



<b>Titre:</b> Title:	Methods for FPGA-based real-time simulation of fast transients in power electronics systems and fault locating applications
Auteur: Author:	Hossein Chalangar
Date:	2021
Туре:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Chalangar, H. (2021). Methods for FPGA-based real-time simulation of fast transients in power electronics systems and fault locating applications [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. <u>https://publications.polymtl.ca/9473/</u>

# **Document en libre accès dans PolyPublie** Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/9473/
Directeurs de recherche: Advisors:	Keyhan Sheshyekani, Jean Mahseredjian, & Tarek Ould-Bachir
Programme: Program:	Génie électrique

# POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

# Methods for FPGA-based real-time simulation of fast transients in power electronics systems and fault locating applications

#### HOSSEIN CHALANGAR

Département de génie électrique

Thèse présentée en vue de l'obtention du diplôme de Philosophiæ Doctor

Génie électrique

Octobre 2021

© Hossein Chalangar, 2021.

### POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

# Methods for FPGA-based real-time simulation of fast transients in power electronics systems and fault locating applications

présentée par Hossein CHALANGAR

en vue de l'obtention du diplôme de *Philosophiæ Doctor* a été dûment acceptée par le jury d'examen constitué de :

Houshang KARIMI, président Keyhan SHESHYEKANI, membre et directeur de recherche Jean MAHSEREDJIAN, membre et codirecteur de recherche Tarek OULD-BACHIR, membre et codirecteur de recherche Ilhan KOCAR, membre Handy FORTIN-BLANCHETTE, membre externe

# DEDICATION

To my family

#### ACKNOWLEDGEMENTS

I would like to express my deep sense of gratitude to my supervisor Dr. Keyhan Sheshyekani for his supervision and support of this Ph.D. project. I would also like to extend my sincere thanks to Dr. Jean Mahseredjian for his trust and guidance during my research. Undoubtedly, special thanks go to Dr. Tarek Ould-Bachir, whose insightful guidance always helped me to follow the right direction during the course of this study. His deep knowledge and experience and enthusiastic encouragement which helped to shape this work definitely deserve my deepest gratitude.

I would also like to express my warm gratitude to all those at Polytechnique Montreal, the jury members, and president who have made this thesis possible: Dr. Ilhan Kocar, Dr. Handy Fortin-Blanchette, and Dr. Houshang Karimi.

I would also thank Andy Gallay whose expertise helped me to conduct experimental tests.

Hossein Chalangar, October 2021.

### **RÉSUMÉ**

La simulation en temps réel avec matériel dans la boucle est une méthode utilisée pour évaluer et vérifier les performances des systèmes de contrôle ou de protection des réseaux électriques durant les phases initiales de leur conception. Ce type de tests sont préférés aux tests de laboratoire conventionnels principalement pour les raisons suivantes: premièrement, le système ciblé peut être testé avant la fabrication des dispositifs ; deuxièmement, l'ensemble de l'installation, à l'exception du matériel testé, est simulée numériquement au même rythme que l'installation physique réelle. Les tests avec matériel dans la boucle permettent de réduire les coûts, augmentent la couverture de test et permettent de tester sans risque l'équipement dans des conditions extrêmes du réseau. Les tests avec matériel dans la boucle impliquent une plate-forme matérielle de traitement qui est responsable de la résolution des équations discrétisées dans le domaine temporel qui, en cas de transitoires rapides, doit exécuter la simulation avec un pas de temps inférieur à la microseconde. Pour atteindre ces petits pas de temps, les FPGA (Field-Programmable Gate Arrays) sont la plate-forme matérielle de traitement de choix, principalement en raison de son parallélisme élevé, de sa puissance de calcul et des entrées/sorties à faible latence qu'elle offre. Mais de nombreux défis se posent dans cette pratique que cette thèse tente de clairement identifier, caractériser et de régler.

Le premier thème traité par cette thèse est celui des convertisseurs à commutation rapide. Les progrès de la technologie des semi-conducteurs permettent d'augmenter l'efficacité et la densité de puissance des convertisseurs électroniques de puissance modernes en les faisant fonctionner à haute fréquence de commutation, c'est-à-dire jusqu'à plusieurs centaines de kHz. Cette tendance, cependant, pose certains défis à la simulation en temps réel basée sur FPGA, principalement attribués aux ressources mémoire limitées de ces dispositifs et à la nécessité d'effectuer des calculs coûteux dans un très petit pas de temps. Ce défi est d'autant plus critique lorsque le réseau simulé comporte plusieurs convertisseurs de puissance.

Cette thèse propose une nouvelle méthode de simulation des convertisseurs de puissance qui permet de simuler des cas de commutation rapide avec une grande précision et un petit pas de calcul. La méthode proposée construit une sorte de cartographie reliant les variables d'état du réseau et les combinaisons de commutateurs, de sorte à produire une solution non itérative exacte aux équations du réseau. Dans le cas de réseaux comportant de multiples convertisseurs de puissances, le procédé proposé a été combiné à des techniques de partitionnement du réseau afin d'en rendre l'utilisation réaliste. La solution a été implémentée sur FPGA et des pas de temps allant de 25 ns à 175 ns sont rapportés, selon la complexité des cas de tests considérés : un convertisseur LLC commuté à 500 kHz et un chargeur de batterie de véhicule électrique commuté à 200 kHz.

Le second thème considéré ici est celui des relais basés sur les ondes progressives. Ce type de relais utilise les transitoires pour localiser les défauts sur les lignes de transmission avec une grande précision. Afin d'effectuer un test avec matériel dans la boucle fiable de ces relais, un modèle du réseau simulé avec des pas inférieurs à la microseconde est nécessaire, notamment à cause d'une fréquence d'échantillonnage élevée des entrées du relais (1 MHz). De plus, le simulateur doit être en mesure de considérer les cas de localisation utilisant des mesures provenant d'un seul ou des deux terminaux de la ligne de transmission.

Cette thèse propose une étude détaillée de l'intérêt et des limites d'un modèle de ligne à paramètres constants pour le test des localisateurs de défaut par ondes progressives. Un banc de test robuste et fiable implémentant le modèle de ligne universel sur FPGA est proposé pour répondre aux limites identifiées dans le banc d'essais utilisant la ligne à paramètres constants. Il est démontré que le banc de test ainsi développé atteint un pas de temps inférieur à 200 ns et qu'il permet de tester les configurations à un et à deux terminaux.

#### ABSTRACT

Real-time simulation (RTS) and hardware-in-the-loop (HIL) testing are employed to evaluate and verify the performance of control and protection systems at the early stage of their design. The HIL testing is preferred over conventional laboratory tests mainly for the following reasons: first, the targeted system can be tested before the actual plant is built; second, the whole plant except for the hardware under test is digitally simulated at the same rate as the actual physical plant. Moreover, HIL test lowers costs, increases test coverage, and provides availability to test under extreme conditions of network in a de-risked environment. The HIL testing involves a hardware processing platform which is responsible for solving the discretized equations in the time domain and, in the presence of fast transients, it should run the simulation with a sub-microsecond time-step. To achieve such small time-steps, FPGAs (Field-Programmable Gate Arrays) are the hardware processing platform of choice, primarily due to their high parallelism, computing power, and low latency I/Os. However, many challenges arise in this practice and this thesis attempts to clearly identify, characterize, and address them.

The first subject dealt with in this thesis is the real-time simulation of fast switching converters. Advances in semiconductor technology allow the increase of the efficiency and power density of modern power electronic converters by enabling them to operate at high switching frequencies, i.e., up to several hundred kHz. This trend, however, poses some challenges to FPGA-based real-time simulation, primarily attributed to the limited memory resources of these devices and the need to perform heavy calculations in a very small time-step. This challenge becomes more convoluted when network under simulation includes multiple power converters, e.g., onboard power system.

This thesis proposes a new method of simulating power converters which enables accurate simulation of high switching frequency converters with small time-step. The proposed method proceeds by building a map between network state variables and switch status providing an exact non-iterative solution to network equations. For networks including multiple power converters, the proposed method is combined with network partitioning techniques to increase its scalability. The proposed method is implemented on FPGA and time steps ranging from 25 ns to 175 ns are reported, depending on the complexity of the test cases considered: an LLC converter switched at

500 kHz, an electric vehicle battery charger switched at 200 kHz, and a resonant boost circuit switch at 100 kHz.

The second topic considered in this thesis is the testing of traveling-wave (TW)-based fault locator relays. This type of relays uses transient waves to locate faults on transmission lines with very high accuracy. In order to conduct a reliable HIL test on these relays, simulated network should run with a sub-microsecond time-step, in particular because of a high sampling frequency of the inputs of the relay (1 MHz). In addition, the simulator must be able to provide conditions to test both single-ended and double-ended fault locating schemes.

The suitability of incorporating constant parameter line model is thoroughly investigated in this thesis. A reliable testbench based on universal line model is proposed. Its is demonstrated that the developed model reaches a time step of less than 200 ns and allows to assess both single- and double-ended schemes of the fault location.

### TABLE OF CONTENTS

DEDICATION	III
ACKNOWLEDGEMENTS	IV
RÉSUMÉ	V
ABSTRACT	VII
TABLE OF CONTENTS	IX
LIST OF TABLES	XIV
LIST OF FIGURES	XV
LIST OF SYMBOLS AND ABBREVIATIONS	XIX
LIST OF APPENDICES	XX
CHAPTER 1 INTRODUCTION	
1.1 Context and Motivations	1
1.1.1 Real-Time Simulation	2
1.1.2 Motivations	4
1.2 Objectives and Contributions	7
1.3 Thesis Outline	7
CHAPTER 2 STATE-OF-THE-ART OF THE REAL-TIME SIMULATION	9
2.1 Introduction	9
2.2 Switch Modeling	9
2.3 Network Solution Approaches	11
2.3.1 Circuit Partitioning	11
2.3.2 Decoupling Approaches	
2.3.3 Multi-Rate Simulation	13
2.4 Dealing with Switches	14

2.5 Real-Time Simulation of Transmission Line	16
2.5.1 TW-based Relay and Fault Locator	16
2.5.2 Existing TW-Based Relay and Fault Locator Test Methods	17
2.5.3 TW-Based Relay RT Test Setup	17
2.6 Conclusion	18
CHAPTER 3 DIRECT MAPPED METHOD	19
3.1 Introduction	19
3.2 Iterative Solution	19
3.3 Direct Mapped Method	22
3.3.1 Demonstration of DMM Concept Using the Single-Phase Rectifier Circuit	22
3.3.2 General Mathematical Formulation of the DMM	26
3.3.3 Building the DMM Function	27
3.4 Case Study: 500 kHz LLC DC-DC Resonant Converter	29
3.4.1 Overview of the LLC Resonant Converter	30
3.4.2 Simulation Algorithm	34
3.4.3 Low-latency Implementation of the DMM	36
3.5 FPGA Implementation of DMM-Based LLC Simulator	37
3.5.1 Hardware Implementation	37
3.5.2 Number Format	38
3.5.3 Design Space Exploration	39
3.5.4 Computational Accuracy	40
3.6 Resonant Boost Converter Case	43
3.7 Conclusion	46
CHAPTER 4 DIRECT MAPPED METHOD FOR MULTI-CONVERTER CIRCUIT	47

4.1	I	ntroduction	47
4.2	Ľ	Decoupling Techniques	47
4.2	.1	Brief Review of Methods from the Literature	47
4.2	.2	SDD	48
4.2	.3	NTT	49
4.3	N	Network Tearing Technique	49
4.3	.1	NTT Formulation	49
4.3	.2	An Illustrative Example for NTT	51
4.3	.3	Sub-circuit Network Equations	53
4.3	.4	NTT Simulation Flow	53
4.4	C	Combination of DMM with SDD and NTT	55
4.4	.1	DMM-SDD	55
4.4	.2	DMM-NTT	55
4.5	C	Case Study: EVBC	56
4.5	.1	EVBC Overview	56
4.5	.2	Modeling Alternatives for the RTS of the EVBC	58
4.5	.3	Offline Simulation Results	60
4.6	F	PGA Implementation	62
4.6	.1	Normalization	63
4.6	.2	Hardware Implementation	63
4.6	.3	DMM Function Implementation	65
4.6	.4	Comparison Between the FPGA Implementations of the Three Methods	66
4.7	R	Real-Time Test	67
4.8	C	Conclusion	69

CHAPTER 5 FPGA-BASED SIMULATION OF TRANSMISSION LINE	70
5.1 Introduction	70
5.2 Overview on TW-Based Fault Locating Methods	70
5.3 CP TL Model	72
5.4 CP TL for TW-Based Fault Locator Testing	74
5.4.1 Effect of the Ground Mode	75
5.5 CP-Based TWFL HIL Test	79
5.5.1 HIL TWFL Testbench Hardware	80
5.5.2 Power System Model	80
5.5.3 Fault Scenarios	80
5.6 Results	82
5.6.1 Observations	82
5.6.2 Analysis	83
5.6.3 Fault-Location Errors	84
5.7 FPGA-Based ULM Computing Unit	86
5.8 Universal Line Model	87
5.8.1 ULM Modelling Approach	87
5.8.2 ULM Time Domain Simulation	90
5.9 ULM Hardware Architecture	91
5.9.1 FPGA-Based ULM Solver	91
5.9.2 Low Latency ULM Computing Engine	91
5.9.3 Low-Latency Custom-Made Floating-Point Operators	93
5.10 Results	95
5.10.1 Test Case 1: Aerial Transmission Line	95

5.1	0.2 Test Case 2: TWFL test network	95
5.1	0.3 Area Occupation and Speed Performance	97
5.11	Conclusion	99
СНАРТ	TER 6 CONCLUSION AND RECOMMENDATIONS	
6.1	Summary of Thesis	100
6.2	List of Publications	101
6.3	Future Work	101
REFER	ENCES	
APPEN	DICES	

# LIST OF TABLES

Table 3.1 TCAM-RAM for the single-phase rectifier	
Table 3.2 Circuit parameters for LLC Circuits [21]	32
Table 3.3 Design exploration targeting the Kintex K325T [21]	40
Table 3.4 Comparison with the existing work on the FPGA-based real-time simulation of converter [21]	resonant 41
Table 3.5 LLC with parameter Set #2 [87]: 2-Norm Relative Errors	43
Table 3.6 FPGA (Kintex K325T) resource consumption of RBC implementation	45
Table 3.7 RBC circuit- FPGA computational accuracy	45
Table 4.1 EVBC parameters	57
Table 4.2 2-Norm relative errors (%) obtained from various simulation methods [77]	62
Table 4.3 FPGA resource consumption of ESU-SDD, DMM-SDD and DMM-NTT [77].	66
Table 4.4 EVBC circuit-FPGA computational accuracy	67
Table 5.1 Data of the aerial TL	75
Table 5.2 Travelling time of CP TL for fault with the distance [8]	79
Table 5.3 Mean and standard deviation of the error for all fault location measurements [8	]85
Table 5.4 Mean value of difference between relays ToA of TW for fault at $DF = 50$ km	[8]85
Table 5.5 ULM test cases FPGA footprints [68]	99

# LIST OF FIGURES

Figure 1.1 Schematic of all the required processes within a time-step of HIL simulation. (a): case
with normal operation; and (b): case with an overrun occurrence
Figure 1.2 General view of (a): CHIL; and (b): PHIL [11]4
Figure 1.3 Illustration of switching/transient frequency and number of state variables for various
FPGA-based applications
Figure 3.1 Network solution flowchart with iterative switch update
Figure 3.2 Single-phase rectifier circuit [21]
Figure 3.3 Single-phase rectifier: Mapping history currents to diode status [21]25
Figure 3.4 Single-phase rectifier DMM function. Representation of cone and hyper-plane
Figure 3.5 (a): LLC converter used as the case study; and (b): voltage gain of LLC tank versus <i>F</i> for different quality factors [21]
<ul> <li>Figure 3.6 LLC with Parameter Set#1: (a): Resonant current of the LLC converter calculated by FE and BE methods, (b): Close-up view of resonant current during blocking mode. LLC with Parameter Set#2: (c): resonant current of the LLC converter calculated by FE and BE methods, (d): Close-up view of resonant current during blocking mode. In all cases an identical time-step of 15ns is set [21]</li></ul>
Figure 3.7 Data-flow diagram: (a): two-stage algorithm; (b): single-stage algorithm [21]
Figure 3.8 Datapath of the hardware implementation of the LLC simulator using Single-Stage DMM Algorithm [21]
Figure 3.9 The <i>N</i> , <i>m</i> MVM module used in the proposed LLC simulator: The module is composed of $N \times m$ ROMs and <i>N m</i> -Input DP units [21]
Figure 3.10 LLC with Parameter Set #2: (a): Output voltage, (b): Switching frequency and fault period [21]41
Figure 3.11 LLC with Parameter Set #2: Close-up views of output voltage and resonant current,

magnetizing current. (a)-(b): Close-up view 1; (c)-(d): close-up view 2; (e)-(f): close-up view

3; and (g)-(h): close-up view 4. Output voltage close-up views are shown in a wider timespan
than resonant and magnetizing currents [21]42
Figure 3.12 Resonant boost converter circuit [91]43
Figure 3.13 Resonant boost converter circuit-PWM signals
Figure 3.14 Resonant boost converter circuit-resonant inductor current waveform
Figure 3.15 RBC switched at 100 kHz - Real-time test result of resonant inductor current [77]46
Figure 4.1 SDD method scheme-(a): original circuit, and (b): decoupled
Figure 4.2 NTT concept
Figure 4.3 NTT applied to LLC circuit: (a): LLC circuit is teared to three sub-circuit, (b): interconnection of three sub-circuit equivalents
Figure 4.4 (a): Simulation algorithm of the NTT, (b): Simulation algorithm of the relaxed NTT 54
Figure 4.5 (a): DMM-SDD flowchart, (b): DMM-NTT flowchart
Figure 4.6 EVBC test case [77]57
Figure 4.7 Adjacency graph of the feasible switch combinations: (a): diode combination of stage 1; (b): diode combination of stage 2 [77]60
Figure 4.8 EVBC circuit waveforms: (a): inductor current of IBC, (b): phase-a resonant current, (c): phase-a magnetizing current, (d): output voltage, (cv.1)-(cv.8): closed-up views [77]61
Figure 4.9 Datapath of implemented methods, (a): ESU-SDD, (b): DMM-SDD [77]64
Figure 4.10 Datapath of DMM-NTT [77]64
Figure 4.11 General view of the hardware DMM function
Figure 4.12 Experimental setup [77]
Figure 4.13 EVBC circuit- Real-time test results of phase a and phase b resonant and magnetizing currents of LLC [77]
Figure 5.1 Bewley diagram for TWs initiated from a fault at point <i>F</i> [8]71

Figure 5.2 (a): Loss inclusion to the lossless CP TL model, (b): $n$ -Norton Equivalent circuits of $n$ -
decoupled modes relating to the phase quantities [8]73
Figure 5.3 Geometry of the aerial TL75
Figure 5.4 TWs arriving to the terminals (Phase A) of a TL of length $l = 100$ km due to a single- phase fault with $DF = 25$ km for both CP and WB TL models. The time axis origin corresponds to the fault occurrence instant [8]
Figure 5.5 Comparison of time difference between the ground and aerial modes propagation delays and time difference between the arrival times of the first peak and that of the subsequent false peak at terminal <i>k</i> . Case of CP line model and AG fault for different DFs [8]77
Figure 5.6 TWs arriving to the terminals of the 100 km-TL at phase-C for both CP and WB line models. Case of a BC fault with $DF$ = 25 km. The time axis origin corresponds to the fault occurrence instant [8]
Figure 5.7 TWIA arriving to terminal k due to the AG fault at the distance of $DF=25$ km for different ground resistivity ( $\rho$ ) for the CP TL of length $l = 100$ km. The time axis origin corresponds to the fault occurrence instant [8]
Figure 5.8 TWIA arriving to terminal k due to the AG fault at the distances of $DF = 2$ km, $DF = 5$ km, and $DF = 15$ km. The origin of time axis is the instant of initiating fault [8]79
Figure 5.9 Schematic view of the double-ended fault locating test system, comprised of the OP4510 HIL simulator and the SEL-T400L fault locator [8]
Figure 5.10 Configuration of power system under study [8]81
Figure 5.11 Scatter plots of absolute fault locating errors of the fault 20 km to 80 km away from terminal <i>k</i> of a 100 km TL. Fault types: (a): AG; (b): BG; (c): ABG; (d): BC. Each sub-plot includes three simulation time-steps: $\Delta t = 500$ ns, 1 $\mu s$ , 2 $\mu s$ [8]82
Figure 5.12 Empirical cumulative distribution functions of the absolute fault location error for the three simulation time-steps [8]
Figure 5.13 Current at Phase-a, terminal $k$ following an AG fault applied in the middle of a 100

km TL. The signals generated by the simulator are demonstrated in zoomed views for  $\Delta t =$ 

500 <i>ns</i> , 1 $\mu$ <i>s</i> , 2 $\mu$ <i>s</i> . The bottom figures show the points which are captured by the TWFL according to its 1 MHZ sampling rate [8]
Figure 5.14 (a): General view of a TL showing the reflected and incident currents in its terminals
k and $m$ in frequency domain; (b): The Norton equivalents of the TL in the time domain [68]
Figure 5.15 (a): Nodal Solver connected to <i>N</i> ULM Computing Engines; (b): High-level inside view of an individual ULM Computing Engine [68]90
Figure 5.16 Detailed view of the convolutional kernels for terminal k: (a): Method 1: Direct hardware implementation of Eqs. (5.16)-(5.19); (b): Method 2: Rescheduling of Eqs. (5.16)-(5.19) to reduce latency [68]
Figure 5.17 Timing diagrams for: (a): Method 1; (b): Method 2 [68]94
Figure 5.18 100 km aerial line test case [68]96
<ul> <li>Figure 5.19 Test Case 1: an AG fault is applied at receiving end at t = 100 ms and is cleared at t = 200 ms. (a): Receiving end phase voltages; (b): Close-up view of phase-voltages during fault initiation; and (c): Close-up view of phase voltages during fault clearing [68]</li></ul>
Figure 5.20 Assessment of the accuracy of the FPGA-based simulation. (a): Incident current at the receiving end from test case 1, phase-a; (b): Relative errors compared to EMTP [68]97
Figure 5.21 Network for the TWFL test case including ULMs [68]97
Figure 5.22 Test Case 2: an AG fault is applied at $DF = 20$ km (see Figure 5.21). (a): Currents from protected line at sending end; (b): Currents from protected line at receiving end; (c): Phase-a Travelling-Wave currents from sending and receiving ends of the protected line [68]

# LIST OF SYMBOLS AND ABBREVIATIONS

ADC	Associated Discrete Circuit
BE	Backward Euler
СР	Constant Parameter
DMM	Direct Mapped Method
DEFLE	Double-Ended Fault Locating Scheme
EV	Electric Vehicle
EVBC	Electric Vehicle Battery Charger
EMT	Electromagnetic Transient
EMTP	Electromagnetic Transient Program
FE	Forward Euler
FP	Floating-Point
FLOPS	Floating Point Operations Per Second
FP	Floating-Point
FPGA	Field-Programmable Gate Array
HIL	Hardware-In-the-Loop
HSF	High Switching Frequency
SEFLE	Single-Ended Fault Locating Scheme
MVM	Matrix-Vector Multiplication
NTT	Network Tearing Technique
PEC	Power Electronic Converter
PFC	Power Factor Corrector
PFM	Pulse Frequency Modulation
RSM	Resistive Switch Model
RT	Real-Time
RTS	Real-Time Simulation
TL	Transmission Line
TR	Trapezoidal
TW	Travelling Wave
TWFL	Traveling-Wave Fault Locator
ULM	Universal Line Model
WB	Wide Band

### LIST OF APPENDICES

Appendix A: Switch Model	
Appendix B: MANA Equation	114
Appendix C: Discrete Companion Model	116
Appendix D: Fourier–Motzkin Elimination	118
Appendix E: DMM TCAM-RAM Content for EVBC	

#### CHAPTER 1 INTRODUCTION

### **1.1 Context and Motivations**

Electromagnetic transients (EMTs) are initiated by switching events or external sources such as lightning striking power system components. The EMTs manifest their effect in the form of overvoltages and/or overcurrents which can damage insulation of power system components and consequently impair power system operation. The EMTs are characterized by a wide-frequency bandwidth extended from DC up to several MHz depending on the source of transient. Hence, accurate analysis of EMTs requires detailed modeling of components (e.g., transmission lines, power transformers, etc.) experiencing the associated transient disturbances.

To perform power system EMT analysis, component models must be valid for the wide frequency range of the EMT phenomenon of interest in order to accurately replicate high frequency content of the transient disturbances. This is accomplished by using wide-band models of the components which are solved in time-domain. To perform time-domain simulation, as a common practice, the governing differential equations are discretized and converted to the well-known Norton equivalent form based upon which a global system matrix is constructed and solved at each discrete time-point [1-3]. The EMT simulation is categorized to off-line simulation and real-time simulation (RTS). EMTP-type tools [3, 4] are extensively utilized for offline simulation where simulation time-step can be fixed or variable depending on the type of study and simulation requirements.

Comprehensive testing under a realistic condition of system operation has become an indispensable step to ensure functionality of new designed hardware/algorithm prior to real-life commissioning or implementation. This procedure is referred to as hardware-in-the-loop (HIL) testing and is realized by means of RTS which can accurately emulate system dynamics and transients. RTS must be performed under hard real-time (RT) constraint, which means, unlike offline simulation, internal variables and outputs of the simulated system must be computed within the same time duration that its physical counterpart would require. This is accomplished by establishing a tailored hardware and software platforms designed to perform accurate EMT simulation under RT constraint.

#### **1.1.1 Real-Time Simulation**

RTS is commonly used as a mean to assess and verify the performance of a physical device, i.e., controller or protective device, at its early stage of development or before its installation in the plant. In general, RTS is categorized as software-in-the-loop (SIL) and HIL simulation. In SIL simulation, both power system and control/protective algorithm are modelled and run in the simulator and no external connections are involved. In contrast, in HIL simulation, full feedback is established between RT simulator, which simulates power system, and physical device, also known as hardware under test (HUT). The purpose of HIL simulation is to test HUT under controlled and realistic system condition once the underlying algorithm is well-established. As an example of HIL test, a new developed physical controller for a power electronic converter (PEC) or a motor can be connected to a virtual PEC or motor model executed in RT to thoroughly assess their functionality in a realistic and de-risked condition. Some applications of HIL test are: modeling and designing of distributed generation units [5], energy storage device [6], micro-grid [7], protection and fault locating devices [8, 9], PEC controller [10], electrical machine drive [11, 12], ship power system [13, 14], electrical vehicle and their charging system [15], more-electric aircraft system [16, 17], etc.

To ensure the accuracy and reliability of the HIL RTS test, time-step should be selected based on the dynamic of the system to be simulated. For instance, network including PECs involve fast transients where adopting a sub– $\mu s$  time-step is crucial to accurately replicate system transients. To satisfy this timing constraints, FPGAs have become processor of choice in RTS. Although CPUs offer more computational capabilities than FPGAs, their higher latencies in communication buses and memory access are prohibitive factors to achieve small time-step of simulation [18]. Also, graphical processing units (GPUs) which are widely employed to parallelize transient stability type simulations, are not suitable for this high-frequency application due to high latency of PCI Express links. As to FPGA, the parallel architecture and low latency I/Os along with its hardware reconfigurability which provides designing application-specific computing engine, pave the way to achieve very small simulation time-steps. Heterogenous computing platform which leverages both FPGA and CPU features are employed in commercially available simulators such as RTDS [19] and OPAL-RT [20].

With particular reference to RTS of HSF PECs using combined FPGA-CPU, the FPGA is responsible to solve part of the system which has a fast dynamic while the CPU is used to implement the remainder of the system (i.e., controller of PEC).

Since RT simulator interfaces with an external device in the context of HIL test, the time-step is not solely defined based on the model computation time. In fact, time for receiving and/or reading input signals, time for model computation and time for outputting and/or witting signals all contribute to the determination of the HIL time-step. The RTS is considered erroneous if all operations are not completed within the pre-determined fixed time step; a situation which is referred to as overrun, see Figure 1.1.



Figure 1.1: Schematic of all the required processes within a time-step of HIL simulation. (a): case with normal operation; and (b): case with an overrun occurrence

The HIL itself is divided into two categories: i) control hardware-in-the-loop (CHIL) and ii) power hardware-in-the-loop (PHIL). In the CHIL, low-power signals are exchanged between the simulator and the HUT (controller or relay), and hence analogue to digital (A/D) and digital to analogue (D/A) converters are used to interface the HUT with the simulator. In this scheme, according to Figure 1.2.a, the simulator sends feedback signals (if HUT is a controller) or voltage/current signals (if HUT is relay) to the HUT and in return HUT sends back control signals (if HUT is a controller) or command signals (if HUT is a relay) to the simulator.

As opposed to the CHIL, the PHIL involves high power exchange between the HUT and the simulator. As a result, a source and sink are involved in this process in order to provide or absorb power. Power amplifier is employed to feed simulator's outputs to the HUT at high power level, see Figure 1.2.b. It is worth mentioning that amplifier characteristics such as delay, non-linearity, etc., significantly affect the accuracy and stability of the PHIL test.

#### **1.1.2 Motivations**

High switching frequency (HSF) PECs are the important part of modern energy conversion systems due to their high efficiency and power density [21]. On-board power system and micro-grid are some typical applications which utilize HSF PECs as a part of their energy conversion system. As mentioned earlier, to accurately replicate high transients due to switching or resonances, sub- $\mu s$  time-step is required. In addition, the recent advancements in the realm of power system protection lead to the advent of travelling wave (TW)-based relays [8]. The importance of having very small-time step to replicate high-bandwidth fault launched transients requires FPGA to be the main computing platform for this application.



Figure 1.2: General view of (a): CHIL; and (b): PHIL [11]

In general, the emerging applications of FPGA-based RTS along with the challenges that they introduce can be listed as follows:

• HSF PECs: the switching frequency of these converters are in the range of 50~500 kHz. Their accurate simulation entails heavy computation within a very small time-step.

- On-board power systems: including electrical systems of aircrafts, ships, EVs, etc., which contain multiple HSFs. In addition to the challenges regarding the simulation of HSFs, the network is tightly coupled and conventional decoupling approaches such as transmission line decoupling are not applicable.
- Microgrid system: in this case, switching frequencies of PECs are not as high as on-board power system converters. However, microgrids have high number of nodes and large system matrix which pose challenge for FPGA-based RTS mainly due to the FPGA limited resource.
- TW-based relay testing: FPGA-based HIL RTS is required for testing emerging TW-based relays. The need to have computationally demanding TL model running in a very small time-step is the challenge faced by the FPGA-based simulation.

Figure 1.3 shows the above-mentioned applications in terms of their frequency range and size. In this figure, the area covered by "Existing FPGA-based implementation" refers to the works which have been done in the literature, e.g., [18, 22, 23].

Accurate RTS of HSF PECs, necessitates the adoption of two-valued resistive switch model instead of associated discrete circuit (ADC) switch model to avoid spurious oscillations and fictious losses. Using resistive switch model, the states of naturally commutated switches are forced by the circuit variables in the same time step. Thus, an iterative process is required to obtain accurate switch status and network variables. In this process, iteration will increment as long as no conflict between the obtained variable and the switch logics is detected. As a result, the iteration number is not fixed and leads to variable computational burden. This issue is even more challenging when adopting very small time-steps which is crucial for accurate RTS of HFS PECs. To cope with this challenge, the following well-known approaches have been so far proposed: i) ADC switch model: this model leads to erroneous results and is not practical for HSF PECs, ii) explicit solver: this approach suffers from potential instability and it is unable to model all operating modes of a converter, iii) previous time-point variables to determine diode status: this tends to generate inaccurate results and is invalid for some converters, iv) converter equivalent: the downsides of this approach is its inability to access individual switch variables and represent all operating modes of a converter.

In addition to the aforementioned challenges in the RTS of HSF PECs, the FPGA suffers from limited memory resources which prohibits RTS of converters with high number of switches. In this sense, this thesis proposes a promising algorithm which extends utilization of the FPGA hardware for fast and accurate RTS of HSF PECs.



Figure 1.3: Illustration of switching/transient frequency and number of state variables for various FPGA-based applications

Regarding the microgrid application, the challenge posed by large network size can be addressed by accommodating multi-FPGA or heterogeneous processors in the RTS platform. Features such as, low-latency communication link between computing platforms, selection of accurate converter model, and choosing an accurate/stable decoupling/partitioning method must be considered in RT microgrid simulator.

Regarding the TW-based relay testing, using less computationally demanding transmission line (TL) models may seem a promising approach to address time-step constraint. However, it will affect the simulation accuracy. For TL case, in particular, any model which does not consider the frequency dependence of line parameters yields inaccurate transient response. The determination of how and to what extend TL modelling affects the TW-based relay testing needs a comprehensive evaluation in both offline and experimental contexts.

### **1.2 Objectives and Contributions**

The focus of this thesis lies on the development of efficient and accurate methods for FPGA-based RTS of fast transient power electronics and power systems with particular reference to HSF PECS and TW-based transmission line protection. Regarding HSF PECs, the aim is to propose a method that satisfy RTS requirements including, accuracy, small time-step and resource-efficient FPGA implementation. In addition, the proposed method must be scalable and applicable to systems with multi–HSF PECs or on-board power systems.

The FPGA-based RTS will be proposed for HIL testing of TW-based protection relays. To this aim, the suitability of different TL models is thoroughly investigated and tested versus experimental results for constant parameter (CP) line model. Moreover, the FPGA-based RTS of ULM is presented in order to obtain a higher level of accuracy for HIL testing of TW-based transmission line protection relays.

This thesis contributions are summarized as follows:

- Developing a fast and computationally efficient algorithm to enable accurate RTS of HFS PECs with switching frequencies up to several hundreds of kHz.
- Designing a low-cost FPGA hardware to implement a high-fidelity FPGA-based RTS platform of HSF PECs.
- Devising a method to model and perform accurate RTS of on-board power systems, including multiple HSF PECs with tight coupling, with a very small time-step.
- Evaluation of a CP-based traveling-wave fault locator (TWFL) HIL test setup.
- Developing a high-performance FPGA universal line model (ULM) computing unit achieving sub-µs time-step.

### **1.3 Thesis Outline**

This dissertation is comprised of 6 chapters which are organized as follows:

**CHAPTER 1** which provides some backgrounds on RTS and explains its associated challenges. Also, it defines motivations and objectives of this thesis. **CHAPTER 2** gives an overview on the state-of-the-art in power electronics RTS field. Various aspects including switch models, solvers, partitioning/decoupling techniques, etc., are revisited. This chapter closes with a brief review about the RTS of TL and TWFL testing methods.

**CHAPTER 3** proposes an algorithm for accurate and fast RTS of HSF PECs. In this chapter the iterative simulation of PECs is discussed. Then, the proposed direct mapped method (DMM) is elaborately explained. To this aim, first, DMM is explained using a simple example and then its general mathematical formulation is presented. To demonstrate the effectiveness of the DMM, a 500 kHz resonant LLC converter and a 100 kHz resonant boost converter are considered as the case studies. FPGA implementations of the test cases are elaborately discussed.

**CHAPTER 4** proposes methods which enable applying the DMM to larger circuits including multiple HSF PECs, such as on-board power systems. The proposed methods leverage advantages of both the DMM and decoupling to enhance the scalability of the DMM. An electric vehicle battery charger circuit is considered as the case study and details of its FPGA implementation are provided.

**CHAPTER 5** embodies evaluation of HIL RTS testbench of the emerging TW-based fault locators. To this aim, first, an HIL test setup including CP line model is experimentally evaluated. Then, a high-performance FPGA-based computation unit for ULM enabling very small time-step of RTS is proposed and elaborately discussed.

**CHAPTER 6** highlights the summary of this dissertation and provides some potential future works.

# CHAPTER 2 STATE-OF-THE-ART OF THE REAL-TIME SIMULATION

### 2.1 Introduction

The presence of high frequency transients in a network poses challenges for RTS due to its high computational demand within a small time-step. A great deal of research is being conducted on high switching frequency converters, and some industries, such as automotive and aerospace increasingly tend to utilize these converters in their applications. The real-time simulation should follow this pace and offer accurate and reliable solutions to conduct HIL testing of these converters. Testing of TW-based fault locator relays is another high frequency transients application for RTS. However, due to the very recent emergence of these type of relays, there is little literature about the topic.

To perform real-time EMT simulation of networks including PECs, two important subjects must be considered: i) how to efficiently model a switch; and ii) how to form/solve network equations. Switch model has a pivotal role on the accuracy of simulation and must include two features, i) ability to accurately replicate system level EMTs; and ii) being computationally efficient. With these two considerations, different switch models are proposed and implemented in RTS. This chapter reviews and discusses these models. Throughout this chapter, the term solver refers to any method which covers areas of discretization, forming network equations, handling switches, etc. This chapter revisits different solvers proposed in the literature and discusses their different aspects.

Last part of this chapter reviews literature related to TW-based fault locating. Since TL modeling plays an important role on the accuracy of TW-based fault locators testbench, literature concerning RTS of TL is revisited. Afterward, existing test methods of TW-based relays are reviewed. Finally, features of commercially built TWFL HIL setups is discussed.

### 2.2 Switch Modeling

In the realm of RTS of power electronic circuit, behavioral switch models are the preferred model over detailed models [24]. Although internal details of switch are not modelled, this approach is sufficient to accurately generate circuit level transients of a power electronic-based circuit [25];

and using detailed switch models is not necessary. In that regard, two approaches are followed; one tries to keep the system matrix constant regardless of switching actions and other uses two different resistances for each of switch states and deals with time-varying topology.

In early 90s, the idea of using a small capacitor and a small inductor (L/C), respectively, when the switch blocks and when it conducts, is proposed [26]. Few years later, this concept is used to develop the so-called associated discrete circuit (ADC) model and make the admittance matrix fixed irrespective of the switch status [27, 28]. In ADC, the conductance values associated with the capacitor and inductance of the model are set to be equal ( $g_L = g_C$ ). To increase the stiffness of the switch model, an element *R* is added in series with the capacitance during the on state [29]. Another advantage of the ADC is that switch status is determined explicitly from gating signals and variables taken from the previous time-point of the simulation.

Although ADC is computationally efficient and can reach very small time-step in RTS, it suffers from some drawbacks:

- i) <u>False oscillation</u>: in [27], it was proved that the ADC reaches correct steady state values but will cause false oscillatory behavior of the current/voltage of the switch that may lead to wrong simulation results;
- <u>Fictitious losses:</u> the energy stored in the capacitor, or the inductor is lost when switch transition occurs, which leads to false losses in the converter. These losses increase in higher switching frequencies.

In general, there is no specific rule to determine the value of conductance in ADC and approaches such as setting higher damping factor in switch model may lead to suppress oscillation, but it further increases switch losses. In [29], by evaluating impulse response of the RLC circuit (representing switch model), it is proposed to select switch parameter based on the equal lost energies in switch capacitor and inductor while keeping the damping factor in the range of 0.85~1.33. To calculate energy losses, one can use the rated current and voltage of converter. In [30], switch conductance is obtained through an optimization problem. ADC model is improved by adding a compensation source in [31]. These methods, however, are not general and are fully dependent on the converter's topology. In addition, ADC is not practical for some converters such as resonant type converters since the adopted switch model may affect the resonant tank behaviour.

Another switch model which keeps network matrix constant is proposed in [32]. This approach represents a switch with a Norton equivalent with a fixed conductance value. Unlike the ADC, there is no constraint on the selection of the conductance value; but the value of Norton current source must be determined through an iterative solution.

In resistive switch model (RSM), switch is replaced by a high resistance value,  $R_{off}$ , when it is off and by a small resistance,  $R_{on}$ , when it is on. This leads to time-variant network matrix and to compute network variables in RTS i) the inverses of network matrices associated to all switch combinations are precomputed [33, 34] or ii) the inverse of matrix is updated online [33-35]. The first approach is hardly conceivable for a circuit with high number of switches even with the recent advancements in hardware platforms capacity, whereas the second approach solves this issue but with the expense of higher computational cost. In RSM, status of naturally commutated switches implicitly relates to network variables. As a result, an iterative solution is needed to accurately obtain their statuses. Hence, despite the accuracy and reliability offered by RSM, it brings heavier computational load compared to ADC and achieving small time-steps in RTS is more challenging.

### 2.3 Network Solution Approaches

#### 2.3.1 Circuit Partitioning

A diakoptics approach proceeds by tearing a circuit into multiple subcircuits and solving them independently and then interconnecting interfaces according to the original circuit topology to obtain the solution of whole circuit [36]. The interface variables can be voltage, current or both. In fact, diakoptics tries to increase the scalability of the simulation by parallelizing the computational burden to enable the simulation of large networks.

In [37], the hybrid compensation method for non-linear branches is proposed. This method compensates non-linear branches with voltage/current sources across the interface ports of linear circuit to obtain the solution of whole circuit.

State-space nodal (SSN) method is proposed in [38, 39] which enables modular SS modelling of multiple sub-circuits. This approach is hybrid and allows both current/voltage being the interface variables (hybrid); thus both Thévenin/Norton equivalents can be incorporated in the global nodal or MANA [3] equation. This approach circumvents modelling limitations regarding SS formulation

such as synthesis process of large circuit and difficulties related to solution of nonlinear elements [38]. In addition, it enables efficient handing of high numbers of switches.

Multi-area Thevenin equivalents (MATE) approach is another partitioning alternative proposed in [40, 41]. In MATE, Thévenin equivalents of sub-circuits are computed and are utilized to solve for link currents. Link currents are then applied to each sub-network to update the internal states of each sub-circuit. This approach is used in [25] to conduct RTS of a 12/24-pulse bridge serving as HVDC converter. It has been demonstrated in [42] that MATE is actually a direct copy of the compensation method [43-45] which existed before the publication of MATE.

Another network tearing technique (NTT) is proposed in [33]. The main motivation behind this method is to exploit parallel computing capability of FPGA and increase the potential number of switches in RTS of power electronic circuit. The interface ports can be either current or voltage (hybrid) and global matrix is formed using Tableau approach [46].

In general, diakoptics offers the following benefits for RTS application:

- *Circuit dimension*: it enhances parallel computation which specially suits well the FPGA structure.
- Lower latency: it allows reaching small time-step of simulation.
- *Flexibility of solver selection:* it offers the ability to choose suitable solver for each subcircuit.
- *Increases switch number:* it increases the potential number of switches since it deals with independent switch subsets; hence enables RTS of more complex power electronic based circuit.

#### 2.3.2 Decoupling Approaches

Decoupling a large network distributes computational load between multiple computing units. It is also used to connect different simulators in HIL simulations or interfacing simulator to the hardware under test. When a decoupling is applied, the interface variables are usually discretized using an explicit method. It is also possible to apply interface variables with some time delays, depending on the method being used. In power system applications, a TL naturally decouples two networks connected to its terminals. A decoupling approach called Stubline is introduced in [47], and consists in emulating the behaviour of a very short TL. In the Stubline approach, the TL section (stub) is comprised of an inductor and a capacitor chosen in such a way that the travelling time of the TL is equal to one simulation time-step. As a result, incident voltage at each end is explicitly related to the reflected voltage at the other end, which is already known. Stubline adds state variables to the system and changes its transient response. Reference [48] proposes another decoupling technique similar to the Stubline approach, which aims to decouple a circuit using series inductor branches.

Discretizing energy storage element using an explicit integration method, e.g., forward Euler (FE) is another method for decoupling a circuit. In [49], FE is applied to the so-called selected energy storage elements (parallel capacitor/series inductor) to decouple a circuit. In [50], first, the fastest state variable is detected and discretized with BE and then FE is applied to one of its adjacent states in order to decouple the circuit.

In general, decoupling a circuit brings the same benefits as diakoptics methods, but they must be thoroughly investigated prior to simulation as they are susceptible to numerical errors and potential instability.

#### 2.3.3 Multi-Rate Simulation

The stiff nature of power systems and power electronic-based circuits involves a wide range of time constants. To accurately reproduce system transients, some parts of the network require small simulation time-steps (for instance switching circuits), whereas other parts (such as machines) have slower dynamics and can be accurately simulated using larger time-steps. A multi-rate simulation approach assigns distinct simulation time-steps to each sub-circuit according to their time constants. In this condition, accuracy and stability of simulation must be assessed.

In a multi-rate simulation involving two sub-circuits, the effect of the faster sub-circuit on the slower sub-circuit are reflected using averages of interface variables from the fast sub-circuit [51]. On the other side, the solution of the faster sub-circuit involves extrapolations from the results of the slower sub-circuit.

In [41], multi-rate simulation is performed while circuit is partitioned using MATE. In [49], multirate approach is used along with FE to simulate power electronic-based circuits. TL-based decoupling and multi-rate are combined to conduct FPGA-based RTS of a microgrid in [52]. In [53], system including multiple time-steps is represented by a unified state-space formulation and its stability is assessed by its poles location.

### 2.4 Dealing with Switches

As mentioned earlier, the ADC switch model offers a low computational burden since it keeps the system matrix constant. This fact has been proven in literatures [18, 23] where small time-steps of simulation are reported.

Dealing with RSM is more challenging in RTS. Method proposed in [54] separates switch module from the rest of network to deal with smaller matrices to update switches. It uses RSM and thus requires iterative procedure to update naturally commuted switch status. The RTS of a multiconverter system is proposed in [55]. In this work, the compensation method [37, 43-45] is used to handle switching networks and switch status are updated iteratively. However, to reach small timesteps, two assumptions are made while drawing equivalents seen from converter's ports, i) independent sources values are applied with a time-step delay; and ii) other converter's switching interactions are ignored, and they are replaced by zero voltage/current sources. The need for iterative solution is removed in [34] by including small parasitic element in switch model and thus using previous time-step values to determine diode status. A method to conduct RTS of voltage source converter is proposed in [56]. It assumes that switches in one leg of inverter have very weak interaction to other legs and thus each leg is treated separately. By forming equivalent circuit of each leg and using the previous time-point value of dc-link voltage, an algorithm is designed to search through all possible combinations and find the valid one. In [33], NTT is used to tear a circuit into multiple sub-circuits and iterative algorithm runs inside each sub-circuit to obtain accurate switch status. Since number of iterations are unknown, a fixed number of iterations (two) is defined.

Precomputing/storing of all system matrices associated with all switch combinations is not always practical due to the limited on-chip memory capacity. To cope with this issue, Sherman-Morrison-Woodbury formula [57] is utilized in [58] to update the inverse of matrix. This method obtains

15

matrix inversion by applying some changes in the original matrix. Later, [35] utilized this method for RTS of power electronic-based circuit. Gauss-Jordan elimination is another alternative to compute inverse of a matrix during simulation. It is used by [33, 59] for RTS of power electronic-based circuits.

Another approach to deal with a converter in RTS application is using the switching function and time average methods [14] [24, 60]. Latency based linear multistep compound (LB-LMC) [61] uses switching function approach to handle converters. In LB-LMC, a power electronic converter, e.g., an inverter is considered as a component and its switching part is modelled based on the switching function method while its internal states are explicitly discretized. As a result of doing this, non-linear components are contributing to system solution with a time-step delay where their reflected current/voltage sources appear in the injection matrix. In [14], the LB-LMC is used to conduct FPGA-based real time simulation of a ship power system (including AC/DC converters switch at 200 kHz) and time-step of 50 ns was reported. In [62], LB-LMC is exploited to implement a microgrid comprised of multiple 50 kHz converters. In general, the limitations of these can be listed as: i) potential instability, ii) individual switch variables are not accessible, iii) all operational modes of converters cannot be modelled and they are incapable to model internal fault such as commutation malfunctions, iv) they cannot accurately represent transient behaviour of the simulated system, and v) they depend on PEC topology and require knowledge about the PEC operation.

Predictor-corrector (PC) method is another approach of simulating a switch circuit. It uses combination of explicit and implicit solvers, e.g., the FE and trapezoidal integration rules. Using PC, the switch network solution resides outside the linear network and thus their solutions are decoupled. At each time-point, predicted variables and gate signals are used to determine the switch status following which voltage/current of switch network ports are drawn in order to compute corrected states in the corrector stage. Meanwhile, predictor stage computes the predicted states values of the next time-point [63, 64]. In PC method, to derive ports' voltages/currents as a function of switches variables, a pre-processing of circuit is required which adds an extra effort before simulation starts. Furthermore, since the additional computations are required in this method, FPGA hardware resource usage limits the circuit size.
### 2.5 Real-Time Simulation of Transmission Line

Time-domain TL modeling is mainly categorized into two main approaches [65]: i) frequencyindependent models and ii) frequency-dependent models. CP line refers to a distributed parameter line model whose per unit length parameters are considered frequency independent. CP model is very efficient regarding its computational load and may suit well for FPGA-based RTS application due to the FPGA limited processing capabilities. However, it suffers from modelling deficiency which arises from the fact that it ignores the frequency dependency of TL parameters, in particular the series resistance and inductance [66].

ULM [67] is a wide-band frequency-dependent model of TL. The model incorporates full frequency dependency of the TL parameters and is considered the most accurate TL model employed in EMTP [3, 4]. However, it is computationally expensive for RTS [68]. CPU-based RTS of the ULM have been demonstrated to run with the time-steps in the >10  $\mu s$  range. The ability to run the TL model on FPGA with a sub- $\mu s$  is justified by various application requirements such as HIL testing of TWFLs, where accurate regeneration of high frequency transients is crucial. FPGA-based real-time implementations of the ULM that have been reported in the literature fails to report small time-step [8]. In [69], FPGA-based implementation of the ULM for HIL-based testing of TWFLs is presented and a time-step of 1.42  $\mu s$  has been achieved. The TL model implemented in [69] is based on the second-order realization of TL's state-space equations, which deals with only real states.

#### 2.5.1 TW-based Relay and Fault Locator

TWs initiated by a fault on a TL can be captured and processed by TL protection systems to rapidly detect and accurately locate the fault. Such an approach decreases patrolling time of maintenance team to pinpoint the fault location. In addition, due to fast operation of these relays, transient stability margin of power system will be increased. Modern TWFLs can locate a fault within a tower span ( $\pm 300$  m) but require 1 MHz sampling rate to achieve such performance. The TW-based protection and fault location testbench must provide a realistic condition which is as close as possible to the real system. This condition allows a reliable condition for any troubleshooting and calibration of relay or TWFL to attain the expected performance [8].

#### 2.5.2 Existing TW-Based Relay and Fault Locator Test Methods

The existing experimental evaluation of TWFLs is not mature enough and cannot properly address different technical aspects of a given relay prior to installation in a real grid. An external circuitbased setup for reproducing fault launched TWs has been proposed in [70] for testing of TWFLs. This type of testing is not however straightforward and should comply with many criteria including time accuracy, synchronization, sufficient slew rate, etc. [70]. This is not a practical choice in the sense that the generated TW cannot accurately represent a real fault-launched TW due to uncertainties in the grid parameters as well as the stochastic nature of the fault in terms of its location and inception angle [8].

Events simulated or recorded from the actual power system can be used by the relay's event playback function to provide more realistic TW to be applied to the relay. This method does not require any physical test setup and relies on recording the event data and subsequently applying them to the relay in an off-line manner. Furthermore, this type of testing requires data manipulation such as adjustment of signal scaling and sampling frequency, choosing the period of interest and also passing through the anti-aliasing filter to mimic the relay data acquisition system [70]. These processes can be time-demanding and burdensome for the user. These, if not properly done, can also affect accuracy of fault locating of the test system. In addition, the testing process is time-consuming and might limit the repeatability of the test for different network conditions and fault locations [8].

### 2.5.3 TW-Based Relay RT Test Setup

Real-time simulators can be used for providing a comprehensive test-bench for protection relays in a closed loop. Such a setup can be beneficial in the sense that it provides flexibility for selecting power system parameters as well as fault type and location while allowing the regeneration of realtime signals. Moreover, real-time simulators provide the possibility of in-situ testing of protection relays. However, for a HIL setup to function properly as a TWFL testbed, the simulated network necessitates a sub- $\mu s$  time-step as well as accurate frequency-dependent line model [8]. In addition, ULM line model is required to have a reliable TWFL testbench. Recently, [69] presented an HIL testing of TWFLs using the ULM. This work used SEL T400 L time domain relays as a TWFL devices. This system evaluates both internal and external faults with the power system running with 1.42  $\mu s$  time-step.

Commercial solution of TW relay test setup is proposed by RTDS Technologies using frequency dependent line model [71]. To perform a high-speed simulation, TL model is simulated on GTFPGA unit [72] which includes a Xilinx Virtex-7 FPGA board and is running in parallel with the main hardware who is responsible to simulate rest of network. In this TW relay testbench, TL model achieves a time-step of  $1-3 \mu s$ .

Another commercial TW testing solution is proposed by OPAL-RT Technologies which can incorporate both CP and FD TL models [73]. In this testbench, TL model is stimulated by FPGA in eHS environment with a sub- $\mu s$  time-step.

# 2.6 Conclusion

This chapter provided an overview on the state of the art of the literature related to the thesis objectives. Different aspects of RTS of PECs and HIL testing of TW relays have been discussed. This review gives a general insight on what has been presented in the literature. Although certain technicalities have been eluded from this literature review for the sake of readability, more details will be presented in the subsequent chapters, along with their related subject. More basic concepts such as switch models, MANA equations and discretization methods are presented in the appendices.

### CHAPTER 3 DIRECT MAPPED METHOD

# 3.1 Introduction

The ever-increasing trend toward the deployment of HSF PECs in energy conversion systems is challenging for the realm of FPGA-based HIL. The contribution of this chapter is to propose a new method to accurately model and perform FPGA-based RTS of HSF PECs. The proposed method called direct mapped method (DMM) and has two important features: i) It has a lower latency and more deterministic computational time compared to an iterative solution; and ii) It renders an exact solution. These characteristics make DMM suitable for RTS.

The mathematical foundation of the proposed method (DMM) is presented in this chapter. A demonstration of the effectiveness of the DMM is presented with FPGA-based real-time simulation test cases of a 500 kHz LLC resonant converter and a 100 kHz resonant boost converter. Dedicated FPGA computing units are designed for both network solver and the DMM computing unit. To further decrease the latency of the FPGA implementation, data dependencies of different computing stages are recomposed by a proper rewriting of the DMM equations.

# **3.2 Iterative Solution**

Due to its capability to generate accurate circuit level transient response in the simulation of a power electronic circuits, this thesis uses the RSM. In general, based on switch status logic, a semiconductor is either categorized as controlled and uncontrolled switch. The status of a controlled switch is fully determined by its input gate signal and is known at the beginning of each time-step. However, determining the status of an uncontrolled switch is not straightforward since it is forced by network variables at the same time-point. In this condition, iterative solution is used to simultaneously determine the switch status and network variables.

In order to mathematically explain the iterative process, we will use the compensation method [37, 44], which replaces switches with current sources [44].

Let us assume that the following network equations govern the time domain simulation of a power electronic circuit:

$$\begin{bmatrix} \mathbf{A}_{c} & \mathbf{T}^{T} \\ \mathbf{T} & -\mathbf{R}_{sw}^{\sigma} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{i}_{sw} \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{z}$$
(3.1)

in which  $\mathbf{A}_{c}$  is the system MANA matrix of the network excluding switches (linear part), **T** is the connectivity matrix and  $\mathbf{R}_{sw}^{\sigma}$  is the switches resistances matrix for a given switch combination  $\sigma$ . In addition, **x**,  $\mathbf{i}_{sw}$ , **B** and **z** are respectively MANA unknown variables, switch current vector, incidence matrix and MANA input vector. Since the switches are modelled using RSM, the switch voltage vector is given by:

$$\mathbf{v}_{\mathrm{sw}} = \mathbf{R}_{\mathrm{sw}}^{\sigma} \mathbf{i}_{\mathrm{sw}} \tag{3.2}$$

Using the Schur complement of (3.1), and combining with (3.2), one can obtain:

$$\mathbf{v}_{\rm sw} = \mathbf{v}_{\rm th} - \mathbf{R}_{\rm th} \mathbf{i}_{\rm sw} \tag{3.3}$$

with:

$$\begin{cases} \mathbf{v}_{th} = \mathbf{T}\mathbf{A}_{c}^{-1}\mathbf{B}\mathbf{z} \\ \mathbf{R}_{th} = \mathbf{T}\mathbf{A}_{c}^{-1}\mathbf{T}^{\mathrm{T}} \end{cases}$$
(3.4)

 $\mathbf{v}_{th}$  and  $\mathbf{R}_{th}$  are Thévenin equivalent parameters of the linear network seen by the switches. By inserting (3.2) in (3.3), one obtains:

$$(\mathbf{R}_{\rm th} + \mathbf{R}_{\rm sw}^{\sigma})\mathbf{i}_{\rm sw} = \mathbf{v}_{\rm th} \tag{3.5}$$

To solve (3.5),  $\mathbf{R}_{sw}^{\sigma}$  and  $\mathbf{i}_{sw}$  should be coherent. For instance, if a diode is passing, its current should be positive. This implicit dependency between uncontrolled switch variables and its status requires an iterative procedure to solve (3.5) as:

$$(\mathbf{R}_{\rm th} + \mathbf{R}_{\rm sw}^{\sigma})^{(i)} \mathbf{i}_{\rm sw}^{(i+1)} = \mathbf{v}_{\rm th}$$
(3.6)

where superscript i denotes the iteration index at each time-point. The number of iterations is different at each time-point and depends on the circuit conditions.<sup>1</sup>

Figure 3.1 shows the simulation flow of this iterative approach. As can be seen in the flowchart, uncontrolled switch currents are computed at each iteration until all switch status are correctly set. "Check/Update switch combination" block updates switches based on their logic and checks whether they differ from their previous values and set their status accordingly. This iterative process may violate time constraint of RTS due to its nondeterministic computation time; thus, it is not practical for RTS applications. In this condition, accurate RTS of HSF PECs becomes very challenging due to very small-time step requirements, provided that 25-50 samples are typically expected in each half switching cycle to obtain accurate results. This issue is intensified when dealing with a large circuit comprising a high number of switches.



Figure 3.1: Network solution flowchart with iterative switch update

<sup>&</sup>lt;sup>1</sup> It is worth mentioning that switch current or its voltage,  $\mathbf{i}_{sw}$  and  $\mathbf{v}_{sw}$ , can interchangeably be used here.

# **3.3 Direct Mapped Method**

DMM aims to avoid an iterative solution to perform a fast and accurate RTS of power converters. This is realized by building a direct link between state variables and switch statuses of a PEC by using a mapping function. Before delving into the details of the mathematical formulation of the DMM, its principles are presented through a simple yet fundamental example: the single-phase rectifier.

#### **3.3.1** Demonstration of DMM Concept Using the Single-Phase Rectifier Circuit

The single-phase rectifier is widely utilized in HSF conversion systems: it is employed at the input stage of most EVBC circuits and at the rectifying stage of certain LLC converter topologies. Figure 3.2 demonstrates single-phase rectifier circuit, where both AC and DC sides of the rectifier are replaced by a Norton equivalent comprised of a conductance in parallel with a time-dependent current source (a history term).

The following systems of equations associate network equations of the rectifier using MANA approach [21]:

$$\mathbf{A}^{\sigma_{rec}}\mathbf{x} = \mathbf{b} \tag{3.7}$$

$$\mathbf{A}^{\sigma_{rec}} = \begin{bmatrix} g_1 + g_{d_1} + g_{d_2} & -g_1 & -g_{d_1} \\ -g_1 & g_1 + g_{d_3} + g_{d_4} & -g_{d_3} \\ -g_{d_1} & -g_{d_3} & g_2 + g_{d_1} + g_{d_3} \end{bmatrix}$$
(3.8)

$$\mathbf{b} = \begin{bmatrix} i_1^h & -i_1^h & i_2^h \end{bmatrix}^T$$
(3.9)

where  $\mathbf{A}^{\sigma_{rec}}$  is the MANA matrix,  $\sigma_{rec} \in \{0, 1, ..., 15\}$  refers to the 16 diodes status combinations,  $\mathbf{x} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^{\mathrm{T}}$  is the unknown node voltages and **b** is the current injection vector.  $g_{d_i} = 1/R_{d_i}$ ,  $i \in \{1, 2, 3, 4\}$  are the conductances associated with the diodes  $(g_{d_i} = g_{on} \text{ or } g_{off})$ , with respect to the diode combination  $\sigma_{rec}$ ).



Figure 3.2: Single-phase rectifier circuit [21]

The network equations can be rewritten as:

$$\mathbf{A}^{\sigma_{rec}}\mathbf{x} = \mathbf{b} = \mathbf{B}\mathbf{z} \tag{3.10}$$

$$\mathbf{B} = \begin{bmatrix} +1 & 0\\ -1 & 0\\ 0 & +1 \end{bmatrix}$$
(3.11)

$$\mathbf{z} = \begin{bmatrix} i_1^h & i_2^h \end{bmatrix}^{\mathrm{T}} \tag{3.12}$$

In addition, diodes voltages can be defined using a connectivity matrix, **T**, as:

$$\mathbf{v}_{\mathbf{d}} = \mathbf{T}\mathbf{x} \tag{3.13}$$

$$\mathbf{T} = \begin{bmatrix} +1 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & +1 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$
(3.14)

where  $\mathbf{v}_{\mathbf{d}}$  is the vector of diode voltages:  $\mathbf{v}_{\mathbf{d}} = [v_{d_1}, v_{d_2}, v_{d_3}, v_{d_4}]^{\mathrm{T}}$ . From (3.10) and (3.13),  $\mathbf{v}_{\mathbf{d}}$  can be obtained as:

$$\mathbf{v}_{\mathbf{d}} = \mathbf{T}(\mathbf{A}^{\sigma_{rec}})^{-1}\mathbf{B}\mathbf{z}$$
(3.15)

The RSM requires the correct state of each diode (*on/off*) at each time-point. This is done by checking whether the sign of the diode's voltage or its current ( $v_{d_i} = R_{d_i}i_{d_i}$ ,  $R_{d_i} > 0$ ) is positive or negative. From (3.15), it can be found that diode  $d_i$  conducts whenever  $\vartheta_{d_i} > 0$ ,  $i \in \{1, 2, 3, 4\}$  [21]:

$$\vartheta_{d_1} = + (g_{d_2}g_{d_3} + g_{d_3}g_{d_4} + g_{d_3}g_2 + g_{d_4}g_2)i_1^h - (g_{d_2}g_{d_3} + g_{d_2}g_{d_4} + g_{d_2}g_1 + g_{d_4}g_1)i_2^h$$
(3.16)

$$\vartheta_{d_{2}} = + (g_{d_{1}}g_{d_{4}} + g_{d_{3}}g_{d_{4}} + g_{d_{3}}g_{2} + g_{d_{4}}g_{2})i_{1}^{h} - (g_{d_{1}}g_{d_{3}} + g_{d_{1}}g_{d_{4}} + g_{d_{1}}g_{1} + g_{d_{3}}g_{1})i_{2}^{h}$$

$$(3.17)$$

$$\vartheta_{d_3} = + (g_{d_1}g_{d_2} + g_{d_1}g_{d_4} + g_{d_1}g_2 + g_{d_2}g_2)i_1^h - (g_{d_1}g_{d_4} + g_{d_2}g_{d_4} + g_{d_2}g_1 + g_{d_4}g_1)i_2^h$$
(3.18)

$$\vartheta_{d_4} = + (g_{d_1}g_{d_2} + g_{d_2}g_{d_3} + g_{d_1}g_2 + g_{d_2}g_2)i_1^h - (g_{d_1}g_{d_3} + g_{d_2}g_{d_3} + g_{d_1}g_1 + g_{d_3}g_1)i_2^h$$
(3.19)

For each combination  $\sigma_{rec}$ , a system of inequalities combining (3.16)-(3.19) is obtained. Fourier-Motzkin elimination (FME) [74] is used to evaluate the feasibility of each set and diode combination.

The feasibility check reveals that only 4 feasible diode combinations exist, those for which  $d_1 = d_4$  and  $d_2 = d_3$ . A mapping function  $f(i_1^h, i_2^h)$  linking the state variables to diode states,  $(i_1^h, i_2^h) \mapsto \sigma_{rec} \equiv (d_1, d_2, d_3, d_4)$ , can be obtained. Figure 3.3 demonstrates this mapping function in the  $i_1^h i_2^h$ -plane with four half lines dividing the plane into four feasible regions: (0, 0, 0, 0) = Blocked, (1, 0, 0, 1) = Positive, (0, 1, 1, 0) = Negative and (1, 1, 1, 1) = Shorted [21]. The half lines start at the origin and have slopes of  $\pm m_1$  and  $\pm m_2$ , as illustrated in Figure 3.3. The slopes  $m_1$  and  $m_2$  are given by:

$$m_{1} = \frac{g_{off}g_{off} + g_{off}g_{off} + g_{off}g_{off} + g_{off}g_{2}}{g_{off}g_{off} + g_{off}g_{off} + g_{off}g_{1} + g_{off}g_{1}}$$
(3.20)

$$m_{2} = \frac{g_{on}g_{on} + g_{on}g_{on} + g_{on}g_{2} + g_{on}g_{2}}{g_{on}g_{on} + g_{on}g_{on} + g_{on}g_{1} + g_{on}g_{1}}$$
(3.21)

$$m_1 = \frac{g_{off} + g_2}{g_{off} + g_1} \tag{3.22}$$

$$m_2 = \frac{g_{on} + g_2}{g_{on} + g_1} \tag{3.23}$$

Assuming  $g_{off} \cong 0$ , slope  $m_1$  is approximated by:

$$m_1 \cong \frac{g_2}{g_1} \tag{3.24}$$

From (3.24), it is observed that the slope  $m_1$  (which delimits the Blocked from the Positive and Negative states) is approximately independent of diode conductances and is a function of both AC and DC side parameters. It also appears from Eq. (3.24) that explicit solvers such as FE [49], PC [64] and latency insertion method [61] which substitute the Norton equivalent of the AC side by a pure current source ( $g_1 = 0$ ), discard the Blocked mode of the rectifier because they force  $m_1$  to infinity. Such an approximation is not problematic if the slope  $m_1$  is very large but can considerably degrade the simulation results for other parameters [21].



Figure 3.3: Single-phase rectifier: Mapping history currents to diode status [21]

#### **3.3.2** General Mathematical Formulation of the DMM

The mathematical formulation of the DMM will be presented in this section. Assume each switch combination  $\sigma$  is composed of controlled switch combination  $\sigma_c$ , and diode combination  $\sigma_d$ . It is then convenient to define the sets of all possible switch combinations  $\Sigma$ ,  $\Sigma_c$ , and  $\Sigma_d$ , such that:  $\sigma \in \Sigma$ ,  $\sigma_c \in \Sigma_c$ ,  $\sigma_d \in \Sigma_d$ , and  $\Sigma = \Sigma_c \times \Sigma_d$ .

DMM function  $f^{\sigma_c}$  defines for a known controlled switch combination  $\sigma_c$  maps known input and state variables  $\mathbf{z}(t)$  to a diode switch combination  $\sigma_d$ :

$$f^{\sigma_c}: \mathbf{z}(t) \mapsto \sigma_d \tag{3.25}$$

Switch voltage vector  $\mathbf{v}_{\mathbf{d}}$  can be obtained using the following equation:

$$\mathbf{v}_{\mathbf{d}} = \mathbf{Q}^{\sigma} \mathbf{z}(t) \tag{3.26}$$

In (3.26),  $\mathbf{Q}^{\sigma}$  can be obtained using both MANA and state space formalism. The DMM function is then obtained from solving multiple systems of linear inequalities whose purpose is to determine if the diodes conduct or not. Hence, the polarity of the diodes is determined from:

$$\mathbf{S}(\sigma_d)\mathbf{Q}^{\sigma}\mathbf{z} < \mathbf{0} \tag{3.27}$$

where  $\mathbf{S}(\sigma_d)$  is a diagonal matrix with either -1 or +1 along its diagonal, respectively when the associated diode is either *on* or *off*. FME algorithm [74, 75] is used to assess the feasibility of each set. The set of feasible diode combinations  $\Sigma_{fd}^{\sigma_c} \subseteq \Sigma_d$  is comprised of switch combinations  $\sigma_{fd}$  for which (3.27) is feasible:

$$\Sigma_{fd}^{\sigma_c} = \left\{ \sigma_{fd} \in \Sigma_d \mid \mathsf{C}^{\sigma_c}(\sigma_{fd}) \neq \emptyset \right\}$$
(3.28)

$$\mathsf{C}^{\sigma_c}(\sigma_{fd}) = \{ z \in \mathbb{R}^n \mid \mathbf{S}(\sigma_d) \mathbf{Q}^{\sigma} \mathbf{z} < \mathbf{0} \}$$
(3.29)

Each feasible switch combination is associated with a convex set  $C^{\sigma_c}(\sigma_{fd})$  of  $\mathbf{z}(t)$ . The set is shaped by circuit parameters and the simulation time-step  $\Delta t$ . Certain converter topologies bind the logic of subsets of switches, yielding redundant inequalities in (3.27). Removing the redundant inequalities results in a reduced system:  $\mathbf{W}^{\sigma_c}(\sigma_{fd})\mathbf{z} < \mathbf{0}$ , where  $\mathbf{W}^{\sigma_c}(\sigma_{fd})$  rows are comprised of feasible combination hyper-planes. The convex set  $C^{\sigma_c}(\sigma_{fd})$  is a cone delimited by hyper-planes passing through the origin and given by  $\mathbf{W}^{\sigma_c}(\sigma_{fd})\mathbf{z} = \mathbf{0}$ .

Now, let  $\mathbf{W}^{\sigma_c}$  be a matrix obtained by extracting linearly independent rows from the matrices  $\mathbf{W}^{\sigma_c}(\sigma_{\rm fd})$  over all  $\sigma_{\rm fd} \in \Sigma_{\rm fd}^{\sigma_c}$  and let  $\mathbf{p}^{\sigma_c}(\mathbf{z})$  be:

$$\mathbf{p}^{\sigma_c}(\mathbf{z}) = \operatorname{sgn}(\mathbf{W}^{\sigma_c}\mathbf{z}) \tag{3.30}$$

where sgn(.) is the sign function. Then, by construction, we get that  $\forall \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n, \exists ! \sigma_{fd} \in \Sigma_{fd}^{\sigma_c}$ :

$$\left(\mathbf{p}^{\sigma_{c}}(\mathbf{z}_{1}) = \mathbf{p}^{\sigma_{c}}(\mathbf{z}_{2})\right) \Longrightarrow \left(\mathbf{z}_{1}, \mathbf{z}_{2} \in \mathsf{C}^{\sigma_{c}}(\sigma_{fd})\right)$$
(3.31)

Hence, there exists a surjective function  $\sigma_{fd} = \varphi^{\sigma_c}(\mathbf{p}^{\sigma_c}(\mathbf{z}))$  such that:

$$f^{\sigma_c}(\mathbf{z}) = \varphi^{\sigma_c}(\mathbf{p}^{\sigma_c}(\mathbf{z}))$$
(3.32)

### 3.3.3 Building the DMM Function

Building the DMM function is achieved by solving multiple systems of linear inequalities using the FME [74], which determines if a system of inequalities given by (3.27) is feasible. Further details on the FME can be found in the Appendix.

Recall that the single-phase rectifier circuit which admits 4 feasible switch combinations, i.e.,  $\Sigma_{fd}^d = \{0, 6, 9, 15\}$ . Each switch combination is associated with a cone and each cone is delimited by two lines (hyper-planes), see Figure 3.4.

In this case, Eq. (3.30) becomes as:

$$\mathbf{p}(\mathbf{z}) = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \operatorname{sgn} \left( \begin{bmatrix} +m_1 & -1 \\ -m_1 & -1 \\ +m_2 & -1 \\ -m_2 & -1 \end{bmatrix} \begin{bmatrix} i_1^h \\ i_2^h \end{bmatrix} \right)$$
(3.33)

and  $\sigma_{fd} = \phi(\mathbf{p}(\mathbf{z}))$  is:

$$\varphi(\mathbf{p}(\mathbf{z})) = \begin{cases}
0 & \text{when } (p_1, p_2) = (-1, -1) \\
6 & \text{when } (p_2, p_3) = (+1, -1) \\
9 & \text{when } (p_1, p_4) = (+1, -1) \\
15 & \text{when } (p_3, p_4) = (+1, +1)
\end{cases}$$
(3.34)



Figure 3.4: Single-phase rectifier DMM function. Representation of cone and hyper-plane

A DMM function is implemented by combination of a TCAM and a RAM. A content addressable memory (CAM) takes contents as input and returns the address of the found location. When the data in memory represents don't care entries, the memory is referred to as ternary CAM (TCAM) [76]. Table 3.1 outlines the TCAM-RAM content for this DMM function. The first five columns are associated with the TCAM part of the  $\varphi(\mathbf{p}(\mathbf{z}))$  function, which takes content ( $p_i$ values,  $1 \le i \le 4$ ) as an input and returns a memory location (*addr*) as an output. The last two columns are associated with the RAM part of the  $\varphi(\mathbf{p}(\mathbf{z}))$  function and returns for each address (*addr*) a switch combination  $\sigma_{fd}$ .

Table 3.1: TCAM-RAM for the single-phase rectifier	

$p_1$	<i>p</i> <sub>2</sub>	$p_3$	$p_4$	addr	$\sigma_{fd}$
-1	-1	-	-	0	0
-	+1	-1	-	1	6
+1	-	-	-1	2	9
-	-	+1	+1	3	15
		RA	M		

Algorithm 1 presents the process used to build the DMM function and fill the TCAM-RAM. It takes as an input the matrix  $\mathbf{S}(\sigma_d)\mathbf{Q}^{\sigma}$  and returns TCAM addr =TCAM $^{\sigma_c}(\mathbf{p})$  and a RAM filled with feasible switch combinations  $\sigma_{fd}$ =RAM $^{\sigma_c}(addr)$ , as well as the  $\mathbf{W}^{\sigma_c}$  matrix needed to evaluate Eq.(3.30). The Algorithm relies on the following subroutines [77]:

- [feasible, W] ← FME(M) implements the FME algorithm and determines if the system of inequalities Mz(t) < 0 is feasible; if it is, it returns the reduced matrix W, that is M redacted of redundant rows.</li>
- [W<sub>1</sub>, i<sub>1</sub>, i<sub>2</sub>] ← append(W<sub>1</sub>, W<sub>2</sub>) takes two input matrices W<sub>1</sub> and W<sub>2</sub>, and appends to W<sub>1</sub> rows from W<sub>2</sub> that are linearly independent from all rows in W<sub>1</sub>. i<sub>1</sub> are row positions in W<sub>1</sub> that have redundant equivalents in W<sub>2</sub>; i<sub>2</sub> are row positions that have been added to W<sub>1</sub>.
- TCAM<sup>σ<sub>c</sub></sup> ← addEntry(TCAM<sup>σ<sub>c</sub></sup>, addr, i<sub>1</sub>, i<sub>2</sub>) adds a new entry to the TCAM associated with address. The key **p** for the new entry reads as follows: position p<sub>i</sub> = +1 (i ∈ i<sub>1</sub>), position p<sub>j</sub> = −1 (j ∈ i<sub>2</sub>), remaining positions are don't cares.

Algorithm 1: Build DMM function
$addr \leftarrow 0$
$\mathbf{W}^{\sigma_c} \leftarrow []$
$TCAM^{\sigma_c} \leftarrow []$
foreach $\sigma_d \in \Sigma_d$ do
$[feasible, W] \leftarrow FME(M)$
if feasible then
$\sigma_{fd} \leftarrow \sigma_d$
$[W_1, i_1, i_2] \leftarrow append(W_1, W_2)$
$TCAM^{\sigma_c} \leftarrow addEntry(TCAM^{\sigma_c}, addr, i_1, i_2)$
$\mathbf{RAM}^{\sigma_c}(addr) \leftarrow \sigma_{fd}$
addr = addr + 1
end
end
return TCAM $\sigma_c$ , RAM $\sigma_c$ , $\mathbf{W}^{\sigma_c}$

## 3.4 Case Study: 500 kHz LLC DC-DC Resonant Converter

A 500 kHz LLC converter is assumed as the case study. After revisiting some fundamentals of LLC converters, DMM is developed for this circuit and its FPGA implementation results are presented.

#### 3.4.1 Overview of the LLC Resonant Converter

A typical DC-DC resonant converter is comprised of three stages: an inverter, a 2- or 3-element resonant tank, and a rectifier. The inverter consists of a MOSFET half-bridge or full bridge, operated with a fixed 50% duty cycle. The power flow of the converter is controlled by modulating the frequency of the square wave with respect to the tank's resonant frequency. The rectifier stage converts its AC input to a DC output voltage which is filtered by a capacitor. A transformer is often used in the rectification stage for scaling and isolation purposes [21].

The resonant tank modulates the voltage using LC elements such as series LC, parallel LC, LCC, LLC, etc. These resonant tanks are selected depending on the converter application [78]. The focus here is on the LLC converter topology which is comprised of a full-bridge inverter and a full-bridge rectifier, see Figure 3.5.a. The LLC converter has drawn attention for its unique characteristics such as low voltage stress on the secondary rectifier and high efficiency at high input voltages [79, 80]. It is used in many applications such as power electronic-based distributed generation, electric vehicles, computer and communication systems [81, 82], etc.

#### 3.4.1.1 LLC characteristic

The LLC converter is a multi-resonant converter with resonant frequencies  $f_r = f_{r1} = 1/(2\pi\sqrt{L_rC_r})$  and  $f_{r2} = 1/(2\pi\sqrt{(L_r + L_m)C_r})$ . With reference to Figure 3.5.a,  $L_m$  is embodied in the magnetizing inductance of the transformer which helps to reduce the size and the cost of the converter [83]. LLC resonant tank gain, G(F), is defined as the magnitude of its input–output voltage transfer function [21].

Typical voltage gains of the LLC tank for different normalized switching frequencies  $F = f_s/f_r$ , ( $f_s$  being the switching frequency) are obtained from [84]:

$$G(F) = \frac{(m-1)F^2}{\sqrt{(mF^2-1)^2 + (m-1)^2F^2(F^2-1)^2Q^2}}$$
(3.35)



Figure 3.5: (a): LLC converter used as the case study; and (b): voltage gain of LLC tank versus *F* for different quality factors [21]

where Q is the quality factor, and m is the inductance ratio:

$$\begin{cases} Q = (\sqrt{L_r/C_r})/R_{ac} \\ m = (L_r + L_m)/L_r \\ R_{ac} = (8n^2/\pi^2)R_L \end{cases}$$
(3.36)

 $R_{ac}$  denotes the reflected load resistance  $R_L$ , seen from the resonant tank side, and *n* is the turn ratio of the isolation transformer [21].

#### 3.4.1.2 LLC Operation

Figure 3.5.b depicts the LLC voltage gain for different values of quality factor Q, which are used for the analysis and design of the resonant converter.  $f_{r_2}$  delimits the capacitive and the inductive regions of the resonant tank, which are associated with the zero current switching (ZCS) and zero voltage switching (ZVS) regions, respectively. The MOSFET-based LLC converters are preferably operated in the ZVS region, i.e.,  $f_s > f_{r_2}$ , since it significantly mitigates the switching losses [85]. When LLC is working with  $f_{r_2} < f_s < f_{r_1}$ , at the end of one-half resonant cycle, during a certain time interval no power is delivered to the load and the whole resonant current passes through the magnetizing inductance. During this time interval, the rectifier is said to be in blocking mode  $(i_{in}^{rec} = n|i_r - i_m| \approx 0)$ . The region where  $f_s > f_{r_1}$  refers to an operational mode of the LLC where a resonant half cycle is not completed and interrupted by the switching of other MOSFET [21].

#### 3.4.1.3 LLC Simulation

The LLC converter of Figure 3.5.a is considered for which two sets of parameters are chosen from the literature as listed in Table 3.2. In this table, LLC with Parameter Set #1 [86] and Parameter Set #2 [87] has the resonant frequency of 160 kHz, 500 kHz, respectively. It will be shown hereafter why an iterative solution is needed to accurately simulate the LLC converter.

An explicit integration method such as FE can be used to avoid the iterative process. This however necessitates adopting a sufficiently small simulation time-step. Recall the forward Euler method which offers a recursive solution to  $\dot{x}(t) = f(x(t))$  in the following form:

$$x(t) = x(t - \Delta t) + \Delta t f(x(t - \Delta t)).$$
(3.37)

Parameter	$f_{r_1}(kHz)$	$f_{r_2}(kHz)$	n: 1	$V_{in}(V)$	$V_{out}(V)$	$L_r(\mu H)$	$L_m(\mu H)$	$C_r(\mu F)$	$C_o(\mu F)$	P(kW)	$R_L(\Omega)$
Parameter											
Set #1	160	60	1.5:1	600	400	25	150	40	1000	5.3	30
[86]											
Parameter											
Set #2	500	210	33:1	400	12	4.5	21.6	22	3000	1	0.144
[87]											

Table 3.2: Circuit parameters for LLC Circuits [21]

By applying Eq. (3.37) to the resonant inductors of the resonant tank of the LLC, an explicit expression for the rectifier's input current is obtained (see Figure 3.5.a):

$$i_{in}^{rec}(t) = n\{i_r(t) - i_m(t)\}$$
  
=  $n\{i_r(t - \Delta t) + \frac{\Delta t}{L_r}v_{L_r}(t - \Delta t) + i_m(t - \Delta t) + \frac{\Delta t}{L_m}v_{L_m}(t - \Delta t).$  (3.38)

If the sign of input current is used to determine the mode of the full-bridge rectifier; positive mode when  $i_{in}^{rec}(t) > 0$ ,  $(d_1, d_2, d_3, d_4) = (1, 0, 0, 1)$ , and negative mode otherwise —

 $(d_1, d_2, d_3, d_4) = (0, 1, 1, 0)$ . However, such an approach may lead to either inaccurate results or instability, even for very small time-steps. This behavior is partly because this method completely ignores the blocking mode of the converter [21]. Figure 3.6.a gives the resonant current,  $i_r$ , for LLC with Parameter Set #1, calculated by either making use of FE or backward Euler (BE) methods, and assuming a time-step of 15 ns. It is noted that BE method is applied through an iterative solution and in FE case the mode of rectifier is determined based on the sign of  $i_{in}^{rec}$ . The zoomed view of resonant current during the blocking mode is shown in Figure 3.6.b. It is seen from this figure that in contrast with BE method, the current  $i_r$  calculated by the FE is oscillatory during the blocking mode period. The blocking mode is in fact disregarded during the simulation i.e., only positive, and negative modes are considered, and the model alternates between the two modes during the blocking period.

The effects of ignoring the blocking mode for LLC with Parameter Set #2 are observed in Figure 3.6.c and Figure 3.6.d. Similar to the previous case,  $i_r$  is calculated by either BE or FE with the same 15 ns time-step. It is seen from these figures that  $i_r$  calculated by FE is oscillatory during the blocking mode. It is also seen from Figure 3.6.c and Figure 3.6.d that, as opposed to LLC with Parameter Set #1, the results of FE and BE are quite different for LLC with Parameter Set #2. This indicates that the accuracy of the method which uses FE while considering only positive and negative modes of rectifier can be affected by circuit parameters [21].

In LLC with Parameter Set #2, the resonant current calculated by FE method becomes unstable for  $\Delta t = 30 \text{ ns}$ . It is also worth noting that another possible mode of the rectifier stage of the LLC circuit of Figure 3.5.a is the short circuit mode, i.e.,  $(d_1, d_2, d_3, d_4) = (1, 1, 1, 1)$  which is also disregarded by explicit methods.



Figure 3.6: LLC with Parameter Set#1: (a): Resonant current of the LLC converter calculated by FE and BE methods, (b): Close-up view of resonant current during blocking mode. LLC with Parameter Set#2: (c): resonant current of the LLC converter calculated by FE and BE methods, (d): Close-up view of resonant current during blocking mode. In all cases an identical time-step of 15ns is set [21]

#### 3.4.2 Simulation Algorithm

To develop a simulation algorithm for the LLC converter of Figure 3.5.a, it is assumed that the inverter is operated in controlled mode, and DMM is used to determine the diode status.

Network equations for the LLC converter of Figure 3.5.a are formulated using the MANA [3] after discretizing all elements using BE rule.

$$\mathbf{A}^{\sigma}\mathbf{x}(t) = \mathbf{B}[u(t), \mathbf{i}_{ac}^{hist}(t), \mathbf{i}_{dc}^{hist}(t)]$$
(3.39)

where  $\mathbf{A}^{\sigma}$  is the MANA matrix for switch combination  $\sigma$ , **B** is the incidence matrix and  $\mathbf{x}(t)$  is the vector of unknown variables.  $u(t) = v_{dc}(t)$  is the input DC voltage,  $\mathbf{i}_{ac}^{hist}(t)$ , and  $\mathbf{i}_{dc}^{hist}(t)$  are ac and dc side history vectors, respectively. Vector  $\mathbf{x}(t)$  is comprised of node voltages  $\mathbf{v}_n(t)$ , the current entering the DC voltage source  $i_{DC}(t)$  and the current entering the secondary port of the transformer  $i_{DB}(t) = -i_{in}^{rec}(t) : \mathbf{x}(t) = [\mathbf{v}_n(t), i_{DC}(t), i_{DB}(t)]$ . The history vectors are comprised of the history currents that result from the backward discretization of the L/C components in the circuits:  $\mathbf{i}_{ac}^{hist}(t) = [i_{C_r}^{hist}(t), i_{L_r}^{hist}(t)]$ , and  $\mathbf{i}_{dc}^{hist}(t) = i_{C_f}^{hist}(t)$  [21].

To reduce the computational burden of solving Eq. (3.39) at each time-point of the simulation, the following formulation is used:

$$\begin{bmatrix} \mathbf{i}_{ac}^{hist}(t+\Delta t) \\ \mathbf{i}_{dc}^{hist}(t+\Delta t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{ac,u}^{\sigma(t)} & \mathbf{H}_{ac,ac}^{\sigma(t)} & \mathbf{H}_{ac,dc}^{\sigma(t)} \\ \mathbf{H}_{dc,u}^{\sigma(t)} & \mathbf{H}_{dc,ac}^{\sigma(t)} & \mathbf{H}_{dc,dc}^{\sigma(t)} \\ \mathbf{H}_{y,u}^{\sigma(t)} & \mathbf{H}_{y,ac}^{\sigma(t)} & \mathbf{H}_{y,dc}^{\sigma(t)} \end{bmatrix} \begin{bmatrix} u(t) \\ \mathbf{i}_{ac}^{hist}(t) \\ \mathbf{i}_{dc}^{hist}(t) \end{bmatrix}$$
(3.40)

where  $\mathbf{H}_{-,-}^{\sigma(t)}$  are precomputed matrices obtained from simple algebraic manipulations of **B** and inverse of  $\mathbf{A}^{\sigma}$ .  $\mathbf{y}(t)$  is a vector comprised of desired output variables,  $\mathbf{y}(t) = [v_o(t), i_r(t), i_m(t)]$ . Similar rewritings are used to determine the history currents associated with the Norton equivalents of the ac and dc sides:

$$\begin{cases} i_{1}^{h}(t) = \mathbf{H}_{1,u}^{\sigma_{inv}(c(t))} u(t) + \mathbf{H}_{1,ac}^{\sigma_{inv}(c(t))} \mathbf{i}_{ac}^{hist}(t) \\ i_{2}^{h}(t) = \mathbf{H}_{2} \mathbf{i}_{dc}^{hist}(t) \end{cases}$$
(3.41)

where  $\sigma_{inv}$  is the switch combination associated with the inverter's state, which is defined as a function of the gating signal c(t) driving  $s_1$  and  $s_4$  (see Figure 3.5.a), given the controlled operation of the inverter:

$$\sigma_{inv}(c(t)) = \begin{cases} 6, & \text{if } c(t) = 0\\ 9, & \text{if } c(t) = 1 \end{cases}$$
(3.42)

The DMM function is used to determine  $\sigma_{rec}$  and  $\sigma$ :

$$\sigma_{rec}(t) = f\left(i_1^h(t), i_2^h(t)\right), \sigma_{rec}(t) \in \{0, 6, 9, 15\},\tag{3.43}$$

$$\sigma(t) = 16\sigma_{inv}(c(t)) + \sigma_{rec}(t), \qquad (3.44)$$

From the mathematical formulation given above, the following algorithm is used at each time-point to simulate the LLC [21]:

- Determine  $\sigma_{rec}(t)$  using Eqs. (3.41), (3.33) and (3.34).
- Determine  $\sigma(t)$  using Eqs. (3.42) and (3.44).

Compute the output vector y(t) as well as the history vectors i<sup>hist</sup><sub>ac</sub>(t) and i<sup>hist</sup><sub>dc</sub>(t) for the next time-point using Eq. (3.40).

This algorithm is referred to as the two-stage algorithm because it involves two matrix-vector multiplications (MVMs): Step 1) is performed by combining Eqs. (3.41) and (3.33) into a single MVM following which Eq. (3.34) is used to determine  $\sigma_{rec}(t)$ . Step 2) is a simple binary word concatenation and is therefore instantaneous. Step 3) constitutes the second stage MVM of the algorithm [21].

### 3.4.3 Low-latency Implementation of the DMM

Figure 3.7.a illustrates. a dataflow diagram for the two-stage algorithm and shows the data dependency of the two stages (through  $\sigma(t)$ ) that forces their serial execution. Moreover, each stage involves MVMs that hinders reaching to a small time-step.



Figure 3.7: Data-flow diagram: (a): two-stage algorithm; (b): single-stage algorithm [21]

It is possible to reduce the time-step if Eqs. (3.41) are rewritten in the following form:

$$i_{1}^{h}(t) = \mathbf{H}_{1,u}^{\sigma_{inv}(c(t))} \mathbf{H}_{1,u}^{\sigma_{inv}(c(t))} u(t) + \mathbf{H}_{1,ac}^{\sigma_{inv}(c(t))} \mathbf{H}_{ac,u}^{\sigma(t-\Delta t)} u(t-\Delta t) + \mathbf{H}_{1,ac}^{\sigma_{inv}(c(t))} \mathbf{H}_{ac,ac}^{\sigma(t-\Delta t)} \mathbf{i}_{ac}^{hist}(t-\Delta t) + \mathbf{H}_{1,ac}^{\sigma_{inv}(c(t))} \mathbf{H}_{ac,dc}^{\sigma(t-\Delta t)} \mathbf{i}_{dc}^{hist}(t-\Delta t), i_{2}^{h}(t) = \mathbf{H}_{2} \mathbf{H}_{dc,u}^{\sigma(t-\Delta t)} u(t-\Delta t) + \mathbf{H}_{2} \mathbf{H}_{dc,ac}^{\sigma(t-\Delta t)} \mathbf{i}_{ac}^{hist}(t-\Delta t) + \mathbf{H}_{2} \mathbf{H}_{dc,dc}^{\sigma(t-\Delta t)} \mathbf{i}_{dc}^{hist}(t-\Delta t),$$

$$(3.45)$$

According to (3.45), knowing  $u(t - \Delta t)$ ,  $\sigma(t - \Delta t)$ ,  $\mathbf{i}_{ac}^{hist}(t - \Delta t)$ ,  $\mathbf{i}_{dc}^{hist}(t - \Delta t)$ , u(t), and c(t) would suffice to determine  $\sigma(t)$ . Figure 3.7.b illustrates how this rewriting yields a single-stage algorithm version of the DMM by breaking the data dependency. The only limitation of this approach is that two consecutive simulation steps are needed to produce the output vector  $\mathbf{y}(t)$ . The hardware implementation of the single-stage results in an input-output latency of two timesteps, but that it is legitimate since the simulation time-step is halved compared to the two-stage version of the algorithm [21].

## 3.5 FPGA Implementation of DMM-Based LLC Simulator

### **3.5.1 Hardware Implementation**

A hardware implementation of the single-stage DMM algorithm datapath is shown in Figure 3.8. The architecture is almost a one to one map of the dataflow diagram of Figure 3.7.b, and consists of two main computing units: a) The first unit is devoted to updating history terms and computing outputs of interest, i.e., Eq. (3.40); b) The second unit evaluates the DMM function, i.e., Eqs. (3.33), (3.34), (3.44) and (3.45). Eq. (3.40) is implemented by a dedicated MVM module; Eqs. (3.33) and (3.34) are combined and implemented by a dedicated MVM module as well. Eq. (3.34) is a simple lookup table. Eq. (3.44) consists of a bit string concatenation and comes at no hardware cost [21].



Figure 3.8: Datapath of the hardware implementation of the LLC simulator using Single-Stage DMM Algorithm [21]

Each MVM module from Figure 3.8 is shown in Figure 3.9. The MVM module implements the multiplication of an  $n \times m$  matrix by an  $m \times 1$  vector and is made up of a set of  $N \times m$  read-only memories (ROMs) and N *m*-input dot-product (DP) units, where  $N \leq n$  is a parallelization parameter. The two MVM modules of the LLC simulator of Figure 3.8 handle  $n_1 \times m_1 = 7 \times 5$  and  $n_2 \times m_2 = 4 \times 6$  matrices, and as such are defined by parameters ( $N_1, m_1$ ) and ( $N_2, m_2$ ) [21].

### 3.5.2 Number Format

An FPGA can handle real arithmetic using either fixed-point (FXP) or floating-point (FP) format. The FXP format uses less hardware and yields a datapath of lower latency but has a limited dynamic range. The FP number format allows for a larger dynamic range, but its hardware arithmetic operators are costly in terms of FPGA resource consumption and require deeper pipelines than their FXP counterparts. Due to latency considerations, in this implementation solely the FXP number format is considered. The limited dynamic range issue of the FXP format is addressed by normalizing matrix entries using a per-unit scale. The targeted FPGA, a Kintex K325T from Xilinx, uses an asymmetric multiplication block ( $25 \times 18$  signed). Hence, the selected number format used by the LLC simulator will be different whether we are dealing with a vector (FXP 25.23) or a matrix (FXP 35.29) in order to offer a good precision to the precomputed matrices and good computation accuracy, while reducing the area footprint of the simulator [21].

#### 3.5.3 Design Space Exploration

The purpose of design space exploration to evaluate the impact of parameters  $N_1$  and  $N_2$  on area occupation, and simulation time-step. Two implementations are considered in this design exploration, the first approach consists in a fully parallel implementation (FPI) that sets  $N_1 = n_1 = 7$ , and  $N_2 = n_2 = 4$ ; whereas the second approach will resort to a hardware reuse technique (HRT) that consists in setting  $N_1 < n_1$  and  $N_2 < n_2$ , thus time-multiplexing the computations. Here, the adopted HRT approach sets  $N_1 = 1$  and  $N_2 = 1$  [21].



Figure 3.9: The (N, m) MVM module used in the proposed LLC simulator: The module is composed of  $N \times m$  ROMs and N *m*-Input DP units [21]

The effect of the FPGA clock frequency on the timing performance and simulation time-step is assessed. Various clock frequencies are investigated, namely 40 MHz, 200 MHz and 320 MHz. The timing closure is met by varying the pipelining depth of the DP unit accordingly. When the 40 MHz clock frequency is targeted, the DP unit is purely combinational [21].

Table 3.3 presents the design exploration results for the Kintex K325T for each design approach and each target frequency. It shows that the smallest simulation time-steps (25 ns) are obtained when a FPI approach is adopted. FPI is however the most expensive approach in terms of hardware consumption. The HRT trades simulation time-step for area footprint. Hence, this approach allows considerable saving in hardware utilization (up to 4 folds), with a very acceptable impact on the simulation time-steps, which are less or equal to 100 ns. For HRT, smaller time-steps are achieved by increasing the clock frequency, with the smallest time-step (34.375 ns) obtained for the highest clock frequency (320 MHz). The 200 MHz implementations are the most balanced options for both FPI and HRT approaches.

That being said, all reported implementations are characterized by a very low hardware utilization, a small simulation time-step ( $\leq 100 \text{ ns}$ ), and an input-output (In-Out) latency of two time-steps. These notable results are more obvious from, where the 200 MHz FPI and HRT results have been reproduced, the proposed implementations have indeed a very small footprint and offer one of the smallest time-steps ever reported in the literature. It is noteworthy that, thanks to the proposed DMM, these outcomes are obtained using an implicit solver, without decoupling any part of the circuit, and while using a simultaneous and exact switch state solution.

### 3.5.4 Computational Accuracy

The computational accuracy of the hardware LLC simulator (200 MHz FPI) is assessed through a test sequence lasting 0.6s. LLC with Parameter Set #2 is considered. During the test sequence, the input voltage is kept constant at 400 V. The FPGA simulation results are validated against an offline iterative solution taken from EMTP, as shown in Figure 3.10, where the output voltage,  $v_o$ , and the switching frequency are shown [21].

	Item	40 MHz	200 MHz	320 MHz
	Minimum Time-Step	25 ns	25 ns	25 ns
	Input-Output Latency	50 ns	50 ns	50 ns
Ī	Registers	1,224 (0.3%)	1,697 (0.4%)	2,373 (0.6%)
H	LUTs	1,090 (0.5%)	1,095 (0.5%)	1,223 (0.6%)
	DSP Blocks	88 (10.5%)	88(10.5%)	88 (10.5%)
	BRAMs	22 (4.9%)	22 (4.9%)	22 (4.9%)
	Minimum Time-Step	100 ns	40 ns	34.375 ns
HRT	Input-Output Latency	200 ns	80 ns	68.750 ns
	Registers	211 (0.1%)	350 (0.1%)	498 (0.1%)
	LUTs	277 (0.1%)	329 (0.2%)	311 (0.2%)
	DSP Blocks	22 (2.6%)	22 (2.6%)	22 (2.6%)
	BRAMs	5.5 (1.2%)	5.5 (1.2%)	5.5 (1.2%)

Table 3.3: Design exploration targeting the Kintex K325T [21]

The test sequence comprises the following steps:

- t = 0 s-0.1 s: Operation of the inverter at  $f_s = 312.5$  kHz.
- $t = 0.1 \text{ s-}0.13 \text{ s: Load is shorted (fault); } f_s = 312.5 \text{ kHz.}$

- t = 0.13 s-0.25 s: Fault cleared;  $f_s = 312.5$  kHz.
- t = 0.25 s-0.48 s: The switching frequency  $f_s$  is increased from 312.5 kHz to 500 kHz.
- t = 0.48 s-0.6s: Operation at  $f_s = 500$  kHz.

Table 3.4: Comparison with the existing work on the FPGA-based real-time simulation of resonant converter [21]

		Power e	electronic circuit		FPGA Resource Consumption						
	$\Delta t(ns)$	Solver	Application	Sw. Freq. (kHz)	FPGA	Clk. (MHz)	Number Format	LUTs	Registers	BRAMs	DSP 48
FPI	25	BE	LLC	500	K7-325 <sup>1</sup>	320	FXP 32.29/25.22	1,095 (0.5%)	1,697 (0.4%)	22 (4.9%)	88 (1.05%)
HRT	40	BE	LLC	500	K7-3251	320	FXP 32.29/25.22	329 (0.2%)	350 (0.1%)	5.5 (1.2%)	22 (2.6%)
[86]	15	FE	Batt.Charger	160	K7-410 <sup>2</sup>	66.67	FXP 25.23	45,560 (17.9%)	44,277 (8.7%)	106 (13.3%)	43 (2.8%)
[35]	36	BE	AC-DC-AC	50	V7-485 <sup>3</sup>	200	$SFP^4$	116,670 (38%)	68,920 (11%)	1,202 kb (3%)	762 (27%)
[88]	100	FE	LLC	60	NR <sup>5</sup>	NR	NR	NR	NR	NR	NR
[63]	40	PC:BE- FE	NPC	NR	K7-410	25	32	47,028 (25.8%)	42,642 (11.2%)	91 (11.6%)	120 (17.7%)
[89]	40	Sw.fct+ FE	Batt.Charger	100	ZYNQ	200	FXP 32.20	NR	NR	NR	NR
[34]	50	BE	3ph-Inverter	200	V5-50 <sup>6</sup>	200	FXP 35.30/25.16	229 (7%)	3,531 (10.8%)	44 (33.3%)	176 (61.1%)
[14]	50	Sw.fct+ FE	3ph-Inverter	100	V7-485 <sup>3</sup>	200	FXP 72.43	3,943 (1.3%)	893 (0.1%)	NR	170 (6.1%)
<sup>1</sup> K7-32 VC5VS	<sup>1</sup> K7-325: Kintex XC7K325T. <sup>2</sup> K7-410: Kintex XC7K410T. <sup>3</sup> V7-485: Virtex XC7VX485T. <sup>4</sup> SFP: Single precision. <sup>5</sup> NR: Not Reported. <sup>6</sup> V5-50: Virtex VC5VSX50T										



Figure 3.10: LLC with Parameter Set #2: (a): Output voltage, (b): Switching frequency and fault period [21]

From Figure 3.10, one can see that the  $v_o$  gradually decreases and finally settles at ~12 V as  $f_s$  is increased from 312.5 kHz to  $f_s = 500$  kHz. Figure 3.11 offers close-up views of four instants, as identified in Figure 3.10. The FPGA results are overlapped in the same figure and show very good agreements with the reference. At 500 kHz, the output voltage shows a slight dc offset that is not larger that 0.2%.



Figure 3.11: LLC with Parameter Set #2: Close-up views of output voltage and resonant current, magnetizing current. (a)-(b): Close-up view 1; (c)-(d): close-up view 2; (e)-(f): close-up view 3; and (g)-(h): close-up view 4. Output voltage close-up views are shown in a wider timespan than resonant and magnetizing currents [21]

To further assess the accuracy of the FPGA result, each signal is compared to its offline reference. The 2-norm relative error is used as a measure of accuracy [90]:

$$e = \frac{\|\mathbf{f}_s - \mathbf{f}_r\|_2}{\|\mathbf{f}_r\|_2}$$
(3.46)

where  $\mathbf{f}_s$  and  $\mathbf{f}_r$  are FPGA results and offline reference, respectively. Table 3.5 reports the 2-norm errors for the FPGA results for each sub-sequence as well as for the entire test sequence.

All the reported errors are below 0.5% and show the very good performance of the FPGA implementation in different operating mode and condition of the LLC.

Sequence	vo	i <sub>r</sub>	i <sub>m</sub>
00s-0.10s	0.024%	0.383%	0.12%
0.10s-0.13s	0.001%	0.001%	0.016%
0.13s-0.25s	0.021%	0.338%	0.131%
0.25s-0.48s	0.006%	0.391%	0.131%
0.48-0.60s	0.023%	0.499%	0.048%
00s-0.60s	0.018%	0.336%	0.124%

Table 3.5: LLC with parameter Set #2 [87]: 2-Norm Relative Errors

# 3.6 Resonant Boost Converter Case

To further assess the performance of the proposed method, a 100 kHz resonant boost converter (RBC) is considered, see Figure 3.12 [91]. Applying DMM algorithm yields up 10 feasible combinations and a  $14 \times 6 \mathbf{W}^{\sigma_c}$  matrix.



Figure 3.12: Resonant boost converter circuit [91]

In RBC circuit, the main switch  $s_1$  is turned *on* after a delay  $T_1$ . On the other side, the auxiliary switch  $s_2$  is turned *on* at the beginning of each switching cycle and turned off after a fixed time  $T_2$ . Figure 3.13 shows this switching pattern where  $T_1 = 0.54 \ \mu s$  and  $T_2 = 1.43 \ \mu s$  [91].



Figure 3.13: Resonant boost converter circuit-PWM signals

Figure 3.14 shows resonant inductor waveform for a short period of simulation time. In this figure, the waveform obtained from DMM is overlapped with the reference waveform from the EMTP and shows very good agreement.



Figure 3.14: Resonant boost converter circuit-resonant inductor current waveform

For this case, like the LLC converter, two implementation approaches are followed, namely, double-stage and single-stage DMM. Similar hardware architecture as LLC simulator is designed for these implementations. Due to latency considerations, a 35.32 FXP number format is considered. To address FXP limited dynamic range issue, matrix entries are normalized based on per-unit scales.

Table 3.6 reports resource consumption of these two implementation approaches. As can be seen in Table 3.6, double-stage and single-stage implementations could achieve time-steps of 100 ns and 62.5 ns, respectively. However, single-stage version consumes more FPGA resources compared to its double-stage counterpart where it requires approximately 1.5 times more BRAMs and 1.75 times more registers.

Table 3.6: FPGA (Kintex K325T) resource consumption of RBC implementation

	Time-step	LUTs	Registers	BRAMs	DSP Blocks
Double-stage	100 ns	2091 (1.02%)	4173 (1.6%)	29 (6.5%)	216 (25.7%)
Single-stage	62.5 ns	2453 (1.2%)	7202 (1.7%)	45 (10.11%)	232 (27.6%)

Table 3.7 shows the FPGA computational accuracy of 5 state variables of the RBC circuit. It is seen that the 2-norm relative errors are below 0.06%.

Table 3.7: RBC circuit- FPGA computational accuracy

Variable	$i_g$	i <sub>r</sub>	$v_{C_s}$	$v_{C_r}$	$v_o$
%2-norm error	8.25e-3	8.03e-3	5.6e-2	2.22e-2	9.12e-3

Based on this implementation, a physical setup is built to perform real-time test. In this setup, a Zynq-7000 SoC ZC706 board from Xilinx is used. It is worth mentioning that this board has a logical part equivalent to the K325T Kintex. Figure 3.15 presents an oscilloscope capture of the resonant current ( $i_r$ ) obtained by this setup. The oscilloscope indicates  $i_r$  frequency as being 99.993 kHz, which is very close to the 100 kHz switching frequency used [77].



Figure 3.15: RBC switched at 100 kHz - Real-time test result of resonant inductor current [77]

# 3.7 Conclusion

This chapter started by revisiting iterative procedure of simulating power electronic based circuit. To obviates the need for iterations and achieving small time-step of RTS, the DMM is introduced. The DMM renders an exact simulation result while rendering a deterministic computational burden. To demonstrate the effectiveness of the proposed method an FPGA-based simulator is developed for two case studies, namely, a 500 kHz LLC and a 100 kHz resonant boost circuit.

To decrease datapath latency of the proposed simulator, single-stage version of the DMM is proposed and utilized in the implementation. Using this approach, LLC simulator achieved a time-step of 25 ns. It has been shown that the DMM-based RTS simulator offers a very small hardware footprint. In the last section of this chapter, real-time test results of FPGA-based resonant boost circuit is reported. For this case, both double-stage and single-stage DMM versions are implemented, and time-steps of 100 ns and 62.5 ns are achieved.

# CHAPTER 4 DIRECT MAPPED METHOD FOR MULTI-CONVERTER CIRCUIT

# 4.1 Introduction

On-board power systems such as electric vehicle battery charger (EVBC) circuit and more electric aircraft are comprised of multiple HSF PECS. In this type of circuits, building the DMM function becomes difficult because the problem grows exponentially with the number of switches. Hence, offering an efficient and accurate RTS algorithm for a large circuit using DMM is a challenge.

Decoupling techniques such as single delay decoupling (SDD) and NTT, split a large circuit into multiple sub-circuits and give more parallelism. In such techniques, each sub-circuit is comprised of a subset of switches that can be dealt independently, thus reducing the memory requirements. These approaches are well suited to the FPGA-based computations because of its inherent parallel structure.

The main contribution of this chapter is to combine DMM with decoupling methods to enhance the scalability of the DMM for larger circuit. Regarding FPGA-based simulation, the contributions of this chapter are: i) to proposed FPGA implementation of DMM function, and ii) to design a low-cost FPGA hardware to perform RTS of multi-converter system. An EVBC test case is used to demonstrate the high accuracy achieved by the proposed methods. FPGA implementations are proposed and shown to achieve from 75 ns to 175 ns real-time simulation time-steps.

# 4.2 Decoupling Techniques

## 4.2.1 Brief Review of Methods from the Literature

The goal of diakoptics and decoupling is to distribute the computational burden of a large circuit simulation to multiple parallel processing units. Various approaches are presented in the literature which were reviewed in Chapter 2. NTT, MATE, artificial TL, using an explicit solver such as FE, introducing time-delay such as SDD are some examples of these techniques. Among them, SDD [92] and NTT [33] are shown their effectiveness in RTS applications. In the following sections, these two techniques are briefly reviewed.

#### 4.2.2 SDD

SDD is a decoupling technique used for RTS applications [92] which adds a time-step delay between decoupled sub-circuits. The effect of each sub-circuit is modeled with a voltage or current source inserted across the interface ports whose value is taken from the previous time-point.

To clarify the SDD technique, Figure 4.1 is considered. In this example, two sub-circuits are connected at points *a* and *b*, where a capacitor is connected too (see Figure 4.1.a). Figure 4.1.b shows the decoupled circuit using the SDD. SDD reflects subcircuit#1's impact on sub-circuit#2 with a voltage source obtained from the interface link voltage  $v_{ab}$  with a single time-step delay. On the other side, a current source is inserted across sub-circuit#1 to model sub-circuit#2's behaviour. The value of this current source obtained from the sub-circuit#2's input current  $i_{ab}$  with a single time-step delay.



Figure 4.1: SDD method scheme-(a): original circuit, and (b): decoupled

SDD is a straightforward approach and broadly used in the industrial practice, but it is susceptible to numerical errors and potential instability. In addition, using the SDD, the simulation accuracy may deteriorate during fault conditions. This will be more discussed later in this chapter.

#### 4.2.3 NTT

The NTT is a diakoptics which tears a large circuit into separate sub-circuits, and each sub-circuit is dealt with independently. Each Sub-circuit is reduced to its multi-port hybrid equivalent which is interconnected to other sub-circuits through its interface variables. Based on the original circuit topology, network equations are formed to solve for interface variables. The solution of each sub-circuit is found, provided that the interface variables are known. This solution is as accurate as when solving the whole circuit simultaneously since it is known from the compensation theorem that the interface variables reflect the exact solution of the rest of circuit [33, 36, 37].

# 4.3 Network Tearing Technique

### 4.3.1 NTT Formulation

Assume a resistive network is teared into N sub-circuits. Each sub-circuit k can be shown by its multi-port hybrid equivalent which stands for the fact that it includes both voltage ports and current ports. The multi-port hybrid equivalent of sub-circuit k is given as (4.1) [37]:

$$\begin{bmatrix} \mathbf{i}_{k}^{o} \\ \mathbf{v}_{k}^{o} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{iu}^{\sigma_{k}} & \mathbf{B}_{ih}^{\sigma_{k}} \\ \mathbf{B}_{vu}^{\sigma_{k}} & \mathbf{B}_{vh}^{\sigma_{k}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{k} \\ \mathbf{i}_{k}^{h} \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{iv}^{\sigma_{k}} & \mathbf{H}_{ii}^{\sigma_{k}} \\ \mathbf{H}_{vv}^{\sigma_{k}} & \mathbf{H}_{vi}^{\sigma_{k}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{k}^{i} \\ \mathbf{i}_{k}^{i} \end{bmatrix}$$
(4.1)

where  $\mathbf{i}_k^0$  (current of voltage port) and  $\mathbf{v}_k^0$  (voltage of current port) are output variables,  $\mathbf{i}_k^i$  (current of current port) and  $\mathbf{v}_k^i$  (voltage of voltage port) are imposed variables at sub-circuit k ports.  $\mathbf{u}_k$ and  $\mathbf{i}_k^h$  are respectively independent voltage and current sources and history terms; and matrices  $\mathbf{H}_{-k}^{\sigma_k}$  are the characteristic matrices of sub-circuit k for a given switch combination  $\sigma_k$ . The compact form of multi-port hybrid equivalent of (4.1) can be represented as:

$$\mathbf{w}_k^{\mathbf{o}} = \mathbf{w}_k^{\mathbf{h}} - \mathbf{H}_k^{\sigma_k} \mathbf{w}_k^{\mathbf{i}}$$
(4.2)

where:

$$\mathbf{w}_{k}^{\mathrm{o}} = \begin{bmatrix} \mathbf{i}_{k}^{\mathrm{o}} \\ \mathbf{v}_{k}^{\mathrm{o}} \end{bmatrix}, \quad \mathbf{w}_{k}^{\mathrm{i}} = \begin{bmatrix} \mathbf{v}_{k}^{\mathrm{i}} \\ \mathbf{i}_{k}^{\mathrm{i}} \end{bmatrix}$$
(4.3)

$$\mathbf{w}_{k}^{h} = \begin{bmatrix} \mathbf{i}_{k}^{eq} \\ \mathbf{v}_{k}^{eq} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{iu}^{\sigma_{k}} & \mathbf{B}_{ih}^{\sigma_{k}} \\ \mathbf{B}_{vu}^{\sigma_{k}} & \mathbf{B}_{vh}^{\sigma_{k}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{k} \\ \mathbf{i}_{k}^{h} \end{bmatrix}$$
(4.4)

$$\mathbf{H}_{k}^{\sigma_{k}} = \begin{bmatrix} \mathbf{H}_{iv}^{\sigma_{k}} & \mathbf{H}_{ii}^{\sigma_{k}} \\ \mathbf{H}_{vv}^{\sigma_{k}} & \mathbf{H}_{vi}^{\sigma_{k}} \end{bmatrix}$$
(4.5)

Based on generalized post-compensation method [37], at each time-point,  $\mathbf{w}_k^i$  reflects an exact and simultaneous contribution of the remainder of the circuits to sub-circuit k solution.



Figure 4.2: NTT concept

To solve for interface variables, each sub-circuit's equivalent from Eq. (4.2) is inserted into a global system of equations based on the tableau approach [46]:

$$\begin{bmatrix} \mathbf{I} & \mathbf{H}_{\mathrm{d}}^{\sigma} \\ \mathbf{M}_{1} & \mathbf{M}_{2} \end{bmatrix} \begin{bmatrix} \mathbf{w}^{\mathrm{o}} \\ \mathbf{w}^{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{w}^{\mathrm{h}} \\ \mathbf{0} \end{bmatrix}$$
(4.6)

where  $\mathbf{H}_{d}^{\sigma}$  is a block diagonal matrix built from the contribution of all sub-circuits' equivalent matrices  $\mathbf{H}_{d}^{\sigma_{k}}$  ( $1 \le k \le N$ ), i.e.,  $diag(\mathbf{H}_{1}^{\sigma_{1}}, ..., \mathbf{H}_{N}^{\sigma_{N}})$ . Similarly, the vectors  $\mathbf{w}^{o}$ ,  $\mathbf{w}^{i}$  and  $\mathbf{w}^{h}$  are comprised of sub-circuits port variables, i.e.,  $\mathbf{w}_{k}^{o}$ ,  $\mathbf{w}_{k}^{i}$  and  $\mathbf{w}_{k}^{h}$  ( $1 \le k \le N$ ), respectively.  $\mathbf{M}_{1}$  and  $\mathbf{M}_{2}$  are connectivity matrices and **I** is the identity matrix. Eq. (4.6) can be reduced using partial Gaussian elimination to be solved for interface variables only, that is the voltages or currents of connecting ports. In addition, some interface variables are dependent and can be eliminated as well. Hence, the reduced form of (4.6) is given as:

$$\mathbf{H}_{\mathrm{red}}^{\sigma} \mathbf{w}^{\mathrm{if}} = \mathbf{w}_{\mathrm{red}}^{\mathrm{h}} \tag{4.7}$$

with  $\mathbf{w}^{if}$  being a sub-vector of  $\mathbf{w}^{i}$  comprised of independent interface variables (voltage or currents), where  $\mathbf{H}_{red}^{\sigma}$  and  $\mathbf{w}_{red}^{h}$  are obtained from the partial Gaussian elimination process. Eq. (4.7) is henceforth referred to as global equation. The general concept of the NTT is demonstrated in Figure 4.2. At each time-point of simulation, all sub-circuits feed their equivalent circuit to the global equation solver. Then, after solving (4.7), the global equation block sends their imposed variables back.

### 4.3.2 An Illustrative Example for NTT

To better understand the NTT concept, recall LLC circuit from of Figure 3.5.a. Figure 4.3.a shows LLC circuit which is partitioned into three sub-circuits which are grayed out. After obtaining each sub-circuit equivalent, they are connected according to Figure 4.3.b. In this circuit, all equivalents are based on Norton equivalent and thus all ports are voltage port type.

The Norton equivalents of three sub-circuits are as follow:

$$i_1^{0} = i_1^{eq} - H_1 v_1^{i}, \quad i_3^{0} = i_3^{eq} - H_3 v_3^{i}$$
(4.8)

$$\begin{bmatrix} i_{2,1}^{0} \\ i_{2,2}^{0} \end{bmatrix} = \begin{bmatrix} i_{2,1}^{eq} \\ i_{2,2}^{eq} \end{bmatrix} - \begin{bmatrix} H_{2,a}^{\sigma_{2}} & H_{2,b}^{\sigma_{2}} \\ H_{2,c}^{\sigma_{2}} & H_{2,d}^{\sigma_{2}} \end{bmatrix} \begin{bmatrix} v_{2,1}^{i} \\ v_{2,2}^{i} \end{bmatrix}$$
(4.9)

It is noted that sub-circuit #2 is purely resistive and hence  $[i_{2,1}^{eq}, i_{2,2}^{eq}] = [0, 0]$ .

Having known all equivalents, the global equation (4.6) submatrices are:

$$\mathbf{H}_{d}^{\sigma} = \begin{bmatrix} H_{1} & 0 & 0 & 0\\ 0 & H_{2,a}^{\sigma_{2}} & H_{2,b}^{\sigma_{2}} & 0\\ 0 & H_{2,c}^{\sigma_{2}} & H_{2,d}^{\sigma_{2}} & 0\\ 0 & 0 & 0 & H_{3} \end{bmatrix}$$
(4.10)
$$\mathbf{M}_{1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 \\ 0 & 0 & +1 & +1 \end{bmatrix}, \quad \mathbf{M}_{2} = \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & 0 & -1 & +1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
(4.11)



Figure 4.3: NTT applied to LLC circuit: (a): LLC circuit is teared to three sub-circuit, (b): interconnection of three sub-circuit equivalents

By substituting (4.10) and (4.11) in (4.6) and permuting columns 5 and 7 of the resulted matrix, and then performing partial Gaussian elimination to rows 7 and 8, the reduced form (4.7) is given by:

$$\begin{bmatrix} H_1 + H_{2,a}^{\sigma_2} & H_{2,b}^{\sigma_2} \\ H_{2,c}^{\sigma_2} & H_3 + H_{2,d}^{\sigma_2} \end{bmatrix} \begin{bmatrix} v_1^i \\ v_3^i \end{bmatrix} = \begin{bmatrix} i_1^{eq} \\ i_{eq}^{eq} \end{bmatrix}$$
(4.12)

Eq. (4.12) only computes independent interface variables and according to Figure 4.2, two other interface variables are:

$$v_{1,2}^i = v_1^i, \quad v_{2,2}^i = v_3^i$$
 (4.13)

### 4.3.3 Sub-circuit Network Equations

Network equations of each sub-circuit is assembled using MANA. The process is demonstrated in Appendix and is followed here to obtain network equations of sub-circuit k.

To compute hybrid multi-port equivalent of each sub-circuit ((4.1) or (4.2)), the original MANA equation of sub-circuit k is augmented with ports output variables as follows:

$$\begin{bmatrix} \mathbf{A}_{k}^{\sigma_{k}} & \mathbf{D}_{k} & \mathbf{0} \\ \mathbf{D}_{k}^{T} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{k} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k} \\ \mathbf{i}_{k}^{o} \\ \mathbf{v}_{k}^{o} \end{bmatrix} = \mathbf{B}_{k}^{\prime} \begin{bmatrix} \mathbf{u}_{k} \\ \mathbf{i}_{k}^{h} \end{bmatrix} + \mathbf{H}_{k}^{\prime} \begin{bmatrix} \mathbf{v}_{k}^{i} \\ \mathbf{i}_{k}^{i} \end{bmatrix},$$
(4.14)

where  $\mathbf{A}_{k}^{\sigma_{k}}$  is the MANA matrix of sub-circuit k,  $\mathbf{D}_{k}$  is a connectivity matrix to account for current of voltage ports; and  $\mathbf{T}_{k}$  is a matrix to select voltage of current ports.  $\mathbf{B}_{k}^{\prime}$  and  $\mathbf{H}_{k}^{\prime}$  are incidence matrices. By performing partial Gaussian elimination, Eq. (4.14) becomes:

$$\begin{bmatrix} \mathbf{L}_{1,k}^{\sigma_k} & \mathbf{L}_{2,k}^{\sigma_k} \\ \mathbf{L}_{3,k}^{\sigma_k} & \mathbf{L}_{4,k}^{\sigma_k} \end{bmatrix} \begin{bmatrix} \mathbf{i}_k^o \\ \mathbf{v}_k^o \end{bmatrix} = \mathbf{B}_k^{"} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{i}_k^h \end{bmatrix} + \mathbf{H}_k^{"} \begin{bmatrix} \mathbf{v}_k^i \\ \mathbf{i}_k^i \end{bmatrix}$$
(4.15)

matrices  $\mathbf{L}_{-k}^{\sigma_k}$ ,  $\mathbf{B}_{k}^{"}$  and  $\mathbf{H}_{k}^{"}$  are obtained from partial Gaussian elimination process. By doing some algebraic manipulations, Eq. (4.15) can be written to look like as Eq. (4.1) which represents multiport hybrid equivalent of sub-circuit *k*.

### 4.3.4 NTT Simulation Flow

Figure 4.4.a illustrates NTT-based simulation algorithms. According to this flowchart, all subcircuits provide global solver with their hybrid equivalent. Afterward, global equation solver feeds sub-circuits with their associated interface variables which are used to compute switch voltage vector. All sub-circuits are dealt with in parallel and this method renders an exact solution by solving all sub-circuits simultaneously [36, 37]. After all switches are correctly set, network solver inside each sub-circuit computes history terms and output of interest.

The NTT simulation flow introduces two iterative processes, *local* and *global* iterations. The *local* iteration consists in each sub-circuit model using an iterative solution to determine its switch combination for a given instance of the interface variables. The *global* iteration takes place

whenever a change in switch combination is detected on any sub-circuit, after which global solver computes new interface variables after gathering all updated hybrid equivalents. The NTT can enhance parallelization and alleviate FPGA memory requirement, but it fails to achieve a small time-step of simulation.



Figure 4.4: (a): Simulation algorithm of the NTT, (b): Simulation algorithm of the relaxed NTT It is legitimate to remove *global* iteration when setting interfaces variables to circuit states. In this condition, switch status are determined based on the interface variables read from the previous time-point of the simulation [33]. The global equation is solved once all switch status are

determined. At the final stage, local solver of each sub-circuit computes new state variables and outputs. This version of NTT is called relaxed-NTT and its flowchart is presented in Figure 4.4.b. Relaxed-NTT uses iterative solution inside each sub-circuit to determine switch status, hence it still can not be employed to obtain a small time-step RTS.

# 4.4 Combination of DMM with SDD and NTT

As presented in pervious chapter, one main step for obtaining DMM function is to check feasibility of each combination using FME algorithm. This process becomes difficult since the problem grows exponentially with the number of switches. To circumvent this problem, SDD and NTT are used to split a network to multiple sub-circuits with a smaller independent sub-set of switches. Then for each sub-circuit a DMM function can be assigned to directly determine diode status.

### 4.4.1 DMM-SDD

Figure 4.5.a shows the general simulation flow of DMM-SDD method. When SDD is applied, the solutions of sub-circuits are decoupled at each time-point. In this condition, the *global* iteration is automatically removed. Each sub-circuit, however, deals with *local* iteration for its own solution which is removed by applying DMM.

### 4.4.2 DMM-NTT

The DMM-NTT method eliminates both *local* and *global* iterations. Figure 4.5.b presents the DMM-NTT method. To eliminate the *local* iterations, DMM is used at each sub-circuit level. Interface variables are seen by the sub-circuit as input voltage or current sources. On the other hand, the *global* iterations can be avoided using the relaxed version of the NTT if interface variables are tied to circuit states.



Figure 4.5: (a): DMM-SDD flowchart, (b): DMM-NTT flowchart

# 4.5 Case Study: EVBC

### 4.5.1 EVBC Overview

The EVBC is selected as a case study because of its topicality for the EV industry. It consists of two power conversion stages [93]:

- 1. An ac-dc stage that acts as a PFC. This first stage interfaces the universal grid, shapes the input current, minimizes THD, and provides a unity power factor [94];
- 2. A dc-dc stage that galvanically isolates the EV from the grid, regulates current/voltage of the battery, and rejects low frequency ripples [93].

Figure 4.6 shows the target EVBC circuit. Table 4.1 lists the circuit parameters used here [95, 96]. A full bridge rectifier followed by a two-phase interleaved boost converter (IBC) is used at the first stage (Rect-IBC). A  $3-\Phi$  LLC resonant converter is used at the second stage.



Figure 4.6: EVBC test case [77]

The two phases of the IBC operate at a 180° phase shift, which leads inductor ripple currents to cancel each other and thus a reduction in  $i_{in}$  ripple, which is a function of the duty-cycle  $D = 1 - \frac{v_{ac}}{v_{dc}}$ . *D* is a function of phase angle and amplitude of the grid voltage and lies between 0.02 and 1 for PFC applications [97].

Fable 4.1: EVBC parameter	S	
---------------------------	---	--

Parameter	Value
V <sub>ac</sub>	240 V/60 Hz
R <sub>s</sub>	2 Ω
L <sub>s</sub>	1 mH
$f_{s,IBC}$	100 kHz
$L_{1}, L_{2}$	368 µH
$C_{\rm dc}$	1 mF
$f_{r_1}$	205 kHz
$f_{r_2}$	102 kHz
n	16/3
$L_{\mathbf{r}_a}, L_{\mathbf{r}_b}, L_{\mathbf{r}_c}$	20 µH
$L_{m_a}, L_{m_b}, L_{m_c}$	60 µH
$C_{r_a}, C_{r_b}, C_{r_c}$	30 nF
$C_o$	1 mF
$R_L$	25 Ω

Similarly, the dc capacitor ripple current is decreased compared to the conventional boost converter. This allows smaller inductors and capacitors, which in turn increases the converter power density.

The 3- $\Phi$  LLC structure offers higher power and is used in level 2 and level 3 in EVBC applications [96]. The LLC controls the power flow from the intermediate dc-link to the battery using PFM. The frequencies of fixed duty cycle (50%) pulses are adjusted by means of a voltage-controlled oscillator block.

### 4.5.2 Modeling Alternatives for the RTS of the EVBC

The EVBC circuit consists of multiple high-switching frequency converters, so its RTS requires very small time-steps (in the range of tens of ns) while having the ability to handle the simulation of circuits of a significant size.

### 4.5.2.1 Decoupling

The EVBC circuit of Figure 4.6 can be decoupled at the dc bus, splitting the EVBC into two subcircuits: Stage 1 and Stage 2. The decoupling approaches are SDD and NTT. When using SDD, the sub-circuits exchange  $i_{dc}$  and  $v_{dc}$  from the previous time-point of the simulation. On the other hand, when NTT is used, each sub-circuit is reduced to a Norton equivalent, so that the dc voltage is given by:

$$v_{dc}(t) = \left(g_1^{\sigma_1} + g_2^{\sigma_2}\right)^{-1} \left(i_1^h(t) + i_2^h(t)\right),\tag{4.16}$$

where  $i_k^h(t)$  and  $g_k^{\sigma_k}$  ( $k \in \{1, 2\}$ ) are the Norton currents and conductances of each sub-circuit and associated with switch combination  $\sigma_k$  for Stage k. It is worth mentioning that  $\mathbf{H}_k^{\sigma_k} \equiv g_k^{\sigma_k}$ ,  $\mathbf{w}_k^h \equiv i_k^h(t)$  ( $k \in \{1, 2\}$ ), and  $\mathbf{w}^{\text{if}}$ . More specifically,  $i_k^h(t)$  of each sub-circuit is obtained as follows:

$$i_{1}^{h}(t) = \mathbf{C}_{1}^{\sigma_{1}} \left[ \nu_{ac}(t), i_{L_{s}}^{h}(t), i_{L_{1}}^{h}(t), i_{L_{2}}^{h}(t), i_{C_{dc}}^{h}(t) \right]^{T}$$
(4.17)

$$i_{2}^{h}(t) = \mathbf{C}_{2}^{\sigma_{2}} \left[ i_{\mathrm{L_{rabc}}}^{h}(t), i_{\mathrm{L_{mabc}}}^{h}(t), i_{\mathrm{C_{rabc}}}^{h}(t), i_{\mathrm{C_{o}}}^{h}(t) \right]^{T}$$
(4.18)

where  $\mathbf{C}_1^{\sigma_1}$  and  $\mathbf{C}_2^{\sigma_2}$  are coefficient vectors of sub-circuit 1 and sub-circuit 2, respectively and  $i^h_-(t)$  are history terms values of inductors and capacitors.

#### 4.5.2.2 Solving for diodes

To deal with diodes in the circuit, two approaches are considered:

- 1. The status of the diodes is determined based on their current/voltage as read from the previous time-point; this method is denoted hereafter explicit status update (ESU).
- 2. DMM is utilized to obtain the status of the diodes.

The experiments with ESU have shown that the method works satisfactorily for the bridge rectifiers on both stages 1 and 2 of the EVBC. However, it fails with the IBC. Hence, ESU results are drawn with the hypothesis that the IBC is operated in continuous mode, that is diode of each leg ( $d_5$  or  $d_6$ ) blocks whenever its associated controlled switch ( $s_1$  or  $s_2$ ) conducts, and *vice versa*.

Sub-circuit 1 contains 6 diodes and admits 64 possible diode combinations. Let us label these combinations with an integer  $\sigma_d^1$  ranging from 0 to 63 and given by the binary representation  $\sigma_d^1 = d_1 d_2 d_3 d_4 d_5 d_6$ , out of which only 16 are found to be feasible. These combinations are determined by a set of 32 non-redundant hyper-planes. hence,  $\mathbf{W}^{\sigma_c}$  is a 32 × 6 matrix (6 is the size of  $\mathbf{z}_1(t)$  in the Rect-IBC sub-circuit) which yields 32-entry  $\mathbf{p}_1^{\sigma_c}$  vector, for each of the two assumed controlled switch combinations of the IBC.

Sub-circuit 2 contains 6 diodes, yielding a total of 64 diode combinations, labeled  $\sigma_d^2 = d_7 d_8 d_9 d_{10} d_{11} d_{12}$ , out of which only 19 are deemed feasible (assuming  $v_o > 0$ ). These 19 combinations are determined by a set of 30 non-redundant hyper-planes leading to  $30 \times 11 \text{ W}^{\sigma_c}$  matrix (11 is the size of  $\mathbf{z}_2(t)$  in the 3- $\Phi$  LLC sub-circuit) which yields 30-entry  $\mathbf{p}_2^{\sigma_c}$  vector, for each of the 6 controlled switch combinations of inverter.



Figure 4.7: Adjacency graph of the feasible switch combinations: (a): diode combination of stage 1; (b): diode combination of stage 2 [77]

Figure 4.7 shows the adjacency graph of the feasible diode combinations for Stage 1 and 2 of the EVBC. Each labeled node represents a feasible combination (a cone) and edges are hyper-planes delimiting two adjacent cones. It is worth mentioning that the number of edges in Figure 4.7 corresponds to the number of rows in  $\mathbf{W}^{\sigma_c}$  matrix of each sub-circuit.

### **4.5.3 Offline Simulation Results**

The accuracy of the aforementioned simulation methods is assessed using the following test sequence:

- 1. Sequence I#1. Time-span: 0.00s-0.14s; The EVBC starts from zero and is operated normally. The switching frequencies of the IBC and the LLC are 100 kHz and 200 kHz, respectively.
- 2. Sequence I#2: Time-span: 0.14s-0.15s: Load is shorted.
- 3. Sequence I#3. Time-span: 0.15s-0.21s: Fault cleared. Back to normal.
- 4. Sequence I#4. Time-span: 0.21s-0.22s: Fault at the resonant tank of the LLC between phases *a* and *b*.
- 5. Sequence I#5. Time-span: 0.22s-0.33s: Fault cleared. Back to normal.
- Sequence I#6. Time-span: 0.33s-0.40s: The LLC switching frequency is decreased to 150 kHz.



Figure 4.8: EVBC circuit waveforms: (a): inductor current of IBC, (b): phase-a resonant current, (c): phase-a magnetizing current, (d): output voltage, (cv.1)-(cv.8): closed-up views [77]

Figure 4.8 (a)-(d) show the waveforms of four state variables of interest for the total test sequence duration of 0.4s. The four variables are the IBC inductor current  $i_{L_1}$ , the LLC resonant current  $i_{L_{ra}}$ , the LLC magnetization current  $i_{L_{ma}}$ , and the output voltage  $v_o$ . A simulation time-step of 50 ns was selected. The accuracy of the simulation is assessed using the 2-norm relative error, while reference results are drawn from EMTP using iterative solver. The 2-norm relative errors for  $i_{L_1}$ ,  $i_{ra}$ ,  $i_{ma}$  and  $v_o$  for four simulation alternatives are considered here: ESU-SDD, ESU-NTT, DMM-SDD, DMM-NTT are listed in Table 4.2.

Although ESU offers a reasonable computational cost, it does not produce very accurate results, according to Table 4.2. The error of the magnetization current  $(i_{ma})$  reaches over 114% during the first fault (Seq. I#2), whereas an error of over 20% is reached by the resonant current  $(i_{ra})$  once

the fault is cleared during Seq. I#3. Nevertheless, the errors are generally acceptable for RTS applications (often below 3%).

	SDD		NTT						
		$i_{L_1}$	i <sub>ra</sub>	i <sub>ma</sub>	$v_o$	$i_{L_1}$	i <sub>ra</sub>	i <sub>ma</sub>	vo
	Seq. I#1	8.7	2.3	2	2	8.7	2.3	2	2
	Seq. I#2	2.4	2.7	85.3	2.8	2.4	2.7	114.7	2.8
5	Seq. I#3	6	23.1	1.6	1	6	21.4	4.9	1
ISE	Seq. I#4	0.9	1	1	1.5	0.9	1.5	1.5	1.5
щ	Seq. I#5	8.2	2	1.7	1.7	8.2	2	1.7	1.7
	Seq. I#6	39.7	2.5	2.3	2.4	39.7	2.5	2.3	2.4
	Total	6.6	3.9	6.7	2	6.6	3.7	8.9	2
	Seq. I#1	3.8e-5	2.8e-4	1.2e-4	1.2e-4	6.9e-9	1.5e-8	9e-9	2.8e-9
	Seq. I#2	4.3e-4	1.5e-3	1.9e-5	1.4e-4	1.9e-9	3.4e-9	3.9e-9	3.7e-9
7	Seq. I#3	7.2e-4	7.3e-4	3.5e-4	3.5e-4	4.7e-9	1.1e-8	7e-9	1.9e-9
IW	Seq. I#4	1.8e-2	1.3	1.3	3.8e-5	2.1e-8	2.9e-9	4.6e-9	2.7e-9
Q	Seq. I#5	1.1e-2	3.3e-4	1.8e-4	7.2e-5	6.8e-9	4.8e-8	2.6e-8	2.5e-9
	Seq. I#6	5.6e-4	1.9e-4	6e-5	2.7e-5	3.2e-8	4e-9	3.3e-9	3.2e-9
	Total	1.3e-2	4.7e-3	1.2e-2	1.3e-4	1.4e-8	9.5e-9	1.2e-8	2.8e-9

Table 4.2: 2-Norm relative errors (%) obtained from various simulation methods [77]

But this observation should be considered with caution: the test sequence was generated using an open loop control, and results are prone to deteriorate in closed loop control.

The DMM on the other hand offers excellent accuracy. When combined with SDD, DMM barely exceeds 1% during the second fault sequence (Seq. I#4), and generally exhibits errors way below 0.02%. But when combined with NTT, DMM accuracy is perfect. Table 4.2 reports 2-norm errors in the  $10^{-9}\%/10^{-8}\%$  range, which can be interpreted as exact accuracy with numerical artifacts. To understand better on what the numbers in Table 4.2 mean, Figure 4.8(cv.1)-(cv.8) present close-up views of four variables for the two time intervals. Lager error values reported in Table 4.2 specially for  $i_{ma}$  during Seq. I#2 can easily be observed in Figure 4.8(cv.5). It is also observed that

DMM-NTT results are perfectly matched with the reference from EMTP.

## 4.6 FPGA Implementation

FPGA implementation of ESU-SDD, DMM-SDD and DMM-NTT methods are considered. ESU-NTT is excluded since its accuracy is very close to ESU-SDD while rendering more computations. Similar to the LLC simulator of Chapter 3, the targeted FPGA is a Kintex K325T. In all implementations, a 35-bit FXP data format with 32 fractional bits is used to perform all calculations. To address FXP limited dynamic range issue, matrix entries are normalized based on per-unit scales. In the following section, normalization process is explained.

### 4.6.1 Normalization

MANA equations of a given network can be rewritten in the form of (4.19) which only solves for history terms and desired outputs:

$$\left[\mathbf{i}^{\mathrm{h}},\mathbf{y}\right]^{T}=\mathbf{H}^{\sigma}\mathbf{z} \tag{4.19}$$

Assume  $[\mathbf{i}_n^{h}, \mathbf{y}_n]^T$  and  $\mathbf{z}_n$  represent normalized counterparts of respectively  $[\mathbf{i}^{h}, \mathbf{y}]^T$  and  $\mathbf{z}$  based on the normalization matrices  $\mathbf{M}_{iy}$  and  $\mathbf{M}_z$ . The relation of the original vectors with their normalized counterparts are:

$$\begin{cases} \left[\mathbf{i}^{\mathrm{h}}, \mathbf{y}\right]^{T} = \mathbf{T}_{iy} \left[\mathbf{i}^{\mathrm{h}}_{n}, \mathbf{y}_{\mathrm{n}}\right]^{T} \\ \mathbf{z} = \mathbf{T}_{z} \mathbf{z}_{\mathrm{n}} \end{cases}$$
(4.20)

where matrices  $\mathbf{T}_{iy} = \mathbf{M}_{iy}^{-1}$  and  $\mathbf{T}_z = \mathbf{M}_z^{-1}$  are referred to as de-normalization matrices. Replacing Eq. (4.20) in Eq. (4.19) results in:

$$\left[\mathbf{i}_{n}^{\mathrm{h}},\mathbf{y}_{\mathrm{n}}\right]^{T} = (\mathbf{M}_{iy}\mathbf{H}^{\sigma}\mathbf{T}_{z})\mathbf{z}_{\mathrm{n}}$$
(4.21)

 $\mathbf{M}_{iy}\mathbf{H}^{\sigma}\mathbf{T}_{z}$  is the normalized network matrix. Since 35-bit FXP data format with 32 fractional bits is selected, normalization matrices should be properly selected to maintain all computation results in the range of [0, 3]. On criteria to select these matrices elements is to set their values equal to the maximum values of inputs, history terms and output. It is worth mentioning that the same process should be applied to other matrices.

### 4.6.2 Hardware Implementation

Figure 4.9 and Figure 4.10 show the datapath of the hardware implementations of the ESU-SDD, DMM-SDD and DMM-NTT approaches, respectively. MVM modules used in these implementations is demonstrated in Figure 3.9, where each MVM module i (MVM<sub>*i*</sub>) multiplies an

 $n_i \times m_i$  matrix by an  $m_i \times 1$  vector. Similar to the LLC simulator, MVM<sub>i</sub> uses  $N_i m_i$ -input DP units, where  $N_i \le n_i$ , to balance between area occupation and time-step. Also, a set of  $N_i \times m_i$  read-only memories (ROMs) provide precomputed matrices to each MVM<sub>i</sub> module [77].

Figure 4.9.a presents the datapath of the ESU-SDD approach. It is comprised of two computing parts, separated by a dashed line in Figure 4.9.a. Each part deals with a sub-circuit of the EVBC, i.e., Rect-IBC and 3- $\Phi$  LLC. Each computing part determines the switch combination  $\sigma_k$  using a ROM addressed by the gating signals and the sign values of the diode's currents. Both parts exchange their interface variables ( $v_{dc}$  and  $i_{dc}$ ) with a single time-step delay. The MVM unit updates the states of each sub-circuit,  $\mathbf{x}_k(t)$ , and compute desired outputs,  $\mathbf{y}_k(t)$  [77].

Figure 4.9.b shows the datapath of the DMM-SDD approach. Similar to the datapath of the ESU-SDD, it includes two computing parts dedicated to the two sub-circuits of the EVBC (separated by a dashed line in Figure 4.9).  $\sigma_k$  of each sub-circuit is found using the DMM [77].



Figure 4.9: Datapath of implemented methods, (a): ESU-SDD, (b): DMM-SDD [77]



Figure 4.10: Datapath of DMM-NTT [77]

Inside each DMM module, the computed sign function is used to address a ternary content addressable memory (TCAM) to obtain  $\sigma_k$ . In this datapath,  $v_{dc}$  and  $i_{dc}$  are exchanged in the same way as the ESU-SDD approach [77].

The datapath of the DMM-NTT approach is shown in Figure 4.10. It consists of three dedicated computing parts (distinguished by dashed lines): i) Rect-IBC, ii) 3- $\Phi$  LLC and iii) NTT. The NTT computing part, computes Norton currents ( $i_k^h$ ) of each sub-circuit, which are used to obtain the dc-link voltage according to (4.16). In this part, binary word concatenation unit renders the unified switch combination of the EVBC, i.e.,  $\sigma = \{\sigma_1, \sigma_2\}$  to address the ROMs which contain  $(g_1^{\sigma_1} + g_2^{\sigma_2})^{-1}$  values. The updated value of  $v_{dc}$  is then sent to the final-stage MVM module of each sub-circuit devoted to update  $\mathbf{x}_k(t)$  and compute  $\mathbf{y}_k(t)$  [77].

### 4.6.3 DMM Function Implementation

The DMM function is implemented by a combination of a TCAM and a random-access memory (RAM), as illustrated in Figure 4.11. The location in the TCAM is then used to address a RAM that contains switch combinations. In Figure 4.11, sgn( $\mathbf{p}^{\sigma_c}(\mathbf{z}(t))$ ) is used as a key to search through the TCAM. The search keys of both sub-circuits are listed as Tables presented in the Appendix E. Since a single switch combination is feasible at each time-step, an encoder is used to generate the address for the RAM [77].



Figure 4.11: General view of the hardware DMM function

### 4.6.4 Comparison Between the FPGA Implementations of the Three Methods

In order to implement DMM-SDD and DMM-NTT, two approaches are followed. In the first approach, DMM and State Update units, are executed serially. The State Update Unit starts its process as soon as the attached DMM unit finishes its own, double-stage implementation [21]. In this implementation approach, the maximum latency is determined by the sum of the latencies of both units. In the second implementation approach, the equations of DMM and State Update units are manipulated in a such way that they process their computations in parallel, single-stage implementation approach [21]. In this approach, the total latency of the datapath is decreased to achieve a smaller time-step.

Table 4.3 reports time-steps and FPGA resource consumption of each method. All designs are able to sustain a 320 MHz clock frequency. When compared to double-stage implementations of DMM, ESU-SDD offers the smallest time-step and area footprint as expected. The resource consumption of DMM-SDD and DMM-NTT are very close together but slightly higher numbers in the DMM-NTT is due to the fact that it includes an extra level of computation for the NTT. On the other hand, single-stage versions of the DMM offer time-steps of 75 ns and 100 ns for DMM-SDD and DMM-NTT respectively.

As can be seen from Table 4.3, single-stage versions consume slightly more FPGA resources compared to their double-stage counterparts but require approximately twice the number of BRAMs. It is noted that time-steps reported in Table 4.3 can be smaller if the parallelization factors of MVM<sub>*i*</sub> modules,  $N_i$ , are increased.

Item	Method					
	ESU-SDD	DMM-SDD		DMM-NTT		
	n.a	Double-stage	Single-stage	Double-stage	Single-stage	
Time-Step	75 ns	125 ns	75 ns	175 ns	100 ns	
LUTs	1787 (0.8%)	4784 (2.3%)	6485 (3.18%)	5118 (2.5%)	6513 (3.14%)	
Registers	5572 (1.3%)	14119 (3.4%)	16182 (3.9%)	14223 (3.4%)	16835 (4.1%)	
BRAMs	36.5 (8.2%)	74.5 (16.7%)	142 (31.9%)	78.5 (17.6%)	143.5 (32.2%)	
DSP Blocks	156 (18.5%)	428 (50.9%)	476 (56.6%)	448 (53.3%)	480 (57.1%)	

Table 4.3: FPGA resource consumption of ESU-SDD, DMM-SDD and DMM-NTT [77]

In RTS applications where FPGA resource consumption is a concern and very accurate results are not required, ESU-SDD can be employed. If accurate results are expected and higher hardware footprints are not an important issue, DMM-SDD and DMM-NTT can be used. While the accuracy of DMM-SDD is high, the DMM-NTT can be used to obtain almost exact results. In addition, if the behavior of the circuit during the fault condition is targeted, using the DMM-NTT is recommended since the accuracy of DMM-SDD was shown to deteriorate during the fault period [77].

## 4.7 Real-Time Test

The real-time test results of the proposed simulator for the EVBC case are presented in this section. Figure 4.12 shows the physical setup of the proposed real-time simulator: A Zynq-7000 SoC ZC706 board from Xilinx, attached to a DAC3484, a quad-channel, 16-Bit, 1.25-GSPS digital-to-analog converter (DAC) from TI. The ZC706 board has a XC7Z045 Zynq FPGA, comprised of a dual core ARM and a logical part equivalent to the K325T Kintex [77].

The EVBC circuit is implemented on the FPGA to run in the real-time. Table 4.4 shows the computational accuracy of the FPGA implementation in terms of %2-norm error for four state variables of the EVBC circuit. This table shows that the FPGA performs the computations with very small relative errors.

Table 4.4: EVBC circuit-FPGA computational accuracy

Variable	i <sub>L1</sub>	i <sub>ra</sub>	i <sub>ma</sub>	$v_o$
%2-norm error	2.04e-2	9.63e-3	8.51e-3	8.65e-3



Figure 4.12: Experimental setup [77]



Figure 4.13: EVBC circuit- Real-time test results of phase a and phase b resonant and magnetizing currents of LLC [77]

Figure 4.13 presents an oscilloscope capture of the three-phase resonant tank of the LLC using the physical setup, namely  $i_{ra}$ ,  $i_{rb}$ ,  $i_{ma}$ , and  $i_{mb}$ . It is worth mentioning that the oscilloscope indicates  $i_{ra}$ , frequency as 200 kHz, which is equal to the switching frequency used [77].

# 4.8 Conclusion

This chapter addressed the scalability issue of the DMM for complex systems. In doing so, this chapter proposes combination of DMM with decoupling techniques, namely, NTT and SDD for RTS of multi-HSF PECs networks. NTT method was revisited, and it is demonstrated that it brings two levels of iterative solutions, i.e., *local* inside each sub-circuit, and *global* in the global solver level. Afterward, it is shown that a DMM mapping function designated to each sub-circuit removes *local* iteration, and *global* iteration is eliminated using the relaxed version of the NTT and SDD.

An EVBC circuit is considered as the test case study of this chapter. The DMM-based methods are compared with ESU as an alternative method, and it is shown that the proposed methods can achieve the same level of time-step as ESU, but they occupy more FPGA resources. This however is not a serious issue since in most cases in RTS applications a specific circuit such as EVBC is targeted. In addition, a physical setup is built to perform the real-time test simulation of the EVBC circuit, and the experimental results are presented.

### CHAPTER 5 FPGA-BASED SIMULATION OF TRANSMISSION LINE

# 5.1 Introduction

Time-domain relays offer reliable, accurate and fast operation in both protective and fault locating schemes. The performance of these relays should be tested for different fault scenarios in realistic and practical conditions. To fulfill these requirements, FPGA-based HIL test has recently attracted a great deal of attention due its ability to accurately reproduce transient waves and perform simulation with small time-step down to nanoseconds [8]. From the modelling perspective, TL modelling plays a vital role in the reliability and accuracy of this testbench and should be evaluated in detail.

This chapter presents a testbench for HIL testing of TWFL. First, due to moderate computational cost, the CP TL model is used and evaluated. This model allows adopting small simulation time-steps in the order to achieve sub-microsecond simulation time-step and it is particularly suitable for double-ended fault location approaches. However, as it is shown in this chapter, the incorporation of the CP TL model in the single-ended fault location approaches affects the reliability and accuracy of the testbench. To overcome the inadequacy of the CP line model, a more accurate and efficient FPGA-based WB universal TL computing unit is proposed. This model executes WB TL equations within a very small time-step which significantly improves the reliability and accuracy of TWFL testbench.

### 5.2 Overview on TW-Based Fault Locating Methods

To explain the concept of TW-based fault locating, let us consider the Bewley lattice diagram of Figure 5.1 which shows every incident, reflected and transmitted TWs in a TL connecting bus k (local bus also is referred to as sending end) and bus m (remote bus also is referred to as receiving end). In Figure 5.1, the TW is initiated from a fault at point F and arrives to the local bus k and the remote bus m at arrival times  $t_{k1}$  and  $t_{m1}$ , respectively. The TW relays or recorders are installed on the same buses [8].

The location of the fault can be obtained by comparing these initial incident time of arrivals (ToAs):

$$DF = \left(\frac{t_{m1} - t_{k1}}{\tau} + 1\right)\frac{l}{2}$$
(5.1)

In (5.1),  $\tau$  is the TL travel time, *DF* is the fault distance from the local bus *k* and *l* is the length of the TL under protection.



Figure 5.1: Bewley diagram for TWs initiated from a fault at point F [8]

TW-based fault locating approach based on (5.1) is called double-ended approach which is also referred to as Type-D [98]. To measure  $t_{m1}$  and  $t_{k1}$ , fault locating devices which are installed at buses k and m, should acquire data with a common time reference [99]. Thus, a communication link is established between local and remote TW-based devices to ensure their synchronization.

In contrast to double-ended approach, single-ended approach (also known as Type-A) [98] uses time difference between the first incident TW and first reflected TW from the fault point (arriving at  $t_{k2}$ ) at a single terminal to estimate the fault location. Assuming bus k as the measuring terminal, the fault distance *DF* is given by:

$$DF = \left(\frac{t_{k2} - t_{k1}}{2\tau}\right)l\tag{5.2}$$

Type-A approach does not rely on data from other end, thus eliminates the need for communication channel, and is therefore not prone to timing errors [100]. In this scheme the correct identification of first reflection from the fault point among other TWs, which arrive from terminals behind or in front of relay, is very crucial.

## 5.3 CP TL Model

CP refers to a distributed line model whose per unit length parameters are considered frequency independent. In EMTP [3, 4], CP is represented by two lossless TL segments and lumped resistances in three places, namely two at the line ends and one in the middle, to model ohmic losses [2], see Figure 5.2(a). The two-segment CP model is valid if the TL characteristic impedance is much larger than the series resistance.

An *n*-conductor TL is governed by *n* coupled phase-domain equations. As a standard practice, TL phase-domain (ph) quantities are transformed into modal-domain (md), where *n* coupled phase-domain TL equations are represented in the form of *n*-decoupled modal-domain equations. In the case of CP, the transformation between phase and modal quantities is performed using constant voltage and current transformation matrices [2]:  $\mathbf{v}_{ph} = \mathbf{T}_v \mathbf{v}_{md}$ ,  $\mathbf{i}_{ph} = \mathbf{T}_i \mathbf{i}_{md}$ . The modal transformation results in *n* propagation modes characterized by individual modal propagation functions and modal characteristic impedances. It is noted that such transformations increases the model parallelism and reduce the computational burden, which is beneficial for RTS purposes, particularly in the context of FPGA-based RTS [8].

Modal domain equations for the sending end *k* of the CP TL are given by:

$$\mathbf{i}_{k,md}(t) = \mathbf{Z}_{mod,md}^{-1} \mathbf{v}_{k,md}(t) - \mathbf{i}_{k,md}^{hist}(t)$$
(5.3)

where vector  $\mathbf{i}_{k,md}^{hist}$  is the modal domain history term at the line's sending end k and:

$$\begin{cases} \mathbf{Z}_{c,md} = \sqrt{\frac{\mathbf{L}'_{md}}{\mathbf{C}'_{md}}} \\ \mathbf{Z}_{mod,md} = \mathbf{Z}_{c,md} + \mathbf{R}/4 \end{cases}$$
(5.4)

In (5.4),  $\mathbf{Z}_{c,md}$  is the modal characteristic impedance of lossless TL with  $\mathbf{L}'_{md}$  and  $\mathbf{C}'_{md}$  being, respectively, the modal domain per unit length series inductance and shunt capacitance of the TL. In addition,  $\mathbf{Z}_{mod,md}$  is the modal domain modified surge impedance [2]. In (5.3), interchanging subscripts *k* and *m* results in similar expressions for the receiving end *m*.

The modal domain history term at the line's sending end k,  $\mathbf{i}_{k,md}^{hist}$ , is as follows:

$$\mathbf{i}_{k,md}^{hist}(t) = \mathbf{k}_{v1}\mathbf{v}_{k,md}(t - \mathbf{\tau}_{md}) + \mathbf{k}_{i1}\mathbf{i}_{k,md}(t - \mathbf{\tau}_{md}) + \mathbf{k}_{v2}\mathbf{v}_{m,md}(t - \mathbf{\tau}_{md}) + \mathbf{k}_{i2}\mathbf{i}_{m,md}(t - \mathbf{\tau}_{md})$$
(5.5)

with:

$$\begin{cases} \mathbf{k}_{v1} = \frac{\mathbf{I} - \mathbf{h}_{md}}{2} \frac{\mathbf{I} + \mathbf{h}_{md}}{\mathbf{Z}_{mod,md}} \\ \mathbf{k}_{v1} = \frac{\mathbf{I} + \mathbf{h}_{md}}{2} \frac{\mathbf{I} + \mathbf{h}_{md}}{\mathbf{Z}_{mod,md}} \\ \mathbf{k}_{i1} = -\frac{\mathbf{I} - \mathbf{h}_{md}}{2} \mathbf{h}_{md} \\ \mathbf{k}_{i2} = -\frac{\mathbf{I} + \mathbf{h}_{md}}{2} \mathbf{h}_{md} \end{cases}$$
(5.6)

and

$$\mathbf{h}_{md} = \frac{\mathbf{Z}_{c,md} - \mathbf{R}/4}{\mathbf{Z}_{mod,md}}$$
(5.7)



Figure 5.2: (a): Loss inclusion to the lossless CP TL model, (b): *n*-Norton Equivalent circuits of *n*-decoupled modes relating to the phase quantities [8]

In EMT-type simulation, Eq. (5.3) is replaced by two Norton equivalents at both ends of the CP line, as shown in Figure 5.2.b. In Eq. (5.5),  $\tau_{md}$  is not necessarily a multiple integer of the fixed simulation time-step and interpolations are used to calculate the history terms values [8].

## 5.4 CP TL for TW-Based Fault Locator Testing

This section aims to elaborate on the suitability of the CP TL model utilized in TWFL testing. With this aim and without losing any generality, a TL of 100 km length is considered. Table 5.1 lists TL parameters and

Figure 5.3 shows TL configuration. A single phase to ground fault is applied on phase-A (AG), at point *F* located at a distance of DF = 25 km from the sending end *k*, see Figure 5.1. Figure 5.4 shows the Bewley lattice diagram of the current TWs of phase-A (TWIA), launched due to the AG fault, at terminals *k* and *m* for both CP and WB [67] TL models. It is worth mentioning that EMTP is used to obtain simulation results. In Figure 5.4, it is observed that the first peaks of the TWIAs simultaneously arrive at each line end for both CP and WB TL models. Owing to this fact, the Type-D TWFL can accurately determine the fault location irrespective of the adopted line model. From Figure 5.4, an immediate subsequent peak, also referred to as "false peak" is observed for the CP line model. This false peak TW can be confused with TW reflected from the fault point and affect the performance of the Type-A [8].

The so-called subsequent false peaks, which are shown in Figure 5.4, can be reasonably attributed to inadequacy of CP TL in which the frequency dependence of line parameters is totally excluded. In fact, the frequency dependence of the line parameters which caused by skin effect, results in more attenuation and dispersion at higher frequencies. This effect is even more pronounced for the ground mode (zero sequence) propagation, which appears in asymmetric ground faults such as AG [8].

It is known that in the case of AG fault, the ground propagation mode contributes to the fault current and hence to its corresponding TWs. The propagation velocity of the ground mode is lower than that of aerial modes. Hence, the ground mode accounts for all subsequent false peaks shown in Figure 5.4 with no contribution to the initial incident TWs. In contrast, in the WB TL model, the ground propagation mode becomes attenuated, hence there exist no subsequent false peak. The aerial modes are not strongly affected by the frequency dependence of the TL parameters, thus their corresponding TW peaks coincide for both CP and WB line models, as shown in Figure 5.4 [8]. Table 5.1: Data of the aerial TL

Parameter	Value
Outside diameter of phase wire	4.06908 cm
Outside diameter of ground wire	0.98044 cm
DC resistance of phase wire	0.0324 Ω/km
DC resistance of ground wire	1.6216 Ω/km
Ground resistivity	100 Ω. m



Figure 5.3: Geometry of the aerial TL

### 5.4.1 Effect of the Ground Mode

#### 5.4.1.1 Modes Propagation Times vs. ToAs of TWs

To support the discussion presented earlier regarding the contribution of ground mode to the subsequent false peaks, the difference between the ground (0) and aerial mode (1) propagation delays i.e.,  $(\tau_0 - \tau_1)$  for different fault locations, i.e., *DF* ranging from 20 km to 80 km, are obtained using EMTP. In addition, for each fault location, the difference between the arrival times of the first peak  $(t_{k1})$  and that of the subsequent false peak  $(t_{sfp,k})$  for the same *DF*s, i.e.,  $(t_{sfp,k} - t_{k1})$  is calculated. Figure 5.5 shows both  $(t_{sfp,k} - t_{k1})$  and  $(\tau_0 - \tau_1)$  for different fault location for an AG fault. From this figure, it is observed that for any given *DF*,  $(t_{sfp,k} - t_{k1})$  is equal to  $(\tau_0 - \tau_1)$ . This observation further confirms that the subsequent false peaks in the CP model arise from the ground mode [8].



Figure 5.4: TWs arriving to the terminals (Phase A) of a TL of length l = 100 km due to a single-phase fault with DF = 25 km for both CP and WB TL models. The time axis origin corresponds to the fault occurrence instant [8]

#### 5.4.1.2 Line-to-Line (BC) Fault Case

For fault cases which do not include ground mode (such as line-to-line fault), it is expected that subsequent false peak will not appear. Figure 5.6 verifies this claim where the Bewley lattice diagram and the current TWs of Phase-C (TWIC) launched by a phase B-to-phase C fault at DF=25 km are depicted [8].

#### 5.4.1.3 Effect of Ground Resistivity

To evaluate the effect of ground resistivity  $\rho$  on the ground propagation mode of the CP TL, Figure 5.7 shows, for different values of  $\rho$ , the first incident TWIA at terminal k due to an AG fault initiated at DF = 25km. According to this figure, with decreasing the ground resistivity, the difference between propagation velocity of the aerial (first peak) and ground (subsequent peak) modes decreases too. Therefore, subsequent peak coincides with the first peak for the ground resistivity of  $\rho=0$  [8].



Figure 5.5: Comparison of time difference between the ground and aerial modes propagation delays and time difference between the arrival times of the first peak and that of the subsequent

false peak at terminal k. Case of CP line model and AG fault for different DFs [8]



Figure 5.6: TWs arriving to the terminals of the 100 km-TL at phase-C for both CP and WB line models. Case of a BC fault with DF=25 km. The time axis origin corresponds to the fault occurrence instant [8]



Figure 5.7: TWIA arriving to terminal k due to the AG fault at the distance of DF=25 km for different ground resistivity ( $\rho$ ) for the CP TL of length l = 100 km. The time axis origin corresponds to the fault occurrence instant [8]

#### 5.4.1.4 Near-end Fault Cases

To give more evidence on the impact of the ground mode on the TWs, an AG fault is considered at three fault locations, namely, DF = 2 km, DF = 5 km (near-end faults) and DF = 15 km. For the case of near-end faults, due to the small propagation distance, and despite having different propagation velocities, the difference between the travelling time of aerial ( $\tau_1$ ) and ground ( $\tau_0$ ) propagation modes of the CP TL decrease, see Table 5.2. The TWIAs arriving to terminal k for both CP and WB TLs are shown in Figure 5.8. According to this figure, it is observed that for nearend faults (DF = 2 km and 5 km) the subsequent false peak does not appear for both CP and WB TLs. For the CP line, this is attributed to the small difference between the arrival times of the aerial and ground propagation modes. For DF = 15 km, however, it is seen that the subsequent false peak appears for the CP TL. In addition, Figure 5.8 shows that regardless of the fault location, no subsequent false peak appears for the WB TL [8].

According to the evidence provided above, it is concluded that CP TL is only suitable for testing double-ended scheme of TWFL, and single-ended scheme testing requires employment of WB TL.

Fault location	2 km	5 km	15 km
$ au_0$	8.953 µs	22.384 µs	67.154 μs
$ au_1$	6.764 µs	16.910 μs	50.732 µs

Table 5.2: Travelling time of CP TL for fault with the distance [8]



Figure 5.8: TWIA arriving to terminal k due to the AG fault at the distances of DF = 2 km, DF = 5 km, and DF = 15 km. The origin of time axis is the instant of initiating fault [8]

# 5.5 CP-Based TWFL HIL Test

Based on the conclusion drawn from the previous section, a CP-based HIL setup is considered to test the double-ended scheme of TWFL. This section aims to put forward details of this testbench as well as analysis test results.

### 5.5.1 HIL TWFL Testbench Hardware

Figure 5.9 illustrates schematic view of the HIL TWFL testbench. This test system is comprised of an OP4510 HIL simulator and two SEL-T400L fault locators [100], see Figure 5.9. To accurately emulate the fault-launched TW and achieve to a reliable TWFL testbench, sub-µs time-step of simulation is required. To achieve this goal, the power system which shown in Figure 5.10 is implemented on a Kintex 7 FPGA, i.e., XC7K325T from Xilinx. The voltage and current of each TL's end under protection are sent to the relays located at each end through the D/A outputs of the simulator [8].

SEL-T400 TWFLs capture time-aligned TW on both sides of the protected TL with a sampling rate of 1 MHz and an 18-bit resolution through their AIW, BIW, and CIW inputs [8]. Relays located at each TL's ends communicate through a 1 Gbps fiber optic communication channel. According to the document from the manufacturer of the relay, the TW-based fault locating scheme of the relay has the field-proven accuracy of lower than a tower span ( $\pm 300 m$ ) [8]. The length of the protected TL and the propagation velocity of the traveling waves are properly set on to the relays to conform with the testbed specifications [8].

### 5.5.2 Power System Model

Figure 5.10 shows the power system which is considered to evaluate the HIL TW-based fault locating set up. This network has the rated frequency and voltage of 60 Hz and 535 KVRMS, respectively. The length of the TL under protection is 100 km. The power network behind each TL's termination is modeled by a Thévenin equivalent circuits with a resistance of R =0.08929  $\Omega$  and an inductance of L = 1.658 mH. A 15 km TL connects each Thévenin equivalent to the protected TL with matched impedance which avoids any reflections at buses k and m of TL under protection.

### **5.5.3 Fault Scenarios**

To perform a comprehensive analysis, four fault types are considered: phase A to ground (AG), phase B to ground (BG), phase A and B to the ground (ABG) and phase B to C (BC) [8]. The fault locations are varied for DFs ranging from 20 km to 80 km with steps of  $\Delta l = 1$  km [8]. Also, three

simulation time-steps are considered during the test, i.e., 500 *ns*, 1  $\mu s$  or 2  $\mu s$ . For each fault location, fault type and simulation time-step combination, 5 measurements are performed [8].



Figure 5.9: Schematic view of the double-ended fault locating test system, comprised of the OP4510 HIL simulator and the SEL-T400L fault locator [8]



Figure 5.10: Configuration of power system under study [8]

The fault locations are retrieved from the relay's event summary provided in the IEEE COMTRADE header files [8, 100]. A dataset comprised of 3660 fault location measurement results is created. In each measurement at each fault location, the reported fault locations of two relays are checked to ensure that the fault location provided by the relay connected to the sending end is the complement of the fault location given by the relay at the receiving end.

## 5.6 Results

#### 5.6.1 Observations

Scatter plots for all fault locating measurements are shown in Figure 5.11. Each sub-figures of Figure 5.11 is dedicated to a specific fault type, i.e., AG, BG, ABG, and BC, and illustrates the absolute fault locating errors for three different simulation time-steps. The results of these three time-step are distinguished by color and marker (black circle for  $\Delta t = 500 ns$ , blue plus sign for  $\Delta t = 1 \mu s$ , and red asterisk for  $\Delta t = 2 \mu s$ ) [8].



Figure 5.11: Scatter plots of absolute fault locating errors of the fault 20 km to 80 km away from terminal *k* of a 100 km TL. Fault types: (a): AG; (b): BG; (c): ABG; (d): BC. Each sub-plot includes three simulation time-steps:  $\Delta t = 500$  ns, 1  $\mu s$ , 2  $\mu s$  [8]

As can be observed from scatter plots of Figure 5.11, using smaller simulation time-step yields a better accuracy of the fault location, irrespective of the fault type and location [8]. On the contrary, for the simulation time-step of  $\Delta t = 2 \ \mu s$ , larger fault location errors are observed which also causes multiple outliers outside the 300 m accuracy range of relay [8].

Empirical cumulative distribution function (CDF) of the fault location errors for each of the three simulation time-steps is presented in Figure 5.12. In this figure, each empirical CDF is derived, for each time-step, from 1220 fault location errors and includes four fault types. For  $\Delta t = 500 \text{ ns}$  case, all fault locations are within an accuracy range of 40 m from the fault location. For  $\Delta t = 1 \mu s$  case, the accuracy range is increased to 300 m, but still more that 99.3% of the measured fault locations are in the accuracy range of 50 m [8]. However, in the case of  $\Delta t = 2 \mu s$ , only 88.7% of the measured fault locations are in the accuracy range of 50 m, and 1.55 % of them are located outside the 300 m error span [8].

#### 5.6.2 Analysis

It is noted that during the experiments, and for both  $\Delta t = 500$  ns and  $\Delta t = 1 \ \mu s$ , the sum of the reported fault locations by the SEL-T400L relays is always equal to 100 km [8]. In addition, the SEL-T400L relays are always tripped by the TW differential module (TW87). For  $\Delta t = 2 \ \mu s$ , however, the sum of the fault locations reported by two relays is not always equal to line length (the sum is often off by 100 m) [8]. Moreover, the tripping is performed by the TW directional module (TW32) instead of the TW87, which can be the result of the fact that the 2  $\mu s$  simulation time-step is larger than the relay's sampling rate (1 MHz) and relays can not only rely on the current signal [8].

These observations are evidenced by the reported mean and standard deviation ( $\sigma$ ) of the fault location errors listed in Table 5.3, where it clearly appears that the errors are very scattered for  $\Delta t = 2 \,\mu s$ , where larger standard deviations are encountered [8]. These observations show the importance of having a sub- $\mu s$  simulation time-step (1  $\mu s$  being acceptable) to perform an effective and reliable HIL test of TW-based fault locating relays. It is noted that the relay's fault locating accuracy is affected by tolerances in the current readings, and time stamping circuitries [8].



Figure 5.12: Empirical cumulative distribution functions of the absolute fault location error for the three simulation time-steps [8]

### 5.6.3 Fault-Location Errors

In double-ended fault locating testbench, the fault location relies on the time stamping of the ToA of incident TWs [8]. However, minor errors on the time-stamping can cause large fault location errors. In double-ended schme,  $1 \mu s$  error in time-stamping results in an error of approximately 150 m. The error associated with the three simulation time-steps can be assessed by checking the values of  $|t_{k1} - t_{m1}|$  for faults applied at the middle of protected line (here DF= 50 km), which is supposedly  $|t_{k1} - t_{m1}| = 0$ . Table 5.4 lists the actual  $|t_{k1} - t_{m1}|$  values for different fault types obtained from the SEL-T400L logs. As can be seen from this table, the error in  $\Delta t = 2 \mu s$  can be as large as  $\Delta t = 1.54 \mu s$  (equivalent to 230 m) which confirms and explains the many outliers shown in red in Figure 5.11[8]. For this time-step, the relay stamps the arrival time of the first peak with higher degree of uncertainty that can affect the fault-location accuracy [8].

Time-step	Fault Type	Mean ( <i>m</i> )	σ (m)
500 ns	AG	1.90	7.84
	BG	6.77	10.61
	ABG	4.39	7.26
	BC	2.66	5.23
	AG	0.61	12.00
1	BG	3.59	26.28
$1 \ \mu s$	ABG	2.99	14.21
	BC	4.23	6.71
	AG	6.29	75.21
2	BG	-8.49	92.39
$2 \mu s$	ABG	8.66	32.12
	BC	7.32	19.84

 Table 5.3: Mean and standard deviation of the error for all fault location measurements [8]

Table 5.4: Mean value of difference between relays ToA of TW for fault at DF = 50 km [8]

Foult	$ t_{k1} - t_{m1} $ (µs)			
raun	500 ns	1 µs	$2 \mu s$	
AG	0.0208	0.0460	1.5400	
BG	0.0690	0.1840	0.1850	
ABG	0.0330	0.0390	0.1340	
BC	0.0470	0.0692	0.1340	

Another source of error can be attributed to the interpolation required to approximate the history current term of (5.5). As mentioned earlier, the modal delays are not integer multiples of the simulation time-step and linear interpolation is employed to compute the history term values at each time-step [8].

Figure 5.13.a shows the current at Phase *a*, terminal *k* ( $i_k$ ) for a period of  $\approx 15$  ms following an AG fault applied in the middle of a 100 km TL [8]. This transient current is emulated by the simulator using three different time-steps of  $\Delta t = 500 \text{ ns}$ , 1  $\mu s$ , 2  $\mu s$  whose zoomed-in views for two different periods are shown in Figure 5.13.b and Figure 5.13.c [8]. These transient currents are updated at associated  $\Delta t$  in the simulator. Figure 5.13.d and Figure 5.13.e show the transients signals resulting from 1  $\mu s$  sampling period, that are seen at the output of the analog to digital converter (ADC) of the relay [8]. The advantage of using smaller time-steps at the model level (in the simulator) is clearly evidenced in Figure 5.13.d-e, even if the ADC uses a lower sub sampling rate [8].



Figure 5.13: Current at Phase-a, terminal k following an AG fault applied in the middle of a 100 km TL. The signals generated by the simulator are demonstrated in zoomed views for  $\Delta t = 500 \text{ ns}$ , 1  $\mu$ s, 2  $\mu$ s. The bottom figures show the points which are captured by the TWFL according to its 1 MHZ sampling rate [8]

# 5.7 FPGA-Based ULM Computing Unit

Based on the discussions presented in the preceding sections, a reliable TWFL HIL testbench should have two important features [8]; i) ability to simulate network with a sub- $\mu$ s time-step; and ii) utilizes accurate TL model. Although the ULM can be a potential candidate for this purpose, its computational burden hinders to achieve sub- $\mu$ s time-step of simulation.

## 5.8 Universal Line Model

#### 5.8.1 ULM Modelling Approach

The ULM [67] is phase-domain line model which includes full frequency dependency of the TL and cable parameters. The ULM with its various improvements [101, 102] is used in EMT-type programs and currently recognized as being the most accurate wide-band TL model. The aim of this section is to revisit the theoretical basis of the ULM [68].

General view of a typical multi-conductor line (or cable) with length l is depicted in Figure 5.14. This TL has characteristic admittance  $\mathbf{Y}_c$ , and propagation function **H**. The frequency domain relation between voltage and current at sending end k is given by:

$$\mathbf{I}_k = \mathbf{I}_{shk} - 2\mathbf{I}_{ik} \tag{5.8}$$

$$\mathbf{I}_{shk} = \mathbf{Y}_c \mathbf{V}_k \tag{5.9}$$

$$\mathbf{I}_{ik} = \mathbf{H}\mathbf{I}_{rm} \tag{5.10}$$

where  $\mathbf{I}_k$  and  $\mathbf{V}_k$  are the current and voltage at sending end k,  $\mathbf{I}_{ik}$  is the incident current, and  $\mathbf{I}_{rm}$  is the reflected current from receiving end m, see Figure 5.14.a. Similar equations are obtained for the receiving end m by interchanging subscripts k and m [68].

As shown in Figure 5.14.b, TL is represented using a Norton equivalent at each end. The Norton equivalent at sending end k is given by:

$$\mathbf{i}_k(t) = \mathbf{G}_{\mathrm{TL}} \mathbf{v}_k(t) - \mathbf{i}_k^{hist}(t)$$
(5.11)

In (5.11),  $\mathbf{G}_{\text{TL}}$  is the admittance matrix associated with TL.  $\mathbf{Y}_c$  and  $\mathbf{H}$  are TL's characteristics and are defined as:

$$\mathbf{Y}_c = \mathbf{\Gamma} \mathbf{Z}^{\prime - 1} \tag{5.12}$$

$$\mathbf{H} = e^{-\Gamma l} \tag{5.13}$$


Figure 5.14: (a): General view of a TL showing the reflected and incident currents in its terminals *k* and *m* in frequency domain; (b): The Norton equivalents of the TL in the time domain [68]

where  $\Gamma = \sqrt{\mathbf{Y}'\mathbf{Z}'}$  is propagation constant and  $\mathbf{Y}'$  and  $\mathbf{Z}'$  represent, respectively, per unit length shunt admittance and series impedance matrices of TL.

The characteristic admittance  $\mathbf{Y}_c$  and propagation function  $\mathbf{H}$  are rationally fitted in the frequency domain and are represented as:

$$\mathbf{Y}_c \cong \mathbf{G}_0 + \sum_{j=1}^{N_{Y_c}} \frac{\mathbf{R}_j}{s - p_j}$$
(5.14)

$$\mathbf{H} \cong \sum_{i=1}^{N_g} \left( \sum_{j=1}^{N_H^i} \frac{\mathbf{R}_{i,j}}{s - p_{i,j}} \right) e^{-s\tau_i}$$
(5.15)

where  $p_j \mathbf{R}_j$  and  $N_{Y_c}$ , are respectively the poles, residue matrices and the fitting order obtaining from the fitting of  $\mathbf{Y}_c$ . Likewise,  $p_{i,j}$ ,  $\mathbf{R}_{i,j}$ ,  $N_g$  and  $N_H^i$ , represent respectively the poles, the residue matrices, the number of fitting groups and the fitting order of each modal group i ( $i \in \{1, 2, ..., N_g\}$ ) resulting from the fitting of the propagation function  $\mathbf{H}$ .  $\tau_i$  denotes the minimum phase delay associated with each modal group. State vectors  $\mathbf{x}_j^{\mathbf{Y}_c}$  and  $\mathbf{x}_j^{\mathbf{H}}$  are associated with  $\mathbf{Y}_c$  and  $\mathbf{H}$ respectively and are updated at each simulation time-point according to:

$$\mathbf{x}_{j}^{\mathbf{Y}_{c}}(t+\Delta t) = \alpha_{j}^{\mathbf{Y}_{c}}\mathbf{x}_{j}^{\mathbf{Y}_{c}}(t) + \mathbf{\beta}_{j}^{\mathbf{Y}_{c}}\mathbf{v}_{k}(t)$$
(5.16)

$$\mathbf{x}_{i,j}^{\mathbf{H}}(t+\Delta t) = \alpha_{i,j}^{\mathbf{H}} \mathbf{x}_{i,j}^{\mathbf{H}}(t) + \boldsymbol{\beta}_{i,j}^{\mathbf{H}} \{ \mathbf{i}_{rm}(t-\tau_i) + \mathbf{i}_{rm}(t-\tau_i+\Delta t) \}$$
(5.17)

Based on (5.16)-(5.17) and, shunt and incident currents at the sending end k are obtained as:

$$\mathbf{i}_{shk}^{hist}(t+\Delta t) = \sum_{j=1}^{N_{Y_c}} \mathbf{x}_j^{\mathbf{Y}_c}(t+\Delta t)$$
(5.18)

$$\mathbf{i}_{ik}(t + \Delta t) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_H^i} \mathbf{x}_{i,j}^{\mathbf{H}}(t + \Delta t)$$
(5.19)

These two currents then contribute to the history term value as:

$$\mathbf{i}_{k}^{hist}(t+\Delta t) = 2\mathbf{i}_{ik}(t+\Delta t) - \mathbf{i}_{shk}^{hist}(t+\Delta t)$$
(5.20)

It is worth mentioning that in Eqs.(5.11), (5.16) and (5.17), the closed form expressions for  $\mathbf{G}_{\text{TL}}$ ,  $\alpha_j^{\mathbf{Y}_c}$ ,  $\boldsymbol{\beta}_j^{\mathbf{Y}_c}$ ,  $\boldsymbol{\alpha}_{i,j}^{\mathbf{H}}$  and  $\boldsymbol{\beta}_{i,j}^{\mathbf{H}}$  are given as:

$$\mathbf{G}_{\mathrm{TL}} = \mathbf{G}_0 + \sum_{j=1}^{N_{Y_c}} \boldsymbol{\lambda}_j \tag{5.21}$$

$$\boldsymbol{\lambda}_j = \frac{\Delta t \mathbf{R}_j}{2 - \Delta t p_j} \tag{5.22}$$

$$\alpha_j^{\mathbf{Y}_c} = \frac{2 + \Delta t p_j}{2 - \Delta t p_j} \tag{5.23}$$

$$\boldsymbol{\beta}_{j}^{\boldsymbol{Y}_{c}} = (\alpha_{j}^{\boldsymbol{Y}_{c}} + 1) \boldsymbol{\lambda}_{j}$$
(5.24)

$$\alpha_{i,j}^{\mathbf{H}} = \frac{2 + \Delta t p_{i,j}}{2 - \Delta t p_{i,j}} \tag{5.25}$$

$$\boldsymbol{\beta}_{i,j}^{\mathbf{H}} = \frac{\Delta \mathbf{t} \mathbf{R}_{i,j}}{2 - \Delta t p_{i,j}}$$
(5.26)

## 5.8.2 ULM Time Domain Simulation

The following algorithm is used to simulate the ULM in time-domain with a time-step  $\Delta t$  [68].

- Solve network equations to obtain the terminal voltages and the reflected currents at each terminal of the line for the present simulation time-point, i.e., v<sub>k</sub>(t), v<sub>m</sub>(t), i<sub>rk</sub>(t), and i<sub>rm</sub>(t);
- 2. Update the buffers storing reflected currents using the results from Step 1.
- 3. For each terminal of the line, update the state vectors  $\mathbf{x}_{j}^{\mathbf{Y}_{c}}(t + \Delta t)$  and  $\mathbf{x}_{i,j}^{\mathbf{H}}(t + \Delta t)$  using Eqs. (5.16) and (5.17) respectively.
- 4. Compute the shunt history vector  $\mathbf{i}_{sh}^{hist}(t + \Delta t)$  and the incident current vector  $\mathbf{i}_i(t + \Delta t)$  at each terminal using Eqs. (5.18) and (5.19) respectively.
- 5. Compute the total history vectors  $\mathbf{i}_{k}^{hist}(t + \Delta t)$  and  $\mathbf{i}_{m}^{hist}(t + \Delta t)$  for the next time-point using Eq. (5.20).
- 6. Set  $t = t + \Delta t$  and go to Step 1.

This algorithm is shown in Figure 5.15 and used as a foundation for the development of the proposed FPGA-based ULM simulator.



Figure 5.15: (a): Nodal Solver connected to *N* ULM Computing Engines; (b): High-level inside view of an individual ULM Computing Engine [68]

# 5.9 ULM Hardware Architecture

#### 5.9.1 FPGA-Based ULM Solver

The FPGA-based simulation of the ULM is presented in Figure 5.15.a. It is comprised of two main blocks, a Nodal Solver and a ULM Solver made of multiple instances of the ULM Computing Engine. The nodal solver composes network equations using MANA. The two solvers proceed serially, which yields the following formula for the minimum simulation time-step  $\Delta t_{min}$ :

$$\Delta t_{min} = \tau_{min} \{ \ell_{\rm NS} + \ell_{\rm ULM} \}$$
(5.27)

where  $\tau_{min}$  is the clock period that the FPGA model can handle, whereas  $\ell_{NS}$ , and  $\ell_{ULM}$  are the latencies of respectively the nodal and the ULM solvers.

A closer view of the ULM Computing Engine is given in Figure 5.15.b. It is comprised of two convolutional kernels, one for conducting the convolutions involving  $\mathbf{Y}_c$ , the other for conducting the convolutions involving  $\mathbf{H}$ . The two modules being executed in parallel, the latency of the ULM Solver  $\ell_{\text{ULM}}$  is given by:

$$\ell_{\text{ULM}} = \max_{1 \le i \le N} (\ell_{Y_{\text{c}}}^i + \ell_{\text{H}}^i)$$
(5.28)

where *N* is the number of ULM instances,  $\ell_{Y_c}^i$ , and  $\ell_H^i$  are the latencies of respectively Module  $Y_c$  and Module **H** of the ULM Computing Engine instance *i*. These latencies are implementation-specific [68].

## 5.9.2 Low Latency ULM Computing Engine

Figure 5.16 presents two alternative methods for implementing the convolutional kernels [68]. Method 1 is shown in Figure 5.16.a. It is a direct hardware implementation of Eqs. (5.16)-(5.19). The MVM and State Update blocks are used to implement Eqs. (5.16) and (5.17) in respectively the  $Y_c$  and H modules of Figure 5.16.a.

When fully pipelined, Method 1 yields:

$$\ell_{\mathbf{Y}_{\mathbf{C}}} = \ell_{\mathbf{Y}_{\mathbf{C}}}^{dp} + N_{\mathbf{Y}} \tag{5.29}$$

$$\ell_{\mathbf{H}} = \ell_{\mathbf{H}}^{dp} + N_{\mathbf{g}} \max_{1 \le i \le N_g} N_{\mathbf{H}}^i$$
(5.30)

where  $\ell_{\mathbf{Y}_{c}}$ , and  $\ell_{\mathbf{H}}^{dp}$  are the datapath latencies of respectively the  $\mathbf{Y}_{c}$  and  $\mathbf{H}$  modules.

The aim is to reduce these latencies in order to minimize the simulation time-step. This is achieved in Method 2 by means of two complementary optimizations [68]:

• First, it is noted that the MVM precedes the State Update block in Figure 5.16.a. However, the MVM can be executed in parallel to an alternative State Update block in Module  $Y_c$  if both parts are properly synchronized, which is done in Figure 5.16.b, resulting in a lower latency  $\ell_{Y_c}^{op}$ . Such an optimization is harder to obtain for Module **H** because an interpolation of past terms of  $i_{rk}$  needs to be performed, making the rescheduling difficult. Hence, the MVM and the State Update blocks are kept separate in Module **H** of Method 2.



Figure 5.16: Detailed view of the convolutional kernels for terminal k: (a): Method 1: Direct hardware implementation of Eqs. (5.16)-(5.19); (b): Method 2: Rescheduling of Eqs. (5.16)-(5.19) to reduce latency [68]

• The second optimization results from the observation that Eqs. (5.17) and (5.19) involve past terms only. Hence, the datapath latency of Module **H** could be ignored if the history buffer was moved from input to output. An input buffer would still be required nevertheless because Eq. (5.17) combines terms involving different modal delays  $\tau_i$ . However, the input buffer would be less deep than the one needed in Method 1. This optimization is shown in Module **H** of Figure 5.16.b. Hence, when fully pipelined, Method 2 yields:

$$\ell_{\mathbf{Y}_{c}} = \ell_{\mathbf{Y}_{c}}^{dp} + N_{\mathbf{Y}} \tag{5.31}$$

$$\ell_{\rm H} = \ell_{\rm H}^{ob} \tag{5.32}$$

where  $\ell_{\rm H}^{ob}$  is the latency of the output buffer, which is typically a single clock cycle.

## 5.9.3 Low-Latency Custom-Made Floating-Point Operators

To further decrease the simulation time-step, the ULM Solver is built using low-latency custommade FP operators. FXP format uses less hardware and yields a datapath with lower latency, but the number format suffers from a restricted dynamic range. On the other hand, the FP number format allows for a larger dynamic range, but its hardware arithmetic operators are costly in terms of FPGA resource consumption and require deeper pipelines than their fixed-point counterparts. This is even more true when double precision is considered. In general the following measures are considered for FPGA implementation [68]:

- A non-standard FP format with intermediate precision (between single and double) is used. More specifically, we used a 34-bit mantissa, with 8-bit exponent. This choice is a compromise between hardware consumption (only 3 DSP blocks per multiplication are be used vs. 2 for single precision and 12 for double precision) and computational accuracy.
- The arithmetic operators were built using a so-called fused-datapath (FDP) approach [103]. FDP is an approach to the implementation of complex floating-point datapaths which consists in removing all intermediate packing and unpacking stages and performing all the computations jointly within a fused-datapath. The approach does not adhere to IEEE-754 Standard but provides better resource consumption.
- All internal additions (including accumulation) are carried out using an internal FP format called the self-alignment format (SAF) [104]. SAF provides higher numerical accuracy while allowing the implementation of complex fused-path operators with very low latencies. SAF has also the ability to provide single-cycle FP accumulation, which is needed

for the summation unit of our convolutional kernels. FDP/SAF operators are clearly identified in Figure 5.16.b.

The impact of the proposed improvements on the simulation time-step are illustrated in Figure 5.17. Figure 5.17.a shows the timing diagram for Method 1. For each time-point, the Nodal Solver starts by reading **u**,  $\mathbf{i}_{sh}^{hist}$  and  $\mathbf{i}_i$ , then produces **y**, **v**, and  $\mathbf{i}_r$  after a latency of  $\ell_{NS}$ . At that moment, the **Y**<sub>c</sub> Module and **H** Module start their processing and produce  $\mathbf{i}_{sh}^{hist}$ , and  $\mathbf{i}_i$  for the next time-point of the simulation after a latency of  $\ell_{Y_c}$  and  $\ell_H$  respectively. The computations for the following time-point can only start when the results of the convolutional kernels are both available.



Figure 5.17: Timing diagrams for: (a): Method 1; (b): Method 2 [68]

Figure 5.17.b. illustrates how Method 2 manages to reduce the simulation time-step  $\Delta t$ . On the first hand, the FDP/SAF approach reduces the latency  $\ell_{Y_c}$  needed to produce  $\mathbf{i}_{sh}^{hist}$ . On the second hand, the required latency to obtain  $\mathbf{i}_i$  (i.e.,  $\ell_H$ ) is a single clock cycle. The total latency of the ULM Solver becomes  $\ell_{ULM} = \ell_{Y_c}$  [68].

# 5.10 Results

#### 5.10.1 Test Case 1: Aerial Transmission Line

The first test case considered here consists of an TL line of length 100 km, as shown in Figure 5.18. The geometry and data of the line are given in Figure 5.3 and Table 5.1, respectively. The fitting of the line parameters was performed using EMTP for 8 decades starting at  $f_{min} = 0.1$  Hz and resulted in  $N_{\rm Yc}$ ,  $N_g = 2$  and  $N_{\rm H}^i = 4$ . The line is energized by a three-phase sinusoidal source while the receiving end is left open. After reaching steady-state, a single-phase to earth fault is applied on phase-a (AG fault) at the receiving end of the line at t=0.1 s of the simulation time. The fault is cleared after 0.1s. For this test, the simulation time-step was set to 1  $\mu$ s. However, the FPGA design can achieve a time-step as small as 200 ns for this case.

Figure 5.19.a superimposes the voltages at the receiving end from the FPGA and EMTP for the first 0.25s of the simulation time. Close-up views of the fault initiation and clearing instants are given in Figure 5.19.b and Figure 5.19.c respectively. As one can see, the FPGA implementation matches perfectly the EMTP reference during steady state as well as during transients.

The computational accuracy of the ULM Solver is assessed by drawing the incident current computed by Module **H** for the receiving end of the TL and comparing to EMTP. Figure 5.20.a shows the incident current at the receiving end while Figure 5.20.b gives the computed relative errors of the FPGA-based ULM using standard single and double precision arithmetics as well as the proposed FDP/SAF. It is obvious from Figure 5.20.b that FDP/SAF brings better accuracy than strict IEEE-754 compliant single precision, while offering a lower datapath latency, a higher clock frequency at reasonable hardware cost [68].

## 5.10.2 Test Case 2: TWFL test network

Test Case 2 deals with a HIL TWFL testing. An FPGA-based ULM simulator with submicrosecond time-step capability has been shown to be adequate for the HIL testing of TWFLs in both single- and double-ended configurations. This test case considers the power system of Figure 5.21 which consists of a 100 km protected line surrounded by two TLs of 50 km connected to lumped equivalent networks at each end. Each equivalent network consists of 3-phase resistance and inductance of  $R = 0.08929 \Omega$  and L = 1.658 mH, and a 500 kV RMS, 60 Hz 3-phase voltage source. An AG fault is applied at the distance DF = 20 km from the sending end of the protected TL at t = 0.191 s of the simulation time.



Figure 5.19: Test Case 1: an AG fault is applied at receiving end at t = 100 ms and is cleared at t = 200 ms. (a): Receiving end phase voltages; (b): Close-up view of phase-voltages during fault initiation; and (c): Close-up view of phase voltages during fault clearing [68]

Figure 5.22.and Figure 5.22.b show the terminal currents at both ends of the protected line, as executed by the FPGA model. For this test, the simulation time-step was set to 500 ns. The TWs captured at both ends are shown in Figure 5.22.c. As one can see, the relative arrival time of the first two peaks at two terminals can be used to accurately locate the fault in a double-ended configuration:  $202 \ \mu s \equiv DF = 20.003 \text{ km}.$ 

Alternatively, the relative arrival time of the first two peaks of the TW seen at the sending end (*k*) can be used to locate the fault in a single-ended configuration: 135  $\mu s \equiv DF = 20.047$  km. Hence, the FPGA model produced accurate transients that allowed testing single- and double-ended TWFL configurations [68].

## 5.10.3 Area Occupation and Speed Performance

Table 5.5 gives the FPGA resource consumption of the two ULM test cases clarified for each Module  $\mathbf{Y}_c$  and Module  $\mathbf{H}$  as well as the ULM Solver. The designs are fully pipelined and can sustain a clock frequency of up to 250 MHz.



Figure 5.20: Assessment of the accuracy of the FPGA-based simulation. (a): Incident current at the receiving end from test case 1, phase-a; (b): Relative errors compared to EMTP [68]



Figure 5.21: Network for the TWFL test case including ULMs [68]

It is worth mentioning that the ULM Solver consists of two  $\mathbf{Y}_c$  and two **H** modules, which explains the results of the table (for instance, the ULM of Test Case 1 requires 31,042 LUTs, which is twice as much as 5,962 + 9,559). Similarly, Test Case 2 consists of four line segments, hence the numbers of Table 5.5 (124,168 =  $4 \times 31,042$ ).

It is also worth mentioning that  $\ell_{Y_c} = 16$ ,  $\ell_{NS} = 25$  for both test cases. With  $N_Y = 7$ , we get  $\Delta t_{min} = 4 \text{ ns} \times ((16 + 7) + 25) = 192 \text{ ns}$ 

As one can see from Table 5.5, a ULM Solver made of a single Computing Engines (Test Case 1) occupies a little more than 20 %, a number mainly dominated by the consumption of DSP blocks. It is worth mentioning that the FPGA considered here offers 840 DSP blocks, whereas more recent devices can provide thousands DSP blocks.



Figure 5.22: Test Case 2: an AG fault is applied at DF = 20 km (see Figure 5.21). (a): Currents from protected line at sending end; (b): Currents from protected line at receiving end; (c): Phase-a

Travelling-Wave currents from sending and receiving ends of the protected line [68]

Test Case 1															
Item	LUTs	Registers	BRAMs	DSP											
Y <sub>c</sub>	5,962 (2.9%)	6,070 (1.49%)	0 (0%)	36 (4.3%)											
Н	9,559 (4.7%)	11,004 (2.7%)	9 (2.0%)	63 (7.5%)											
ULM	31,042 (15.2%)	34,148 (8.4%)	18 (4.0%)	198 (23.6%)											
	Test Case 2														
Item	LUTs	Registers	BRAMs	DSP											
Y <sub>c</sub>	23,848 (11.7%)	24,280 (6.0%)	0 (0%)	144 (17.1%)											
Н	38,236 (18.8%)	44,016 (10.8%)	36 (8.1 %)	252 (30.0%)											
ULM	124,168 (60.9%)	136,592 (33.51 %)	72 (16.2 %)	792 (94.3%)											

Table 5.5: ULM test cases FPGA footprints [68]

## **5.11Conclusion**

In the first part of this chapter, the suitability of the CP TL model in the TWFL setup has been evaluated. A HIL testbench comprised of two SEL T400L relays and the OPAL-RT simulator was evaluated for the double-ended fault locating scheme. Different fault types have been applied to different locations using different simulation time-steps. Two important observations have been made from this experiment: i) TL models which are more rigorous than the CP model are required, and ii) RTS should have a sub- $\mu$ s time-step to meet the relay's sampling rate and to reduce the undesirable effects resulting from interpolating history terms. With that said, the second part of this chapter was dedicated to developing a low-latency FPGA-based ULM computing engine which achieves simulation time-step in the order of 200 ns. This is obtained through: i) increasing the clock frequency of FPGA and increasing the pipeline, and ii) rescheduling of the ULM equations to obtain more FLOPS while having a lower latency. The efficacy and accuracy of the proposed FPGA-based ULM implementation was evaluated through simulation results.

#### CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

## 6.1 Summary of Thesis

This purpose of this thesis is to develop methods to perform fast real-time simulation of networks including fast transients whose require sub-microsecond time-steps. These methods are of particular interest for FPGA-based real-time simulation primarily due to its high parallelism capability. The proposed algorithms targeted two increasingly topical applications of HIL real-time simulation, namely, high switching frequency converters, and travelling-wave fault locating applications.

This thesis developed a direct mapped method (DMM) for the fast and accurate modelling of systems including fast switching frequency converters operating in close electrical proximity. The DMM achieves very small simulation time-step without compromising the simulation accuracy. This is crucial to faithfully reproduce high frequency waveforms due to switching ripples or resonance behaviour, during the HIL test of converter. In addition, it has more deterministic computational time compared to an iterative solution. A dedicated FPGA hardware has been designed to implement the proposed algorithm with a very low latency and hardware footprint. The proposed FPGA-based simulator improves RTS of HSF PECs in terms of accuracy, reliability, scalability, and hardware cost. The proposed DMM and its mathematical foundation have been elaborately discussed in Chapter 3. The DMM obviates the need for iterative solver by directly mapping PEC states to diode status. Based on the DMM, FPGA-based simulation of a 500 kHz LLC and a 100 kHz resonant boost converter achieved time-steps of 25 ns and 62.5 ns, respectively. In Chapter 4, the DMM has been extended to systems of multi-HSF PECs. The customized hardware to implement the proposed method has been presented by using EVBC as the test case for implementation on FPGA. In the hardware design process, the FXP number format was utilized with all matrices and inputs normalized. It was shown that this approach can further reduce simulator's latency. A physical setup was built based on the proposed method and its results are presented in Chapter 4. The contents presented in Chapters 3 and 4 are original contributions of this thesis in the context of RTS of HSF PECs.

This thesis also addressed another emerging topic in the field of FPGA-based HIL test, the TWbased relay testing. This type of relay operates based on time-domain high frequency transients to protect a TL and/or locate the fault. A comprehensive evaluation of this setup has been presented in Chapter 5, for the first time. The CP-based HIL test setup was built and tested for different system conditions. It was shown that this model cannot fulfill all the requirements of TW relay testing. To circumvent the inadequacy of CP line models for real-time TW-based relay testing, a fast FPGA-based ULM model has been proposed. The model is customized for FPGA and can execute all the ULM computations in 200 ns. A customized FP format was utilized for the FPGA implementation.

## 6.2 List of Publications

1) **Hossein Chalangar**, Tarek Ould-Bachir, Keyhan Shesheykani, and Jean Mahseredjian, "Methods for the Accurate Real-Time Simulation of High Frequency Power Converters", IEEE Transactions on Industrial Electronics, pp. 1-1, 2021.

2) **Hossein Chalangar**, Tarek Ould-Bachir, Keyhan Shesheykani, and Jean Mahseredjian, "A Direct Mapped Method for Accurate Modeling and Real-Time Simulation of High Switching Frequency Resonant Converters", IEEE Transactions on Industrial Electronics, vol. 68, no. 7, pp. 6348-6357, 2021.

3) Tarek Ould-Bachir\*, **Hossein Chalangar**\*, Keyhan Shesheykani, and Jean Mahseredjian, "High performance computing engines for the FPGA-based simulation of the ULM", Electric Power Systems Research, vol. 190, 2021.

\* These authors contributed equally to this work.

4) **Hossein Chalangar**, Tarek Ould-Bachir, Keyhan Shesheykani, Shijia Li, and Jean Mahseredjian, "Evaluation of a Constant Parameter Line-based TWFL Real-Time Testbed", IEEE Transactions on Power Delivery, vol. 35, no. 2, pp. 1010-1019, 2020.

## 6.3 Future Work

To further complete the ideas and methods presented in this thesis, the following topics are suggested for future studies:

#### 1) Creating a nano-second time-step FPGA-based RT library based on the DMM

FPGA-based HIL RTS must comply with the requirements of HSF PECs systems. With this regard, a reliable FPGA-based library which offers both accuracy and small time-step should be developed. The DMM is the cornerstone of this library to avoid iterative solution and achieve very small simulation time-steps. To avoid time-consuming process of finding DMM function, mapping functions of different HSF converters, such as resonant converter, rectifier, buck, boost, etc., are precomputed and added to the library. An automatic algorithm will separate each converter from the circuit and network equations will be formed based on the NTT. A dedicated fixed hardware will be designed for the solution of network and DMM equations in real-time.

### 2) Fixed network matrix in combination with DMM

As an alternative to the two well-known behavioral switch models, RSM and ADC, a Norton switch model with an arbitrary fixed conductance value can be considered. The nonlinear characteristics of the switch can be modeled with the current source which has an implicit relation with switch voltage and current, unlike ADC switch model. For such a switch model, the determination of the value of current source at each time-point is challenging. The DMM can be used to overcome this problem by establishing a map between state variables and current source values at each time-point.

### 3) Multi-FPGA or heterogeneous computing platform

For FPGA implementation of large on-board power systems such as more electric aircraft networks, multi-FPGA platform must be utilized due to limited computing resources of a single FPGA. To this aim, the modeling and implementation details such as partitioning, communication, synchronization, etc., must be evaluated. Moreover, in the presence of slow dynamic models, the possibility of using CPU-FPGA (heterogenous platform) needs to be investigated.

#### 4) ULM-based TW-based relay HIL testing setup

A HIL TW-relay testbench can be built based on the ULM engine proposed in Chapter 5. To perform similar stochastic study as CP line-based testbench, the model should have the ability to arbitrary change the fault location without the need to stop simulation and perform time-consuming tasks including recompiling hardware code and changing bitstream of the FPGA.

#### 5) Evaluation of using FD line model in TW-based relay HIL testing setup

The frequency-dependent line model has a lower computational cost compared to the ULM and it achieves a small time-step of simulation. Thus, it can be used instead of CP for the TWFL HIL application. The suitability of the FD model for the TWFL HIL application should be investigated for different cases. In addition, applying more accurate integration methods for transmission lines can also be a subject of future work.

#### REFERENCES

- [1] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis Of Electronic Circuits: Algorithms And Computational Techniques*. New Jersey: Prentice-Hall, 1975.
- [2] H. W. Dommel, *EMTP Theory Book*. Microtran Power System Analysis Corporation, 1992.
- [3] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian, and L. Gérin-Lajoie, "On a new approach for the simulation of transients in power systems," *Electric Power Systems Research*, vol. 77, no. 11, pp. 1514-1520, 2007.
- [4] J. Mahseredjian, U. Karaagac, S. Dennetiere, and H. Saad, "Simulation of electromagnetic transients with EMTP-RV," in *Numerical Analysis of Power System Transients and Dynamics*, A. Ametani, Ed.: The Institution of Engineering and Technology (IET), 2015.
- [5] O. Nzimako and R. Wierckx, "Modeling and Simulation of a Grid-Integrated Photovoltaic System Using a Real-Time Digital Simulator," *IEEE Transactions on Industry Applications*, vol. 53, no. 2, pp. 1326-1336, Mar-Apr 2017.
- [6] Y. J. Kim and J. H. Wang, "Power Hardware-in-the-Loop Simulation Study on Frequency Regulation Through Direct Load Control of Thermal and Electrical Energy Storage Resources," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2786-2796, Jul 2018.
- [7] J. Wang, Y. Song, W. Li, J. Guo, and A. Monti, "Development of a Universal Platform for Hardware In-the-Loop Testing of Microgrids," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2154-2165, 2014.
- [8] H. Chalangar, T. Ould-Bachir, K. Sheshyekani, S. J. Li, and J. Mahseredjian, "Evaluation of a Constant Parameter Line-Based TWFL Real-Time Testbed," *IEEE Transactions on Power Delivery*, vol. 35, no. 2, pp. 1010-1019, Apr 2020.
- [9] "NETOMAC real-time simulator a new generation of standard test modules for enhanced relay testing," 2004.
- [10] D. Majstorovic, I. Celanovic, N. D. Teslic, N. Celanovic, and V. A. Katic, "Ultralow-Latency Hardware-in-the-Loop Platform for Rapid Validation of Power Electronics Designs," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4708-4716, 2011.
- [11] M. Steurer, C. S. Edrington, M. Sloderbeck, W. Ren, and J. Langston, "A Megawatt-Scale Power Hardware-in-the-Loop Simulation Setup for Motor Drives," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1254-1260, Apr 2010.
- [12] P. Saenger and M. Hilairet, "Hardware-In-the-Loop simulation of a DC-machine with INTEL FPGA boards," presented at the IECON 2018 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018.
- [13] M. Bosworth, D. Soto, M. Sloderbeck, J. Hauer, and M. Steurer, "MW-scale power hardware-in-the-loop experiments of rapid power transfers in MVDC naval shipboard power systems," presented at the 2015 IEEE Electric Ship Technologies Symposium (ESTS), 2015.

- [14] M. Milton, A. Benigni, and J. Bakos, "System-Level, FPGA-Based, Real-Time Simulation of Ship Power Systems," *IEEE Transactions on Energy Conversion*, vol. 32, no. 2, pp. 737-747, 2017.
- [15] L. Herrera, C. Li, X. Yao, and J. Wang, "FPGA-Based Detailed Real-Time Simulation of Power Converters and Electric Machines for EV HIL Applications," *IEEE Transactions on Industry Applications*, vol. 51, no. 2, pp. 1702-1712, 2015.
- [16] V. Venkataramanan, A. Mallikeswaran, and A. Srivastava, "Analysis of aircraft electric microgrid system with Auxiliary Power Unit using real time simulation," presented at the 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), Buzios, Brazil, 2015.
- [17] Z. Huang and V. Dinavahi, "An Efficient Hierarchical Zonal Method for Large-Scale Circuit Simulation and its Real-Time Application on More Electric Aircraft Microgrid," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5778 - 5786, 2019.
- [18] T. Ould Bachir, C. Dufour, J. Bélanger, J. Mahseredjian, and J. P. David, "A fully automated reconfigurable calculation engine dedicated to the real-time simulation of high switching frequency power electronic circuits," *Mathematics and Computers in Simulation*, vol. 91, pp. 167-177, 2013/05/01/ 2013.
- [19] *RTDS*. Available: https://www.rtds.com/
- [20] OPAL-RT. Available: https://www.opal-rt.com/
- [21] H. Chalangar, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, "A Direct Mapped Method for Accurate Modeling and Real-Time Simulation of High Switching Frequency Resonant Converters," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6348-6357, 2021.
- [22] M. Matar and R. Iravani, "FPGA Implementation of the Power Electronic Converter Model for Real-Time Simulation of Electromagnetic Transients," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 852-860, 2010.
- [23] M. Dagbagi, A. Hemdani, L. Idkhajine, M. W. Naouar, E. Monmasson, and I. Slama-Belkhodja, "ADC-Based Embedded Real-Time Simulator of a Power Converter Implemented in a Low-Cost FPGA: Application to a Fault-Tolerant Control of a Grid-Connected Voltage-Source Rectifier," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1179-1190, Feb 2016.
- [24] H. Jin, "Behavior-mode simulation of power electronic circuits," *IEEE Transactions on Power Electronics*, vol. 12, no. 3, pp. 443-452, May 1997.
- [25] S. Acevedo, L. R. Linares, J. R. Marti, and Y. Fujimoto, "Efficient HVDC converter model for real time transients simulation," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 166-171, 1999.
- [26] S. Y. R. Hui and C. Christopoulos, "A discrete approach to the modeling of power electronic switching networks," *IEEE Transactions on Power Electronics*, vol. 5, no. 4, pp. 398-403, 1990.
- [27] P. Pejovic and D. Maksimovic, "A method for fast time-domain simulation of networks with switches," *IEEE Transactions on Power Electronics*, vol. 9, no. 4, pp. 449-456, 1994.

- [28] S. Y. R. Hui and S. Morrall, "Generalised associated discrete circuit model for switching devices," *IEE Proceedings - Science, Measurement and Technology*, vol. 141, no. 1, pp. 57-64, 1994.
- [29] T. Maguire and J. Giesbrecht, "Small Time-step ( < 2µSec ) VSC Model for the Real Time Digital Simulator," presented at the International Conference on Power Systems

Transients (IPST), Montreal, Canada, June 19-23, 2005.

- [30] K. Wang, J. Xu, G. Li, N. Tai, A. Tong, and J. Hou, "A Generalized Associated Discrete Circuit Model of Power Converters in Real-Time Simulation," *IEEE Transactions on Power Electronics*, vol. 34, no. 3, pp. 2220-2233, 2019.
- [31] Q. Mu, J. Liang, X. Zhou, Y. Li, and X. Zhang, "Improved ADC Model of Voltage-Source Converters in DC Grids," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 5738-5748, 2014.
- [32] P. Pejovic and D. Maksimovic, "A new algorithm for simulation of power electronic systems using piecewise-linear device models," *IEEE Transactions on Power Electronics*, vol. 10, no. 3, pp. 340-348, 1995.
- [33] T. OuldBachir, H. Blanchette, and K. Al-Haddad, "A Network Tearing Technique for FPGA-Based Real-Time Simulation of Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3409 3418, 2015.
- [34] H. F. Blanchette, T. Ould-Bachir, and J. P. David, "A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4555-4567, 2012.
- [35] A. Hadizadeh, M. Hashemi, M. Labbaf, and M. Parniani, "A Matrix-Inversion Technique for FPGA-Based Real-Time EMT Simulation of Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1224-1234, 2019.
- [36] L. Chua and C. Li-Kuan, "Diakoptic and generalized hybrid analysis," *IEEE Transactions* on Circuits and Systems, vol. 23, no. 12, pp. 694-705, 1976.
- [37] J. Mahseredjian, S. Lefebvre, and D. Mukhedkar, "Power converter simulation module connected to the EMTP," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 501-510, 1991.
- [38] J. Mahseredjian, C. Dufour, U. Karaagac, and J. Bélanger, "Simulation of power system transients using state-space grouping through nodal analysis," presented at the International Conference on Power Systems Transients (IPST), Delft, the Netherlands, 2011.
- [39] C. Dufour, J. Mahseredjian, and J. Belanger, "A Combined State-Space Nodal Method for the Simulation of Power System Transients," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 928-935, 2011.
- [40] J. R. Marti, L. R. Linares, J. Calvino, H. W. Dommel, and J. Lin, "OVNI: an object approach to real-time power system simulators," presented at the POWERCON '98. 1998 International Conference on Power System Technology. Proceedings Beijing, China, 1998.
- [41] F. A. Moreira, J. R. Mart, J. L. C. Zanetta, and L. R. Linares, "Multirate Simulations With Simultaneous-Solution Using Direct Integration Methods in a Partitioned Network

Environment," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 12, pp. 2765-2778, 2006.

- [42] J. Mahseredjian, *Compensation, Diakoptics, MATE and Bordered-Block-Diagonal methods for network solution parallelization.* Polytechnique Montreal, March 18, 2019.
- [43] W. Tinney, "Compensation Methods for Network Solutions by Optimally Ordered Triangular Factorization," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-91, no. 1, pp. 123-127, 1972.
- [44] H. Dommel, "Nonlinear and Time-Varying Elements in Digital Simulation of Electromagnetic Transients," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 6, pp. 2561-2567, 1971.
- [45] J. Mahseredjian, S. Lefebvre, and X.-D. Do, "A New Method for Time-Domain Modelling of Nonlinear Circuits in Large Linear Networks," presented at the Power Syst. Comput. Conf. (PSCC), Avignon, France, 1993,
- [46] G. Hachtel, R. Brayton, and F. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," *IEEE Transactions on Circuit Theory*, vol. 18, no. 1, pp. 101-113, 1971.
- [47] K. K. Fung and S. Y. R. Hui, "Fast simulation of multistage power electronic systems with widely separated operating frequencies," *IEEE Transactions on Power Electronics*, vol. 11, no. 3, pp. 405-412, 1996.
- [48] S. C. Verma, H. Odani, S. Ogawa, K. Kuroda, and Y. Kono, "Real time interface for interconnecting fully digital and analog simulators using short line or transformer," presented at the 2006 IEEE Power Engineering Society General Meeting, 2006.
- [49] T. Kato, K. Inoue, T. Fukutani, and Y. Kanda, "Multirate Analysis Method for a Power Electronic System by Circuit Partitioning," *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 2791-2802, 2009.
- [50] L.-A. Gregoire, H. F. Blanchette, J. Belanger, and K. Al-Haddad, "Real-Time Simulation-Based Multisolver Decoupling Technique for Complex Power-Electronics Circuits," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2313-2321, 2016.
- [51] S. D. Pekarek *et al.*, "An Efficient Multirate Simulation Technique for Power-Electronic-Based Systems," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 399-409, 2004.
- [52] Z. Wang, C. Wang, P. Li, X. Fu, and J. Wu, "Extendable multirate real-time simulation of active distribution networks based on field programmable gate arrays," *Applied Energy*, vol. 228, pp. 2422-2436, 2018.
- [53] L.-A. Gregoire, H. F. Blanchette, J. Belanger, and K. Al-Haddad, "A Stability and Accuracy Validation Method for Multirate Digital Simulation," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 512-519, 2017.
- [54] B. DeKelper, H. F. Blanchette, and L. A. Dessaint, "Switching Time Model Updating for the Real-Time Simulation of Power-Electronic Circuits and Motor Drives," *IEEE Transactions on Energy Conversion*, vol. 20, no. 1, pp. 181-186, 2005.

- [55] R. Mirzahosseini and R. Iravani, "Small Time-Step FPGA-Based Real-Time Simulation of Power Systems Including Multiple Converters," *IEEE Transactions on Power Delivery*, vol. 34, no. 6, pp. 2089-2099, 2019.
- [56] T. Maguire, S. Elimban, E. Tara, and Y. Zhang, "Predicting Switch ON/OFF Statuses in Real Time Electromagnetic Transients Simulations with Voltage Source Converters," presented at the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), 2018.
- [57] H. P. William, A. T. Saul, T. V. William, and P. F. Brian, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* Cambridge University Press, 2007.
- [58] S. A. Sudha, A. Chandrasekaran, and V. Rajagopalan, "New approach to switch modelling in the analysis of power electronic systems," *IEE Proceedings B Electric Power Applications*, vol. 140, no. 2, 1993.
- [59] 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 Transactions on Power Electronics*, vol. 34, no. 4, pp. 3124-3138, 2019.
- [60] K. L. Lian and P. W. Lehn, "Real-Time Simulation of Voltage Source Converters Based on Time Average Method," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 110-118, 2005.
- [61] A. Benigni and A. Monti, "A Parallel Approach to Real-Time Simulation of Power Electronics Systems," *IEEE Transactions on Power Electronics*, vol. 30, no. 9, pp. 5192-5206, 2015.
- [62] M. Milton, A. Benigni, and A. Monti, "Real-Time Multi-FPGA Simulation of Energy Conversion Systems," *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2198-2208, 2019.
- [63] C. Liu, H. Bai, R. Ma, X. Zhang, F. Gechter, and F. Gao, "A Network Analysis Modeling Method of the Power Electronic Converter for Hardware-in-the-Loop Application," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 3, pp. 650-658, 2019.
- [64] C. Liu, H. Bai, S. Zhuo, X. Zhang, R. Ma, and F. Gao, "Real-Time Simulation of Power Electronic Systems Based on Predictive Behavior," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8044-8053, 2020.
- [65] O. Ramos-Leanos, J. L. Naredo, J. Mahseredjian, C. Dufour, J. A. Gutierrez-Robles, and I. Kocar, "A Wideband Line/Cable Model for Real-Time Simulations of Power System Transients," *IEEE Transactions on Power Delivery*, vol. 27, no. 4, pp. 2211-2218, 2012.
- [66] J. A. Martinez, B. Gustavsen, and D. Durbak, "Parameter determination for modeling system transients-Part I: overhead lines," *IEEE Transactions on Power Delivery*, vol. 20, no. 3, pp. 2038-2044, 2005.
- [67] A. Morched, B. Gustavsen, and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1032-1038, 1999.

- [68] T. Ould-Bachir, H. Chalangar, K. Sheshyekani, and J. Mahseredjian, "High performance computing engines for the FPGA-based simulation of the ULM," *Electric Power Systems Research*, vol. 190, 2021.
- [69] R. Mirzahosseini, R. Iravani, and Y. Zhang, "An FPGA-Based Digital Real-Time Simulator for Hardware-in-the-Loop Testing of Traveling-Wave Relays," *IEEE Transactions on Power Delivery*, vol. 35, no. 6, pp. 2621-2629, 2020.
- [70] G. S. A. Guzmán, Z. Sheffield, and D. Taylor, "Testing Traveling-Wave Line Protection and Fault Locators," presented at the 14th International Conference on Developments in Power System Protection, Belfast, United Kingdom, 2018.
- [71] *RTDS Technologies, TWRT: Traveling Wave Relay Testing.* Available: https://knowledge.rtds.com/hc/en-us/articles/360037052734-TWRT-Traveling-Wave-Relay-Testing
- [72] *RTDS Technologies, GTFPGA*. Available: https://knowledge.rtds.com/hc/enus/sections/360005269653-GTFPGA
- [73] OPAL-RT Technologies, Power System Protection. Available: https://www.opalrt.com/protection-system-overview/
- [74] J. Matouek and B. Gärtner, *Understanding and Using Linear Programming*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [75] G. Stehr, H. E. Graeb, and K. J. Antreich, "Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier–Motzkin Elimination," *IEEE Transactions* on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 10, pp. 1733-1748, 2007.
- [76] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, 2006.
- [77] H. Chalangar, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, "Methods for the Accurate Real-Time Simulation of High Frequency Power Converters," *IEEE Transactions on Industrial Electronics*, pp. 1-1, 2021.
- [78] D. Huang, F. C. Lee, and D. Fu, "Classification and selection methodology for multielement resonant converters," presented at the 2011 Twenty-Sixth Annual IEEE Applied Power Electronics Conference and Exposition (APEC), 2011.
- [79] Y. Bo, F. C. Lee, A. J. Zhang, and H. Guisong, "LLC resonant converter for front end DC/DC conversion," presented at the APEC. Seventeenth Annual IEEE Applied Power Electronics Conference and Exposition (Cat. No.02CH37335), 2002.
- [80] X. Xie, J. Zhang, C. Zhao, Z. Zhao, and Z. Qian, "Analysis and Optimization of LLC Resonant Converter With a Novel Over-Current Protection Circuit," *IEEE Transactions on Power Electronics*, vol. 22, no. 2, pp. 435-443, 2007.
- [81] X. Sun, Y. Shen, Y. Zhu, and X. Guo, "Interleaved Boost-Integrated LLC Resonant Converter With Fixed-Frequency PWM Control for Renewable Energy Generation Applications," *IEEE Transactions on Power Electronics*, vol. 30, no. 8, pp. 4312-4326, 2015.

- [82] H. Li, Z. Zhang, S. Wang, J. Tang, X. Ren, and Q. Chen, "A 300-kHz 6.6-kW SiC Bidirectional LLC Onboard Charger," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1435-1445, 2020.
- [83] L. E. Zubieta and P. W. Lehn, "A high efficiency unidirectional DC/DC converter for integrating distributed resources into DC microgrids," presented at the 2015 IEEE First International Conference on DC Microgrids (ICDCM), 2015.
- [84] Resonant LLC Converter:Operation and Design 250W 33Vin 400Vout Design Example. Available: https://www.infineon.com/dgdl/Infineon-Design\_example\_resonant\_LLC\_converter\_operation\_and\_design-AN-v01\_00-EN.pdf?fileId=db3a30433a047ba0013a4a60e3be64a1
- [85] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*. Springer US, 2001.
- [86] H. Bai, H. Luo, C. Liu, D. Paire, and F. Gao, "Real-Time Modeling and Simulation of Electric Vehicle Battery Charger on FPGA," presented at the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), 2019.
- [87] C. Fei, Q. Li, and F. C. Lee, "Digital Implementation of Light-Load Efficiency Improvement for High-Frequency LLC Converters With Simplified Optimal Trajectory Control," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 6, no. 4, pp. 1850-1859, 2018.
- [88] J. Fan, F. Hongtao, and S. Yaojie, "Modelling a FPGA-based LLC converter for real-time hardware-in-the-loop (HIL) simulation," presented at the 2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), 2016.
- [89] T. Gherman, D. Petreus, and R. Teodorescu, "A Real Time Simulator of a PEV's On Board Battery Charger," presented at the 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), 2019.
- [90] W. Gautschi, Numerical analysis : an introduction. MA, Boston: Birkhauser, 1997.
- [91] L. Yang and C. Q. Lee, "Analysis and design of boost zero-voltage-transition PWM converter," presented at the Proceedings Eighth Annual Applied Power Electronics Conference and Exposition, 1993.
- [92] C. Dufour and J. Belanger, "A PC-Based Real-Time Parallel Simulator of Electric Systems and Drives," in *Parallel Computing in Electrical Engineering, 2004. International Conference on*, 2004, pp. 105-113.
- [93] A. Khaligh and S. Dusmez, "Comprehensive Topological Analysis of Conductive and Inductive Charging Solutions for Plug-In Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3475-3489, 2012.
- [94] F. Musavi, M. Edington, W. Eberle, and W. G. Dunford, "Evaluation and Efficiency Comparison of Front End AC-DC Plug-in Hybrid Charger Topologies," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 413-421, 2012.

- [95] W. Haoyu, S. Dusmez, and A. Khaligh, "Design and Analysis of a Full-Bridge LLC-Based PEV Charger Optimized for Wide Battery Voltage Range," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1603-1613, 2014.
- [96] S. A. Arshadi, M. Ordonez, W. Eberle, M. A. Saket, M. Craciun, and C. Botting, "Unbalanced Three-Phase LLC Resonant Converters: Analysis and Trigonometric Current Balancing," *IEEE Transactions on Power Electronics*, vol. 34, no. 3, pp. 2025-2038, 2019.
- [97] M. O'Loughlin, "An interleaving PFC pre-regulator for high-power converters," in "Tech. Rep. slua 746," Texas Instruments2018, Available: http://www.ti.com/lit/pdf/slua746.
- [98] S. Marx, "Traveling Wave Fault Location in Protective Relays : Design, Testing, and Results," 2015.
- [99] F. V. Lopes and E. J. S. Leite, "Traveling Wave-Based Solutions for Transmission Line Two-Terminal Data Time Synchronization," *IEEE Transactions on Power Delivery*, vol. 33, no. 6, pp. 3240-3241, 2018.
- [100] "SEL-T400L Instruction Manual," ed.
- [101] I. Kocar, J. Mahseredjian, and G. Olivier, "Improvement of Numerical Stability for the Computation of Transients in Lines and Cables," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 1104-1111, 2010.
- [102] B. Gustavsen, "Passivity Enforcement for Transmission Line Models Based on the Method of Characteristics," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2286-2293, 2008.
- [103] M. Langhammer, "High performance matrix multiply using fused datapath operators," presented at the 2008 42nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2008.
- [104] T. Ould-Bachir and J. P. David, "Self-Alignment Schemes for the Implementation of Addition-Related Floating-Point Operators," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 6, no. 1, pp. 1-21, 2013.
- [105] A. Schrijver, *Theory of linear and integer programming*. West Sussex, England: John Wiley Sons Ltd, 1986.

### **APPENDIX A: SWITCH MODEL**

As discussed earlier, ADC and RSM are the two behavioral switch models which are widely employed in RTS application. In what follows, these two models are briefly reviewed.

#### 1) ADC model

ADC proceeds by modelling a switch as a Norton equivalent resulting from the discretization of L/C components: a small inductor  $L_{sw}$  when switch conducts, and a small capacitor  $C_{sw}$  when switch blocks. ADC yields a constant admittance matrix independent of switch status by setting equal conductance for *on* state inductance and *off* state capacitor, i.e.,  $g_{L_{sw}} = g_{C_{sw}} = g_{sw}$  or  $\Delta t/L_{sw} = C_{sw}/\Delta t$  (Assuming that time-domain equations of  $L_{sw}$  and  $C_{sw}$  are discretized based on the BE rule). Figure A. 1 demonstrates Norton equivalent representation of the ADC switch, where history current source value  $i_{sw}^h$  is updated as follows:

$$i_{sw}^{h}(t + \Delta t) = \begin{cases} i_{sw}(t), & s(t + \Delta t) = 0\\ -g_{sw}v_{sw}(t), & s(t + \Delta t) = 1 \end{cases}$$
(A. 1)

In this model, controlled switch logic is quite straightforward, and its status is determined by its gate signal, i.e.,  $s(t + \Delta t) = g(t + \Delta t)$ . For naturally commutated switch, previous time-point switch variables are used to update the switch status. For example, diode status is updated as,

$$s(t + \Delta t) = s(t)(i_{sw}(t) > 0) + s(t)(v_{sw}(t) > 0)$$
(A. 2)

Based on (A. 2), the update formula for different switches (such as IGBT-diode pair) can be derived as well.



Figure A. 1: ADC switch model

#### 2) RSM

As explained in section 3.2, in the RSM, uncontrolled switch status update is implicit and requires iterative solution. The update formula for such a switch reads,

$$s(t + \Delta t) = s(t)(i_{sw}(t + \Delta t) > 0) + \overline{s(t)}(v_{sw}(t + \Delta t) > 0)$$
(A. 3)

Similar formula can be developed for other type of switches. Figure A. 2 depicts RSM and its v - i characteristics. By assuming  $R_{off} = \infty$  (open circuit) and  $R_{on} = 0$  (short circuit) during *off* and *on* states, an ideal switch model is obtained. The ideal switch model is not widely used since it leads to a variable admittance matrix which does not always yield a proper tree.



Figure A. 2: RSM switch model

#### **APPENDIX B: MANA EQUATION**

In this thesis, network equations are formulated using the modified augmented nodal analysis (MANA) [3]. This method offers a more generic approach than the conventional nodal method and allows inclusion of complex models. The generic form of the MANA equation is given by,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{B. 1}$$

where  $\mathbf{x}$  is comprised of node voltages and branch currents or other type of variables, and vector  $\mathbf{b}$  contains known voltages and injected currents or other types of known quantities, including history terms. The detailed form of Eq. (B. 1) is as follows (using only currents as supplementary unknowns):

$$\begin{bmatrix} \mathbf{Y} & \mathbf{V}_{c} & \mathbf{D}_{c} & \mathbf{S}_{c} \\ \mathbf{V}_{r} & \mathbf{V}_{d} & \mathbf{V}_{VD} & \mathbf{V}_{VS} \\ \mathbf{D}_{r} & \mathbf{D}_{DV} & \mathbf{D}_{d} & \mathbf{D}_{DC} \\ \mathbf{S}_{r} & \mathbf{S}_{SV} & \mathbf{S}_{SD} & \mathbf{S}_{d} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{n} \\ \mathbf{i}_{V} \\ \mathbf{i}_{D} \\ \mathbf{i}_{S} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_{n} \\ \mathbf{v}_{b} \\ \mathbf{d}_{B} \\ \mathbf{s}_{b} \end{bmatrix}$$
(B. 2)

where **Y** is the admittance matrix,  $\mathbf{v}_n$  and  $\mathbf{i}_n$  are respectively node voltage and injected current vectors.  $\mathbf{V}_r$  ( $\mathbf{V}_r = \mathbf{V}_c^T$ ),  $\mathbf{D}_r$  ( $\mathbf{D}_r = \mathbf{D}_c^T$ ) and  $\mathbf{S}_r$  ( $\mathbf{S}_r = \mathbf{S}_c^T$ ) represent voltage sources, dependent branches (e.g., transformer) and ideal switches equations, respectively. Other sub-matrices are often equal to **0**. For more detail on MANA see [3].

MANA equation of (B. 2) can be manipulated in order to perform efficient computation on FPGA. To do so, first, an incidence matrix of **B** is introduced and the MANA equation can be rewritten as:

$$\mathbf{x}(t) = \mathbf{A}^{-1} \mathbf{B} \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{i}^{\mathbf{h}}(t) \end{bmatrix}$$
(B. 3)

In (B. 3),  $\mathbf{u}(t)$  is the vector comprised of independent sources and  $\mathbf{i}^{\mathbf{h}}(t)$  holds history terms. It is noted that in the presence of switch, to avoid matrix re-factorization, the inverse of matrix **A** can be percomputed for all switch combinations. History terms can be anticipated for the next timepoint of simulation,  $\mathbf{i}^{\mathbf{h}}(t + \Delta t)$ , since they only depend on  $\mathbf{i}^{\mathbf{h}}(t)$  and  $\mathbf{u}(t)$ :

$$\mathbf{i}^{\mathbf{h}}(t + \Delta t) = \mathbf{H}_{\mathbf{i},\mathbf{u}}\mathbf{u}(t) + \mathbf{H}_{\mathbf{i},\mathbf{i}}\mathbf{i}^{\mathbf{h}}(t)$$
(B. 4)

 $\mathbf{H}_{i,u}$  and  $\mathbf{H}_{i,i}$  are drawn by simple algebraic manipulation of MANA matrix. In addition, desired output vector,  $\mathbf{y}(t)$ , can be computed as well. Thus, the final form of the equation, which is solved on FPGA reads,

$$\begin{bmatrix} \mathbf{i}^{\mathbf{h}}(t + \Delta t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{\mathbf{i},\mathbf{u}} & \mathbf{H}_{\mathbf{i},\mathbf{i}} \\ \mathbf{H}_{\mathbf{y},\mathbf{u}} & \mathbf{H}_{\mathbf{y},\mathbf{i}} \end{bmatrix} \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{i}^{\mathbf{h}}(t) \end{bmatrix}$$
(B. 5)

#### **APPENDIX C: DISCRETE COMPANION MODEL**

The MANA equation is assembled after discretizing all elements using a proper numerical integration method, i.e., TR and the BE integration rules. In the realm of RTS of power electronics, the BE is preferable since it avoids numerical oscillation in switching events. The principle of discrete companion model is based on replacing each element with Norton equivalent [1, 2]. Figure C. 1 illustrates discrete companion models of lumped elements, resistance, capacitor and inductor. The companion model of resistance *R*, is similar to its continuous model and does not include any history current,  $i_{km}(t) = \frac{1}{R}v_{km}(t)$ .

The governing equation of an inductor L, connected between k and m is:

$$\frac{di_{km}(t)}{dt} = \frac{1}{L}v_{km}(t) \tag{C.1}$$

By applying the BE rule, the discretized form of (C. 1) is obtained as follows:

$$i_{km}(t) = \frac{\Delta t}{L} v_{km}(t) + i_{km}(t - \Delta t)$$
(C. 2)

where the term  $i_{km}(t - \Delta t)$  is the history current source of inductor  $i_{hist}$  while the inductor equivalent conductance  $g_{eq}$  is  $\frac{\Delta t}{L}$ .

The companion model of a capacitor C can be derived in a similar manner.

$$\frac{dv_{km}(t)}{dt} = \frac{1}{C}i_{km}(t) \tag{C.3}$$



Figure C. 1: (a): R, L, C elements; and (b): their companion models

Discretizing (C. 3) using the BE rule yields:

$$i_{km}(t) = \frac{C}{\Delta t} v_{km}(t) - \frac{C}{\Delta t} v_{km}(t - \Delta t)$$
(C. 4)

where the term  $-\frac{c}{\Delta t}v_{km}(t-\Delta t)$  is the history current source of the capacitor  $i_{hist}$  while the equivalent conductance  $g_{eq}$  of the capacitor is given by  $\frac{c}{\Delta t}$ . By following similar approach, it is feasible to obtain the companion models of combined RC, RL, LC, and RLC branches.

EMTP uses TR as the main discretization method; and halved time-step BE method is used to avoid numerical oscillation caused by discontinuities of state variable [3].

### **APPENDIX D: FOURIER-MOTZKIN ELIMINATION**

In the context of linear inequalities, the Fourier–Motzkin elimination (FME) can be assumed similar to the Gaussian elimination, and it eliminates a variable from a system of linear inequalities. The FME method determines whether a linear system of inequalities has a solution, and if so, finds one of its solution [105]. Geometrically, the FME proceeds by projecting all inequalities along a specific axis in order to remove a variable from the set. By doing so, a set of inequalities with m variables is reduced to a new set of m - 1 variables. If this process is successively repeated, an equivalent system of inequalities with only one variable will be obtained and is assessed to check the feasibility of the original set [75]. In addition, from any solution of the final system of inequalities, a solution for the original system can be derived by backward substitution of removed variables [105]. The FME method is a good alternative to find feasible switch combinations in the DMM. This section aims to provide a general overview of the FME.

The solution of *n* linear inequalities with *m* variables is formed by the intersection of half-spaces; and can be defined as a polyhedron p in  $\mathbb{R}^m$  space:

$$p = \{ \mathbf{x} \in \mathbb{R}^m \mid \mathbf{A}\mathbf{x} \le \mathbf{b} \}$$
(D. 1)

where **x** is a set of *m* variables, **A** is the coefficient matrix in  $\mathbb{R}^{n \times m}$  space, and **b** is a constant vector in  $\mathbb{R}^n$  space.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{1}^{\mathsf{T}} \\ \vdots \\ \mathbf{a}_{n}^{\mathsf{T}} \end{bmatrix}; \ \mathbf{a}_{1}^{\mathsf{T}} = \begin{bmatrix} a_{i1} & \cdots & a_{im} \end{bmatrix}$$
(D. 2)

$$\mathbf{b} = [\mathbf{b}_1 \ \mathbf{b}_2 \dots \mathbf{b}_b]^T \tag{D. 3}$$

For  $k \in m$ ,  $p^k = \{(x_1, x_2, x_{k-1}, x_{k+1}, \dots, x_m) \in \mathbb{R}^{m-1} | (x_1, x_2, \dots, x_m) \in p\}$  is the projection of p along the  $x_k$ -axis. This projection refers to the calculation of new inequality set with  $a_{ik} = 0$  and is mainly realized by utilizing three important inequality properties:

$$\begin{cases} (\alpha_1 \le \beta_1) \land (\alpha_2 \le \beta_2) \Rightarrow \alpha_1 + \alpha_2 \le \beta_1 + \beta_2 \\ (\alpha \le \beta) \land (\gamma \ge 0) \Rightarrow \alpha\gamma \le \beta\gamma \\ (\alpha \le \beta) \land (\gamma \le 0) \Rightarrow \alpha\gamma \ge \beta\gamma \end{cases}$$
(D.4)

The process of obtaining  $p^k$  has, in general, two steps:

1) Sorting: At first step, each inequality must be sorted according to the sign of  $x_k$  coefficients as follows:

$$\begin{cases} I_{+} = \{ \mathbf{a}_{P}\mathbf{x} \le \mathbf{b}_{P} \mid a_{Pk} > 0, P \in S_{+} \} \\ I_{-} = \{ \mathbf{a}_{N}\mathbf{x} \le \mathbf{b}_{N} \mid a_{Nk} < 0, N \in S_{-} \} \\ I_{0} = \{ \mathbf{a}_{E}\mathbf{x} \le \mathbf{b}_{E} \mid a_{Ek} = 0, E \in S_{0} \} \end{cases}$$
(D. 5)

where:

$$\begin{cases} S_{+} \cap S_{-} = S_{+} \cap S_{0} = S_{-} \cap S_{0} = \emptyset \\ S_{+} \cup S_{-} \cup S_{0} = \{1, 2, \dots, n\} \end{cases}$$
(D. 6)

It is noted that inequalities with  $a_{Ek} = 0$ , represent hyperplanes which are parallel to  $x_k$ -axis.

2) *Eliminating*: In this step, all pairwise linear combinations of  $I_+$  and  $I_-$  are obtained. Since  $I_0$  does not contain  $x_k$  variable, it is added to the resulting set:

$$I_0 \cap \left\{ \left( a_{jk} \mathbf{a}_i - a_{ik} \mathbf{a}_j \right) \mathbf{x} \le \left( a_{jk} \mathbf{b}_i - a_{ik} \mathbf{b}_j \right) \middle| i \in S_+ \land j \in S_- \right\}$$
(D. 7)

In this stage, a new set of inequalities with  $k^{th}$  variable removed can be shown as:

$$\mathbf{A}_k \mathbf{x}_k \le \mathbf{b}_k \tag{D. 8}$$

It is noted that (D. 8) is independent of  $x_k$ .

This process is repeated until m - 1 variables are removed. Then, it is trivial to determine whether the resultant set of inequalities is feasible or not. According to the FME, if there is a contradiction in the resulting set (e.g.,  $x_i > 0$  and  $-x_i > 0$ ), the original system of inequalities is *infeasible*. It is worth mentioning that during this two-step process, the process is terminated if any contradiction is detected. In this process, several redundant inequalities are generated and should be removed prior to each step \_\_\_\_\_.

# **APPENDIX E: DMM TCAM-RAM CONTENT FOR EVBC**

Table E. 1 TCAM-RAM for stage#1 of EVBC circuit.

																Т	CAM																
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$	$p_{10}$	$p_{11}$	$p_{12}$	<i>p</i> <sub>13</sub>	<i>p</i> <sub>14</sub>	<i>p</i> <sub>15</sub>	<i>p</i> <sup>16</sup>	<i>p</i> <sub>17</sub>	$p_{18}$	$p_{19}$	$p_{20}$	$p_{21}$	<i>p</i> <sub>22</sub>	<i>p</i> <sub>23</sub>	<i>p</i> <sub>24</sub>	<i>p</i> <sup>25</sup>	<i>p</i> <sup>26</sup>	p <sub>27</sub>	$p_{28}$	<i>p</i> <sup>29</sup>	$p_{30}$	$p_{31}$	<i>p</i> <sub>32</sub>	addr.	$\sigma_{fd}$
-1	-1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
-	-	-	+1	-1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
-	-	+1	-	-	-	-	-1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2
-	-	-	-	-	-	+1	-	-	+1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	3
-	-	-	+1	-	-	-	-	-	-	-	-	-1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	24
-	-	-	-	-	+1	-	-	-	-	-	-	-	-	+1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	25
-	-	-	-	-	-	-	-	+1	-	-	-	-	+1	-	-	-	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	6	26
-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	+1	-	+1	-1	-	-	-	-	-	-	-	-	-	-	-	-	7	27
+`	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1	-1	-1	-	-	-	-	-	-	-	-	-	8	36
-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-1	-1	-	-	-	-	-	-	-	9	37
-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-1	-1	-	-	-	-	-	10	38
-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	+1	-1	-	-	-	-	11	39
-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-1	-1	-	-	12	60
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	+1	-1	-	13	61
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-	-	+1	-	-	-1	14	62
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-	-	+1	+1	15	63
													+1:1	Positive	sign, -1	: Negati	ve sign,	-: Doi	n't care													RA	М

															TCAM	ſ															
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$	$p_{10}$	$p_{11}$	<i>p</i> <sub>12</sub>	<i>p</i> <sub>13</sub>	<i>p</i> <sub>14</sub>	<i>p</i> <sub>15</sub>	<i>p</i> <sub>16</sub>	<i>p</i> <sub>17</sub>	$p_{18}$	<i>p</i> <sub>19</sub>	$p_{20}$	<i>p</i> <sub>21</sub>	<i>p</i> <sub>22</sub>	<i>p</i> <sub>23</sub>	<i>p</i> <sub>24</sub>	$p_{25}$	$p_{26}$	p <sub>27</sub>	$p_{28}$	<i>p</i> <sub>29</sub>	<i>p</i> <sub>30</sub>	addr.	$\sigma_{fd}$
-1	-1	-1	-1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
-	-	-	-	-	+1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
-	-	-	-	+1	-	-	-	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2
-	-	-	+1	-	-	-	-	-	-	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	4
-	-	-	-	-	-	-	-	-	+1	-	+1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	6
-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	8
-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-1	-1	-	-	-	-	-	-	-	-	-	-	-	-	6	9
-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1	-1	-	-	-	-	-	-	-	-	-	-	7	16
-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	+1	-1	-1	-	-	-	-	-	-	-	-	8	18
-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	9	22
-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	+1	-	-	-	-1	-1	-	-	-	-	-	-	10	24
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	+1	-	-	-	-	-	-	11	25
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	+1	-	-	-	-	-	-	-	12	26
+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-1	-1	-	-	-	-	13	32
-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-1	-1	-	-	14	33
-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-1	-1	15	36
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	+1	16	37
-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	17	38
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-	-	-	+1	-	-	-	18	41
												+1: Posi	tive sign	., -1: Ne	egative si	ign, -: I	Oon't car	e												RA	M

Table E. 2 TCAM-RAM for stage #2 of EVBC circuit.