices, computing smaller matrix inversions and correcting them with the **Q** matrix will introduce extra latencies, which are larger than the benefit of decomposition and simply violate the *Principle 2*.

From the above multi-level decomposition configurations, it can be observed that the maximum matrix that is actually required to be inverted is 4×4 for block matrices and 6×6 for **Q** matrices ($k_{(2)} = 6$). The 15×15 **Q** matrix ($k_{(1)} = 15$) is not required to be inverted due to the proposed improved LDE computation procedure (5.7); the 30×30 and 28×28 block matrices inversions are actually assembled using the inverted 4×4 and 3×3 block matrices of the 4^{th}-level decomposition, as shown in Fig. 5.6. Therefore, the computational effort of H-LDE is greatly reduced compared to the original LDE method that computes the 30×30 and 28×28 block matrices inversions directly.

5.5 Simulation Results and Verification

In this section, the matrix equations of the IEEE 39, 57, 118, and 300 bus benchmark test power systems [83] and the auto-generated 400/500/600-bus power systems are solved using the H-LDE method, and the speed-ups are evaluated on both the Intel® i5-7300HQ 2.GHZ CPU with 8G RAM and the NVIDIA® Tesla V100 GPU platform with 5012 cores [82] by comparing with the Gauss-Jordan method, original LDE method, LU factorization with Gauss's algorithm [5] and KLU sparse matrix equation solution [79] method.

5.5.1 Speed-Up of GPU-Based Parallel H-LDE Computation

The performance evaluation of GPU-based parallel H-LDE computation is separated from the CPU-based sequential computation, because they have different orders of magnitude in latencies and different application contexts. In this case, the conductance is regarded as changeable during simulation, therefore, the Gauss-Jordan (GJ) method is chosen as the base, since the pure LU factorization without re-ordering has a slightly larger complexity for a changeable conductance matrix and could not expose much parallelism possibilities compared to the GJ method. The matrix equation solver latency of the proposed H-LDE method is compared with the traditional Gauss-Jordan (GJ), Schur Complement (SC) and original LDE (O-LDE) method. In this chapter, 2-level dynamic parallelism is exploited: the computation procedure of the GJ method could be not expose much parallelism, although some rows and columns of the pivoting or reduction operations can be computed in parallel; the matrix inversion of the block matrices generated by the SC and O-LDE method could be computed in parallel for the first level parallelism, but the application

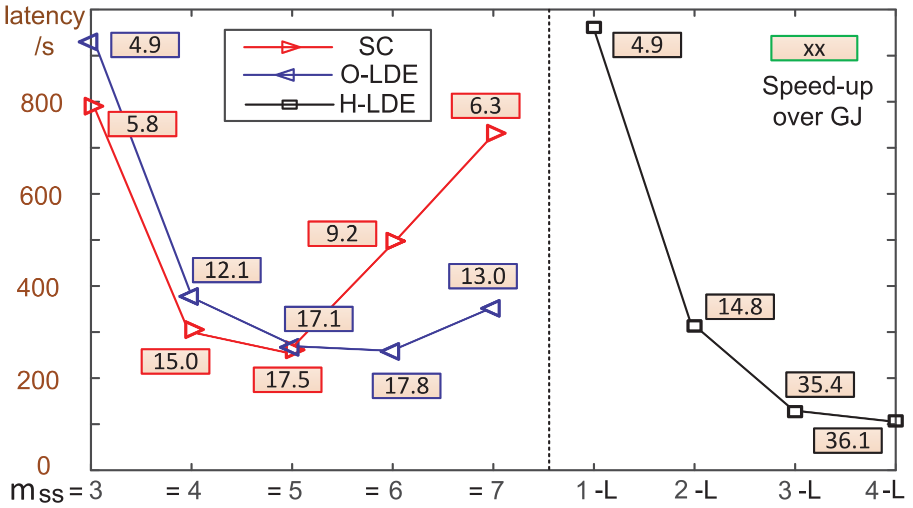


Figure 5.7: GPU-based computational time comparison between the SC, O-LDE and H-LDE method under different numbers of decomposed subsystems and different decomposition levels (Latencies of 5000 time-steps of matrix equation solution).

of the second level parallelism for computing the decomposed block matrices in parallel should be evaluated for different sizes of the block matrices due to the overhead of launching child kernels. For the H-LDE method, the 4-level decomposition and 2-level dynamic parallelism are already described in Section 5.4.2. Note that since the history item updating also occupies a considerable time for each power equipment, in this comparison, only the matrix equation solution time is recorded and compared.

The computational latencies under different number of decomposed subsystems for SC and O-LDE, and under different levels of H-LDE are shown in Fig. 5.7. Note that the GJ method was selected as the base, which is not shown in the figure. The time-step size is set at 20μs, and 5000 steps of matrix equation solution latency in total was recorded. From Fig. 5.7 we can see that the SC and O-LDE method can achieve their maximum speed-ups over the GJ method when m_{ss} reaches to 5 and 6 respectively. That is their full potential because they are both one-level decomposition methods. However, H-LDE could achieve the maximum speed-up of 36.1 over the GJ method at fourth level (4-L), nearly 2 times of performance over the original LDE method, which is quite significant.

Besides, it can also be observed that as the number of levels increases, the improvement of speed-up slows down, which means that the hierarchical LDE could only divide the topology with a certain number of levels to achieve the maximum performance, and when r is greater than that number, the performance will slow down, as analyzed in Section 5.3.3.

5.5.2 Speed-Up of CPU-Based Sequential H-LDE Computation

The sequential computation is commonly used in EMT power system simulators, and in this case, the IEEE 39, 57, 118, and 300-bus benchmark test power systems are evaluated using the H-LDE method. To extend the system scale, the 400, 500 and 600-bus topologies are also generated using the randomized link generation with the row density of 4, which is in the typical row density range of power system conductance matrices [79]. Typically, there are two types of circuits that may influence the selection of proper solvers: circuit with constant conductance matrix, such as the IEEE 118-bus system; circuit with changeable conductance matrix, such as the AC power system with switches installed or the multilevel modular converter (MMC) circuit in AC/DC grids. All of the IEEE benchmark AC test power systems have constant conductance matrices, in this chapter, to obtain a changeable conductance matrix, several time-varying loads are installed in the power system. For example, in the IEEE 118-bus power system, the original consumed active and reactive power of the load on Bus 3 are 0.414pu and 0.1062pu; in this case study, the consumed power is changing from [0.8-1.2] times of the original load every 1ms, that is, every 50 time-steps with the 20μs time-step size.

Constant Conductance Matrix. For a constant conductance matrix, the LU factorization and KLU sparse solver have obvious advantages over the GJ method, since the L and U matrices can be computed in advance, and in the subsequent time slots solving $\mathbf{LU}\boldsymbol{x} = \boldsymbol{b}$ can be simplified into the forward and backward substitution: solving $\mathbf{L}\boldsymbol{y} = \boldsymbol{b}$ and $\mathbf{U}\boldsymbol{x} = \boldsymbol{y}$. Similarly, the corresponding matrices ($\mathbf{G}_d^{-1}$, $\mathbf{C}$, and $\mathbf{Q} = (\Lambda^{-1} + \mathbf{C}^T\mathbf{G}_d^{-1}\mathbf{C})^{-1}$) in the H-LDE method (5.6) can also be obtained in advance. Note that in this case, the procedures (5.7)(5.8) are not required since they are targeting reducing the computational effort for a changing $\mathbf{Q}$ matrix but not reducing the storage cost. Therefore, for a constant conductance matrix, the H-LDE can be executed as:

$$
\begin{aligned}
v &= [\mathbf{G}_d^{-1} - \mathbf{G}_d^{-1}\mathbf{CQC}^T\mathbf{G}_d^{-1}]i_{eq} \\
&= \mathbf{G}_d^{-1}i_{eq} - \mathbf{G}_d^{-1}\mathbf{CQC}^T\mathbf{G}_d^{-1}i_{eq} \\
&= v_{DBM} - (\mathbf{G}_d^{-1}\mathbf{C})\mathbf{Q}(\mathbf{C}^T v_{DBM})
\end{aligned}
\tag{5.21}
$$

In this process, only the matrix multiplication operations are required, and since $\mathbf{G}_d^{-1}$ is a block diagonal matrix, and $\mathbf{C}$ only contains a small number of 1/-1 elements with the rest of 0 elements, the computation of each time-step will be extremely fast. And compared to storing the entire matrix inversion, the storage cost is also reduced a lot, as analyzed in Section 5.2.2.

The computational latencies (ms) of different test power systems are shown in Table 5.1, the duration of simulation is 0.1s, which is 5000 steps with 20μs time-step size. Note that the latency is the pure matrix equation solution latency without the power equipment circuit and history item updating latency involved for a pure comparison of different computational methods. In this case, the matrix input format for the KLU program is transferred

Table 5.1: Computational Latency of 5000 Steps with Constant Matrix

Scale	LU	KLU	H-LDE	Sp-LU	Sp-KLU
39-bus	12 ms	12 ms	4 ms	3.00	3.00
57-bus	24 ms	23 ms	9 ms	2.67	2.56
118-bus	94 ms	59 ms	26 ms	3.62	2.27
300-bus	591 ms	161 ms	150 ms	3.94	1.07
400-bus	1,024 ms	249 ms	237 ms	4.32	1.05
500-bus	1,638 ms	298 ms	396 ms	4.14	0.75
600-bus	2,549 ms	361 ms	707 ms	3.61	0.51

Sp-LU: H-LDE speed-up over LU factorization with Gauss's algorithm;
Sp-KLU: H-LDE speed-up over KLU.

into column compressed format in advance [79]. As can be observed in the results, the computation latency of the H-LDE method is always lower than the LU factorization with Gauss's algorithm in the 600-node system scale, because without the re-ordering and pivoting techniques involved, the generated L+U matrix is a dense matrix and thus requires more computational effort compared to the simple multiplication operations in H-LDE.

However, since the sparse techniques are included in the KLU package, the H-LDE shows less scalability than KLU due to the sparsity of the generated L+U matrix. When the system scale is smaller 400-bus, H-LDE can achieve lower computation latencies; but when the system scale increases larger, the decomposed block matrix sizes increase, and then the influence of large storage and I/O cost could not be omitted, although the H-LDE method requires much less storage than the original LDE method. The scalability of H-LDE on different processors with different RAM sizes may be different, but this result at least shows that the H-LDE method can only achieve lower latencies than the sparse LU methods such as the KLU method within a certain system scale.

Changeable Conductance Matrix. For a changeable conductance matrix, the entire H-LDE computation procedure (5.6)(5.7)(5.8) is required; for example, the maximum size of matrix equation to be solved in the IEEE 118-bus configuration is 15×15 in solving (5.8), and the maximum size of matrix to be inverted is 6×6 as discussed in Section 5.4.2, which are much smaller than the 118×118 conductance matrix. Besides, if only the values of matrix elements change but the node connections do not change (that is, the non-zero elements locations do no change), the computation of the H-LDE method can also be accelerated a lot like the KLU method. The pre-processing of KLU including the permutation to block triangular form (BTF) and fill-reducing ordering could be reused for each time-step due to the same matrix pattern; and the H-LDE can hold the same multi-level partition at each time-step, which means the structure of assembling the high-level small inverted block matrices remains the same.

The computational latencies of 5000 time-steps of different test power systems are

Table 5.2: Computational Latency (ms) of 5000 Steps with Changeable Matrix

Scale	LU	KLU	H-LDE	Sp-LU	Sp-KLU
39-bus	74 ms	138 ms	31 ms	2.39	4.45
57-bus	196 ms	339 ms	90 ms	2.18	3.77
118-bus	1,379 ms	953 ms	331 ms	4.17	2.88
300-bus	19,808 ms	2,536 ms	1,371 ms	14.45	1.85
400-bus	37,814 ms	3,493 ms	2,476 ms	15.27	1.41
500-bus	89,133 ms	4,538 ms	5,702 ms	15.63	0.80
600-bus	218,758 ms	5,101 ms	13,115 ms	16.68	0.39

shown in Table 5.2, the H-LDE speed-up over the LU factorization without sparse techniques involved is increasing as the system scale increases, which shows that the H-LDE method is more scalable than the pure LU factorization. For the KLU, since the pre-processing procedures are involved, it could not obtain good performance in the small scale system. But as the fill-in reducing algorithms are applied, the generated L+U matrix is still a sparse matrix and thus the method can scale very well when the system scale expands. For the H-LDE method, the computational procedure for computing the block matrices inversions can be significantly accelerated, and thus within the 400-bus system scale it can even achieve lower latencies than the sparse KLU method. It can be expected that if the CPU multi-core parallel architecture is utilized, the block matrices inversions can be computed in parallel to achieve a faster speed. Therefore, although sparse techniques are suitable for large-scale circuit simulation, the H-LDE method can also be applied for the fast and parallel EMT power system simulation within a certain power system scale.

5.6 Summary

In this chapter, the capability of LDE method is fully exploited by the proposed hierarchical LDE (H-LDE) method. First, the LDE computation procedure is improved to avoid storing the entire inverted conductance matrix to reduce the storage cost. Based on this, the detailed decomposition configuration for a specific number of decomposition levels is presented. Then, the complexity of H-LDE decomposition is analyzed, based on which the two decomposition principles are proposed to improve the decomposition performance. Based on the proposed decomposition principles, the IEEE 118-bus test power system is decomposed into four levels to demonstrate the application of the H-LDE method. The simulation results and speed-ups over the original LDE method, the classical LU factorization and the KLU sparse matrix equation solvers show that the proposed H-LDE method could achieve a lower latency in the 400-bus level power system. Therefore, the H-LDE method can be applied for the fast and parallel power system circuit simulation within a certain power system scale.

6

Real-Time Co-Emulation Framework for EMT-Based Power System and Communication Network on FPGA-MPSoC Hardware Architecture

With the expansion of smart grid infrastructure world-wide, modeling the interaction between power systems and communication networks becomes paramount and has created a new challenge of co-simulating the two domains before commissioning. Existing co-simulation methods mostly concentrate on the off-line software-level interface design to synchronize messages between the simulators of both domains. Instead of simulating in software with a large latency, this chapter proposes a novel real-time co-emulation (RTCE) framework on FPGA-MPSoC based hardware architecture for a more practical emulation of real-world cyber-physical systems. The *discrete-time* based power system electromagnetic transient (EMT) emulation is executed in programmable hardware units so that the transient-level behaviour can be captured in real-time, while the *discrete-event* based communication network emulation is modeled in abstraction-level or directly executed on the hardware PHY and network ports of the FPGA-MPSoC platform, which can perform the communication networking in real-time. The data exchange between two domains is handled within each platform with an extremely low latency, which is sufficiently fast for real-time interaction; and the multi-board scheme is deployed to practically emulate the communication between different power system areas. The hardware resource cost and emulation latency for the test system and case studies are evaluated to demonstrate the validity and effectiveness of the proposed RTCE framework.

6.1 Introduction

With the innovations in electric power grids and information and communication technologies (ICT), the traditional power systems are evolving into complex cyber-physical systems (CPS), which increasingly leverages the capabilities of data communication and computation to enhance the flexibility of power transfer [6] in the smart grid (SG). One major concern in smart grid research is to simulate the entire behavior of the system to adequately evaluate the influence of the interplay between digital world and physical equipment. However, since the power system and communication network have distinctly different working principles and simulation tools, simulating the entire behavior of the two domains (so-called co-simulation) is quite challenging [6, 43].

Various co-simulation frameworks have been proposed in the recent past since the first interface for the PSCAD/EMTDC® simulator [36] to integrate an agent-based distributed application into the simulation was developed. Most of these works are not to design a complete simulator that could finish all required functions in one software package but attempt interfacing two existing simulators of each domain, because there are already various mature power system and communication network simulators available. Unfortunately, there does not exist universally accepted interfaces for data exchange between simulators of the two domains due to their different working principles or even operation systems. Thus, the main concern of these co-simulator frameworks [37–42] is to properly handle data exchange and synchronization for related events in both domains at runtime. Most co-simulators mainly distinguish from each other by the type of power system/communication network simulators used, the methods applied for synchronization, and the application scenarios. For example, EPOCHS [40] is mainly used for wide area monitoring, which uses PSCAD/EMTDC® and PSLF® for power system simulation, and NS-2 for communication network simulation; the periodic synchronization mechanism is adopted. GECO [41] interfaces the power system simulator PSLF® and network simulator NS-2, and the synchronization is based on a global event driven on-demand mechanism. INSPIRE [42] uses dynamic synchronization points for the interfacing of DigSilent® and OPNET simulators. In [84] the Matlab and C++ based platforms are applied for power system and cyber system simulation respectively, and the how a cyber-contingency affects power system operations was investigated. The co-simulation platform presented in [85] was implemented based on OPAL-RT® real-time simulator and Riverbed Modeler to examine the power grid vulnerabilities to cyber-physical attacks.

However, the performance of the software-based simulators is relatively low compared with actual power and communication network devices even without taking the data exchange and synchronization time between two simulators into account. If the electromagnetic transient (EMT) is concerned, the simulation process will be extremely slow due to the small time-step size and massive synchronization requirements. It is therefore difficult to practically simulate and test the adequacy of manufactured protection and control equipment responding to potential damage created by transient events in real-time. Although the real-time power system simulation approaches have been extensively studied in past works, the real-time co-simulation of power system and communication network has rarely been investigated. In [86] and [87], the real-time co-simulation frameworks were discussed. Similar with the other software-based co-simulators, in these two works, the interface design and synchronization scheme are still the main concerns. To achieve real-time power system simulation, the commercial simulator RT-LAB® was utilized; thus the implementation details of the entire co-simulator are absent. In [88] and [89], the hybrid hardware and software platforms are applied: the communication network simulation ran on the OPNET software simulator on PC computers, while the real-time power system simulation was conducted on the RTDS® hardware simulator. The interface design between the two different platforms (dedicated hardware and PC software) is even more complicated with loss of generality.

Different from the pure software-based or hybrid hardware and software based *co-simulation* discussed above, there is no relevant work available for the more practical hardware based *co-emulation* on FPGAs, due to the complexity and specificity of the cyber physical power system and FPGA/MPSoC platforms. Although the FPGA-MPSoC platform that enables flexible programmability and highly paralleled computing environment has been used in multi-rate mixed-solver [50], it was purely applied for power system EMT simulation and no communication networks were involved. If the entire resources including the fast and parallel computation capabilities of programmable logics and soft-processors, and the physical ethernet network ports could be leveraged for the co-emulation of the power system and communication network, it can be expected that the testing and evaluation process of cyber-physical energy systems would be more practical and be accelerated significantly.

Based on the above observations, this work proposes a real-time co-emulation (RTCE) framework executed on hardware FPGA-MPSoC platform instead of interfacing two software based simulators. To the best of our knowledge, this is the first work that conducts real-time co-emulation on hybrid FPGA-MPSoC hardware platform. In this work, the entire hardware architecture and implementation details for co-emulating the power system and communication network are given, which utilize all the programmable logics, soft-processor and networking port resources on the FPGA and MPSoC platform, and provide a new vision of co-emulation of emerging cyber-physical systems. The advantages and features of the proposed real-time co-emulator are as follows:

1. Power system EMT emulation is carried out on programmable hardware units so that transients can be captured in real-time;

2. Communication network emulation is modeled in abstraction for transmission-level networking and directly conducted on real-world physical ports to form the function-level networking;

3. Power system and communication network emulation modules are embedded within each FPGA/MPSoC board so that the data exchange and synchronization between the two domains are sufficiently fast for real-time interaction;

4. System monitoring and control applications are executed within the soft processors of FPGA-MPSoC platform so that the system control can be programmed flexibly to respond to the physical system events such as overcurrents or communication breakdown such as link failures.

Based on the above advantages, the proposed RTCE framework can achieve real-time co-emulation of power system and communication network, which resembles a real-world cyber-physical system. The IEEE 39-bus system is chosen as the test case emulated on the hybrid FPGA-MPSoC hardware platform, and the real-time emulation results are captured for the overcurrent and communication link failure case studies.

6.2 Co-simulation Background

In general, simulation plays an important role in design and test of both power system and communication networks. Power system simulation is mainly used for system planning and operating, while the network simulation is mainly used for developing and testing new architecture or protocols. This section introduces the basic concepts and commonly used methods in both simulation domains, and then presents the main challenges of power-communication co-simulation.

6.2.1 Power System Simulation

The power system simulation methods can be classified into two categories: steady-state simulation and continuous-time simulation. Steady-state simulation is mainly used for power flow calculation, which uses large time steps (typically seconds/minutes) and is not able to capture the transient dynamics. Continuous-time simulation is commonly used to capture the transient behaviour through continuous time model, which requires solving the system equations within a very small time-step (typically at microsecond level). Since in most cases co-simulation is used to analyze the impact of short-term communication failure or the instant response of system control, continuous-time simulation is mainly applied for dynamic power system simulation and therefore is the main focus of this work.

In EMT simulation, each power equipment can be modeled into an equivalent circuit using basic power elements such as the resistor, capacitor and current source. Then the whole power system can be described using a set of differential equations obtained by gathering all the power equipment models and applying circuit laws, where the state variables of differential equations are to be solved. The numerical integration algorithms such as Trapezoidal Rule and Backward Euler methods are usually applied to those differential equations to solve the state variables within each discrete time-step. Therefore, the EMT power system simulation is discrete-time based simulation, and a small time-step size (typically at microsecond level) is always required to capture the transient level behaviours.

6.2.2 Communication Network Simulation

Different from power systems, the elements in communication networks are modeled not based on physical principles but on their *functionality*, and thus there are no differential equations or matrix equations to be solved. Instead, the components in communication networks simply receive, modify and send data packets, and thus the process of each network element is always modeled into a sequence of *discrete events* that occur unevenly in time, where each event represents an operation of packet processing or transmission. In this context, the communication network simulator refers to a program running on software or hardware to simulate the behaviour of a user-defined network topology, and each node in the simulator executes the packet processing procedure similar to real-world equipment.

The basic network component in the cyber-physical power system is the sampling and reporting device, or a smart meter, which is responsible to measure the electrical values of the power system and report the measurements to the system controller. The phasor measurement unit (PMU) [90] is a commonly used basic network component that is deployed in each single process bus or substation model to measure the electrical quantities; and after sampling the data values it computes the corresponding phasor values and then reports them to the phasor data concentrator (PDC), which is responsible to monitor a power area and collects the measurements from the PMUs in the area. In fact, the main function of the communication network is to provide a two-way communication between smart meters, data concentrators and controllers to modify the related circuit parameters according to the service requirements.

Over the past, many communication network simulators have been adopted in co-simulation research, of which the representative examples are NS-2/NS-3, OMNeT++, OPNET, etc. NS-2/3 (from network simulator) is a series of discrete-event network simulators that are primarily used in academic research. The network protocols such as TCP/UDP, routing algorithms, multicast schemes are supported in this simulator, thus the traditional network function can be well evaluated. For example, both NS-2 and NS-3 were used in [91–95] for communication network modeling in the co-simulation. OMNeT++ is an open-source discrete event based simulator for communication networks, which was used to model smart grid in [96–98]. Different from NS-2/3 and OMNeT++, OPNET is a more powerful discrete event network simulator mainly for commercial use, enabling abundant network models such as wired networks, wireless networks and ad-hoc networks. OPNET offers a visual high-level user interface to access the C/C++ based blocks that are used for different models and functions, which also enables users to customize network applications in multiple co-simulation approaches [99–101].

6.2.3 Co-Simulation

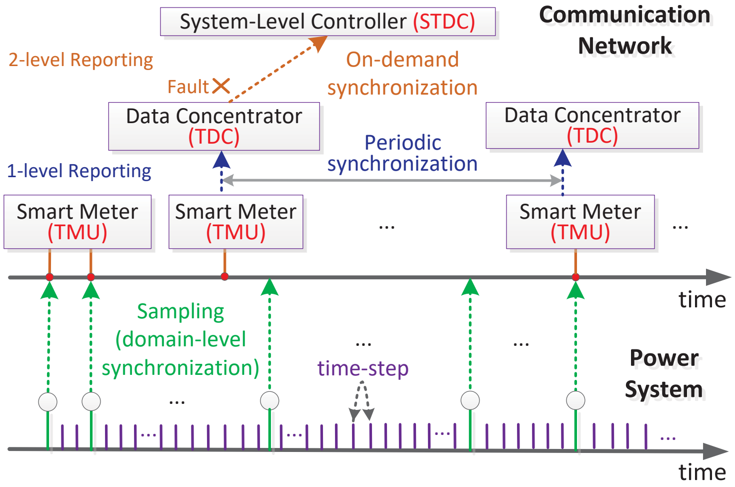


Figure 6.1: Example of interaction between power system and communication network simulation.

Power system and communication network co-simulation refers to simulating the power system operations while taking the communication network layer into consideration, so that the impact of interaction between the two domains can be evaluated and the entire system features can be revealed. However, power system simulation is time-continuous and the simulation proceeds step-by-step in discrete time, while the communication network simulation is event-driven where the events are unevenly distributed in time. Thus, the main challenge in co-simulation is how to interface the two domain simulators with significantly different work principles.

The interaction between the two domains includes the sampling and reporting operations. The sampling operation refers to measuring and digitizing the values of the concerned electrical quantities, which is also called *domain-level synchronization*, since it is used to exchange the data of the two domain simulators. The domain-level synchronization does not involves the network packet encapsulation, which is usually performed periodically at a constant sampling rate for a stable measurement. The reporting operation refers to reporting the measured electrical values from a smart meter to a power area data concentrator, and from a data concentrator to the system controller. It is called the *application-level synchronization*, since it is based on the network data packet and is handled within the communication network domain for the application-level purpose. The application-level synchronization contains a two-stage reporting. The representative approaches for synchronization in each level are the periodic and on-demand methods: when applying the periodic synchronization strategy, the reporting only happens at each synchronization point; for the on-demand synchronization, the data-exchange is only initiated if an event of interest is detected, and under normal conditions the corresponding measurement values are regarded as unchanged. An example is shown in Fig. 6.1, it can be seen that the domain-level synchronization is performed at a constant sampling rate; while the two stages of the application-level synchronization are performed using the periodic and on-demand strategies respectively.

6.3 Proposed Real-Time Co-Emulation (RTCE) Framework

Based on the co-simulation background, this section describes the architecture and work principles of the proposed hardware-based real-time co-emulator.

6.3.1 Motivation

To conclude from the state of the art, most leading research in this area is either built on existing simulators or the development of a middleware to interface simulators from the two domains; thus their performance is dramatically limited by the software environment and achieving fast simulation for large-scale cyber-physical power systems becomes difficult. There comes an alternative strategy into mind that combines both programmability and high-speed computation: hardware-based co-emulation. Furthermore, the real-time power system simulation has already been extensively carried on FPGA, and communication network functions such as Ethernet, TCP/IP are also available as vendor specific IP cores or soft-codes built on FPGA. The multiprocessor system-on-chip (MPSoC) [54] integrates the FPGA based programmable hardware logic and the ARM® based multi-core processor systems within one platform, which enable complex embedded applications to be developed. So why not deploy FPGA-MPSoC platform for implementing power-communication co-simulation? First, the FPGA has a large amount of hardware programmable logic resources that can be customized by the engineer to achieve specific functions; secondly, there are rich I/O transceivers in FPGA that enable several prototyping boards to be interconnected to extend the compute capability for real-time power system EMT emulation or for emulating real communication network functions. In addition, the processors integrated on the FPGA chip (such as Microblaze processor in Xilinx® FPGAs) or the additional ARM® based APU processors in the MPSoC board provide the ability to develop monitor and control systems for the cyber-physical system. Thus in this work, a real-time co-emulation (RTCE) framework based on an integrated FPGA-MPSoC hardware platform is proposed to take advantage of the above mentioned features.

6.3.2 RTCE Hardware Architecture

The top-level architecture of RTCE framework is shown in Fig. 6.2, wherein the power system EMT emulation and the communication network emulation are integrated on each board. In power system domain, each board emulates a subsystem of the entire system, and emulation results are exchanged with connected boards in each time-step; and in communication network domain, each board emulates the information networking between the buses located in its emulated area. Instead of using the commonly used PMU and PDC concepts, in this work, the concepts of transient measurement unit (TMU) and transient data concentrator (TDC) are used, which is to emphasize that the measurement data is obtained from the instantaneous transient level EMT power system simulation; based on the transient level simulation results, the sampling operations of PMUs are more practically emulated than using the results from steady-state equations. The TDC is responsible to monitor the emulated subsystem and collect the measurements from the TMUs; TDC does not have a global view and control applications of the whole power system, thus the super TDC (STDC) is also modeled to receive messages from TDCs and compute the centralized control strategies when detecting abnormal conditions. The corresponding TMU, TDC, and STDC components are marked in Fig. 6.1.

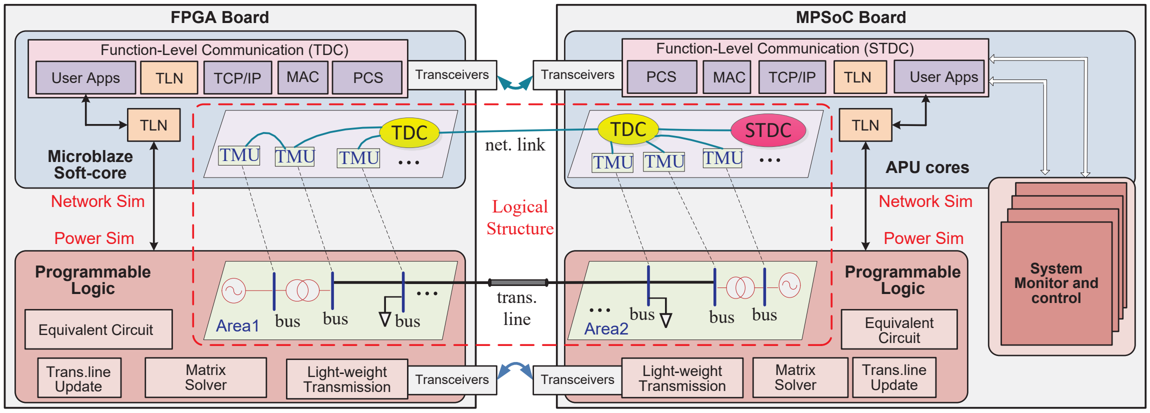


Figure 6.2: Demonstration of real-time co-emulation (RTCE) architecture on FPGA-MPSoC platform.

Power system EMT emulation: For large power systems, it is difficult to emulate the whole circuit topology in one board due to the requirements of huge amount of hardware logic resources and memory. Thus, a common approach is to decompose the system into several subsystems using the inherent latency of the widely distributed transmission lines, and these decomposed subsystems can be allocated to different boards for parallel processing. After the results are obtained in each time-step, they should be exchanged among adjacent subsystems using fast data exchange path, as shown in Fig. 6.2. It is enough to use light-weight communication such as the Xilinx® Aurora protocol for fast data exchange. In EMT emulation, each subsystem can be abstracted into an equivalent circuit along with a matrix equation to be solved, and transmission line data should also be calculated and exchanged between the adjacent subsystems. Thus the power subsystem emulation can divided be into three parts: equivalent circuit parameters updating, matrix equation solution and transmission line data updating, as illustrated in Fig. 6.2.

Communication network emulation: Since each FPGA/MPSoC board has limited I/O resources and soft-processors, it can only operate as one network device. However, the network nodes (TMUs) in a smart grid can even have the same number as the buses, so it is impossible to emulate all of the network nodes with bit-level details in hardware and obtain the correct transmission parameters between each TMU and STDC. Thus in this work, RTCE emulates the communication network with the two-level model: transmission-level networking and function-level networking. The transmission-level networking is used to model the end-to-end packet transmission characteristics such as delay and loss, while the function-level networking is used to emulate the TDCs and model the specific network functions or protocols such as Ethernet or TCP/IP. The transmission-level emulation is inspired by work [102], which simply models individual network components such as routers, hosts, links, and protocols with specific abstractions rather than completely implementing all the details to avoid complex hardware-based networking executions that are not concerned in power system analysis. In this work, the transmission-level networking models all of the transmission characteristics between TMUs and TDCs and between TDCs and STDC using parameters such as the link losses and delays, which can be obtained through software simulators such as NS-2/3. The function-level networking is carried out directly on the I/O resources of FPGA/MPSoC boards, because the physical coding sublayer (PCS) and media access controller (MAC) functions can be implemented using hardware logic resources, and the Internet protocol (IP) or other upper network layer functions (TCP, UDP, etc.) can run in the soft-processors (such as Xilinx® MicroBlaze) to handle the complex control tasks such as routing, connection establishment and flow control.

As shown in Fig. 6.2, each TMU receives the measurement from power system emulation as the sampling operation, and then sends the data packet to the corresponding function-level TDC module via the transmission-level networking (TLN) module as the reporting operation. Then the TDCs send data packets to the function-level STDC module via the real-world communication link if detecting abnormal conditions, but still need to pass the TLN module in the function-level networking to emulate the transmission process between the TDC and STDC since TDCs/STDC are not directly connected in practical network. Using this method, the specific network functionalities can be emulated to observe the interaction between power and communication domains, while the detailed transmission processes between TMUs, TDCs and the STDC are simplified and abstracted as crucial and practical transmission parameters.

Co-emulation: In each board, power system EMT emulation and communication network emulation run concurrently. Thus, the data exchange and synchronization between the two domains are simplified: the results of power system emulation in each bus and each time-step can be output to the network emulation module directly without any delays to emulate the sampling process of the TMU in a single process bus. That means, at every hardware clock the results of the two domains can be exchanged within the FPGA logic. If the sampling rate is assigned practically, this data measuring process can also be done for every specific time space. The TMU in each bus is connected with the TDC via the TLN module to send measurement data packets periodically, and TDCs check the received measurement data to find if there exists abnormal conditions; if some transients are detected, the TDC will send messages to STDC for system control command. This on-demand TDC-to-STDC reporting strategy could reduce the amount of generated data packets and thus reduce the congestion of the communication network. The control command from the STDC that may change the circuit topology or other circuit parameters can also be sent to the controllable power devices in the same way. TDCs can run in the soft-processors of FPGA platform, but the system-level monitor and control of STDC should run in the soft-processors of MPSoC platform because the soft-processors in the MPSoC board have more computation resources and a higher clock frequency.

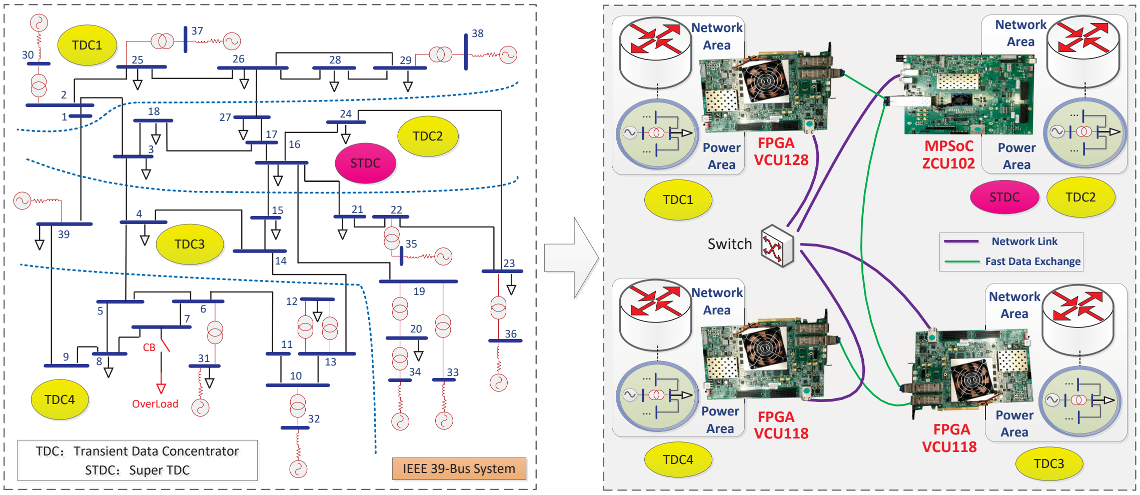


Figure 6.3: Example of co-emulating a cyber-physical system on multi-board hardware platform.

6.4 Hardware Implementation of RTCE

To test and verify the advantages of the proposed RTCE framework, the IEEE 39-bus system [57] is selected as the test system, as shown in Fig. 6.3(a). The test system is partitioned into four areas, and TDCs are deployed in each area (at Bus 25, 24, 4, 9 respectively) to accumulate the measurements from each bus and inform the STDC deployed in area 2 (at Bus 24) if detecting abnormal conditions. The hardware RTCE emulator is developed to co-emulate the test system in real-time. As shown in Fig. 6.3 (b), there are 3 Xilinx® VCU118/VCU128 FPGA boards and 1 Xilinx® ZCU102 MPSoC board used in this emulator, which emulate the partitioned four areas concurrently. The communication network links between RJ-45 ports of each board are connected via a switch, while the fast data exchange links for power system emulation are connected on the SFP/SFP+ ports via the 10G optical fiber. The detailed hardware block design in each board is shown in Fig. 6.4.

6.4.1 Multi-Board EMT Emulation

In this work, the applied power equipment models (synchronous machine, power transformer, transmission line and loads, etc.) are just the same as those of PSCAD/EMTDC® [12] for verification. For real-time multi-board EMT emulation, the power system is usually partitioned by the long transmission lines. Then the fast data exchange path is required to exchange the transmission line model data among adjacent subsystems. In this work, the lightweight communication module is used: Xilinx® 64-bit Aurora core. Instead of using the complex communication functionalities such as the Ethernet protocol, the simple high-speed Aurora communication core enables direct data transfer through different types of transceivers on FPGA/MPSoC boards with up to 10Gb/s speed.

To interact with the communication networking emulation, the abnormal condition detection is the prerequisite to determine the data packets generation. State estimator is usually used to predict the next state variable values and be compared with the real value to determine if the system is running in control. For transient-level analysis of this work, the local truncation error (LTE) is used to estimate the perturbation, and the abnormal condition is found once the LTE or the measured current and voltage exceed the predetermined threshold. After the main conductance matrix is solved in each time-step, the state variables of linear equipment are obtained, and then the LTE of the n^{th} time-step is computed for the linear equipment, given by [25]:

$$LTE(t_n) \approx C_{p+1}\Delta t_n^{p+1}(p+1)!\,\boldsymbol{g}[t_n, ..., t_{n-1-p}] \tag{6.1}$$

where C_{p+1} is the error constant of a specific discretization method, p is the order, and $\boldsymbol{g}[t_{n-1}, ..., t_{n-1-k}]$ can be calculated as:

$$\boldsymbol{g}(t_{n-1}) = \boldsymbol{x}_{n-1} \tag{6.2}$$

$$\boldsymbol{g}[t_{n-1}, ..., t_{n-k}] = \frac{\boldsymbol{g}[t_{n-1}, ..., t_{n-k+1}] - \boldsymbol{g}[t_{n-2}, ..., t_{n-k}]}{t_{n-1} - t_{n-k}} \tag{6.3}$$

For the subsystems where nonlinear equipment exists, the iterative approach is involved. The standard method is to first use an explicit method or interpolation polynomial (called the predictor) to compute a candidate value of the state variable to be solved, and then use it as the initial solution to apply Newton's method for the implicit integrator (called the corrector) until convergence is achieved. Then the LTE can be estimated by comparing the initial solution $\boldsymbol{x}_n^0$ and final solution $\boldsymbol{x}_n$ [25]:

$$LTE(t_n) \approx \frac{C_{p+1}}{1 - C_{p+1}}(\boldsymbol{x}_n - \boldsymbol{x}_n^0) \tag{6.4}$$

6.4.2 Communication Protocol and Implementation

In order to practically emulate the communication network, the standardized communication protocol, IEC 61850 - Communication Networks and Systems in Substations [103], is applied in this work. IEC 61850 is used to manage the large number of communication devices. The existing communication protocols such as the Ethernet, TCP/IP typically define how data bytes are transmitted on the wire, however, they did not specify the application layer data organization. The 61850 model defines how the application data bits should be arranged and how the created data items/objects and services are mapped to any other existing under layer protocols that can meet the service requirements. In this work, all the measurement data packets are encapsulated based the IEC 61850 standard that defines how to transmit synchrophasor information according to IEEE C37.118 [104].

In the communication network emulation, each bus is alligned with a TMU, which is able to digitize the base measurement quantities at the source and transmit the resulting sample values to the TDC. The measurement data includes voltages, currents, and some status information (LTE). As introduced in Section 6.3.2, in RTCE architecture the data is transferred from power emulation to communication networking emulation in each time-step to do real-time EMT analysis, which means the sampling rate of TMU is related to the actually applied time-step size. Different from the TMU measurement gathering that does not involve network packet generation, the TDC collects data from the TMUs in its area based on a specific reporting rate. In this work, the reporting rate from TMU to TDC is set at 60Hz. As mentioned before, each TDC is required to compare the LTE, current or voltage with the specific threshold; once the measurement exceeds the threshold, the abnormal condition is detected and then it is responsible to create a measurement data packet to be sent to the STDC.

Transmission-level networking. The data packet networking is achieved in two modes: transmission-level networking and function-level networking. In the transmission-level emulation, the transmission process between TMUs and the TDC and between TDCs and the STDC is modeled as bandwidth, communication delay and loss rate. The transmission-level networking is implemented in the soft-processor of FPGA/MPSoC boards, because the transmission parameters can be flexibly configured in software without modification of the hardware design. As illustrated in Fig. 6.4, for example, if the TMU in Bus 6 generates a packet to be sent to the TDC at Bus 9, then the data packet is generated in the hardware module and then is sent to the soft-processor and passes the TLN function; and if the TDC (run as the function-level networking module) at Bus 9 detects an abnormal condition and send a data packet to the STDC at Bus 24, the packet needs to first pass the TLN function in the application layer of TCP/IP stack and then be sent to the lower TCP/IP and MAC layer. The implementation of the TLN function is quite straightforward: based on the Xilinx® timer() function, the TLN function detects the source and destination of the input packet and searches for the corresponding end-to-end transmission delay; then, after the delay outputs the input data packet. The real communication delay is related to the distance and hops between the source and destination, while the loss rate refers to the possibility of an unsuccessful transmission process. In this implementation, the values chosen for the delay and loss are determined based on the testing results from the real network simulator NS-3 [105].

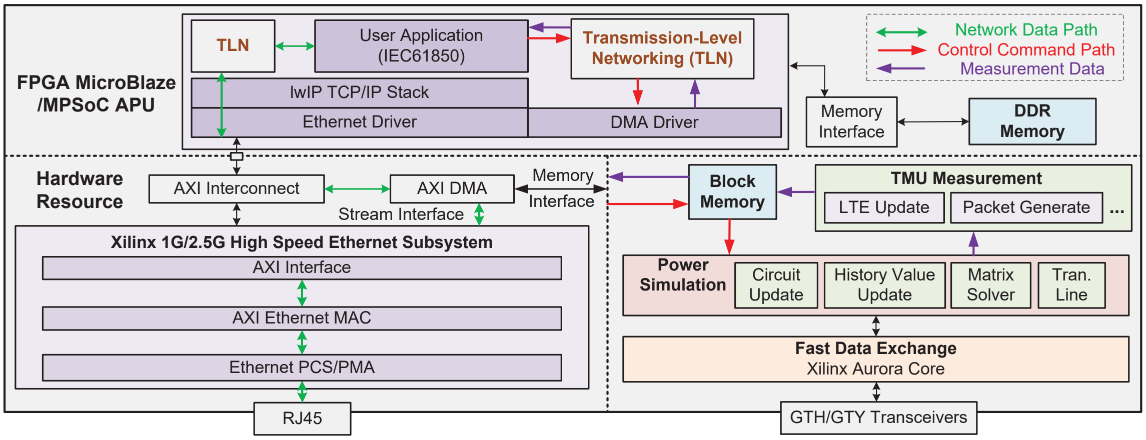


Figure 6.4: Illustration of detailed block design on a single FPGA/MPSoC board.

Function-level networking. For function-level communication network emulation, the high-level view is shown in Fig. 6.4. The Xilinx® 1G/2.5G High Speed Ethernet Subsystem core is used to implement the Media Access Controller (MAC) with a Physical Coding Sublayer (PCS) based on the hardware logic and I/O resources. The upper network layer functions are implemented in the Lightweight IP (lwIP) stack [106], which is an open source TCP/IP networking stack for embedded systems and is available in the Xilinx® programming tools. The Xilinx® Software Development Kit (SDK) provides lwIP software customized to run on various embedded systems that can be MicroBlaze soft-processor in FPGA chip or the ARM®-based SoC devices. Through the lwIP application programming interface, users can add networking capability to an embedded system. Since the echo server application that can create a TCP connection and sets up a callback on a connection being accepted is already provided by Xilinx® SDK, the IEC 61850 protocol and data encapsulation are implemented based on the existing echo server code to achieve the specific communication patterns of smart grid.

The function-level networking data path is marked as green arrows in Fig. 6.4, which is achieved by the stream interface between direct memory access (DMA) module and Ethernet module; the control command path that is used to send control commands from the controller in the soft-processor to the power emulation module is marked as red arrows in Fig. 6.4, which is achieved though the memory interface between DMA module and the block memory; the measurement data path that is used to send measurement data from each TMU to the TDC is marked as the purple arrows, which is transmitted via the transmission-level networking module. Note that the measurement data in the block memory on FPGA can only be passively read from the DMA driver function; that means the reporting operation from TMU to TDC is achieved by the application on the lwIP stack to use the driver function to read the values from the block memory periodically. As mentioned before, this reporting rate is set at 60Hz.

6.5 Real-Time Emulation Results and Verification

Based on the implemented RTCE hardware emulator, the test system is emulated to evaluate how the communication failure affects the power system, and how the power system transient affects the communication network.

Table 6.1: Hardware Resource Consumption of the Case Study

Resource	VCU128	ZCU102	VCU118-1	VCU118-2
LUT	40.3%	95.2%	57.1%	55.2%
FF	36.1%	70.1%	50.3%	48.9%
BRAM	73.3%	86.9%	77.4%	75.5%
DSP	35.6%	87.2%	64.1%	62.8%
Latency	10.1μs	8.7μs	16.2μs	13.1μs

6.5.1 Processing Delay and Hardware Resource Cost

For the hardware emulation, the FPGA boards (including the Microblaze softprocessor) run at the clock frequency of 100 MHz. According to the system partition and RTCE implementation described above, the hardware resource consumption and maximum processing latency for one time-step are presented in Table 6.1. The VCU118-1/2 boards emulate the power areas 3/4, while the VCU128 and ZCU102 boards emulate the areas 1 and 2 respectively.

The processing latency of one time-step includes the latency of power equipment model computation, matrix solution, transmission line updating, history item exchange (including board-to-board fast data exchange), and LTE computation, which indicates the minimum time-step size that can be applied for real-time EMT emulation. In this work the power areas in the four boards are emulated using the same time-step, 20μs, which is larger than the latencies of each subsystem and is sufficient for EMT-level analysis.

For the communication network emulation that runs concurrently, since the transmission-level communication behaviors are parameterized by real emulator results and the function-level emulation is just executed on the hardware transceivers for real world networking, the communication network emulation also runs in real-time. However, the real-time networking performance is actually confined by the transmission-level parameters and function-level hardware capabilities. In addition, the TCP/IP protocol execution and system-level control performance are limited by the computational capabilities of soft-processors in FPGA and APU cores on the MPSoC board. In this work, the tested function-level networking performance is 984Mbit/s based on the lwIP performance testing program; the transmission delay of each communication link in the NS-3 simulator is set at 1ms/200km, which is a classical value used in communication simulation; and the forwarding rate of each network device that impacts the delay of packet processing within a device is set at 100Mbit/s, which is also a commonly used value.

For the interaction between the two domains, the delay actually refers to the delay of the reporting operation between the FPGA MicroBlaze/MPSoC APUs soft-processor and the block memory on FPGA, since the sampling operation is performed directly on the programmable logic with zero delay. On the FPGA board, the tested throughput of the reading data operation of the driver function is 186Mbit/s; on the MPSoC board, the test throughput is 214Mbit/s. Note that the driver read function does not influence the power system simulation, which means, the data exchange delay between the two domains is sufficiently small for real-time interaction.

6.5.2 Case Study 1: Overcurrent of Load

To study how the power system abnormal conditions affects the entire cyber-physical system, a sudden load increase case is applied on Bus 7 at emulation time 2s, which results in a overcurrent condition. Upon detecting this condition, the STDC will perform the centralized algorithm for protection. The emulation results are shown in Fig. 6.5.

As shown in the results, after the abnormal condition is initiated, the corresponding message (including the bus voltage, load current, and LTE) is sent to STDC due to the sudden transient-level increase of LTE. Subsequent messages are generated because the load current exceeds the threshold value. And about 50ms after the fault, the controllable circuit breaker at Bus 7 receives the control command from STDC to open the load circuit for protection. The time delay between the fault and the response message mainly includes the transmission time and computation latency for control command. In this work, the control program is just developed for this case study thus it consumes a very low time cost. The transmission time consists of the in-board communication (referred to TLN delay between Bus 7 and TDC4 and between TDC4 and STDC) and board-to-board communication latency. Since the board-to-board communication delay is extremely low based on the fast speed of physical network ports, it can be omitted compared to the TLN delay that is measured through the software network simulator. The real-time emulation results indicate that the interaction between the two domains is quite fast. Besides, as shown in Fig. 6.5(b)(c), when co-emulating the two domains, the transient-level waveforms can also be captured and output in real-time to observe the details of the value change due to the small time-step applied.

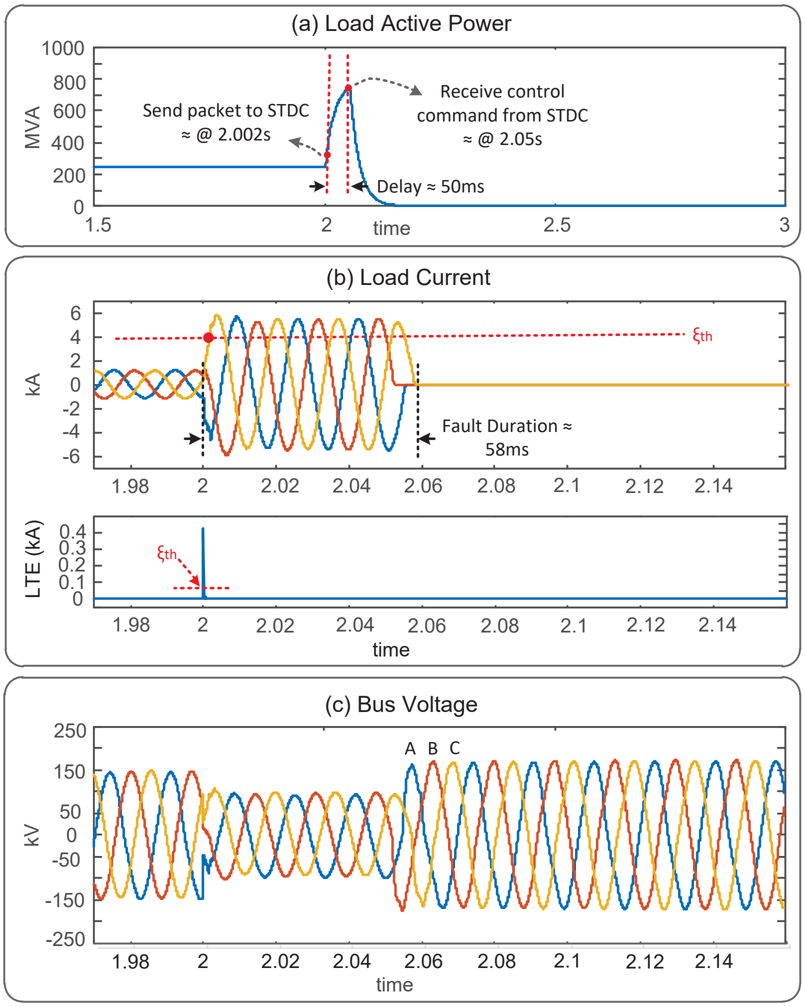


Figure 6.5: Overcurrent fault case study: (a) active power of load at Bus 7; (b) total load current at Bus 7; (c) voltage of Bus 7.

6.5.3 Case Study 2: Communication Link Failure

To investigate the effect of network link failure, the link between Bus 8 and Bus 9 that is essential for the communication between TMUs and TDC breaks down at emulation time 2s, while the sudden load increase is also applied at the same time. Since the on-demand synchronization strategy is applied, during normal conditions the influence is small, because the power system is regarded as working normally if there is no message sent from TDCs. However, when the overcurrent condition happens, this impact is essential to the system response. The system controller in STDC is required to respond to this situation according to the messages received from TDCs, however, since the link broke down, re-computing a new route costs a considerable time, and the generated new route has a much longer distance between the TMU at Bus 7 and TDC at Bus 9, which also results in an increased communication delay.

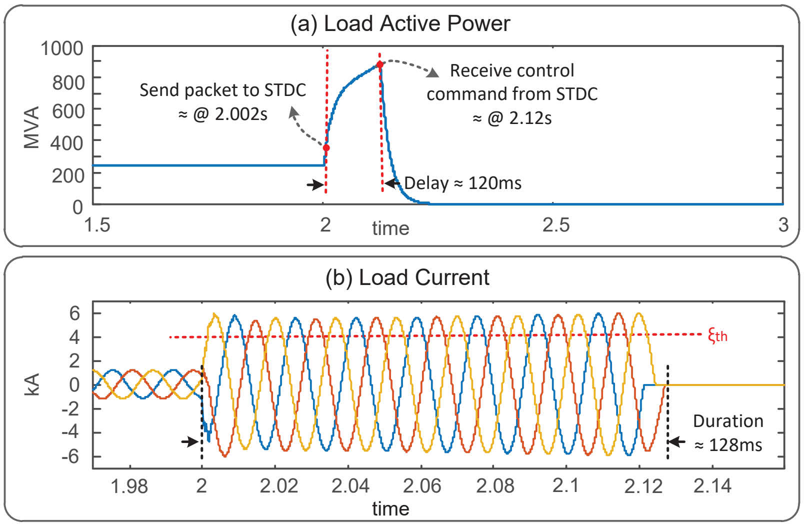


Figure 6.6: Communication link failure case study: (a) active power of load at Bus 7; (b) total load current at Bus 7.

As shown in Fig. 6.6, the system is supposed to respond to the overcurrent condition caused by the sudden load change within a short delay, however, due to the transmission link failure, the system response becomes relatively slow. Since the link failure impacts the transmission-level communication delay, the increased latency is actually obtained from the NS-3 simulator. From Fig. 6.6(b) it can be observed that the total duration of overcurrent condition raises to 128ms, which greatly increases the duration of damage.

6.6 Summary

For fast co-simulation of cyber-physical systems, this work proposes to use the FPGA-MPSoC based hardware platform instead of the software-based architecture to conduct the simulation. In the proposed real-time co-emulation (RTCE) framework, the power system EMT emulation is carried out on programmable hardware units so that transients can be captured in real-time; and the communication network emulation is modeled in abstraction for transmission-level characteristics and directly conducted on real-world physical ports to form the specific networking functions, which can emulate the communication networking in real-time. The power system and communication network emulation modules are embedded within each FPGA/MPSoC board so that the data exchange and synchronization between the two domains are sufficiently fast for real-time interaction. The IEEE 39-bus system test system is implemented and emulated on FPGA-MPSoC platform. The effect of power system faults and communication link faults are studied, and the real-time emulation results show the effectiveness of the RTCE emulator. The proposed RTCE framework can be utilized for the study of emerging cyber-physical systems as well as fast emulation of test systems.

7

Heterogeneous Real-Time Co-Emulation for Communication-Enabled Global Control of AC/DC Grid Integrated with Renewable Energy

The information and communication technologies (ICTs) are increasingly merging with the conventional power systems. For the design and development of modern AC/DC grids with integrated renewable energy sources, the system-level control schemes with ICTs involved should be evaluated in a co-simulation framework. In this chapter, a heterogeneous hardware real-time co-emulator composed of FPGAs, many-core GPU, and multi-core CPU devices is proposed to study the communication-enabled global control schemes of hybrid AC/DC networks. The electromagnetic transient (EMT) power system emulation is conducted on the Xilinx® FPGA boards to provide nearly continuous instantaneous waveforms for cyber layer sampling; the communication layer is simulated on the ARM® CPU cores of the embedded NVIDIA® Jetson platform for flexible computing and programming; and the control functions for modular multi-level converters are executed on GPU cores of the Jetson® platform for parallel calculation. The data exchange between FPGAs and Jetson® is achieved via the PCI express interface, which simulates the sampling operation of the AC phasor measurement unit (PMU) and DC merging unit (DC-MU). The power overflow and DC fault cases are investigated to demonstrate the validity and effectiveness of the proposed co-emulation hardware architecture and global control schemes.

7.1 Introduction

With the development of modern power systems, the information and communication technologies (ICT) are increasingly involved in the control, protection and normal operation of power system equipment and infrastructure, which enables the evolution of the so called cyber-physical system. In such a system, various power equipment such as the generator, circuit breaker and power electronics converter can be controlled by the centralized system-level controller through communication links [107]. The ICT-enabled power system operation and control needs to be evaluated in a co-simulation platform, which creates new challenges to the existing power system simulators [6, 43]. To simulate the entire behaviours of a cyber-physical system, the precise evaluation of power and communication system is required while the interaction between the two domains should also be considered. The existing co-simulation research is mostly focused on the software-based approaches to combine the simulators in the two domains together by designing specific data exchange interface and synchronization policies. Such software-based co-simulators mainly distinguish from each other by the type of power/communication system simulators used, the scheme of data exchange between the two domains, and the application scenarios, as discussed in Chapter 6.

On the other hand, the AC/DC grid with integrated renewable energies proposes challenges to the system-level control for stability, security and fault recovery [108, 109]. The power generated by renewable generators such as offshore wind farms is transmitted to the AC grid through the high voltage DC (HVDC) transmission, which should be controlled dynamically to match the generation and consumption. The existing AD/DC control schemes mostly focused on the control algorithm for a single modular multilevel converter (MMC) [110, 111], or the control strategy in a small scale system without communication networks involved [112–114]. However, the local measurement based control is not optimized in the system-scale and easily delayed without a global view [115]. The cyber layer creates the required capabilities to perform the global measurement and control, and ICT-enabled power electronic converters that play important roles for power flow control can also receive the control command from the system-level controller to change the corresponding parameters. Some utilities/consultants have used real time simulations with actually communication systems implemented in their labs in planning stages for complex control schemes for remedial action scheme for stability enhancing control. This type of control for the large scale AC/DC grid with integrated renewable energies should also be evaluated in terms of system stability, but few works have been concentrated on this topic and no integrated simulation platforms that contain both the power system and communication network simulation have been developed.

Based on the above observations, this work proposes an heterogeneous real-time EMT-based power and communication system co-emulator realized on multiple Jetson® and FPGA platforms for global control of hybrid AC/DC networks. The NVIDIA® Jetson AGX

Xavier embedded platform [116] is an artificial intelligence (AI) computer for autonomous machines, delivering the performance of a 512-core GPU workstation within an embedded module, which can perform heavy computation tasks. Taking advantage of the PCI express (PCIe) interface based connection between Jetson® and the FPGA, a complete hardware solution is developed to provide all the required capabilities for cyber-physical system co-emulation. The contributions of this work are summarized as follows:

1. The EMT-based power system emulation is executed on FPGA boards with a microsecond level time-step, which can provide a nearly continuous measurement to the AC and DC sampling unit; the MMC converter control functions and communication modules run on the Jetson® embedded platform, which perform fast and flexible computation for complex tasks;

2. The interaction between the two domains are emulated in a realistic way: power system measurement data sampling and converter controller sampling operation is achieved via the read operation from the FPGA memory, and the control and protection commands are pushed to the power system emulation via the write operation to the FPGA memory;

3. Based on the proposed heterogeneous co-emulation architecture, the global control schemes are studied on the AC/DC grid integrated with wind farms. By utilizing the communication network to perform global control strategies with fast responses, the influence of power overflow and DC fault can be reduced efficiently.

7.2 ICT-Enabled Hybrid AC/DC Grid

This section introduces the AC/DC test power system with integrated renewable energies, the corresponding communication network, and the ICT-enabled power equipment.

7.2.1 Hybrid AC/DC Power System

To investigate the global control and protection of modern power systems, a hybrid AC/DC grid composed of onshore and offshore generation, AC and DC transmission, AC/DC and DC-DC converters as well as renewable energy sources is required. In this work, the AC/DC grid composed of an onshore IEEE 39-bus system [117], a subsystem of the CIGRÉ B4 DC test grid [117] and two offshore wind farms is chosen as the test system, which can emulate a modern AC/DC power system practically. The CIGRÉ DC grid consists of three interconnected DC systems (DCSs) called DCS1, DCS2, and DCS3, and the bipole HVDC meshed grid DCS3 with ±200kV is selected for DC test system, as shown in Fig. 7.1. The MMC converter topology is applied to the AC/DC converters. The offshore wind farms are responsible to power generation, and the generated power is transmitted to the AC grid through the AC/DC and DC/AC conversion. Such a complex AC/DC power system not only creates challenges for system-level control and protection under different contingencies, but also makes the real-time co-emulation of EMT-based power system and communication network quite challenging.

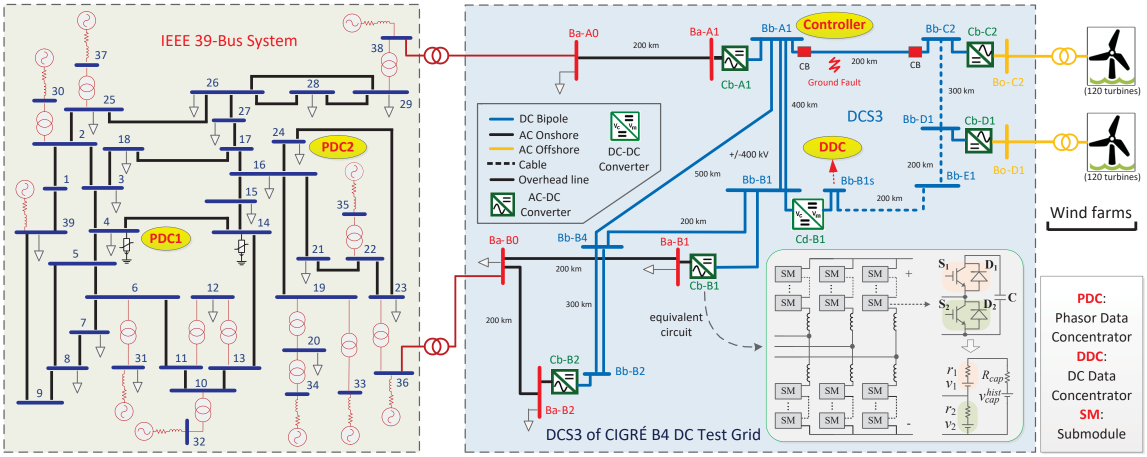


Figure 7.1: Hybrid AC/DC test power system used in this work.

7.2.2 Communication Network

With the cyber layer involved, the real-time status of the power system can be measured and gathered, and then the global view based control strategies can be performed. For example, in each bus or substation, the merging unit (MU) is deployed to gather the signals from voltage and current sources at a synchronized sampling rate. Although the communication links are not revealed in Fig. 7.1, in this work, each AC bus is assumed to have a phasor measurement (PMU) [118] installed, and each DC bus is assumed to have a DC measuring unit (DC-MU) installed, which compose the basic communication element in the cyber layer.

The PMU at AC buses can use the digital samples obtained by MU to compute the phasor values and periodically report to the control system, where a phasor is a complex quantity to present the sinusoidal wave of an electrical signal. For example, a sinusoidal signal is given as:

$$x(t) = X_m \cos(\omega t + \varphi), \tag{7.1}$$

where X_m is the magnitude of the waveform, and φ is the angular starting point. Then the corresponding phasor is expressed using the RMS value:

$$X = \frac{X_m}{\sqrt{2}} \angle \varphi = \frac{X_m}{\sqrt{2}} (\cos \varphi + j \sin \varphi) = X_r + jX_i \tag{7.2}$$

The rule of PMU is to estimate the magnitude and angle of a signal according to the sampling measurements. For DC buses, the DC voltage, current, and power are measured by DC-MU since there is no phasor data.

The control system is built on the cyber layer, in which the local controllers and centralized system-level control center are connected via specific control network architectures such as the tree-based topology and mesh topology. The local controller can be a phasor data concentrator (PDC) that collects AC phasor measurements from PMUs in its region, or a DC data concentrator (DDC) that collects DC measurements from DC-MUs. The measurement collection operation of PDC is performed using a specific reporting rate [118]. In fact, the main function of the communication network is to provide a two-way communication between smart meters and controllers to modify the related circuit parameters according to the service requirements. In this work, two PDCs (PDC1 and PDC2) are deployed at AC Bus4 and Bus24, and one DDC is deployed at DC bus Bb-B1s, as shown in Fig. 7.1. The system-level controller is deployed at Bb-A1.

7.2.3 ICT-Enabled Power System Equipment

In the conventional power system, the power equipment is generally controlled by electrical quantities locally. In cyber-physical power systems, the ICT-enabled devices can receive control commands from remote controllers to change the corresponding parameters. By utilizing the ICT-enabled controllable power equipment, flexible strategies and fast response to faults can be realized. Typically, the AC/DC circuit breakers (CBs) are ICT-enabled for protection. For example, when a DC line ground fault is detected by a DC breaker, the line could be isolated by the DC breaker immediately; but the corresponding fault information is required to be sent to the centralized controller for subsequent stability control. The communication functions are also increasingly deployed in generator and controllable load for flexible power supply and load balancing. For example, if the power generated from a machine exceeds the consumption, the corresponding control command from the control center can guide the machine to reduce power generation or just close the supply for a period. Another important power system equipment is the converter in DC grid, which is typically controlled by the outer/inner loop control and modulation schemes for DC voltage or power regulations. However, with ICT involved, the power electronics converter can also receive the control command from remote controllers to change the values of regulated quantities according to system-level purpose [119].

7.3 Heterogenous Real-Time Co-Emulation Architecture on Multiple Jetson®-FPGA Platform

Based on the above introduced aspects in cyber-physical power systems, this section describes the proposed real-time co-emulation architecture on the Jetson®-FPGA embedded platform.

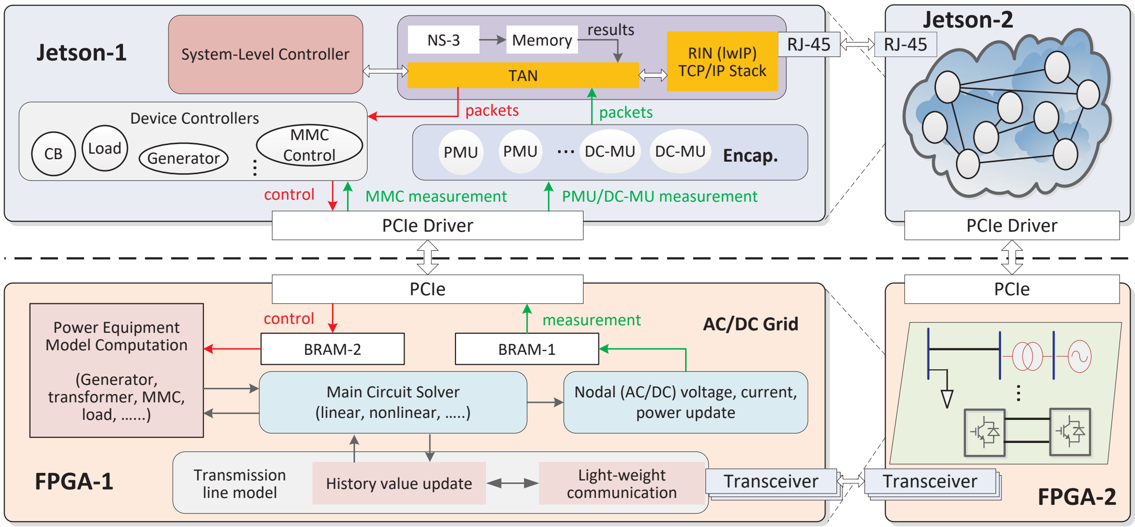


Figure 7.2: Top level architecture of the heterogeneous co-emulation hardware architecture on the multiple Jetson®-FPGA platform.

7.3.1 Co-Emulation Architecture

The top-level architecture of the proposed co-emulation architecture is shown in Fig. 7.2, wherein the power system EMT emulation and communication network simulation are separated to the FPGA and Jetson® respectively. Since the two FPGA and Jetson® embedded platforms have the same processing architecture, only one detailed architecture of the Jetson®-FPGA platform is shown in Fig. 7.2. For the interaction between the two domains, the PCIe bus is used to connect the FPGA and Jetson® platform. The two block RAMs (BRAM-1 and BRAM-2) are used to store the measurement data and control command respectively: after the calculation of each time-step (at microsecond level for EMT emulation), the nodal voltages, currents and other power quantities of interest are written to BRAM-1; while the BRAM-2 is used to receive the control command from the control center or the gate signals from the MMC control functions. The multi-board scheme is also be applied to extend the resources for large-scale system simulation.

The data exchange and synchronization between the two domains mainly refers to two operations: measurement data sampling, and control command provisioning. These two operations become quite convenient in the proposed co-emulation architecture: for the measurement data sampling operation, the simulated PMUs and DC-MUs in the Jetson® embedded platform can read the measurement data from BRAM-1 via the PCIe driver function at a configurable rate; for the control command delivery, the corresponding instructions can be written to BRAM-2 at any time to change the circuit parameters in the FPGA computation. These operations do not influence the normal simulation in the power system domain, which makes the real-time EMT emulation possible. Since the proposed heterogeneous co-emulator is targeting the real-time co-emulation of power system and communication network, the hybrid Jetson®-FPGA hardware platform is utilized, which may not be as flexible as the pure software-based co-simulator and more programming effort needs to be invested when the emulated test system changes.

7.3.2 Hybrid AC/DC Grid EMT Emulation

In this work, the EMT emulation is conducted for the power system, because it not only can capture the transient-level waveforms, but can also provide continuous signals due to the small time-step sizes so that the measured data to PMUs and DC-MUs is accurate enough. Typically, real-time emulation can be achieved on FPGA boards even for large-scale AC/DC grids [120]. However, in co-emulation, the extra operations are introduced: writing measurement data to BRAM-1 and reading control command from BRAM-2. The number of measured data is large if the electrical quantities of each bus are required, then it costs a considerable latency to write data to BRAM-1 since in each clock cycle only one data can be written. In this work, this operation is delayed for one time-step, which means, the results of the last time-step calculation are written to the BRAM-1 while the circuit is computed for the current time-step. Since the interval of sampling is larger than the time-step size of EMT emulation, the one time-step delay is acceptable. The same policy can be applied to the read operations to achieve real-time EMT emulation.

When the multi-board solution is adopted, the power system is decomposed into subsystems using the traveling-wave line model (TLM) or frequency-dependent line model (FDLM). Then the two ends of a transmission line can be computed in two FPGA boards concurrently and exchange their data after each time-step. This type of data exchange between the two FPGA boards should be executed via fast transmission protocol and transceivers to ensure the transmission delay is minimized for real-time emulation.

7.3.3 Communication Network Emulation

The communication network is built on communication devices with specific protocols, thus the task of communication system simulation is to simulate the behaviour of network forwarding devices and transmission links. In the related works, various network simulators such as NS-2/NS-3, OpenNet, OMNet++ have been utilized to conduct the network-domain simulation. However, the existing solutions are not applicable to the proposed co-emulation architecture, because: 1) using the created virtual network, new interfaces are required to be developed for synchronizing the measurement data, which is time-costly and make the real-time co-emulation infeasible; 2) for large-scale AC/DC test power systems, hundreds of nodes need to be instantiated if each node is regarded with a PMU installed, then each node needs to receive the corresponding sampling data and generate data packets, which greatly increases the simulation latency.

Different from the pure network simulation that aims to investigate the packet transmission or test new protocols, the essential goal of the co-emulation is to study the influence of the cyber layer on the physical layer, i.e., how the power system is improved under the support of communication network. Therefore, the end-to-end communication parameters such as the transmission latencies and packet losses are of the greatest concern. Based on this observation, in this work, the communication module is implemented in a hybrid manner as shown in Fig. 7.2: transmission abstraction based networking (TAN) and real network interface based networking (RIN). TAN on a Jetson® embedded platform is used for the networking within the corresponding power system area emulated on the connected FPGA board, which reads the results from the NS-3 simulator in advance and then uses the resulting transmission parameters to abstract the packet transmission. This simplification is to accelerate the co-emulation process to real-time, while the used transmission parameters obtained by the network simulator are also practical and reasonable. RIN is used for inter-board networking, which is executed on the real-world network interfaces of the Jetson® embedded platform to simulate the packet transmission between different power system areas. The measurement data packet generation function is also included in the architecture, which is used to simulate the behaviour of PMUs and DC-MUs that sample measurement data from power system and encapsulate the measurement into network packets to be sent to the PDC.

7.4 Hardware Implementation of Test System

The test system is implemented on the Jetson®-FPGA platform. Two Xilinx® VCU118 FPGA boards and two NVIDIA® Jetson embedded platforms are utilized.

7.4.1 Heterogeneous Co-Emulator Hardware Resources and Set-Up

The Xilinx® VCU118 evaluation board provides a hardware environment for developing and evaluating designs targeting the UltraScale+ XCVU9P-L2FLGA2104 device. The VCU118 evaluation board provides features common to many evaluation systems, including the dual small form-factor pluggable (QSFP+) connector and sixteen-lane PCI express interface. The 16-lane PCIe edge connector performs data transfers at the rate of 8.0 GT/s for Gen3 applications, and the XCVU9P-L2FLGA2104 device is deployed on the VCU118 to support up to Gen3 x8 channels. The two quad (4-channel) QSFP+ (28 Gb/s) connectors accept 28 Gb/s QSFP+ optical modules. Each connector is housed within a single 28 Gb/s QSFP+ cage assembly.

The NVIDIA® Jetson embedded platforms provide the performance and power efficiency to run autonomous machines software. Each Jetson® embedded platform is a complete System-on-Module (SOM), with CPU, GPU, PMIC, DRAM, and flash storage—saving development time and money. The 512-core Volta GPU with tensor cores, the 8-core ARM® v8.2 64-bit CPU, 32GB 256-bit memory, x8 PCIe interface and RJ45 gigabit ethernet interface enable the proposed heterogeneous co-emulation architecture to be completed implemented.

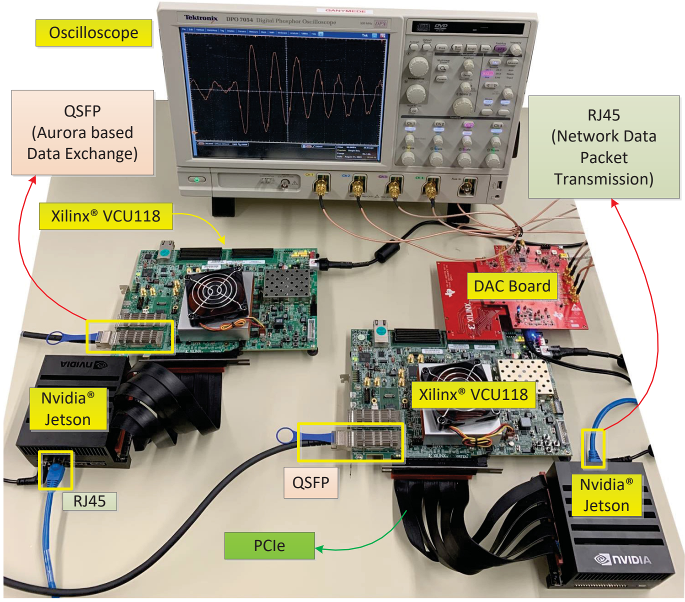


Figure 7.3: Hardware setup for the heterogeneous co-emulator.

The two Xilinx® FPGA boards run at the clock frequency of 100 MHz. The Ubuntu 18.04 Linux operating system runs on the NVIDIA® Jetson embedded platform at 2GHz frequency. According to the top level co-emulation architecture, the heterogeneous hardware platform set up is shown in Fig. 7.3. The digital-to-analog converter (DAC) adapter that connects the VCU118 board and oscilloscope is used to show the real-time waveforms; the two NVIDIA® Jetson embedded platforms are connected via the RJ-45 ethernet port for network data packet transmission, while each Jetson® embedded platform is connected with a VCU118 board via the PCIe cable; and the two VCU118 boards are connected via the QSFP+ transceivers for fast data exchange.

7.4.2 FPGA Implementation

The AC/DC grid is partitioned into two parts to be allocated to the two FPGA boards: IEEE 39-bus system and its connected AC buses in DCS-3 (Ba-A0, Ba-B0), and, the rest part of the grid. The applied AC equipment models are the same as those of PSCAD/EMTDC® [12] used for verification: the synchronous machine is modeled as a Norton current source, and since the current source representation uses the terminal voltages to calculate the injected currents, a characteristic impedance is used to terminate the machine to the network; the AC4A type exciter control model [12] is attached with the machine to provide a feedback for the field voltage; the transformer model uses a conductance matrix generated by the equivalent RLC circuit to compute the voltages of the coupled winding terminals given a known equivalent current injection; the Bergeron transmission line model is a traveling wave line model, which utilizes the transmission latency of a line to decouple the two connected subsystems and make the concurrent computation of the two line ends possible.

For the MMC converter model, the two FPGA boards do not have enough resources to compute all the detailed models, and thus a hybrid modeling scheme is applied: the AC-DC and DC-AC converters use the ideal switch based equivalent circuit model, while the DC-DC converter is modeled as an ideal transformer, which is the same as the applied model in the PSCAD/EMTDC® example case [121]. The windfarm generator model is also simplified as duplication of a single generation unit [122], since the complex wind farm model involving hundreds of generation units consumes significant computation resources and is not the object of interest in this work. In the ideal switch based equivalent MMC model, the IGBT and diode nonlinear switching transients are ignored while only the electrical model is presented. The Thévenin equivalent circuit for each submodule (SM) is represented using $r_{sm,eq}$ and $v_{sm,eq}$ in series [73] as shown in Fig. 7.1:

$$r_{sm,eq} = \frac{r_2(r_1 + R_{cap})}{r_1 + r_2 + R_{cap}} \tag{7.3}$$

$$v_{sm,eq}(t) = \frac{r_{sm}}{r_2} v_2 + \frac{r_{sm}}{r_1 + R_{cap}} (v_{cap}^{Hist}(t - \Delta t) - v_1) \tag{7.4}$$

where $R_{cap} = \frac{\Delta t}{2C}$ is the equivalent resistance of the capacitor, r_1 and r_2 are equivalent resistances of the two switches, and the values are equal to R_{ON} or R_{OFF} depending on the gate signals (1 or 0). The capacitor voltage $v_{cap}(t)$ can be derived using Trapezoidal rule:

$$v_{cap}(t) = R_{cap} i_{cap}(t) + v_{cap}^{Hist}(t - \Delta t) \tag{7.5}$$

where

$$v_{cap}^{Hist}(t - \Delta t) = R_{cap} i_{cap}(t - \Delta t) + v_{cap}(t - \Delta t) \tag{7.6}$$

After the main network equation composed of the equivalent circuit together with other electrical elements is solved, $i_{sm}(t)$ is known. Then the current through the capacitor of each SM is updated:

$$i_{cap}(t) = \frac{r_2 i_{sm}(t) + v_1 + v_2 - v_{cap}^{Hist}(t - \Delta t)}{r_1 + r_2 + R_{cap}} \tag{7.7}$$

Similarly, the Thévenin equivalence for one converter arm is the superposition of equivalent resistance and voltage of all SMs in the arm. In this work, the 51-level MMC converter structure is applied.

The PCIe connected memory BRAM-1 and BRAM2 are implemented as Xilinx® IP cores with AXI-4 interfaces. BRAM-1 with size of 32bit×1024 is used to store the measurement data, including the three phase nodal voltages and line currents, which is enough to for storing measurements of the subsystem in one FPGA board. For BRAM-2, the control command is mainly used for the converters and generators. The control command for generators instructs if the turbines of the wind farms should power off or not; while the control command for converters refers to the reference values of the controlled quantities and the gate signals for ideal switch based equivalent MMC model. Since a 32-bit control command can contain several switch signals, the size of 32bit×1024 is also adopted.

Table 7.1: FPGA Hardware Resource Consumption of the Test System

Board	LUT	FF	BRAM	DSP	Latency
VCU118-1	90.3%	95.2%	57.1%	91.2%	15.7μs
VCU118-2	93.1%	89.1%	50.3%	96.9%	18.6μs

The fast data exchange of the transmission line history items between the two FPGA boards is achieved by using the simple lightweight communication core, the Xilinx® Aurora core. The AXI4-stream user interface enables convenient connection between other modules. The experimental test shows that after about 60 clocks of link initialization, the 64-bit data can be transferred between two boards continuously, which means that the latency of transferring n 64-bit data is about $(n + 60)$ clocks.

The hardware resource consumption and maximum processing latency for one time-step computation are presented in Table 7.1. It can be observed that the resource costs of the two boards are relatively balanced to fully exploit the FPGA programmable resources. The processing latency of one time-step indicates the minimum time-step size that can be applied for real-time EMT emulation. Note that the BRAM read and write operations are one time-step delayed, which can be done when the other calculations are being performed. Therefore, the two boards can use the same time-step size of 20μs.

7.4.3 Jetson® Implementation

The MMC controller function is computed in the Jetson®, which is composed of the outer loop control, inner loop control, and the value-level control, as shown in Fig. 7.4. The outer loop control uses the terminal DC voltage or active power as the reference to generate control signals; the inner current loop control is to generate the modulation signals, which uses the phase-disposition sinusoidal pulse width modulation (PD-SPWM) method [123]; the value-level control is used to generate gate signals for each switch in MMC. In this work, the MMC converters (Cb-A1, Cb-B1, and Cb-B2) connecting to the on-shore 39-bus system are utilized for DC voltage regulation, while the other two converters control the active power flow. This configuration can also be changed during the emulation according to system-level operations, since the controlled quantities and reference values are configurable by the system-level controller, as shown in Fig. 7.4. These MMC controllers of different MMC converters are computed in the GPU cores of the Jetson® embedded platform to fully leverage the parallel capabilities of the heterogeneous platform. The MMC controller measurement sampling rate is set as same as the PMU and DC-MU sampling rate for simplicity: every 60μs the values are sampled as shown in Fig. 7.5, which is about

278 samples per cycle (16.67kHz) for the 60Hz power system.

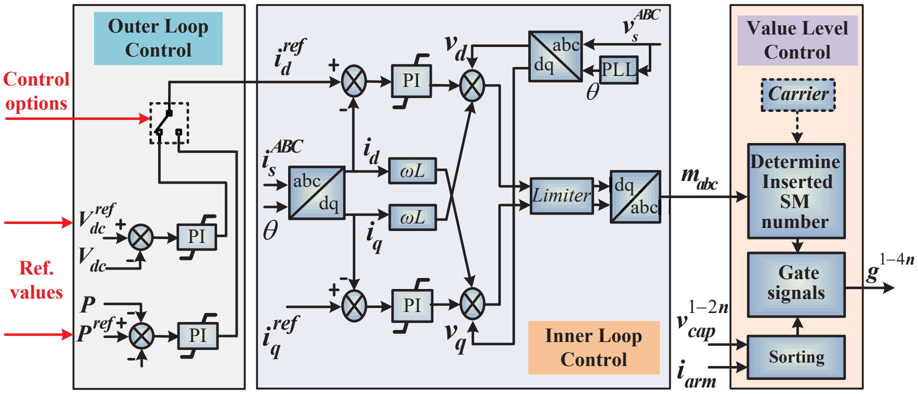


Figure 7.4: The ICT-enabled PD-SPWM MMC control scheme.

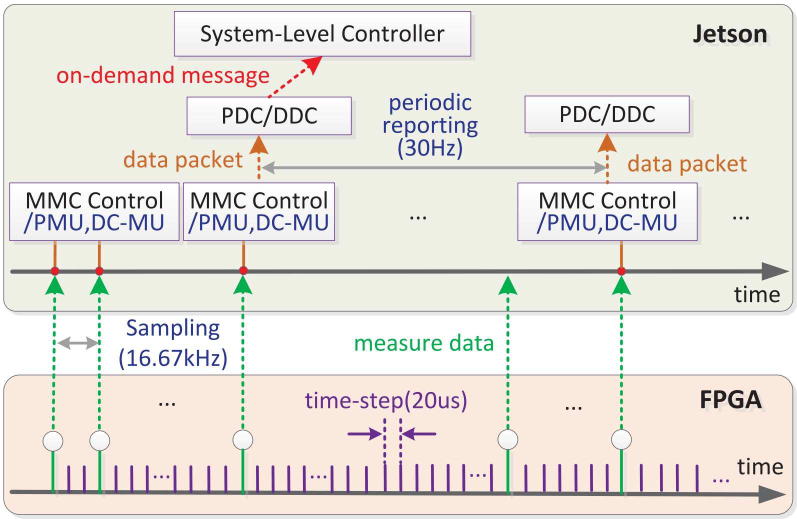


Figure 7.5: Demonstration of interaction between power system emulation and communication network simulation.

The packet generation module uses the PCIe driver functions to read the measurement data in BRAM-1 of the FPGA at the sampling rate. Then the corresponding phasors are computed using the measurement voltage and current, and the phasor data are encapsulated as the network packets and sent to the PDC functions with a reporting rate of 30Hz. This process is to simulate the measurement operation of PMUs and DC-MUs. The IEEE Std. C37.118.2 [90] primarily describes the presentation of synchrophasor data packet in a bit-mapped format. The standardized communication protocol, IEC 61850 - Communication Networks and Systems in Substations [103], also describes a similar approach to presentation and has been mainly used in the communication between protection relays and control systems. In this work, the bit-map of the packet follows IEC 61850. The PDC/DDC function checks the received phasor data or DC data to see if there are abnormal conditions; and if abnormal conditions are detected, the corresponding messages are sent to the center system-level controller in an “on-demand” pattern, as shown in Fig. 7.5.

In the transmission abstraction based networking (TAN), the transmission process between PMUs and PDCs and between PDCs and the system-level controller is handled based on the resulting parameters: end-to-end communication delay and loss rate. For example, if the PMU at Bus 7 generates a data packet to be sent to the PDC at Bus 4, then the data packet is generated in packet generation module and then is sent to the PDC function by passing the TAN function. In this implementation, the values chosen for the delay and loss are determined based on the testing results from the real network simulator NS-3 [105]. The corresponding capabilities of the network devices and links are configurable in the NS-3 simulator. In fact, there are two essential parameters in the NS-3 setup that may affect the end-to-end transmission delay: the transmission delay of each link, and, the forwarding rate of each device. In this work, the forwarding rate of each device is set at 100Mbps, which is a common value for devices in cyber-physical systems; and the transmission delay is set at 1ms/200km, which is also a practical value for the signal transmission. Then the link delays of different transmission lines with different length are computed accordingly.

In the real network interface based networking (RIN), the light-weight IP (lwIP) stack [106] is installed. Through the lwIP application programming interface, users can add customized networking functions. In this work, the IEC 61850 packet format and data encapsulation are implemented based on the existing echo server code to achieve the specific communication patterns of cyber-physical systems. Since the TAN module uses the resulting end-to-end parameters output by NS-3 and the measurement sampling is also performed using the system timer, the network simulation actually runs in real-time. By coordinating with the power system emulation, the real-time co-emulation can be achieved on the heterogeneous Jetson®-FPGA platform.

7.5 Real-Time Hardware Emulation Results for Communication-Enabled Global Control

Based on the implementation of the proposed hardware co-emulator, the global control schemes for the AC/DC test power system is studied and emulated for the two cases: power overflow and DC fault protection. Since the existing co-simulators are not fully open-sourced and are implemented on disparate platforms, it is extremely difficult to evaluate the test power system using the existing co-simulators for comparison. However, the emulation results and the proposed ICT-enabled global control scheme are still reasonable due to the standard test power systems, commonly used power equipment EMT models, practical fault cases, and realistic control strategies.

7.5.1 Case Study 1: Power Overflow Protection

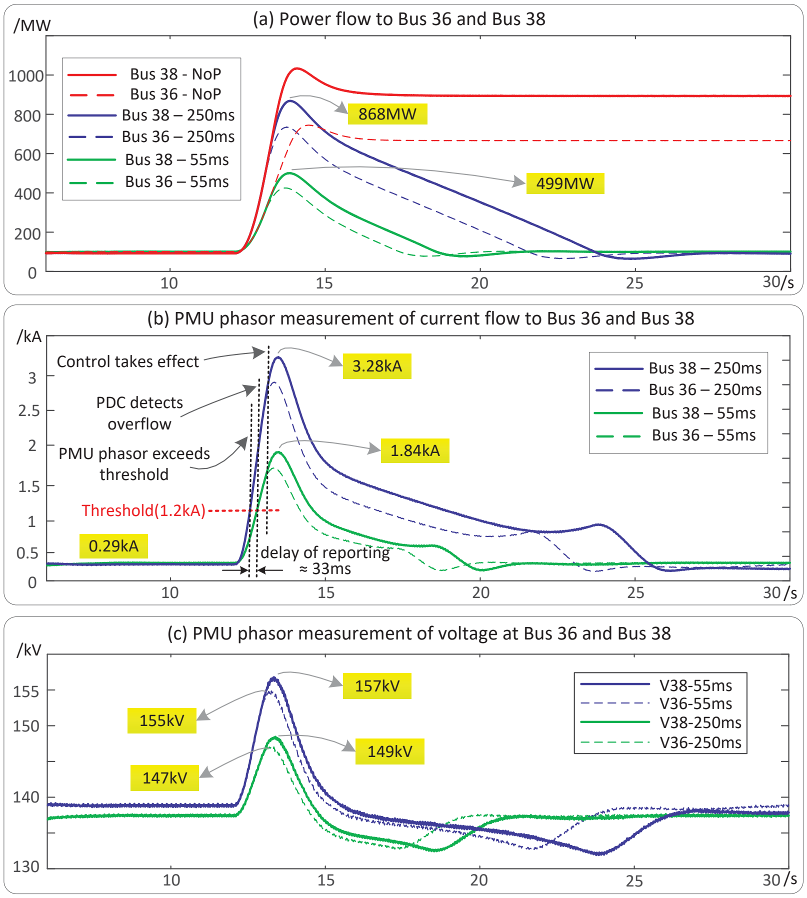


Figure 7.6: Power overflow case: a) comparison of the power flowing to Bus36 and Bus38 with 55ms, 250ms response delay and without protection; b-c) comparison of phasor magnitude of the current flowing to Bus36 and Bus38 and bus voltages with 55ms and 250ms response delay.

One major concern of the hybrid AC/DC grid integrated with renewable energies is to control the generated power by the offshore wind farms to precisely match the onshore consumption. When the power generated by the wind farms far exceeds the consumption of the AC side, the extra power can easily cause grid congestion, damage to the power equipment, and even blackouts. With the help of communication networks, the power flow at each bus can be monitored in real-time and the system-level control strategies can be generated to respond to the abnormal conditions quickly. In this case, the original power generated by each wind farm is 100MW, and total 200MW power flows to the Bus38 and Bus36 of the AC grid. At simulation time of 12s, the power generated by two wind farms starts to increase to 900MW and 700MW respectively, which causes the power overflow on the AC side. After the PMU measurement on the Bus38 and Bus36 are collected by PDC2 and sent to the system-level controller located on DC Bus Bb-A1, the corresponding control commands that instruct the wind farms to decrease the generation are sent to the two wind farms; and the commands that change the reference values of power flows are sent to the converter Cb-C2 and Cb-D1.

This process was emulated on the heterogeneous hardware co-emulator, and the results with 55ms and 250ms response delay and without protection (NoP) are shown in Fig. 7.6. Note that the response delay is between the time when the PMU phasor magnitude exceeded the threshold and the time when the corresponding power equipment received the control command. The 55ms delay was obtained by the real-time co-emulation, which included the reporting delay (30Hz, about 33ms), transmission delay (13ms), control command generation delay, and the control taking effect delay. The 250ms delay is chosen to be compared to the 55ms delay to show the influence of longer response, which can be regarded as the situation where a communication link fault happens and a larger response latency is generated due to the latency of data packet re-routing. The waveforms of power and current phasor magnitude in Fig. 7.6 are all from the view of PMUs and DC-MUs. It can be observed from Fig. 7.6(a) that without the system-level control, the power overflow could cause huge increase of the power to Bus38 and Bus36.

For the case with protection, the PDC2 detected the current flow to Bus38 and Bus36, and the threshold of phasor magnitude (RMS value) is set at 1.2kA, which is 4 times of that under the normal condition. After the current phasor magnitude of PMU38 at Bus38 exceeded the threshold, the PDC2 detected the abnormal condition after a reporting delay. In this case, since the 30Hz reporting rate is used, the reporting delay is at most 33ms. Then the PDC2 generated related data packets and sent to the controller via the communication network. The total communication delay included the delay from PMU38 to PDC2, the delay from PDC2 to controller, and from controller to windfarm and converter Cb-C2/Cb-D1, which was about 13ms resulting from the NS-3 simulator. The system-level controller held a global view of the system topology and made a decision that shut down some turbines of the wind farms to reduce the generated power to 100MW and reduces the reference values of the converters. From Fig. 7.6(b)(c) it can be observed that when the power generated by the wind farms increased quickly, the impact of difference of response delay was amplified on the resulting current and voltage. Therefore, a small communication delay can benefit the global stability. Besides, it can be also seen that after the control command took effect, there was still a short increase of the current, because it took time to release the power stored in the capacitors of MMC submodules.

7.5.2 Case Study 2: DC Fault Protection

The DC line ground fault is an important type of faults that can cause inreversible damages to the converters and should be protected against prior to the commissioning of HVDC grid. DC circuit breakers can be installed on the two ends of the DC lines, and they can isolate the fault when it is detected. The fast transient signal detection scheme [124] can be

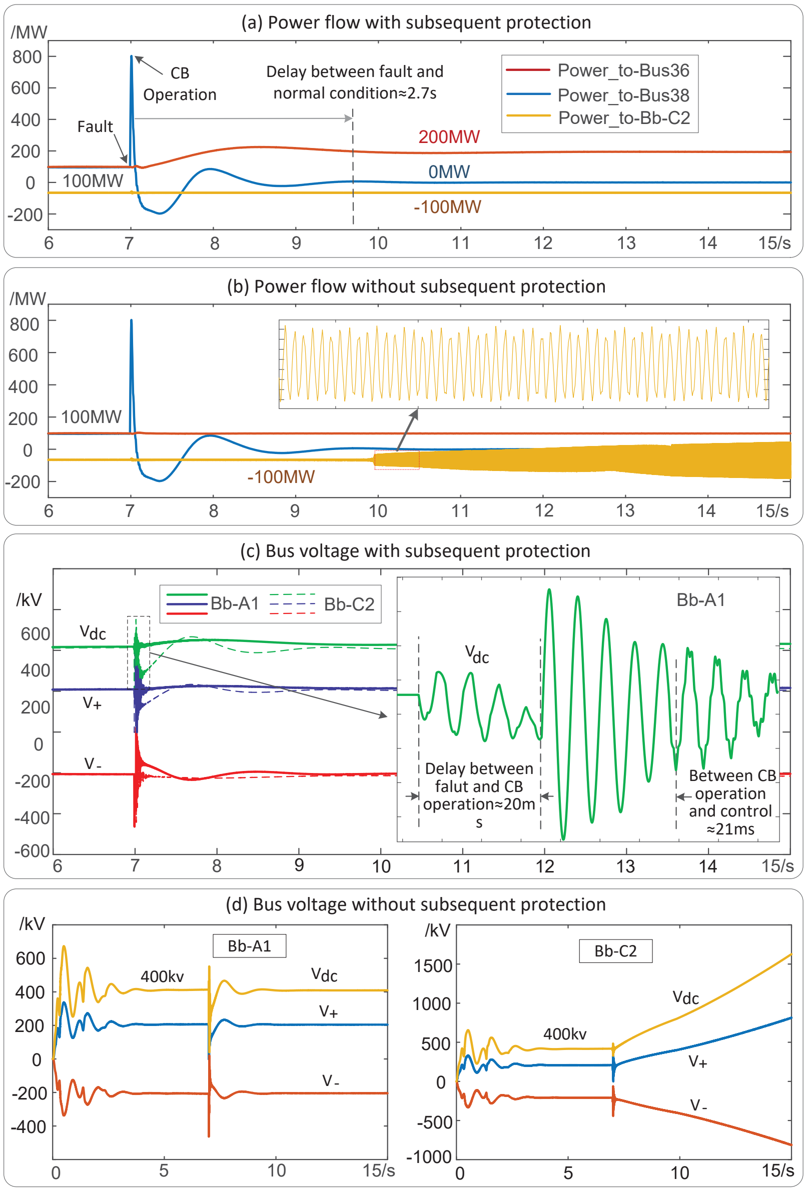


Figure 7.7: DC fault protection case: (a-b) power flowing to different buses with and without subsequent protection; (c-d) positive, negative pole voltage and DC voltage of converter Bb-A1 and Bb-C2 with and without subsequent protection.

applied in this process. However, after a circuit breaker cuts the corresponding DC line, the original power flowing along the DC line requires to be redirected to reduce the impact on the other DC power equipment. In this case, the DC line ground fault occurred on the line connecting the bus Bb-A1 and Bb-C2; then the DC circuit breakers at the two ends turned off to cut the line down when detecting the fault (20ms detection and action delay is assumed). After the line was isolated, the corresponding fault messages were sent to the controller, and the control command to change the reference power of the converters was generated to redirect the power flow. To demonstrate the effect of power flow redirection, the line between bus Bb-C2 and Bb-D1 was open at the beginning, and it was closed when the controller sent command to bus Bb-C2 and Bb-D1 after the DC fault.

The process was emulated on the hardware co-emulator and the results are presented in Fig. 7.7. It can be observed from Fig. 7.7(a)(b) that with system-level control, the original 100MW power flowing on the DC line between Bb-A1 and Bb-C2 was redirected to DC bus Bb-D1, and thus the power flowing to Bus36 increased to 200MW; although it took about 2.7s to return to the normal condition. Without the system-level control, the extra power could cause instability of the converter Cb-C2. From the DC voltages shown in Fig. 7.7(c), it can be seen that after the CB operation, there is a 21ms of transmission and control computation delay before the control command took effect. The power redirection did not cause a big perturbation to the V_{dc}, which indicates the global control strategy is reasonable to reduce the system instability compared to purely cutting off the line without subsequent protection operations.

7.6 Summary

To study the ICT-enabled global control for AC/DC grids integrated with renewable energy, the co-simulation platform is required to evaluate the interaction between the power system and communication network. In this chapter, a heterogeneous hardware real-time co-emulator is proposed and built on the multiple Jetson®-FPGA platform. The EMT-based power system simulation is executed on FPGA boards to provide a continuous measurement to the phasor measurement unit (PMU) and DC merging unit (DC-MU); the MMC converter control functions and communication modules run on the Jetson® embedded platform to perform fast and flexible computational tasks; and the interaction between the two domains are simulated through the read and write operations via the PCIe connector. The multi-board scheme is applied to extend the computational capabilities and resources to accommodate large-scale systems. The hybrid AC/DC grid with wind farms is emulated on the FPGA-Jeston based co-emulator, and the ICT-enabled global control schemes are investigated for the two case studies: power overflow and DC fault. The proposed heterogeneous co-emulator can be applied in fast emulation of practical large-scale cyber-physical power systems, and the investigated global control schemes can be applied to improve the system stability and controllability.

8
Conclusions and Future Work

With the development of AC power equipment, AC/DC power electronic converters, ICT-based smart meters, and controllable devices in modern power systems, fast and parallel hardware based simulation techniques are increasingly needed to deal with the large-scale complex power systems. The challenges in the EMT simulation area come from both computational methods and implementation architectures: in the computation level, new power equipment models are to be derived and applied for higher accuracy, and new computing methods are to be studied for faster matrix equation solutions; in the implementation level, the advancements in the integrated circuit (IC) technology and computing science should be leveraged to accommodate large systems and accelerate the simulation process.

In this thesis, three computational methods (multi-rate scheme, variable time-step-ping scheme and domain decomposition) and hardware (FPGA, MPSoC and GPU) based implementation architectures for fast and parallel power system EMT simulation are studied and improved. For the multi-rate scheme, an extended multi-rate mixed-solver (MRMS) hardware architecture is first proposed for real-time EMT simulation of hybrid AC/DC networks. For the variable time-stepping scheme, the new mathematical computational processes of the universal line model (ULM) and universal machine (UM) model are proposed to improve the stability. For the domain decomposition, a novel linking-domain extraction (LDE) decomposition method is proposed to solve the matrix equation in parallel. The hierarchical LDE method is further proposed to fully improve the LDE method. For the hardware based co-simulation architecture, the novel real-time co-emulation frameworks of the EMT-based power system and communication networks are proposed and conducted on the FPGA-MPSoC platform and Jetson®-FPGA platform respectively to accelerate the co-simulation for cyber-physical power systems.

The proposed computational methods and implementation architectures can be applied in different aspects of the current EMT simulation research, and also benefit the development of the software or hardware based simulation tools. However, more work is required to be conducted in the future to follow the path of the completed works and extend the corresponding research. The specific and detailed contributions of this thesis, corresponding applications, and the directions of future work are described in this chapter.

8.1 Contributions of This Thesis

The main contributions of this thesis are summarized as follows:

- **Computation-Level Contributions**
  (1) *Variable Time-Stepping Universal Line and Machine Models*. The proposed "process-reverse" computational procedure and equivalent circuit for the ULM model and exciter of UM model can perform a stable computation no matter how the time-step changes, which greatly improve the stability of the traditional ULM and UM models for VTS simulation.

  (2) *Linking-Domain Extraction (LDE) Based Domain Decomposition Method*. The proposed LDE method can find the general formula of the matrix inversion to compute the matrix inverse in parallel, which is a new mathematical approach. The LDE method can not only be used in computing matrix inversion in parallel, but can also used in solving matrix equations after decomposing the system into subsystems.

  (3) *Hierarchical LDE (H-LDE) Decomposition Method*. The proposed H-LDE method eliminates the necessity of computing the entire conductance matrix and can further decompose the decomposed subsystems into sub-subsystems to accelerate the process of inverting the decomposed block matrices, which can achieve much lower storage costs and computation latencies than the original LDE method.

- **Implementation-Level Contributions**
  (1) *Real-Time Multi-Rate Mixed-Solver Emulation Architecture on FPGA-MPSoC Platform*. The multi-rate mixed-solver emulation architecture is proposed to simulate the AC/DC power systems in real-time. By integrating the ARM® based processor system and FPGA based parallel computing, the proposed emulation architecture can combine AC and DC power systems simulation together with hybrid linear and nonlinear solvers.

  (2) *Faster-than-Real-Time Emulation Architecture on FPGA and 4-Level Parallel Simulation Architecture on GPU for VTS Simulation*. The FPGA-based and GPU-based parallel simulation architectures are proposed for VTS simulation. Through elaborate configuration to the time-step sizes of different subsystems, the "faster-than-real-time"

mode is achieved on FPGA; using the dynamic parallelism features and hierarchical decomposition, the massively parallel VTS simulation is achieved on GPU.

(3) *Co-Emulation Hardware Architecture for Cyber-Physical Systems*. The proposed real-time co-emulation (RTCE) framework is the first FPGA/MPSoC based emulation architecture to accelerate the co-emulation process of the EMT based power system and communication networks. The Jetson®-FPGA based heterogeneous co-emulation platform is also proposed to practically study the communication enabled global control of hybrid AC/DC grids.

8.2 Applications of the Proposed Works

This section describes the potential applications of the proposed computation and implementation techniques:

- The proposed multi-rate mixed-solver emulation architecture can be applied in the EMT simulation of large-scale AC/DC networks: the system decomposition and time-step coordination, the multi-board solution, and collaboration between soft-processors and programmable logics can be utilized in various FPGA and embedded architectures.

- The proposed VTS based universal line and machine models can be applied in the VTS power system simulation combining with other existing power equipment models. The proposed "faster-than-real-time" simulation architecture on FPGA and 4-level hierarchically parallel simulation architecture on GPU can be applied in different platforms to conduct fast VTS-based EMT simulation.

- The proposed linking-domain decomposition method is a matrix-based decomposition method, which can be applied in cases where the latency-based decomposition methods cannot be used. The (hierarchical) LDE method can also be applied in transient stability or power flow simulation within a certain power system scale.

- The proposed real-time co-emulation framework can be applied in emulating the modern cyber-physical power systems on programmable hardware platforms such as FPGA and MPSoC, or the heterogeneous platforms such as the Jetson®-FPGA platform, which can benefit the design, planning, and testing of the system level control schemes or new communication techniques in cyber-physical power systems.

8.3 Directions for Future Work

The following topics are proposed for future work:

- In the multi-rate simulation, different subsystems use different time-step sizes. How to decompose a given power system into different subsystems automatically and how to determine proper time-step sizes for the decomposed subsystems to guarantee reasonable accuracies remain to be investigated for the complete application of the multi-rate scheme.

- Similar to the automatic time-step configuration for multi-rate simulation, the variable time-stepping simulation also faces the problem of assigning proper time-step sizes during the hardware based simulation. Although the LTE-based time-step control schemes have been studied, how to determine the candidate time-step set prior to the simulation remains to be investigated.

- The automation problems also exit in the proposed LDE/H-LDE method. Although there are already some topology partition algorithms available, they require to be integrated with the LDE computation procedure. Besides, when the LDE decomposition comes to multi-levels, finding the automatic multi-level partition is required to make the H-LDE method really usable to users.

- The LDE method not only can be used in EMT simulation, but can also be used in the transient stability and power flow simulation, which is the future application work. Besides, the application of the LDE method in finite element based power equipment simulation also needs to be further exploited.

- Since the proposed real-time co-emulators for the power system and communication network are implemented on hardware, the configuration for a test power system is not as flexible as the software-based co-simulators, thus more programming effort needs to be invested to provide an interface similar to what users use in practice.

- The conventional centralized power system structure is transforming into a more distributed and autonomous pattern, which proposes new scenarios and challenges in co-simulating the cyber-physical systems such as ICT-enhanced microgrids and power distribution systems. The corresponding co-simulation architectures require to be designed and implemented.

Bibliography

[1] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.,* vol. 22, no. 2, pp. 1235-1246, Apr. 2007.

[2] B. Lu, X. Wu, H. Figueroa and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls," *IEEE Trans. Ind. Electron.,* vol. 54, no. 2, pp. 919-931, Apr. 2007.

[3] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE Trans. Ind. Electron.,* vol. 57, no. 4, pp. 1146-1158, Apr. 2010.

[4] S. U. Grabić, *et al.*, "Ultralow latency HIL platform for rapid development of complex power electronics systems," *IEEE Trans. Power Electron.,* vol. 27, no. 11, pp. 4436-4444, Nov. 2012.

[5] F. N. Najm. *Circuit Simulation.* USA, Hoboken: John Wiley & Sons, 2010.

[6] K. Mets, J. A. Ojea, and C. Develder, "Combining power and communication network simulation for cost-effective smart grid analysis," *IEEE Commun. Surveys Tuts.,* vol. 16, no. 3, pp. 1771-1796, 2014.

[7] B. Wu, *High-Power Converters and AC Drives.* Hoboken, NJ, USA: John Wiley & Sons, 2006, pp. 3-13.

[8] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electron.,* vol. 58, no. 6, pp. 2547-2555, Jun. 2011.

[9] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gener. Transm. Distrib.,* vol. 7, no. 5, pp. 451-463, May 2013.

[10] C. Deml and P. Turkes, "Fast simulation technique for power electronic circuits with widely different time constants," *IEEE Trans. Ind. Appl.,* vol. 35, no. 3, pp. 657-662, May. 1999.

[11] J. J. Sanchez-Gasca *et al.*, "Extended-term dynamic simulation using variable time step integration," *IEEE Comput. Appl. Power,* vol. 6, no. 4, pp. 23-28, Oct. 1993.

[12] *EMTDC® User's Guide: A Comprehensive Resource for EMTDC,* Version 4.7, Manitoba HVDC Research Centre, Winnipeg, Manitoba, Canada, 2010.

[13] Alternative Transient Program (ATP), http://www.emtp.org

[14] EMTP Alliance®, http://emtp-software.com/Overview

[15] PSpice® User's Guide, Cadence Design Systems, Inc., 2000

[16] HSPICE® User Guide: Simulation and Analysis, Version B-2008.09, Synopsys, Inc., 2008

[17] Saber® User Guide, Version V-2004.06-SP1. Synopsys, Inc., 2004

[18] S. Hui, K. Fung, M. Zhang and C. Christopoulos, "Variable time step technique for transmission line modeling," *IEE Proceedings A - Science, Measurement and Technology,* vol. 140, no. 4, pp. 299-302, Jul. 1993.

[19] J. J. Sanchez-Gasca, R. D'Aquila, W. W. Price and J. J. Paserba, "Variable time step, implicit integration for extended-term power system dynamic simulation," *Proc. IEEE Power Ind. Comput. Appl. Conf.,* Salt Lake City, USA, pp. 183-189, May 1995.

[20] S. Henschel, A. I. Ibrahim and H. W. Dommel, "Transmission line model for variable step size simulation algorithms," *International Journal of Electrical Power & Energy Systems,* vol. 21, no. 3, pp. 191-198, Mar. 1999.

[21] A. Ramirez and R. Iravani, "Frequency-domain simulation of electromagnetic transients using variable sampling time-step," *IEEE Trans. Power Del.,* vol. 30, no. 6, pp. 2602-2604, Dec. 2015.

[22] F. Camara, A. C. S. Lima and K. Strunz, "Full-frequency dependent models for variable time-step simulations," *CIGRE Session Papers & Proceedings,* no. C4-302, pp. 1-11, 2018.

[23] N. Lin and V. Dinavahi, "Variable time-stepping modular multi-level converter model for fast and parallel transient simulation of multi-terminal DC grid," *IEEE Trans. Ind. Electron.,* vol. 66, no. 9, pp. 6661-6670, Sept. 2019.

[24] Z. Shen and V. Dinavahi, "Dynamic variable time-stepping schemes for real-time FPGA-based nonlinear electromagnetic transient emulation," *IEEE Trans. Ind. Electron.,* vol. 64, no. 5, pp. 4006-4016, Jan. 2017.

[25] F. N. Najm, *Circuit Simulation.* USA, Hoboken: John Wiley & Sons, pp. 241-252, 2010.

[26] A. Morched, B. Gustavsen and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables," *IEEE Trans. Power Del.,* vol. 14, no. 3, pp. 1032-1038, Jul. 1999.

[27] H. K. Lauw and W. Scott Meyer, “Universal machine modeling for the representation of rotating electric machinery in an electromagnetic transients program,” *IEEE Trans. Power Appl. Syst.,* vol. 101, no. 6, pp. 1342-1351, Jun. 1982.

[28] T. F. Chan and T. P. Mathew. *Domain decomposition algorithms.* Cambridge University Press, 1994.

[29] A. Toselli and O.B. Widlund. *Domain decomposition methods: algorithms and theory.* Springer-Verlag Berlin Heidelberg, 2005.

[30] S. Y. R. Hui, K. K. Fung, and C. Christopoulos, “Decoupled simulation of multi-stage power electronic systems using transmission-line links,” *in Proc. Rec. IEEE Power Electron. Specialists Conf.,* Toledo, Spain, 1992, pp. 1324-1330.

[31] S. Y. R. Hui and K. K. Fung, “Fast decoupled simulation of large power electronic systems using new two-port companion link models,” *IEEE Trans. Power Electron.,* vol. 12, no. 3, pp. 462-473, May 1997.

[32] J. E. Schutt-Aine, “Latency insertion method (LIM) for the fast transient simulation of large networks,” *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.,* vol. 48, no. 1, pp. 81-89, Jan. 2001.

[33] S. N. Lalgudi, M. Swaminathan, and Y. Kretchmer, “On-chip power-grid simulation using latency insertion method,” *IEEE Trans. Circuits Syst. I,* vol. 55, no. 3, pp. 914-931, Apr. 2008.

[34] P. Aristidou, D. Fabozzi and T.V. Cutsem, “Dynamic simulation of large-scale power systems using a parallel Schur complement-based decomposition method,” *IEEE Trans. Parallel Distrib. Syst.,* vol. 25, no. 10, pp. 2561-2570, Oct. 2014.

[35] P. Aristidou, S. Lebeau and T.V. Cutsem, “Power system dynamic simulations using a parallel two-level Schur complement decomposition,” *IEEE Trans. Power Syst.,* vol. 31, no. 5, pp. 3984-3995, Sept. 2016.

[36] M. Baran, R. Sreenath, and N. Mahajan, “Extending EMTP for simulating agent based distributed applications,” *IEEE Power Eng. Rev.,* pp. 52-54, Dec. 2002.

[37] J. Bergmann, C. Glomb, J. Goandtz, J. Heuer, R. Kuntschke, and M. Winter, “Scalability of smart grid protocols: Protocols and their simulative evaluation for massively distributed DERs,” in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. 2010 (SmartGrid-Comm’10),* Oct. 2010, pp. 131-136.

[38] T. Godfrey, S. Mullen, R. C. Dugan, C. Rodine, D. W. Griffith, and N. Golmie, “Modeling smart grid applications with co-simulation,” in *Proc. 1st IEEE Int. Conf. on Smart Grid Commun. 2010 (SmartGrid-Comm’10),* 2010.

[39] V. Liberatore and A. Al-Hammouri, “Smart grid communication and co-simulation,” in *Proc. IEEE Energytech 2011,* may 2011, pp. 1-5.

[40] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, “EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components,” *IEEE Trans. Power Syst.,* vol. 21, no. 2, pp. 548-558, May 2006.

[41] H. Lin, S. Veda, S. Shukla, L. Mili, and J. Thorp, “GECO: Global event-driven co-simulation framework for interconnected power system and communication networks,” *IEEE Trans. Smart Grid,* vol. 3, no. 3, pp. 1444-1456, Sep. 2012.

[42] H. Georg, S. C. Müller, C. Rehtanz, and C. Wietfeld, “Analyzing cyberphysical energy systems: The INSPIRE co-simulation of power and ICT systems using HLA,” *IEEE Trans. Ind. Informat.,* vol. 10, no. 4, pp. 2364-2373, Nov. 2014.

[43] S. C. Müller, et al., “Interfacing power system and ICT simulators: challenges, state-of-the-art, and case studies,” *IEEE Trans. Smart Grid,* vol. 9, no. 1, pp. 14-24, Jan. 2018.

[44] V. Jalili-Marandi, Z. Zhou and V. Dinavahi, “Large-scale transient stability simulation of electrical power systems on parallel GPUs,” *IEEE Trans. Parallel Distrib. Syst.,* vol. 23, no. 7, pp. 1255-1266, July 2012.

[45] Z. Zhou and V. Dinavahi, “Parallel massive-thread electromagnetic transient simulation on GPU,” *IEEE Trans. Power Del.,* vol. 29, no. 3, pp. 1045-1053, June 2014.

[46] Z. Zhou and V. Dinavahi, “Fine-grained network decomposition for massively parallel electromagnetic transient simulation of large power systems,” *IEEE Power Energy Technol. Syst. J.,* vol. 4, no. 3, pp. 54-64, Sept., 2014.

[47] S. Yan, Z. Y. Zhou and V. Dinavahi, “Large-scale nonlinear device-level power electronic circuit simulation on massively parallel graphics processing architectures,” *IEEE Trans. Power Electron.,* vol. 33, no. 6, pp. 4660-4678, June 2018.

[48] Y. Chen and V. Dinavahi, “FPGA-based real-time EMTP,” *IEEE Trans. Power Del.,* vol. 24, no. 2, pp. 892-902, Apr. 2009.

[49] Z. Shen and V. Dinavahi, “Design and implementation of real-time Mpsoc-FPGA-based electromagnetic transient emulator of CIGRÉ DC grid for HIL application,” *IEEE Power Energy Technol. Syst. J.,* vol. 5, no. 3, pp. 104-116, Sept. 2018.

[50] T. Duan, Z. Shen and V. Dinavahi, “Multi-rate mixed-solver for real-time nonlinear electromagnetic transient emulation of AC/DC networks on FPGA-MPSoC architecture,” *IEEE Power Energy Technol. Syst. J.,* vol. 6, no. 4, pp. 183-194, Dec. 2019.

[51] NVIDIA Corporation, *Dynamic parallelism in Cuda,* downloaded in June 2018.

[52] *VCU118 Evaluation Board User Guide (UG1224),* Xilinx Inc., San Jose, CA, USA, Oct. 2018.

[53] Vivado Design Suite User Guide, High-Level Synthesis, UG902 (v2015.1). Xilinx, Inc., 2015

[54] *ZCU102 Evaluation Board User Guide (UG1182),* Xilinx Inc., San Jose, CA, USA, Oct. 2018.

[55] L. O. Chua, "Efficient computer algorithms for piecewise linear analysis of resistive nonlinear networks," *IEEE Trans. Circuit Theory,* vol. 18, no. 1, pp. 73-85, Jan. 1971.

[56] H. W. Dommel, "Nonlinear and time-varying elements in digital simulation of electromagnetic transients," *IEEE Trans. Power App. Syst.,* vol. 90, no. 6, pp. 2561-2567, Nov. 1971.

[57] *PSCAD^{TM} IEEE 39 Bus System,* Revision 1, Manitoba HVDC Research Centre, Winnipeg, Manitoba, Canada, 2010.

[58] *Aurora 64B/66B LogiCORE IP Product Guide, PG074 (V11.2),* Xilinx Inc., San Jose, CA, USA, Apr. 2018.

[59] Protection Against Lightning Electromagnetic Impulse - Part I : General Principles, IEC Standard 1312-I, Feb. 1995.

[60] N. R. Tavana and V. Dinavahi, "Real-time nonlinear magnetic equivalent circuit model of induction machine on FPGA for hardware-in-the-loop simulation," *IEEE Trans. Energy Convers.,* vol. 31, no. 2, pp. 520-530, June 2016.

[61] J. Liu and V. Dinavahi, "Detailed magnetic equivalent circuit based real-time nonlinear power transformer model on FPGA for electromagnetic transient studies," *IEEE Trans. Ind. Electron.,* vol. 63, no. 2, pp. 1191-1202, Feb. 2016.

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

[63] 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 Del.,* vol. 33, no. 3, pp. 1337-1347, June 2018.

[64] B. Gustavsen, G. Irwin, R. Mangelrod, D. Brandt and K. Kent, "Transmission line models for the simulation of interaction phenomena between parallel AC and DC overhead lines," *IPST'99 International Conference on Power Systems Transients,* Budapest Hungary, pp. 61-68, Jun. 1999.

[65] B. Gustavsen and A. Semlyen, "Simulation of transmission line transients using vector fitting and modal decomposition," *IEEE Trans. Power Del.,* vol. 13, no. 2, pp. 605-614, Apr. 1998.

[66] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.,* vol. 59, no. 2, pp. 1300-1309, Feb. 2012.

[67] L. Wang and J. Jatskevich, "A voltage-behind-reactance synchronous machine model for the EMTP-type solution," *IEEE Trans. Power Syst.,* vol. 21, no. 4, pp. 1539-1549, Nov. 2006.

[68] J. Liu and V. Dinavahi, "A real-time nonlinear hysteretic power transformer transient model on FPGA," *IEEE Trans. Ind. Electron.,* vol. 61, no. 7, pp. 3587-3597, Jul. 2014.

[69] *PSCAD^{TM} IEEE 118 Bus System,* Revision 1, Manitoba HVDC Research Centre, Winnipeg, Manitoba, Canada, 2010.

[70] M. A. Woodbury. *Inverting modified matrices.* Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.

[71] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gener. Transm. Distrib.,* vol. 7, no. 5, pp. 451-463, May 2013.

[72] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids," *IEEE Trans. Ind. Informat.,* vol. 10, no. 1, pp. 373-381, Feb. 2014.

[73] Z. Shen and V. Dinavahi, "Real-time device-level transient electrothermal model for modular multilevel converter on FPGA," *IEEE Trans. Power Electron.,* vol. 31, no. 9, pp. 6155-6168, Sept. 2016.

[74] Z. Huang and V. Dinavahi, "A fast and stable method for modeling generalized nonlinearities in power electronic circuit simulation and its real-time implementation," *IEEE Trans. Power Electron.,* vol. 34, no. 4, pp. 3124-3138, Apr. 2019.

[75] 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, Feb. 2018.

[76] J. J. Grainger, W. D. Stevenson, *Power System Analysis.* McGraw Hill Inc., 1994.

[77] K. Sun, Q. Zhou, K. Mohanram and D. C. Sorensen, "Parallel domain decomposition for simulation of large-scale power grids," *2007 IEEE/ACM International Conference on Computer-Aided Design,* no. 9789742, San Jose, CA, USA, Nov. 2007

[78] T. Duan and V. Dinavahi, "A novel linking-domain extraction decomposition method for parallel electromagnetic transient simulation of large-scale AC/DC networks," *IEEE Trans. Power Del.,* early-access, May 2020.

[79] T. A. Davis, and E. P. Natarajan, "Algorithm 907: KLU, a direct sparse solver for circuit simulation problems," *ACM Trans. Mathematical Software,* vol. 37, no. 6, pp. 36:1-36:17, 2010.

[80] X. Chen, Y. Wang, and H. Yang, "NICSLU: an adaptive sparse matrix solver for parallel circuit simulation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* vol. 32, no. 2, pp. 261-274, Feb 2013.

[81] EECS Department of the University of California at Berkeley, "SPICE: Simulation Program with Integrated Circuit Emphasis," http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/

[82] NVIDIA Tesla V100 Datasheet. Dec. 2017. [Online]. Available: http://www.nvidia.com/content/PDF/Volta-Datasheet.pdf

[83] S. Peyghami, P. Davari, M. Fotuhi-Firuzabad, and F. Blaabjerg, "Standard test systems for modern power system analysis: an overview," *IEEE Ind. Electron. Mag.,* vol. 13, no. 4, pp. 86-105, Dec. 2019.

[84] Y. Cao, X. Shi, Y. Li, Y. Tan, M. Shahidehpour and S. Shi, "A simplified co-simulation model for investigating impacts of cyber-contingency on power system operations," *IEEE Tran. Smart Grid,* vol. 9, no. 5, pp. 4893-4905, Sept. 2018.

[85] A. A. Jahromi, A. Kemmeugne, D. Kundur and A. Haddadi, "Cyber-physical attacks targeting communication-assisted protection schemes," *IEEE Tran. Power Syst.,* vol. 35, no. 1, pp. 440-450, Jan. 2020.

[86] Q. Wang, W. Tai, Y. Tang, Y. Liang, L. Huang and D. Wang, "Architecture and application of real-time co-simulation platform for cyber-physical power system," *in Proc. 7th IEEE Int. Conf. Cyber Technol. in Autom., Control and Intell. Syst.,* Hawaii, USA, pp. 81-85, July 31-August 4, 2017.

[87] G. Cao, W. Gu, C. Gu, W. Sheng, J. Pan, R. Li and L. Sun, "Real-time cyber-physical system co-simulation testbed for microgrids control," *IET Cyber-Phys. Syst., Theory Appl.,* Vol. 4, Iss. 1, pp. 38-45, Feb. 2019.

[88] B. Chen, K. L. Butler-Purry, A. Goulart and D. Kundur, "Implementing a real-time cyber-physical system test bed in RTDS and OPNET," *2014 North American Power Symposium (NAPS),* Pullman, USA, pp. 1-6, Sept. 2014.

[89] H. Tong, M. Ni, L. Zhao and M. Li, "Flexible hardware-in-the-loop testbed for cyber physical power system simulation," *IET Cyber-Phys. Syst., Theory Appl.,* Vol. 4, Iss. 4, pp. 374-381, Aug. 2019.

[90] "IEEE Standard for Synchrophasor Measurements for Power Systems," *IEEE Std C37.118.1,* IEEE Power & Energy Society, *New York, NY, 2011.*

[91] J. Bergmann, C. Glomb, J. Goandtz, J. Heuer, R. Kuntschke, and M. Winter, "Scalability of smart grid protocols: Protocols and their simulative evaluation for massively distributed DERs," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. 2010 (SmartGrid-Comm'10),* Oct. 2010, pp. 131-136.

[92] V. Liberatore and A. Al-Hammouri, "Smart grid communication and co-simulation," in *Proc. IEEE Energytech 2011,* may 2011, pp. 1-5.

[93] T. Godfrey, S. Mullen, R. C. Dugan, C. Rodine, D. W. Griffith, and N. Golmie, "Modeling smart grid applications with co-simulation," in *Proc. 1st IEEE Int. Conf. on Smart Grid Commun. 2010 (SmartGrid-Comm'10),* 2010.

[94] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Proc. IEEE/PES Innovative Smart Grid Technol. 2011 (ISGT'11),* Jan. 2011, pp. 1-6.

[95] F. Aalamifar, A. Schlogl, D. Harris, and L. Lampe. "Modelling power line communication using network simulator-3." http://www.ece.ubc.ca/ faribaa/paper.pdf.

[96] J. Nutaro, "Designing power system simulators for the smart grid: combining controls, communications, and electro-mechanical dynamics," in *Proc. IEEE Power and Energy Society General Meeting 2011 (PES'11),* 2011, pp. 1-5.

[97] K. Mets, T. Verschueren, C. Develder, T. L. Vandoor, and L. Vandevelde, "Integrated simulation of power and communication networks for smart grid applications," in *Proc. IEEE 16th Int. Workshop Computer Aided Modeling and Design of Commun. Links and Networks 2011 (CAMAD'11),* Kyoto, Japan, 10-11, June 2011.

[98] R. Mora et al., "Demand management communications architecture," in *Proc. 20th Int. Conf. Electricity Distribution (CIRED'09),* Prague, 2009, pp. 8-11.

[99] T. Sidhu and Y. Yin, "Modelling and simulation for performance evaluation of IEC61850-based substation communication systems," *IEEE Trans. Power Del.,* vol. 22, no. 3, pp. 1482-1489, July 2007.

[100] W. Li, A. Monti, M. Luo, and R. Dougal, "VPNET: A co-simulation framework for analyzing communication channel effects on power systems," in *Proc. IEEE Electric Ship Technologies Symp. 2011 (ESTS'11),* April 2011, pp. 143-149.

[101] K. Zhu, M. Chenine, and L. Nordstrom, “ICT architecture impact on wide area monitoring and control systems’ reliability,” *IEEE Trans. Power Del.,* vol. 26, no. 4, pp. 2801-2808, Oct. 2011.

[102] E. Moradi-Pari, N. Nasiriani, Y. P. Fallah, P. Famouri, S. Bossart and K. Dodrill, “Design, modeling, and simulation of on-demand communication mechanisms for cyber-physical energy systems,” *IEEE Trans. Ind. Informat.,* vol. 10, no. 4, pp. 2330-2339, Nov. 2014.

[103] “Communication networks and systems for power utility automation - Part 1: Introduction and overview,” *IEC TR 61850-1:2013,* 2013.

[104] “Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118,” *IEC TR 61850-90-5,* Jan. 2012.

[105] F. Aalamifar, A. Schlogl, D. Harris, and L. Lampe. “Modelling power line communication using network simulator-3.” http://www.ece.ubc.ca/faribaa/paper.pdf.

[106] Xilinx Doc. XAPP1026 (v5.1), “LightWeight IP Application Examples,” Nov. 21, 2014.

[107] R. Abe, H. Taoka, and D. McQuilkin, “Digital grid: communicative electrical grids of the future,” *IEEE Trans. Smart Grid,* vol. 2, no. 2, pp. 399-410, June 2011.

[108] M. Davari, and Y. A. I. Mohamed, “Dynamics and robust control of a grid-connected VSC in multiterminal DC grids considering the instantaneous power of DC- and AC-side filters and DC grid uncertainty,” *IEEE Trans. Power Electron.,* vol. 31, no. 3, pp. 1942-1958, Mar. 2016.

[109] X. Li, L. Guo, C. Hong, Y. Zhang, Y. W. Li, and C. Wang, “Hierarchical control of multiterminal DC grids for large-scale renewable energy integration,” *IEEE Trans. Sustain. Energy,* vol. 9, no. 3, pp. 1448-1457, July 2018.

[110] 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.

[111] X. Shi, B. Liu, Z. Wang, Y. Li, L. M. Tolbert, and F. Wang, “Modeling, control design, and analysis of a startup scheme for modular multilevel converters,” *IEEE Trans. Ind. Electron.,* vol. 62, no. 11, pp. 7009-7024, Nov. 2015.

[112] P. Wang, C. Jin, D. Zhu, Y. Tang, P. C. Loh, and F. H. Choo, “Distributed control for autonomous operation of a three-port AC/DC/DS hybrid microgrid,” *IEEE Trans. Ind. Electron.,* vol. 62, no. 2, pp. 1279-1290, Feb.. 2015.

[113] C. Qi, K. Wang, Y. Fu, G. Li, B. Han, R. Huang, and T. Pu, "A decentralized optimal operation of AC/DC hybrid distribution grids," *IEEE Trans. Smart Grid,* vol. 9, no. 6, pp. 6095-6105, Nov. 2018.

[114] 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, April 2019.

[115] J. Zhai, et al., "Hierarchical and robust scheduling approach for VSC-MTDC meshed AC/DC grid with high share of wind power," *IEEE Trans. Power Syst.,* early-access, pp. 1-12, July 2020.

[116] NVIDIA®, "NVIDIA Jetson AGX Xavier developer kit: powering AI in autonomous machines," Data Sheet, Oct. 2018.

[117] S. Peyghami, P. Davari, M. Fotuhi-Firuzabad, and F. Blaabjerg, "Standard test systems for modern power system analysis: an overview," *IEEE Ind. Electron. Mag.,* vol. 13, no. 4, pp. 86-105, Dec. 2019.

[118] M. Adamiak, B. Kasztenny, and W. Premerlani, "Synchrophasors: definition, measurement, and application," *presented at the 59th Annu. Georgia Tech Protective Relaying,* Atlanta, GA, Apr. 27-29, 2005.

[119] R. Burgos and J. Sun, "The future of control and communication: power electronics-enabled power grids," *IEEE Power Electron. Mag.,* vol. 7, no. 2, pp. 34-36, June 2020.

[120] Z. Shen, T. Duan and V. Dinavahi, "Design and implementation of real-time MPSoC-FPGA based electromagnetic transient emulator of CIGRÉ DC grid for HIL application," *IEEE Power Energy Technol. Syst. J.,* vol. 5, no. 3, pp. 104-116, Sept. 2018.

[121] PSCAD/EMTDC®, "CIGRE B4-57 working group developed models," Example Case, available at: https://www.pscad.com/knowledge-base/article/57

[122] Z. Shen and V. Dinavahi, "Comprehensive electromagnetic transient simulation of AC/DC grid with multiple converter topologies and hybrid modeling schemes," *IEEE Power Energy Technol. Syst. J.,* vol. 4, no. 3, pp. 40-50, Sept. 2017.

[123] G. Carrara, S. Gardella, M. Marchesoni, R. Salutari and G. Sciutto, "A new multilevel PWM method: a theoretical analysis," *IEEE Trans. Power Electron.,* vol. 7, no. 3, pp. 497-505, Jul. 1992.

[124] L. Liu, M. Popov, P. Palensky, and M. V. D. Meijden, "A fast protection of multi-terminal HVDC system based on transient signal detection," *IEEE Trans. Power Del.,* early-access, pp. 1-9, 2020. DOI: 10.1109/TPWRD.2020.2979811.