

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MANUSCRIPT-BASED THESIS PRESENTED TO  
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
Ph.D.

BY  
Luc-André GRÉGOIRE

MULTI-RATE REAL-TIME SIMULATION OF MODULAR MULTILEVEL CONVERTER  
USING CPU AND FPGA.

MONTREAL, MAY 5<sup>TH</sup>, 2016



Luc-André Grégoire, 2016



This Creative Commons license allows readers to download this work and share it with others as long as the author is credited. The content of this work cannot be modified in any way or used commercially.

## **BOARD OF EXAMINERS**

**THIS THESIS HAS BEEN EVALUATED**

**BY THE FOLLOWING BOARD OF EXAMINERS:**

Mr. Kamal Al-Haddad, Thesis Supervisor  
Department of Electrical Engineering, École de technologie supérieure

Mr. Handy Fortin Blanchette, Thesis Co-supervisor  
Department of Electrical Engineering, École de technologie supérieure

Mrs. Ruxandra Botez, Chair, Board of Examiners  
Department of Automated Manufacturing Engineering , École de technologie supérieure

Mr. Pierre Jean Lagacé, Examiner  
Department of Electrical Engineering, École de technologie supérieure

Mr. Innocent Kamwa, External Examiner  
Institut de Recherche d'Hydro-Québec

**THIS THESIS WAS PRESENTED AND DEFENDED**

**IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC**

**ON APRIL 26<sup>TH</sup>, 2016**

**AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**



## **ACKNOWLEDGEMENTS**

I wish to first thank my supervisor, professor Kamal Al-Haddad, for his support and guidance during the last few years. Special notice should be made to professor Handy Fortin-Blanchette, co-supervisor of this thesis, for following my work on daily bases.

I would also like to acknowledge my colleagues at Groupe de Recherche en Électronique de Puissance et Commande Industrielle (GREPCI) for providing a flourishing work environment.

Finally, all this work would not have been possible without the support of my family. I thank my wife, Julia, for her support, her patience and her understanding for all the hours I spent at work.



# **SIMULATION EN TEMPS-RÉEL D'UN CONVERTISSEUR MULTINIVEAUX DE TYPE MULTICELLULAIRE UTILISANT UNE SIMULATION MULTI-TAUX.**

Luc-André GRÉGOIRE

## **SUMMARY**

Cette thèse présente la simulation temps-réel des convertisseur modulaire multiniveaux (MMC) utilisant un réseau de portes programmables in situ (Field Programmable Gates Array ou FPGA). La réalisation d'un tel projet représente de nombreux défis en raison du nombre élevé de composants d'électronique de puissance du convertisseur. Le choix de la plateforme de simulation est justifié par cette problématique particulière. En utilisant un FPGA, un très grand nombre d'entrées et de sorties peuvent être facilement utilisées. En simulant le convertisseur sur FPGA, la latence entre les entrées/sorties et le modèle simulé est réduite. De plus, les FPGA permettent de grandement réduire le pas de simulation, ce qui assure un haut niveau de précision sur la détection des signaux de gâchette du convertisseur. Le FPGA est seulement utilisé pour la simulation du convertisseur, alors que le reste du modèle, comme le réseau CA et CC, est simulé sur microprocesseur (CPU). Ceci permet l'utilisation de logiciel spécialisé avec de grandes bibliothèques de composants. En utilisant deux plateformes de simulation, CPU et FPGA, ce qui requière un modèle découplé électriquement, mais aussi simulé en utilisant différents pas de calcul.

Cette thèse débute par la présentation de la problématique. Le pas de calcul minimum pour la simulation est ensuite démontré. Afin d'atteindre un pas de calcul aussi faible, une méthode de découplage et sa validation sont proposées. Cette méthode est ensuite généralisée pour être appliquée à la simulation multitaux. En utilisant ces outils développés, une implémentation détaillée du convertisseur est proposée en utilisant les simulateurs temps-réel d'OPAL-RT technologies. Finalement, la validation du modèle est présentée.

**Mots clés:** Simulation Temp-Réel, Discrétisation Multi-taux, Multi-niveau, FPGA, Stabilité numérique





# **MULTI-RATE REAL-TIME SIMULATION OF MODULAR MULTILEVEL CONVERTER USING CPU AND FPGA.**

Luc-André GRÉGOIRE

## **ABSTRACT**

This thesis presents the real-time simulation of a modular multilevel converter (MMC) using Field Programmable Gates Array (FPGA). Undertaking such a project raises challenges due to the very high number of components in MMC. The choice of the hardware used is justified by this particular problematic. Using FPGA, a very large number of inputs and outputs can be easily managed. By simulating the converter on FPGA reduces latency and the delays between the IOs and the MMC. It also allows using very small time-step ensuring accuracy for pulses detection. Only the converter is simulated on FPGA and the remaining component of the simulation, such as the AC system and its distribution network are simulated on CPU. Doing so gives the user access to large library of component from commercial software. Using two distinct platforms, CPU and FPGA, then requires the model not only to be decoupled, but also to use different sampling time.

This thesis debuts by a presentation of the problematic. Then, the required sampling time for accurate simulation of MMC is demonstrated. In order to achieve such a small time-step, a decoupling method and its validation is proposed. The method is then generalized and applied to multi-rate simulation. Using those methods, a details implementation of the converter, using OPAL-RT technologies real-time simulator, is given. Finally, numerical and experimental validation of this model are presented.

**Keywords:** Real-time simulation, Multi-rate simulation, Discrete solver, decoupling, FPGA, Multilevel, Modular multilevel converter



## CONTENTS

	Page
INTRODUCTION .....	1
CHAPTER 1 REAL-TIME SIMULATION OF MODULAR MULTI-LEVEL CONVERTERS (MMC'S) .....	3
1.1 Introduction .....	3
1.1.1 Industrial applications of MMC converters .....	4
1.1.2 Constraint introduced by RT simulation of power electronics converter in general .....	5
1.1.3 MMC Topology presentation .....	8
1.1.4 Constraints of simulating MMC converters .....	11
1.1.4.1 Solving large state-space system .....	11
1.1.4.2 Solving I/O management problem .....	11
1.1.4.3 System-wide simulation simulated faster than real-time .....	12
1.2 Choice of modeling for MMC and its limitations .....	13
1.2.1 Detailed model .....	13
1.2.2 Switching function .....	14
1.2.3 Average model .....	15
1.3 Hardware technology for real time simulation .....	16
1.3.1 Simulation using micro-processors .....	16
1.3.2 Simulation using parallel programming with FPGA devices .....	17
1.4 Implementation for real-time simulator using different approach .....	20
1.4.1 Sequential programming for average model algorithm .....	21
1.4.2 Parallel programming for switching function algorithm .....	24
1.5 Conclusion .....	26
CHAPTER 2 MODULAR MULTILEVEL CONVERTERS OVER-VOLTAGE DIAGNOSIS AND REMEDIAL STRATEGY DURING BLOCKING SEQUENCES .....	29
2.1 Introduction .....	29
2.2 MMC basic topology and control strategy .....	31
2.3 Analytical Study For Failing and Faults Conditions .....	33
2.4 Simulation results .....	40
2.5 Experimental Results .....	42
2.6 Remedial Strategy .....	43
2.7 Conclusion .....	47
2.8 Acknowledgement .....	47

CHAPTER 3	REAL-TIME SIMULATION BASED MULTI-SOLVER DECOUPLING TECHNIQUE FOR COMPLEX POWER ELECTRONICS CIRCUITS .....	49
3.1	Introduction .....	49
3.2	Time and solvers constraints of real-time simulation .....	51
3.2.1	Deterministic simulation .....	52
3.2.2	Numerical integration method .....	53
3.3	Multi-solver method .....	54
3.3.1	Introduction to multi-solver .....	54
3.3.2	Testing stability of multi-solver method applied to large network .....	57
3.3.3	Applying multi-solver method to large network .....	60
3.4	Numerical example .....	61
3.5	Conclusion .....	69
3.6	Acknowledgement .....	69
CHAPTER 4	GENERALIZED VALIDATION METHOD FOR MULTI-RATE DIGITAL SIMULATION .....	71
4.1	Introduction .....	72
4.2	Mutli-rate simulation .....	74
4.2.1	Proposed state-space matrix representation .....	75
4.3	Numerical example .....	78
4.3.1	Multi-rate simulation results .....	83
4.4	Conclusion .....	86
4.5	Acknowledgement .....	86
CHAPTER 5	CONVERTER MODELING FOR MULTI-RATE/REAL-TIME .....	87
5.1	Parallel and multi-rate simulation .....	89
5.2	Sub-Module .....	92
5.3	Discrete power system model .....	97
5.4	Discrete converter .....	98
5.5	Numerical stability and accuracy .....	102
5.5.1	Pole location analysis .....	103
5.5.2	Conducting mode .....	103
5.5.3	Blocking mode .....	105
5.6	Simulation results .....	107
5.6.1	Steady-state .....	108
5.6.2	Converter initialization .....	110
5.6.3	DC glitch .....	113
5.7	Conclusion .....	114
GENERAL CONCLUSION	.....	115
APPENDIX I	.....	117

BIBLIOGRAPHIE .....	123
---------------------	-----



## LIST OF TABLES

	Page
Table 1.1	Switching function of MMC SM..... 25
Table 2.1	Experimental Setup Parameters ..... 44
Table 3.1	Simulation parameters of the circuit ..... 62
Table 3.2	Poles for different methods ..... 65
Table 4.1	Simulation parameters ..... 79
Table 4.2	Poles for different methods ..... 83
Table 5.1	Model parameters of figure 5.1 ..... 102
Table 5.2	Poles of the system during conducting mode ..... 105





## LIST OF FIGURES

	Page
Figure 1.1	Circuit illustrating state-space versus nodal approach ..... 7
Figure 1.2	a) Single MMC SM with its protection b) MMC limb ..... 8
Figure 1.3	Standard exemple of one MMC arm for HVDC link ..... 9
Figure 1.4	THD vs the number of levels ..... 10
Figure 1.5	Example of logic propagation ..... 18
Figure 1.6	Chronogram of time-multiplexed process ..... 18
Figure 1.7	FPGA integrator using pipelining ..... 19
Figure 1.8	Chronogram of pipelined integrator..... 20
Figure 1.9	Equivalent decoupled circuit ..... 21
Figure 1.10	Block diagram for average MMC model..... 22
Figure 1.11	CIGRE B4-57 HVDC grid ..... 23
Figure 1.12	CIGRE benchmark AC fault ..... 24
Figure 1.13	Block diagram of FPGA implementation..... 25
Figure 1.14	Real-time simulator with IOs chassis ..... 27
Figure 1.15	Results of a step on power reference ..... 27
Figure 2.1	A typical half-bridge cell ..... 32
Figure 2.2	Schematic of a 3 phase MMC showing n level ..... 32
Figure 2.3	a) Sequence 1 : $I_{AU}<0$ , $I_{BU}>0$ & $I_{CU}>0$ b) Sequence 4 : $I_{AL}<0$ , $I_{BL}>0$ & $I_{CL}>0$ ..... 35
Figure 2.4	a) Sequence 2 : $I_{AU}<0$ , $I_{BU}<0$ & $I_{CU}>0$ b) Sequence 5 : $I_{AL}<0$ , $I_{BL}<0$ & $I_{CL}>0$ ..... 35
Figure 2.5	a) Sequence 3 : $I_{AU}=0$ , $I_{BU}<0$ & $I_{CU}>0$ b) Sequence 6 : $I_{AL}=0$ , $I_{BL}<0$ & $I_{CL}>0$ ..... 36

Figure 2.6	Laplace equivalent circuit of sequence 1 and 4.....	37
Figure 2.7	Laplace equivalent circuit of sequence 2 and 5.....	37
Figure 2.8	Laplace equivalent circuit of sequence 3 and 6.....	38
Figure 2.9	Simulation results of a blocking sequence.....	41
Figure 2.10	Photograph of the 3.25kVA experimental prototype of a five-level MMC .....	43
Figure 2.11	Limb currents and DC bus voltage transients at blocking as obtained from experimental setup.....	45
Figure 2.12	Control scheme for a single limb to block at zero-crossing.....	45
Figure 2.13	Timing diagram to block at zero-crossing .....	46
Figure 2.14	Limb currents and DC bus voltage for blocking at zero-crossing for each limb .....	46
Figure 3.1	Execution timeline of real-time simulation a) using a single core b) using multi-core.....	52
Figure 3.2	Example of two networks coupled with a transmission lines .....	57
Figure 3.3	Companion model a) using equivalent current source b) using equivalent voltage source .....	60
Figure 3.4	Grid connected inverter circuit with LCL filter.....	61
Figure 3.5	a) Companion model coupling $i_{L1}$ to the inverter b) companion model coupling $v_{C2}$ to $i_{L2}$ .....	64
Figure 3.6	a) Current $v_{C1}$ for the reference and the three methods b) relative errors of $v_{C1}$ for cases 1 to 3 .....	66
Figure 3.7	a) Current $i_{L2}$ for the reference and the three methods b) relative errors of $i_{L2}$ for cases 1 to 3 .....	66
Figure 3.8	a) Current $i_{L1}$ for the reference and the three methods b) relative errors of $i_{L1}$ for cases 1 to 3 .....	67
Figure 3.9	a) Voltage $v_{C2}$ for the reference and the three methods b) relative errors of $v_{C2}$ for cases 1 to 3 .....	67

Figure 3.10	a) Current $i_{L1}$ during fault for the reference and the three methods b) relative errors of $i_{L1}$ during fault for cases 1 to 3 .....	68
Figure 3.11	a) Voltage $v_{C2}$ during fault for the reference and the three methods b) relative errors of $v_{C2}$ during fault for cases 1 to 3 .....	68
Figure 4.1	Signal integrated with two different sampling rates .....	75
Figure 4.2	Single phase 3-buses system with STATCOM .....	79
Figure 4.3	a) Simulation results for $i_{L1}$ b) relative errors for Case 1 and 2 .....	84
Figure 4.4	Zoom on $i_{L1}$ for the reference and the MR model .....	84
Figure 4.5	a) Simulation results for $v_{C1}$ b) relative errors for Case 1 and 2 .....	85
Figure 4.6	a) Simulation results for $i_{L4}$ b) relative errors for Case 1 and 2 .....	85
Figure 5.1	Complete model to be simulated .....	87
Figure 5.2	Stubline example .....	89
Figure 5.3	Chronogram for parallel computing using a) two CPU with single-rate b) CPU/FPGA with single-rate c) CPU/FPGA with multi-rate .....	91
Figure 5.4	Half-bridge sub-module .....	93
Figure 5.5	Power system decoupled with stubline .....	97
Figure 5.6	Decoupled equivalent modular multilevel converter .....	99
Figure 5.7	Poles location during steady-state a) for the unity circle b) zoom in .....	104
Figure 5.8	Poles location during NRM for positive current in a) 2 upper limbs & 1 lower limb b) 2 upper limbs & 2 lower limbs c) 1 upper limb & 1 lower limb d) 1 upper limb & 2 lower limbs .....	105
Figure 5.9	Poles location during blocking for a) $k_1=10$ & $k_2=0.5$ b) $k_1=25$ & $k_2=0.5$ c) $k_1=10$ & $k_2=1.5$ d) $k_1=25$ & $k_2=1.5$ .....	107
Figure 5.10	Simulation results without STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power .....	109

Figure 5.11	Simulation results with STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power .....	110
Figure 5.12	Inrush current in phase A during initialization a) bus 1 b) absolute error at bus 1 c) bus 2 d) absolute error at bus 2 e) bus 3 f) absolute error at bus 3 .....	111
Figure 5.13	Simulation results with STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power .....	112
Figure 5.14	Simulation results with STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power .....	112
Figure 5.15	a) Virtual DC bus during DC glitch b) zoom on the effect of numerical snubbers .....	113

## **LIST OF ABBREVIATIONS**

CPU	Central Processing Unit
CRSNG	Conseil de recherches en sciences naturelles et en génie du Canada
DSP	Digital Signal Processor
FPGA	Field Programmable Gates Array
IOs	Inputs and outputs
MMC	Modular Multilevel Converter
MR	Multi-Rate
MS	Multi-Solver
PWM	Pulse Width Modulation
RTS	Real-Time Simulator
SPS	SimPowerSystems



## LISTE OF SYMBOLS AND UNITS OF MEASUREMENTS

<b>G</b>	Giga or $10^9$
<b>M</b>	Mega or $10^6$
<b>k</b>	Kilo or $10^3$
<b>m</b>	Milli or $10^{-3}$
$\mu$	Micro or $10^{-6}$
<b>n</b>	nanosecond or $10^{-9}$
<b>F</b>	Capacitor unit Farad
<b>H</b>	Inducator unit Henry
$\Omega$	Resistor unit Ohm
<b>S</b>	Conductance unit Siemens
<b>Hz</b>	Frequency unit Hertz
<b>V</b>	Voltage unit Volt
<b>A</b>	Current unit Amp
<b>VA</b>	Apparent power unit Volt-Amp
<b>W</b>	Active power unit Watt
<b>VAR</b>	Reactive power unit Volt-Amp reactive





## INTRODUCTION

In recent years, the number of industrial projects dedicated to energy transportation has greatly increased. Although some of these were intended for completely new facilities, most of them were intended to mesh existing networks. In the past, AC networks have been preferred over DC networks for practical and economical reasons. For instance in AC, voltage rating can easily be changed through power transformers with high efficiency. Such level of efficiency can nowadays be achieved using multilevel converter. A good example of multilevel converter applied to power system is the modular multilevel converter (MMC) topologies that was introduced by (Lesnicar and Marquardt, 2003b). MMC allows to connect two AC networks through a high voltage direct current (HVDC) link; which offers more flexibility. MMC works as voltage source converter, its voltage output can be controlled in amplitude and phase regardless of the output current; allowing therefore complete control of active and reactive power. MMC are made of multiple identical sub-modules connected in series. Every sub-modules output a fraction of the total voltage, yielding multiple voltage levels and reducing stress on each component. In the case of HVDC application, thousands of sub-modules can be used, making its simulation highly challenging, and even more when it comes to real-time (RT) simulation. RT simulation allows the user to test, develop and iterate on a controller without requiring a bulky, and somewhat dangerous setup. Unlike offline simulation, numerous timing constraints have to be satisfied in order to achieve RT simulation. The main constraints come from the use of a fixed-step solvers. Since RT simulation cannot use variable-step solvers, it is therefore imperative to determine the most appropriate sampling time for the simulation. Another key aspect of RT is the use of decoupling techniques, allowing a faster parallel processing of challenging and complex circuit. And finally, in the case of MMC, due to the large number of inputs and outputs inherent to the converter, a very efficient acquisition system is required. Those different issues are covered in this thesis which is divided as follows. Chapter 1 is dedicated to literature review of the present problematic, and raises the issues and challenges to be solved. This chapter was published as a chapter in (Grégoire *et al.*, 2014a). The minimum sampling time required for accurate simulation of MMC is discussed in chapter 2; which was published as a transaction paper in (Grégoire *et al.*, 2015a). In order to achieve real-time simulation of such a converter,

the model needs to be decoupled allowing parallel processing. A newly proposed decoupling method, with a numerical validation method is presented in chapter 3. This chapter was submitted to the special issue “Advances in the Simulation of Power System Transients” of IEEE Transactions on Power Delivery. Real-time simulation also requires a multi-rate simulation approach, and chapter 4 proposes a new method to test numerical stability of such simulation approach, which was submitted to IEEE Transactions on Industrial Informatics. Finally, a multi-rate model for real-time simulation of MMC is presented in chapter 5. Its numerical stability and accuracy is demonstrated using the different methods that were presented in the previous chapters.

## CHAPTER 1

### REAL-TIME SIMULATION OF MODULAR MULTI-LEVEL CONVERTERS (MMC'S)

Luc-André Grégoire<sup>1</sup>, Jean Bélanger<sup>2</sup>, Christian Dufour<sup>2</sup>, Handy Fortin Blanchette<sup>1</sup>,  
Kamal Al-Haddad<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

<sup>2</sup> OPAL-RT technologies Inc.,  
1751 Richardson, Suite 2525, Montréal, Québec, Canada H3K 1G6

Chapter published in « Power Electronics for Renewable Energy Systems, Transportation and Industrial Applications » in May 2014.

#### Abstract

This chapter presents the MMC topology. Its applications and origins are presented as well as the challenges faced when it comes to its simulation in RT. Different methods, their limitations, and how to overcome those challenges are reviewed. A brief introduction to simulation platforms is also made. Finally a proposed implementation method is presented.

#### 1.1 Introduction

RT simulation offers several advantages to speed up the development of new product. One of these advantages being the possibility to test and develop controllers when the hardware is not yet available. This is a serious advantage in the case of high order multilevel converter, like MMC topology. When considering building a full-size converter, its physical size could raise serious issues for most laboratories, without even mentioning the cost to build such a complex structure. Simulation can also be useful to analyse the interaction between several MMC and conventional HVDC systems installed on the same power grid. Furthermore, it can

perform factory acceptance test of the controller before its installation in the field. Nowadays, real-time simulator (RTS) are often used simply to accelerate simulations, as several hours of simulation can be required to run a few seconds simulation, for a power grid having two or three converter stations using conventional single processor simulation software. This chapter introduces fundamentals of RT simulation; its advantages and constraints. Using these fundamentals, RT simulation of an MMC will be undertaken. This topology was first introduced in (Lesnicar and Marquardt, 2003b), it is made of many identical sub-modules (SM) connected in series. Its modularity makes it suitable for various applications from medium voltage in a drive system, using only a few SM (Hiller *et al.*, 2009), to large HVDC transmission system containing a wide range of SM (Rajasekar and Gupta, 2012; Peralta *et al.*, 2012a). Connecting many of these SM in series reduces the voltage level that each sustains, decreasing the price of each component, reducing the switching losses, and smaller  $dV/dt$  at its AC bus, while producing a sinusoidal waveform with a very low total harmonic distortion (THD) eliminating the use of bulky reactive component filter.

### 1.1.1 Industrial applications of MMC converters

This topology was first tested in large scale by ABB in 1997. It consisted in a 10 km overhead transmission line with a 3MW capability at  $\pm 10\text{kV}$  between Hällsjön and Grängesberg in Sweden. It was used as proof of concept and established the capability of this new topology. The MMC, was named HVDC light by ABB, and was first used in a commercial project in Australia between Mullumbimby and Bungalore. Its voltage rating was  $\pm 80\text{kV}$  with a power rating of 180MW commissioned in 2000. Not long after, Siemens commercialised a similar topology as HVDC PLUS. Its first commercial project was a submarine HVDC link connecting San Francisco city center to a substation in the Pittsburgh area, it was commissioned in 2010 (Zhang *et al.*, 2012).

As of today, MMC projects being built are point-to-point converters only. Though actual HVDC network have been discussed theoretically, protection system for such network still need to be developed. ABB announced in November 2012 that they achieved a HVDC breaker

called hybrid HVDC breaker(Callavik *et al.*, 2012). Now that it has been used in a point-to-point setup, it will be tested in HVDC grid and should soon be commercialized. These new developments could change the future of power transportation.

### **1.1.2 Constraint introduced by RT simulation of power electronics converter in general**

Until now, big differences still exist between what can be achieved with standard, or offline simulation software, and RTS. The major constrain is in the time available to solve the differential equations of power electronic circuit. Offline simulation usually uses variable-step solver which works as follow. For every simulation time-step, two solutions are found using two different orders of discretization; one higher than the other. The solver iterates, reducing the time-step at every iteration, until the difference between the two solutions is within a pre-set tolerance (Hartley *et al.*, 1994). This process is very efficient for typical simulation of system with few disturbances. However, it becomes very slow in power electronic application, where stiff system with repetitive switching of semi-conductor needs to be solved. RT simulation, on the other hand, uses several processors, operating in parallel, with fixed-step solver, and uses a fix period of time to solve the differential equations. If a time-step of  $50\mu s$  is chosen to discretize a system, the RTS has to solve the differential equations within that period. Larger model, with more state-space equations will naturally take more time to be solved; in this case there are very few solutions to obtain acceptable results. One can increase the chosen simulation time-step, risking instability or inaccuracy. Having a more power full RTS allows to compute larger system. Computing power of RTS has increased exponentially over the last decade following Moore's law (Schaller, 1997), and are suited to simulate relatively small model. However, the RT simulation of very large power system requires to decouple the system in smaller subsystems that can be solved in parallel (Baracos *et al.*, 2001; Abourida *et al.*, 2002). Nowadays most RTS achieves time-step between  $10\mu s$  to  $50\mu s$  when using general-purpose processors, and between  $100ns$  to  $1\mu s$  when using field gate programmable array (FPGA).

In the case of power electronic circuit which contains fast switching devices, the chosen time-step is very important as it determines the accuracy that can be achieved by the pulse width modulation (PWM) circuit to generate the gating signals. A switching frequency of 10 kHz has a period of  $100\mu\text{s}$ . If one chooses the RTS time-step equal to  $50\mu\text{s}$ , then there is a maximum of 50% error on the time of occurrence of the switching event. Such inaccuracy may produce unrealistic transients and harmonics that could be confused with faulty controllers. This has motivated the usage of super-fast computer subsystem, where the time-step can be reduced further, or an interpolation scheme in order to achieve accurate switching frequency (Dufour *et al.*, 2005). Moreover, this is one of the reasons justifying the use of FPGA to solve such a problem, and consequently increases the popularity of the technology. FPGA chips operate at a 100 to 400 MHz clock frequency much lower than 2 to 4 GHz used for the general purpose processors. Although, processors can achieve computation time of 100 nanoseconds within the processor, achieving parallel processing using multiple processors is at least 50 times slower when using the most powerful commercial computer due to processor communication latency. Furthermore, when comparing processor and FPGA, firing signal of power electronic devices, generated from actual power electronic controller, can be connected directly to FPGA digital input pins. This results in a low latency between the firing order, measured at the controller output, and the resulting currents computed by an FPGA model, which can be less than 200ns. In such a case, the accuracy will be as good as variable-step solver. Such low latency and accuracy cannot yet be achieved using general-purpose computer because of the typical latency of the PCI communication system, 2 to 3  $\mu\text{s}$ .

Moreover, another important parameter to be considered is the type of modeling technique chosen to discretize the circuit. The two main and well-known are the nodal and the state-space method. Depending on the circuit topology, one can be more advantageous than the other. Taking short-cut and making this simpler than it actually is, time of execution in RT simulation comes down to the size of the matrix and its sparsity; since the latter needs to be inverted each time there is a change in the circuit topology, caused by a switching event. In the case of the circuit illustrated in figure 1.1, a state-space approach would generate a two by two matrix

to discretize or inverse, as shown in (1.1), since there is only two state variables. The nodal approach would yield a four by four matrix to solve, as shown in (1.2). This simple example illustrates the same concept that can be applied to larger circuits.

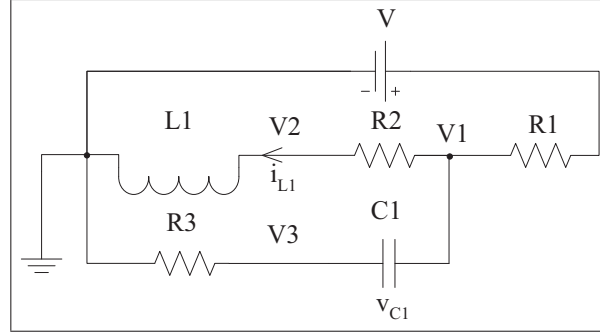


Figure 1.1 Circuit illustrating state-space versus nodal approach

$$\begin{bmatrix} v_{C1} \\ i_{L1} \end{bmatrix} = \begin{bmatrix} \frac{-1}{(R1+R3)C1} & \frac{-R1}{(R1+R3)C1} \\ \frac{R1}{(R1+R3)L1} & \frac{-R2-(R1//R3)}{L1} \end{bmatrix} \begin{bmatrix} v_{C1} \\ i_{L1} \end{bmatrix} + \begin{bmatrix} \frac{-1}{(R1+R3)C1} \\ \frac{-R1}{(R1+R3)L1} \end{bmatrix} V \quad (1.1)$$

$$\begin{bmatrix} i_1 \\ i_2 \\ i_{L1} \\ i_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{R1} & \frac{-1}{R1} & 0 & 0 \\ \frac{-1}{R1} & \frac{1}{R1} + \frac{1}{R2} + C1s & \frac{-1}{R2} & -C1s \\ 0 & \frac{-1}{R2} & \frac{1}{L1s} + \frac{1}{R2} & 0 \\ 0 & -C1s & 0 & C1s + \frac{1}{R3} \end{bmatrix} \begin{bmatrix} V \\ V1 \\ V2 \\ V3 \end{bmatrix} \quad (1.2)$$

Most simulation software uses one or the other method without giving the choice to the user. It is only brought up here to stay as broad and general as possible. Also when it comes to FPGA implementation, very few off-the-shelf tools are now available (Typhoon-HIL, 2013; OPAL-RT, 2013b). Many users still have to develop their own FPGA model despite of its complexity and researchers are still trying to develop general purpose electrical solvers, which would eliminate this complex task of implementing models and solvers in FPGA chip. Furthermore, one must keep in mind that one of the most difficult operation to be accomplished on FPGA is the

division, therefore, it renders inverting a matrix is an important research topic to complete it in a timely fashion suitable for RT simulation.

### 1.1.3 MMC Topology presentation

A MMC topology is constituted of an equal number of SM, presented in figure 1.2 a), distributed in the upper and lower limb, shown in figure 1.2 b). The SM includes power switches S1, S2 and the DC bus capacitor. When a SM is ON, the capacitor voltage is applied to its output using the upper switch S1 of the SM. When a SM is OFF, it is bypassed using the lower switch S2.

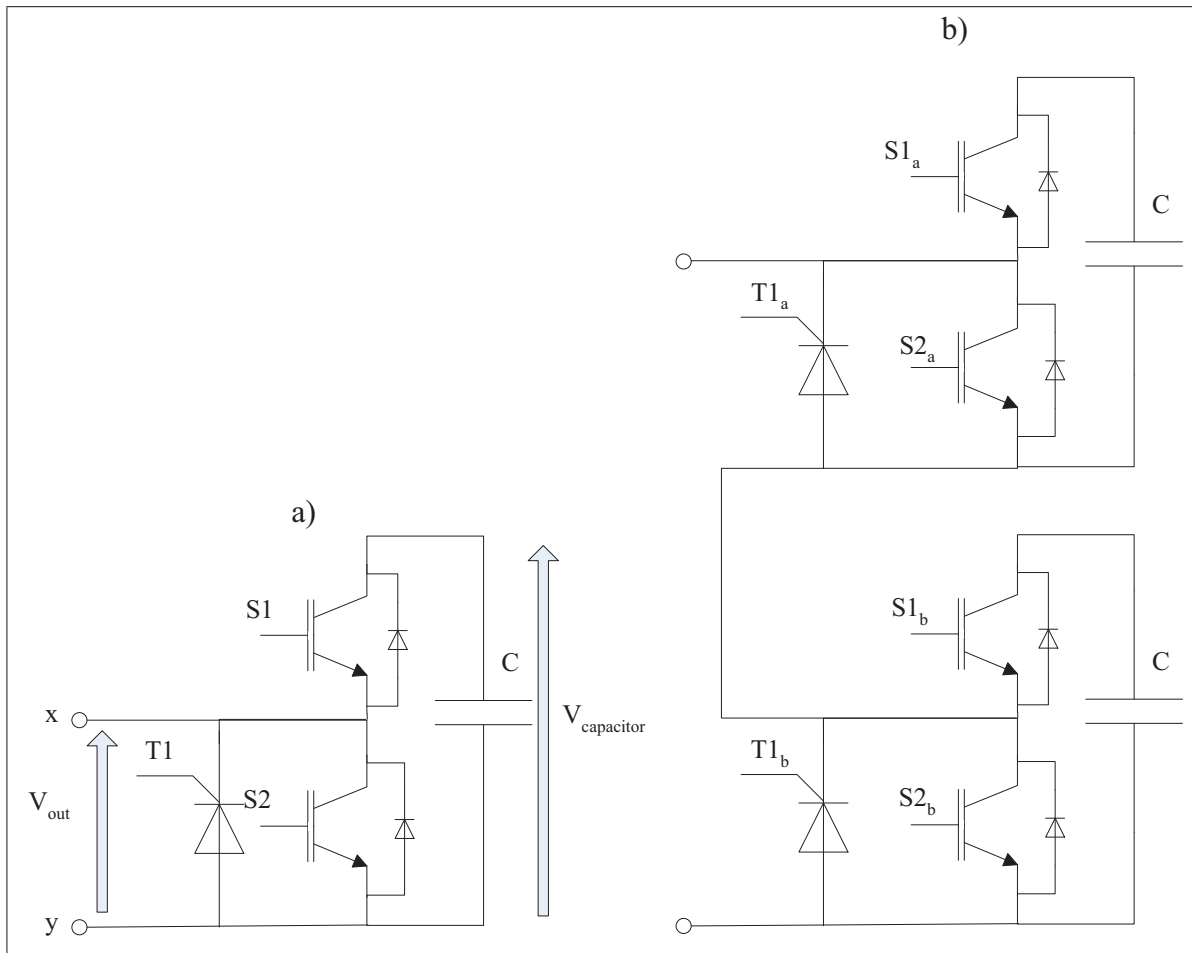


Figure 1.2 a) Single MMC SM with its protection b) MMC limb



The voltage obtained at the mid-point of the converter arm is given by the number of conducting SM in each limb. In steady-state, only half of the SM from one arm are conducting at any given time. For instance if a converter contains 100 SM, only 50 SM distributed between the upper and lower limb of an arm are conducting. Figure 1.3 shows the voltage seen at mid-point when:

- 1 SM in the upper-limb and 49 in the lower-limb are conducting, mid-point is near HVDC+;
- 25 SM in the upper-limb and 25 in the lower-limb are conducting, mid-point is zero;
- 49 SM in the upper-limb and 1 in the lower-limb are conducting, mid-point is near HVDC-.

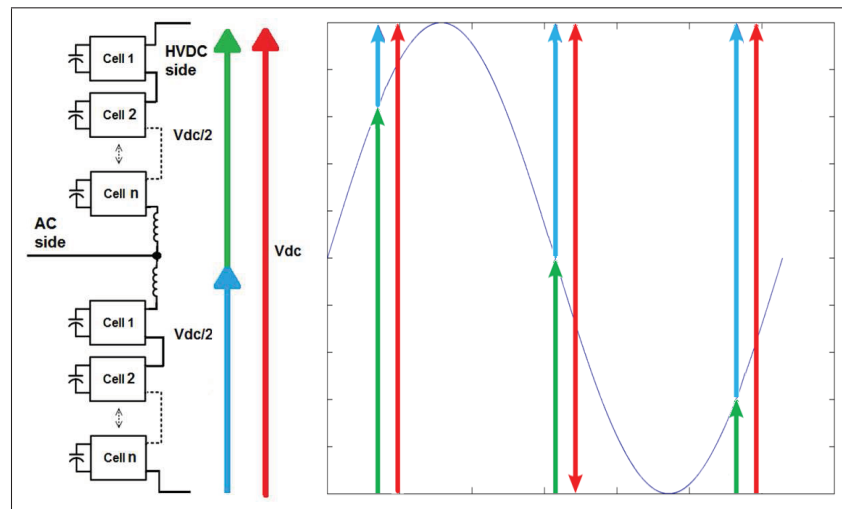


Figure 1.3 Standard example of one MMC arm for HVDC link

When a SM is conducting, its voltage will vary according to the limb current. The voltage is then regulated with rigorous algorithm to choose which SM to turn ON or turn OFF. Though this topology was proposed a few years back, only the increase in the computation power of controller have made it possible to accurately control it. Since then, many method have been proposed to control this converter topology requiring individual control of each SM capacitor voltage,(Hagiwara and Akagi, 2009; Antonopoulos *et al.*, 2009; Yan *et al.*, 2010) to cite only a few.

The number of SM plays two roles in this topology. It is linked to the quality of produced voltage, although when more than 12 levels are used it has been demonstrated that the gain on the THD becomes almost negligible, as shown in figure 1.4 (Arrillaga *et al.*, 2007).

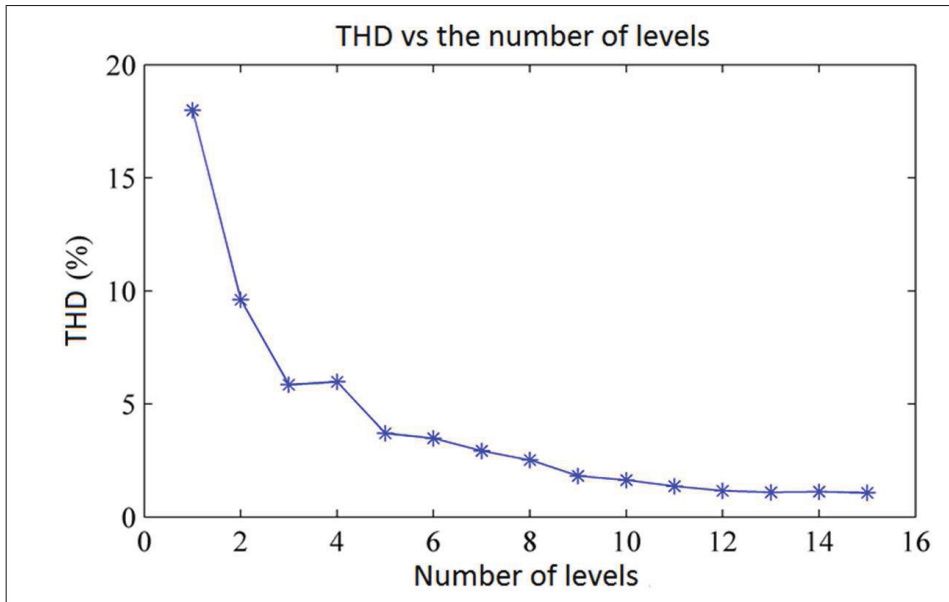


Figure 1.4 THD vs the number of levels

In the case of MMC used in HVDC application, the high number of SM used reduces the stress on each component and also offers a redundancy improving reliability. Furthermore, increasing the number of SM reduces the switching frequency of each individual SM up to once per cycle (Peralta *et al.*, 2012a).

Since this topology is a voltage source converter (VSC), it has complete control on the power flow, both active and reactive. Also unlike most HVDC classic topology, it does not require a network to synchronise itself, allowing to make a black start, since the voltage is imposed by the converter. But all of these advantages come with the price of complex control laws that need to be optimized and tested.

### **1.1.4 Constraints of simulating MMC converters**

When it comes to RT simulation of MMC converters there are two major problems to resolve. The first being the considerable size of the model, whether a state-space or a nodal approach is taken; and the second being the tremendous amount of input/output required to control the converter. Bear in mind that the main purpose to RT simulation is to be interfaced with a real controller. When it comes to thousands of SM to be controlled, it can only be assumed that even more signals are required for control of the converter. It can be assumed that most of the processing time in such a model is due to the I/O management and data transfer between external controller and the MMC model. It is therefore obvious that MMC simulators with 1000 SM per limb would require an I/O processing time much larger than  $25\ \mu\text{s}$ , which is unacceptable.

#### **1.1.4.1 Solving large state-space system**

One way to overcome the first problem is to exploit certain advantages of MMC topologies. Having a rather large inductance in each limb, this generates a very “strong state” on the AC side; where the current variation is rather slow. The DC side being often connected to a DC cable, the capacitance of the cable also generate a “strong state” on the DC side; where the HVDC voltage variation is slow. There two “strong state” can be used to decouple each limb. Once decoupled, it is possible to spread the computation burden over multiple computing units achieving parallel processing. Furthermore, one limb can be divided into smaller sub-circuit without any extra efforts on computation time.

#### **1.1.4.2 Solving I/O management problem**

Having met the requirement to decouple large system, the only problem remaining is the one concerning the amount of IO to be dealt with. Architecture of RTS will be discussed further down, but for now what need to be understood is that most real controller and simulator platforms use custom made card with a communication link to its computation unit. More IO

implies more data to be sent over the communication link and therefore requires more time. If more time is required for IO communication, this leaves less time for computation of the model. What have been done in the first part will actually help resolve the second issue. Spreading the model across more computation unit reduces the amount of data that each must exchange with the IO solving the second problem. Furthermore, simulating MMC SM directly ON FPGA chips, which are managing I/O channels, also minimizes data transfer between external controllers and simulator main processors. Such technique is now used by most advanced RTS.

Separating the large state-space systems formed by the MMC converters coupled to the AC network in order to achieve parallel processing can be achieved in several ways but might involve the use of artificial delays (Hui and Fung, 1997) or multi-rate simulation (Grégoire *et al.*, 2012). As of now, there are no formal and easy methods to achieve parallel processing of complex power electronic circuits coupled to large AC circuits.

#### **1.1.4.3 System-wide simulation simulated faster than real-time**

Several studies target the behaviour of several converters in a large network or the development of controllers before the manufacturing of controller prototype boards. In these cases the controller algorithm can also be simulated in the same simulator simulating the MMC system and the grid. Consequently no external IO are required and it is then possible to simulate faster than RT; i.e.: a typical simulation run of 60 seconds takes 60 seconds with a RT simulator or only a few second with a simulator running faster-than RT. In RT simulation, the acceleration factor is one, where the time-step used and the time required to execute the model has a ratio of one. For faster than RT simulation, the acceleration factor is greater than one since the time required to solve the equation of the model is smaller than the time-step used. Like before, if the model is decoupled and spread over many computation units, its acceleration factor is increased above one while the use of conventional signal-processor simulation software may have an acceleration factor much lower than one, i.e. the simulation time of a ten seconds case could take several minutes or even hours depending on the network and MMC size. Since that control development requires to analyse hundreds of contingencies and to optimize several

parameters, it is obvious that fast simulation tools exploiting multi-core processors and FPGAs will become essential as model complexity increases.

## 1.2 Choice of modeling for MMC and its limitations

As mention before, time is a very important constrain in RT simulation; choosing the appropriate level of modeling for a specific application help reducing the required computation time. The level of modeling can be classified in three main categories namely: Detailed model, switching function model and average model. Each of them will give accurate results but have some level of limitations.

### 1.2.1 Detailed model

There exists different level of modeling in the so called detailed model. Most of them offer too much detail which is not useful for RT simulation. The highest level of details could be qualified as “SPICE” modeling; where all the parasitic capacitors of the power switch and strain inductance of the PCB are taken into consideration. This type of modeling is used to calculate losses that will occur during switching. Even though this is a very important part of a real design, RT simulation should not, but also cannot, be used to evaluate switching losses and electromagnetic interference (EMI). Taking a numerical approach, the time constant of such component, pico-farad and nano-Henry, is around nanoseconds; these kind of time-step cannot be achieved today in RT even with FPGA.

The model where the switch and diode are considered as linear components can also be considered as detailed model. Every semi-conductor is represented by an impedance; small when conducting and high when blocking. Whether a state-space approach or a nodal approach is used, a new set of matrices need to be computed and inverted each time there is a change in switch status. This approach has been demonstrated using Hypersim(Le-Huy *et al.*, 2011) or the State-Space Nodal solver (SSN)(Dufour *et al.*, 2011b; Saad *et al.*, 2013) with 100 SM/arms MMC at time step in the  $30\mu\text{s}$  range.

### 1.2.2 Switching function

Switching functions or event based dynamic system (Zeigler *et al.*, 2000) can be interpreted as a switch case; for a certain input, certain behaviours are expected. In the case of figure 1.2, the switching function is given by (1.3).

$$V_{out} = \begin{cases} V_{capacitor} & \text{when } S1=1 \text{ and } S2=0 \\ 0 & \text{when } S1=0 \text{ and } S2=1 \\ 0 & \text{otherwise} \end{cases} \quad (1.3)$$

This implies that the switching is complementary and that there are no conducting losses; ideal switch. To introduce the switching losses the current must be taken into account. Therefore, the switching function becomes (1.4).

$$V_{out} = \begin{cases} V_{capacitor} - I_S \cdot R_{on} & \text{when } S1=1 \text{ and } S2=0 \\ -I_S \cdot R_{on} & \text{when } S1=0 \text{ and } S2=1 \\ 0 & \text{otherwise} \end{cases} \quad (1.4)$$

The flexibility of switching function makes it a very powerful tool, but it requires a very good understanding of the circuit in order to predict and have a contingency for every possible case. Unlike detailed model this can results into unnatural behaviour and discontinuity that is not present in real life. The gain is in the rapidity of execution which makes it very good candidate for RT simulation. Also when the limitations are known, it does not prevent the use of this model in all other supported mode. In this model, the number of states is not reduced, meaning that an integrator is required for every simulated capacitor SM. A detailed example is given in section 1.4.2.

### 1.2.3 Average model

The term average model here is not only intended like in the classical way. In classic average model, the duty cycle is given as input instead of PWM, but here the overall voltage of every SM capacitor is also averaged out across all the SM; making an ideal regulation of all the SM voltage. This type of modeling is the easiest to implement but it is also the one offering the most limitation. The main interest of this implementation is to study the behaviour of the converter in a larger network where the regulation of each SM is of little interest. Similar to the switching function, the SM output is given by a simple equation decoupling it from the large system. The rest of the system will see the converter as a variable impedance like it would with a detailed model. Again, a detailed example of the implementation is given in the section 1.4.1.

One drawback of this modeling is that it needs a special implementation to support the high impedance mode occurring when no pulses are applied to the converter. In this mode the output of the converter, a voltage source, is only controlled by its current when no pulses are applied to the switches. Normally, if the voltage applied to the limb is higher than the sum of all the capacitor voltage of this limb, current should circulate through the anti-parallel diode of the switches of the SM, charging the SM capacitor to voltage applied to the input of the limb. Once all the SM are charged, the current should become zero, since the anti-parallel diode are not polarize anymore, and stay at zero until either pulses are applied again to the converter or the anti-parallel diodes are polarized. Different schemes can be used to achieve this behaviour, a voltage source controlled by a voltage, but if it is not well implemented the response of the model can become erratic.

All three different types of modeling presented here serve a specific purpose, understanding the limitation of each model helps one to determine whether or not this implementation is suited for his application.

### 1.3 Hardware technology for real time simulation

In the mid 60's, RT simulation was achieved using analog simulator, where real linear and non-linear components were used to model and simulate a circuit (Hudson *et al.*, 1966). Not long after hybrid simulator, part analog part digital, were introduced and then with the evolution in the micro-processor speed, fully digital simulators were achieved. Even though the first digital simulators were limited, their smaller size and versatility made them more attractive and their popularity was powered by the increase of computer power capability over the last 15 years. For these reasons analog and hybrid simulators are hardly used nowadays and won't be explained further here. As for digital simulators, two main technologies divide them; the first one uses sequential programming embedded on micro-processor (CPU) and the second type makes use of parallel programming on FPGA. Because of their differences and their complementarity, it is not rare to see both technologies in one simulator, taking advantage of each one of them. Their respective features are discussed here below.

#### 1.3.1 Simulation using micro-processors

CPU are optimized for certain applications. It receives sequential programming; a series of instruction which are executed subsequently and repeated in a loop. These instructions need to be understood by the processor, what can be called low-level language. But it has to be entered by a user high-level language. The gap between those two levels is the different programming language, such as C, C++, java etc. Every manufacturer has a different machine code which can only be understood by their hardware. Using a common language by the user, like C, manufacturers make compilers that are compatible with their hardware. Nowadays, high-end processor can execute billions of instructions per second. In order to achieve further more computation power, as mentioned earlier, it is possible to execute different set of instruction in parallel using multiple processors sharing a high speed communication link.

In RT simulation, the most sophisticated processors are used in specially designed hardware. The code required is generated using software like Matlab/Simulink, so users do not require



to bother writing code. When multiple processors are available in parallel, users also rely on software to easily distribute the computational burden among them.

One of the greatest examples of RT simulation in parallel is the Hypersim simulator, developed at IREQ, the engineering department of Hydro-Quebec (OPAL-RT, 2013a). It can simulate large network, thousands of nodes, cluster of hundreds of CPU while the allocation of the processor unit to simulate each network subsystems is fully automated (Gagnon *et al.*, 2012). Other RTS would normally require the intervention of advanced user in order to distribute the computation load over multiple computing unit (OPAL-RT, 2013c; RTDS, 2013)

### 1.3.2 Simulation using parallel programming with FPGA devices

FPGA offers much more flexibility when it comes to executing instructions; it actually allows user to develop its own instruction set. Logic operators like NAND-gate or XOR-gate, basic arithmetic like sum, multiplication are some of the component available. By using these functions, users can make an optimized set of instruction for a specific application. On older FPGA generation, only fixed-point representation was available, but since 2009 built-in operators, supporting floating point, are now available.

In FPGA implementation, signals travel as fast as their propagation allows. The number of operations that can be done will depend on the design; the route that signals need to take for a desired logic. On FPGA, clock signals are used to ensure that the expected result has reached its destination and is synchronised with other signals. Figure 1.5 shows the concept of propagation to a simple circuit and the synchronisation of its output.

From Figure 1.5, since only one level of logic gate is required to obtain A, it can be supposed that it will be ready before B that needs two levels of logic. C might changes when A is ready and changes again when B is ready. To avoid uncertain value at the output, a register is added and synchronised with a clock signal. By doing so, D will be synchronised with the clock and its value will be accurate as long as the period of the clock is long enough for the inputs 1, 2 and 3 to pass through the logic resulting into C. In Figure 1.5, the results of A and B are being

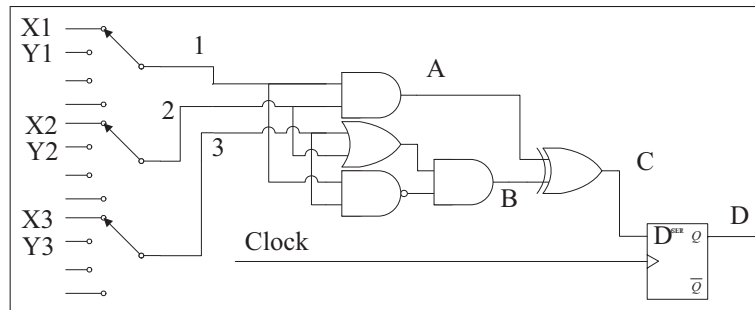


Figure 1.5 Example of logic propagation

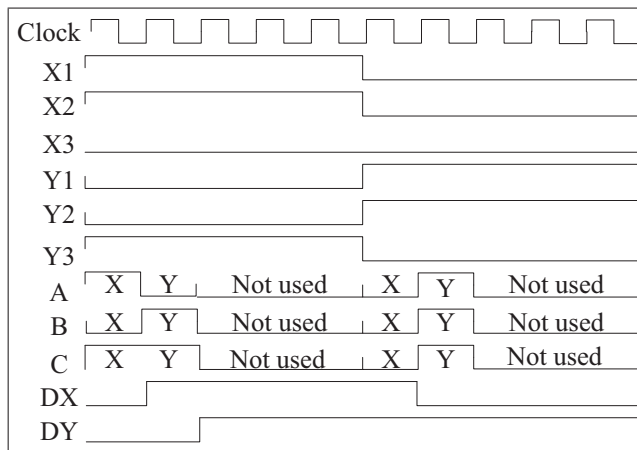


Figure 1.6 Chronogram of time-multiplexed process

processed simultaneously and independently; this is the major advantage of FPGA referred as parallel processing. From simple logic to large matrix multiplication can be performed in parallel and the result for the global solution is found in the end where all the different solution are joined and synchronised.

Understanding that the process can be synchronised to a specific clock, it is also possible to use time multiplexing or pipelining; allowing the same logic to be used for different process. If the clock of the process is slower than the clock of the FPGA, it becomes possible to execute the same process using the same logic. For example, if the process in figure 1.5 can be obtained in one FPGA clock period, but its inputs are only ready every 5 FPGA clock. By multiplexing the input, using selector, and demultiplexing the output, the same logic could be used up to five times to calculate the same process. The chronogram of figure 1.6 shows the time multiplexing

with only two different processes. Process X has input X1, X2, X3, process Y has input Y1, Y2, Y3 and each process, X and Y, yield the results of A, B and C in time. At every clock, the value from the different process X and Y are applied to the logic and their results are shown on the chronogram. After the two processes, the logic is not used and its result is not registered. The result of DX and DY are updated when available and stay there until the next clock of the slow process. Such design can ensure that none of the resources are left idling during the different processes, but it requires very accurate synchronisation and design.

The next example is more related to simulation; the implementation of a forward Euler integrator. The FPGA has a clock of 5 nanoseconds and the integrator time-step is 20 nanoseconds; it is then possible to use pipelining. Figure 1.7 shows the block schematic used in this example. The input A receives the multiplexed in time values to be integrated. B is the result of the values multiplied by the integration time-step, 20 ns. D is actually the output of the sum C with a four-step delay, making the forward Euler integrator. The result in D can then be demultiplexed to send the integrated values to the right process. Here the integration time-step was chosen to facilitate the representation in a chronogram of the system in figure 1.8. Such a small time-step is unlikely to be chosen since it would require a very high level of precision, whether one chooses to use a fixed-point or a floating-point representation.

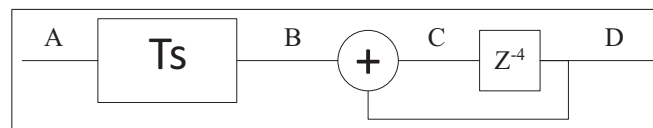


Figure 1.7 FPGA integrator using pipelining

The great versatility of the FPGA also creates its main drawbacks: Complexity to implement models and excessive time to generate the bitstream. The example given above clearly demonstrate that the programming complexity is much larger than using high-level language like C++ or very-high-level language like SIMULINK and code generators like real-time workshop (RTW). Such complexity limits the number of specialists who can develop and maintains models. The debugging is also very difficult and time consuming.

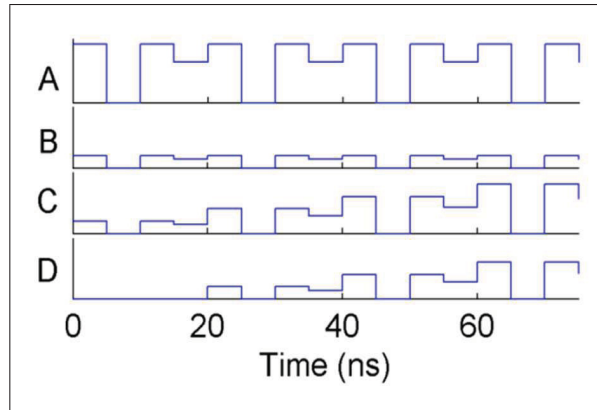


Figure 1.8 Chronogram of pipelined integrator

Because it is a field-programmable gate array, each individual gate need to be programmed and interconnected when generating the code. Generating the code for an FPGA, also known as bitstream, requires a software to analyse each possible path and find the optimal one. With the size of FPGA and models getting larger and larger, the required time to compile the code, or bitstream, also increases. Meaning that if the configuration of your model change, you need to recompile a new version of your bitstream, which may take several hours.

One option to avoid these two drawbacks is the use of embedded solver on the FPGA (Dufour *et al.*, 2012a). This allows testing many different circuit configurations and if needed it is also possible to make some changes and recompile a new bitstream.

#### 1.4 Implementation for real-time simulator using different approach

These are simple examples to give the reader fundamentals allowing him to implement its design. Matlab/Simulink was used to implement and test these implementations, but similar results could be achieved with any other simulation software. For both examples, all SM from a limb are represented by an equivalent voltage source. The only difference is in how the voltage is computed. The equivalent circuit is shown in figure 1.9. VUA, VUB, and VUC represent the upper limb equivalent voltage whereas VLA, VLB, and VLC represent the lower limb equivalent voltage.

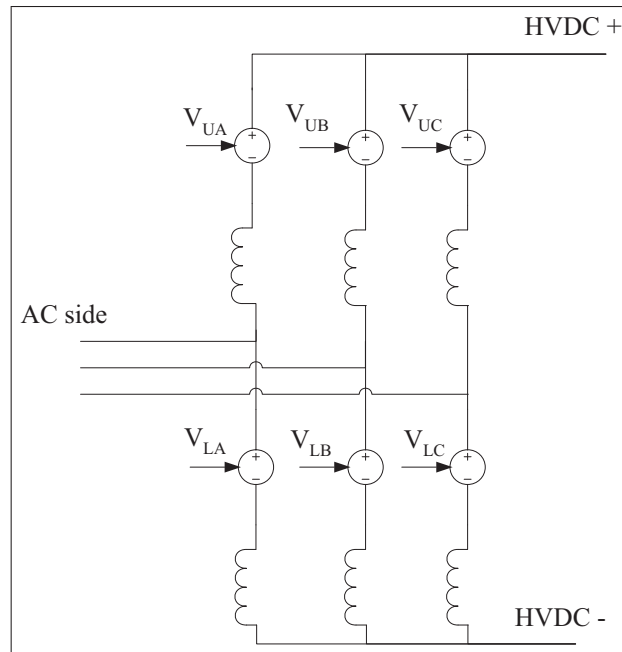


Figure 1.9 Equivalent decoupled circuit

This method of decoupling is adequate since there are two very large states in the model; the large arm inductance ensures a slow variation of the current and the large SM capacitors a slow varying voltage. Measuring the current from the arm inductance, the equivalent voltage from all the conducting SM is computed. In order to break an algebraic loop, a forward Euler integration method is used; this won't affect much the stability of the circuit since it is introduced at a point where there is dominating poles.

#### 1.4.1 Sequential programming for average model algorithm

This type of modeling can be used to test the inner and outer control for converters that would be connected to a larger network. This allows estimating the load flow, verifying contingency test or general behaviour of the overall network without having to bother regulating each individual capacitor SM.

For this model the following assumption are made:

- The current in one limb is the same for all the SM forming that limb; naturally because all are connected in series;
- All the capacitors have the same value; the integration of the current will result to the same voltage variation for all the capacitor of conducting SM;
- Only the number of conducting SM is required as input to the model, it is assumed that the choice of which SM is turned ON within a limb is made by a local and independent controller, who is not part of the model.

Figure 1.10 shows a block diagram for one limb. It has the limb current and the number of SM ON as input and the sum of all conducting SM voltage as output.

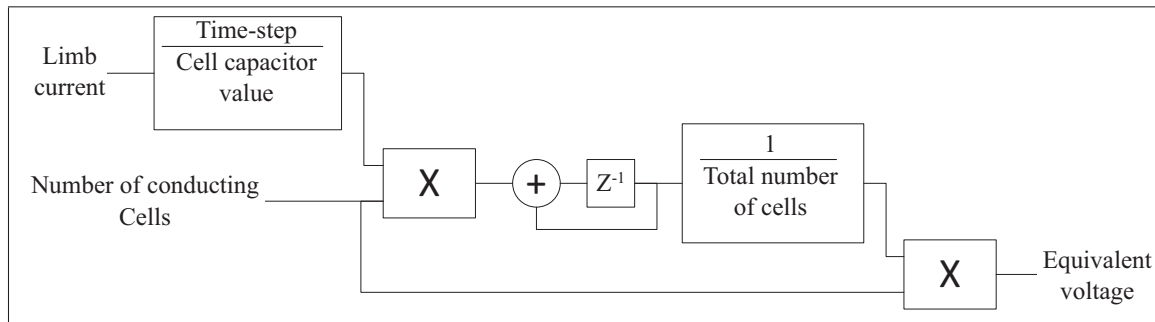


Figure 1.10 Block diagram for average MMC model

The limb current is multiplied by the time-step and it is divided by the capacitor value, giving the voltage variation of any conducting SM. Then this voltage variation is multiplied by the number of conducting SM and the result is added to the previous voltage value of all SM. The total voltage value is divided by the total number of SM obtaining the capacitor voltage of a single SM; this is how the regulation of all the SM is made to the same voltage. Finally the voltage of a single SM is multiplied by the number of conducting SM generating the equivalent voltage for a single arm.

This technic is simple and could even be implemented in a variable-step solver with small modification. One of its limitation is that natural rectification, using anti-parallel diodes from

the SM switches, is not supported in this implementation as well as the possibility to simulate faults inside the limb, and to test the individual SM voltage regulator.

Using this implementation, the HVDC grid of figure 1.11 was simulated faster than RT simulation. This configuration is the DC grid benchmark proposed by the CIGRE work group B4-57. The converter A1 is connected to a larger network, modeled by two voltage source. Converters B1, B2 and B3 are connected to a different network but also have an AC link between one and other. Converter C1, C2, D1 and F1 are offshore wind farm and E1 is an isolated offshore load. All the offshore converters are connected through underground cables for their HVDC link. Converters on land use overhead power lines transmission to interconnect among them.

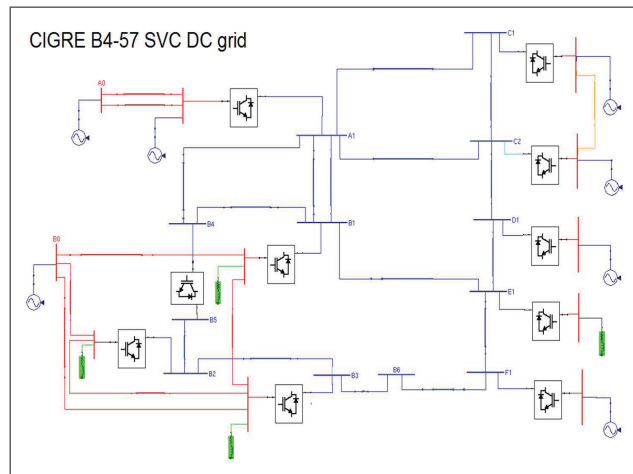


Figure 1.11 CIGRE B4-57 HVDC grid

Figure 1.12 shows the response at the converter C1 and C2. Only the phase A is monitored in this case but all phases are available. In this test there is a three-phase fault on the AC side between C1 and C2. When the fault occurs, line between C1 and C2 is opened at each end for 2 cycles then it is reclosed. At this point the fault has been cleared. Figure 1.12 shows the voltage at each converter. On C2 side, at reclosing an overvoltage is seen. This overvoltage can vary according to the angle at which the breaker is reclosed. Using this model and a sequencer, a series of tests can be generated to make a Monte Carlo study to identify the V2% (Paquin *et al.*, 2009).

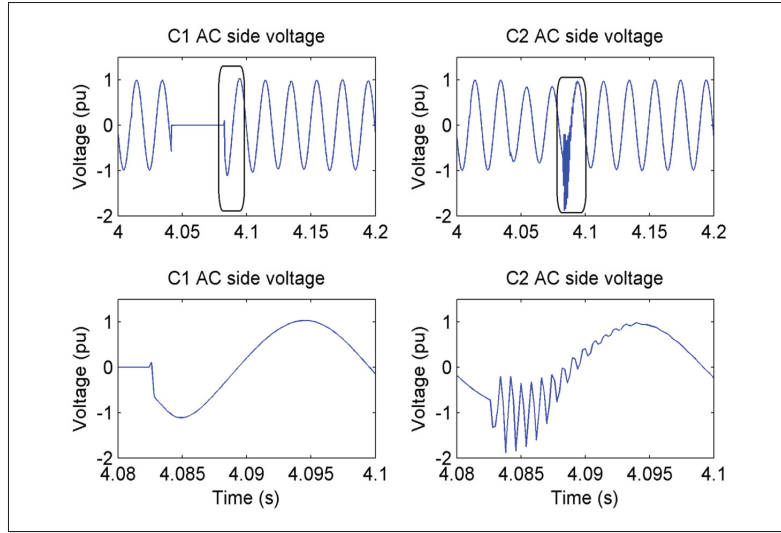


Figure 1.12 CIGRE benchmark AC fault

Since no IOs were used in this model, it was possible to simulate it faster than RT. Using 11 processors of an eMEGAsim simulator, an acceleration factor of 4 was achieved. In the case of Monte Carlo study, where thousands of simulations are required, this acceleration factor is very significant.

#### 1.4.2 Parallel programming for switching function algorithm

As it has been previously discussed, parallel programming can be implemented on FPGA. Taking advantage of both parallel processing and time-multiplexing, a very large MMC converter can be simulated on FPGA with a very small time step of 250 ns. The choice of the time step of 250 ns is not based on stability of the circuit but rather to have very accurate firing instant for each SM.

Table 1.1 gives the switching function for figure 1.2 that will be implemented on FPGA.

One can note that the mathematic behind this model is still relatively simple; the challenge comes in the implementation to achieve the small computation step. The arm current is obtained from the model running on CPU, where the complete network can easily be implemented using standard simulation software. The gate signals, S1 and S2, come from digital



Table 1.1 Switching function of MMC SM

Cases	Arm current	S1	S2	SM's voltage $V_{out}(T)$	Capacitor's voltage $V_c(T)$
1	X	0	1	0	$V_c(T-T_s)$
2	X	1	0	$V_c(T)$	$V_c(T-T_s) + 1/C * I(T-T_s) * d$
3	X	1	1	Not considered	
4	$> 0$	0	0	$V_c(T)$	$V_c(T-T_s) + 1/C * I(T-T_s) * d$
5	$< 0$	0	0	0	$V_c(T-T_s)$
6	$= 0$	0	0	High impedance	$V_c(T-T_s)$

input connected to the FPGA. The simulation time-step on CPU is 100 time slower than the one of the FPGA, therefore instead of sending the instantaneous voltage output of all the SM, only the average over CPU time-step is send; in a similar way that only the duty cycle of PWM can be applied when the simulation step is slower than the PWM period.

There is two distinct processes that need to be implemented; the switching function and the integrator. The integrator uses the same method as the one used in figure 1.7, in this case 10 signals are pipelined over 250 ns or 25 FPGA steps. During the demultiplexing of the integrator results, the capacitor voltages of the conducting SM are summed to achieve the equivalent voltage for the limb. Another important part of the logic is the implementation of the switching function which determines which SM is conducting and which capacitor is charging. Figure 1.13 shows the block diagram of the process and the number of FPGA step each process requires.

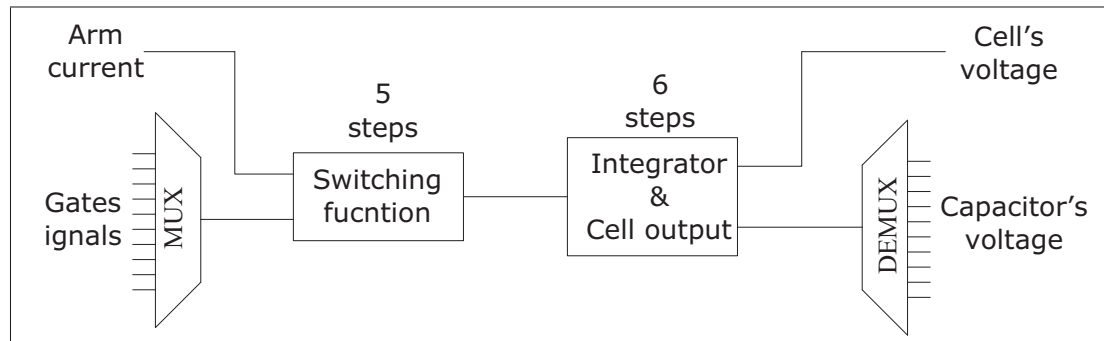


Figure 1.13 Block diagram of FPGA implementation

Note that the overall process takes 11 time step, using an internal clock of 10 ns for the FPGA, which means that the first capacitor value will be available after 110 ns. Since all of them are time multiplexed by group of 10, the last capacitor voltage is available after 21 time step or 210 ns. Here the advantage of pipelining is very clear, by adding more capacitor in the pipeline, only 1 more time step is required to obtain the value. In this case, there is still 4 steps available to add more logic if required, allowing more flexibility as it have been demonstrate in (Grégoire *et al.*, 2012).

The implementation from figure 1.13 was used to simulate a converter with 500 SM per half-limb, for a total of 3000 SM. It can either use an internal controller, embedded on the FPGA, or an external controller, via optical fiber. In this example, every SM is using two optical fibers for communication, one for receiving and one to send data. Figure 1.14 shows simulator used to simulate the converter. In the centre of the picture is the main simulator where the model is computed with a 250 ns time-step. The others racks on each side are only used to manage all the optical fiber that are needed to control the simulation.

Figure 1.15 shows results obtained when changing the power reference. Reactive power is stable at -0.3 pu and the active power changes from 0 to 0.5 pu. Looking at the voltage and current, one can see the phase shift of the current as the active power increases.

This is only one of many tests that can be applied to such a system. Using the FPGA implementation allows a very low latency between the IOs and the model. In this case only the MMC converter is simulated on FPGA and the remaining of the network is simulated using processors. In 2011, Nari-Relays Electric Co. in China used the HIL results for the Nanhui MMC demonstration project, 20 MVA/60kV 2-terminal MMC HVDC project.

## 1.5 Conclusion

This chapter presented an overview of RT simulation with a practical application of the different technology. As discussed, the digital simulators are widely used but different technologies

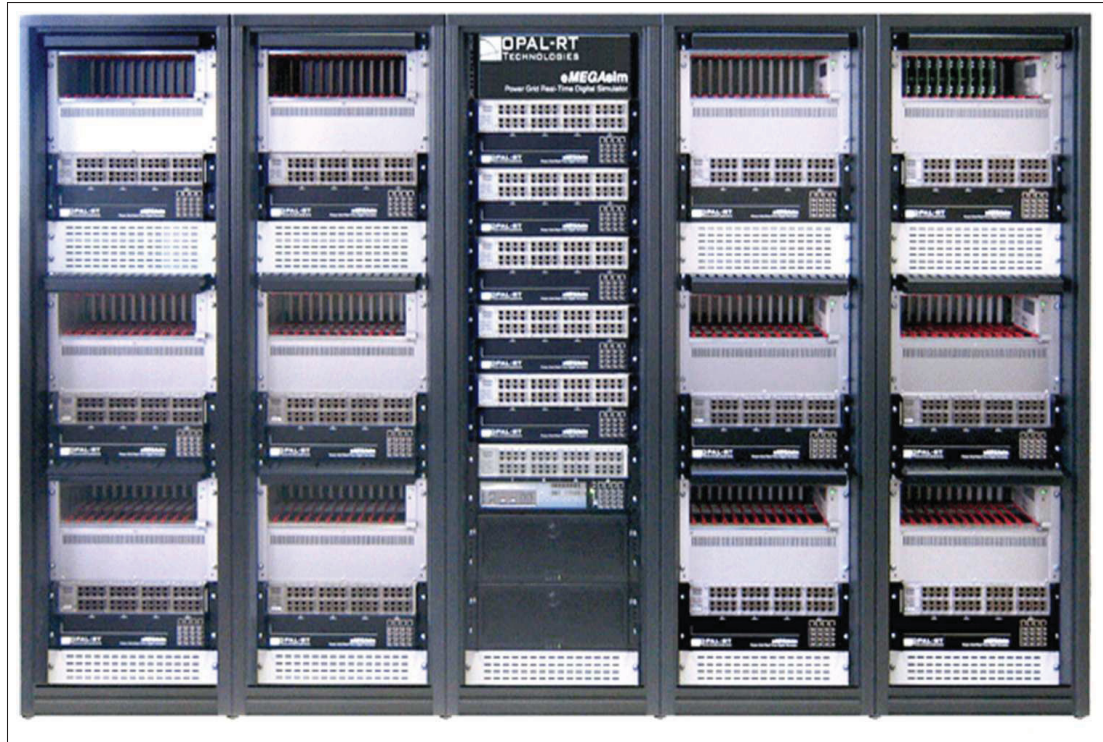


Figure 1.14 Real-time simulator with IOs chassis

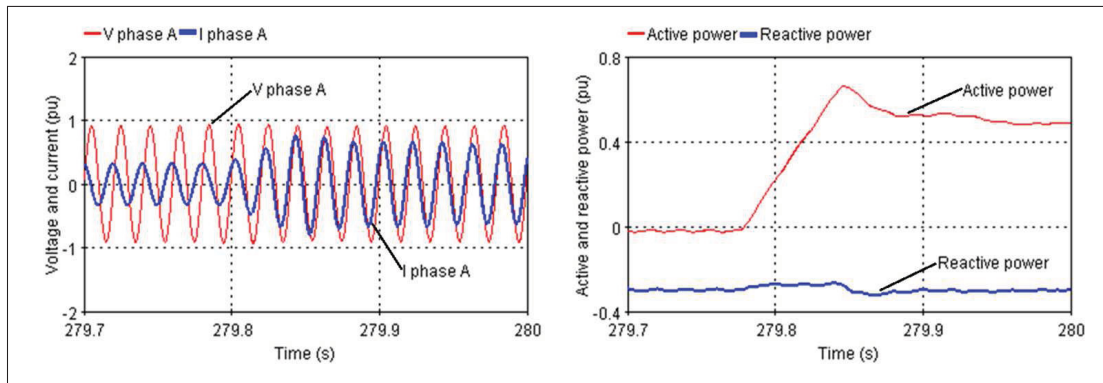


Figure 1.15 Results of a step on power reference

are available. Nowadays, understanding the application before acquiring a RTS can help identify the best suited type for the application.

Standard single-processor offline simulation tool does not offers adequate solution to achieve RT simulation, but it is possible to implement its own design using the different method pro-

posed in this chapter. Multicore micro-processors and FPGA are evolving very fast, and therefore so does RT simulation.

General purpose electrical solvers are available and being developed to facilitate the use of FPGA technologies by abstracting the inner construction of FPGA chips, as this is done with general purpose micro-processors. Such FPGA-based solvers should evolve very fast over the next years. This chapter mainly focus on EMTP simulation, which is the best suited for power electronic simulation, but some software are now offering a mix simulation ETMP/phasor; slow components like transmission network are simulated with phasor algorithm and this simulation is coupled with an EMTP simulation where fast systems, like power electronics, are simulated. Looking to the last ten years, one can expect that the use of real-simulation will keep growing and it seems like it is only limited by the need of the industries.

## CHAPTER 2

### MODULAR MULTILEVEL CONVERTERS OVER-VOLTAGE DIAGNOSIS AND REMEDIAL STRATEGY DURING BLOCKING SEQUENCES

Luc-André Grégoire<sup>1</sup>, Handy Fortin Blanchette<sup>1</sup>, Wei Li<sup>2</sup>, Antonios Antonopoulos<sup>3</sup>,  
Lennart Ängquist<sup>3</sup>, Kamal Al-Haddad<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

<sup>2</sup> Simulation specialist, OPAL-RT technologies Inc.,  
1751 Richardson, Suite 2525, Montréal, Québec, Canada H3K 1G6

<sup>3</sup> Electrical Energy Conversion Laboratory, Royal Institute of Technology,  
Brinellvägen 8, Stockholm, Sweden

Article published in « IEEE Transactions on Power Electronics » in May 2015.

#### **abstract**

In this paper the authors first highlight an existing over-voltage phenomenon that is inherent to the Modular Multilevel Converter (MMC) topology. The latter occurs during the blocking sequences of semiconductor devices if the converter needs to be stopped due to circulating current, loss of control or unexpected faults. An analysis based on time domain expressions describing each operating sequence during normal and faulty blocking conditions is used to demonstrate the origin of this over-voltage. Thereafter, system behaviour is obtained when devices gating signals are withheld as well as the exact over-voltage cause. Real-time simulation, with sub-microsecond time-steps, and experimental results validate the over-voltage phenomena and the proposed remedial strategy to avoid uncontrolled faulty conditions.

#### **2.1 Introduction**

Modular multilevel converter (MMC) topologies are gaining a lot of interest when it comes to high voltage AC drives and HVDC applications, as well as for many renewable energy plants

where accessible DC sources are available for energy harvesting (Perez *et al.*, 2013). These new technologies, which contain numerous components compare to traditional two-level voltage source converters (VSC), require new methods to detect and cope with the different faults they may encounter. Previous literature has already reported various faults and methods to identify them. In (Liu *et al.*, 2013), authors first categorize faults by their locations, sub-module level, converter level, and power system level. After using measurements that are already required by the controller, control layers are added to identify or mitigate faults. Similar methods are used in (Shao *et al.*, 2013) where a control layer is added for detection and to remediate faults. In this paper, adding a similar control layer to an existing controller is proposed to cope with a common recurrent problem. This problem is the uncontrolled over-voltage which occurs during pulse-stopping of semiconductor devices. Such over-voltages have not yet been reported in previous literature for MMC using half-bridge sub-module (HB-SM). A similar phenomenon, when using a full-bridge sub-module (FB-SM), was reported during DC-fault blocking, which is different from the case presented in this paper. Converter using FB-SM gains the capability to naturally block DC-fault without operating AC-breakers, which is one of its advantages. During DC-fault, the AC currents from the FB-SM converter are blocked by stopping sub-modules pulses, and sub-modules of each limb are then forced in series and create an over-voltage. In (Marquardt, 2010), this problem is reported and a new double-clamped sub-module (D-CSM) is proposed to overcome this issue during DC-fault. However, FB-SM and D-CSM increase the losses and the complexity of the controller which justify why HB-SM is still widely used and why fault-tolerant controllers are still required for this topology. In (Guan and Xu, 2012; Teodorescu *et al.*, 2013), proposed controllers that can withstand non permanent faults on the AC-side. In (Shi *et al.*, 2014), the redundancy of the MMC topology is demonstrated by reconfiguring the communication protocol of the converter on-the-fly. It is assumed that faults are not permanent or that the controller will successfully regulate the converter. In the cases when the control is lost or when a permanent fault occurs in the converter, forcing the converter to be stopped are not covered at all. The possibility to stop the converter during fault is raised in (Yan *et al.*, 2013), where the protection coordination for a DC network is proposed. Faults occurring on the DC side are presented and are mitigated using solid-state

circuit breakers (SSCB). When one MMC has to be isolated from the DC network because of a fault, authors only indicate that pulses are stopped. During this operation, neither the impact on the converter, nor a proper procedure has been proposed. In this paper, a detailed analysis of the MMC topology, using HB-SM, is done when gating signals of the converter are stopped. In some cases, this results in an over-voltage of the DC-bus which is also demonstrated in simulation and on an experimental setup. Also, a new blocking sequence is proposed to avoid over-voltage.

This paper is divided as follows. Basic operation principles of MMCs are briefly recalled in section 2.2. Section 2.3 presents an analytical study of the converter in order to investigate deeply the over-voltage phenomenon. The existence of such a transient will be confirmed by simulation results in section 2.4 and by an experimental test on a physical converter in section 2.5. Finally, section 2.6 proposes an emergency stop procedure in order to avoid this transient followed by a conclusion in section 2.7.

## 2.2 MMC basic topology and control strategy

MMCs are obtained by connecting in series numerous identical cells called "sub-modules" (SM); figure 2.1 shows a typical half-bridge cell. In normal operation mode, the two switches are complementary making the  $V_{cell}$  equal either to 0, when the cell is turned OFF, or equal to  $V_{cap}$ , when it is turned ON. One converter's arm has two limbs, upper-limb and lower-limb, with the same number of cells. Three arms make a complete AC/DC converter as shown in figure 2.2. Although modulation methods may vary, the most popular ones are the nearest level control (NLC) (Li and Zhao, 2010), or pulse width modulation ( $n+1$  or  $2n+1$ ) (Li *et al.*, 2012), making the average number of cells turned ON in the upper-limb complementary to the average number of cells turned OFF in the lower-limb; the total voltage stored in the capacitors of each limb is equal to the pole-to-pole voltage. The average capacitor voltage of each cell is then given by (2.1).

$$V_{cap} = \frac{V_{DC}}{n}, \quad (2.1)$$



where  $V_{DC}$  is the voltage between the positive and negative poles of the converter and  $n$  is the number of cells in one limb or half the number of cells in one arm.

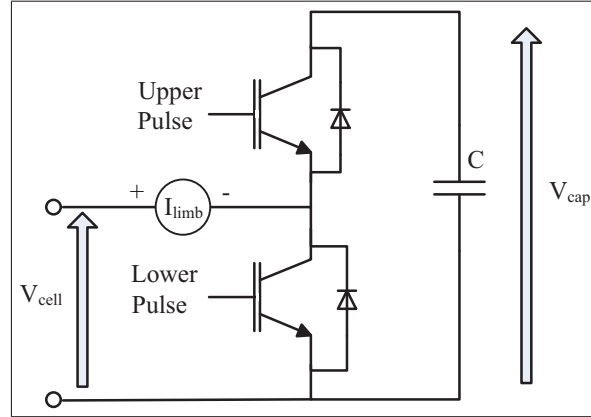


Figure 2.1 A typical half-bridge cell

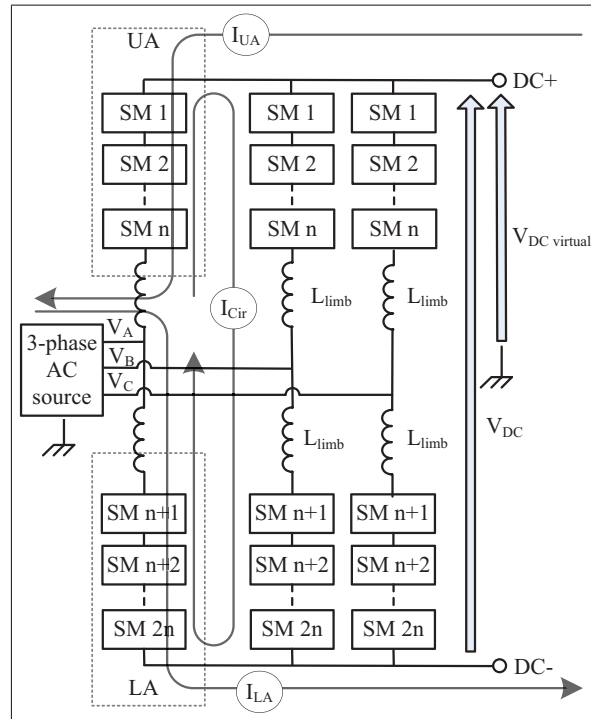


Figure 2.2 Schematic of a 3 phase MMC showing n level



The main difficulty of this topology is regulating the different cell capacitors. Different methods for regulation have been presented over the last years (Rohner *et al.*, 2010; Saeedifard and Iravani, 2010; Angquist *et al.*, 2011; Solas *et al.*, 2013a; Perez *et al.*, 2012; Ilves *et al.*, 2012b; Li *et al.*, 2011; Vahedi *et al.*, 2014; Debnath *et al.*, 2015) and will not be discussed in detail. The main objective of those controllers is to produce the desired output voltage at the AC side while regulating each cell capacitor voltage to its desired value. In normal operation, the output current from one phase should be equally divided between its upper-limb and lower-limb. In figure 2.2, the upper-limb of phase A is referred as UA and the lower one as LA, in the same manner, the current flowing in each limb is noted  $I_{UA}$  and  $I_{LA}$ . To achieve high efficiency, it is also important to minimize the circulating current  $I_{cir}$  between different limbs. As for the control algorithm, numerous methods were proposed to suppress  $I_{cir}$ , and only a few of them are suggested in this paper (Tu *et al.*, 2011; Ilves *et al.*, 2012c; Zhang *et al.*, 2013; Wang *et al.*, 2013; Zhang *et al.*, 2014).

### 2.3 Analytical Study For Failing and Faults Conditions

During the course of its operation, if the MMC operates as a STATCOM, or if the DC line is disconnected, in both situations, only reactive power is exchanged between the AC supply and the converter. Since no current is flowing toward the DC bus, it means that the sum of currents going into the three upper or lower limbs is zero. At this point, it is assumed that no capacitor, nor inductive line is connected to the DC bus; it is then referred to as a virtual DC bus. The virtual DC bus voltage is measured using the neutral point of the AC voltage, as shown in figure 2.2. Choosing this point as a reference voltage allows measuring the positive and the negative DC poles from the same reference point. If pulses are stopped when the current in any of the limbs is different than zero, the anti-parallel diodes of the switches will start conducting. During this so-called free-wheeling mode, depending on the current polarity flowing in the limbs, equivalent MMC cells will be either short-circuited or equal to the sum of all DC capacitor voltage in series. In this case, the sum of the currents from the upper limbs is independent of the ones from the lower limbs; therefore the analysis can be done

independently for lower and higher limbs. Whether it is in the upper or lower limb, a limb current is considered positive if it is charging the cell's capacitor, as shown by  $I_{limb}$  in figure 2.1. Considering the different possibilities, the limb currents can either be:

- two positive limb currents and one negative;
- two negative limb currents and one positive;
- one positive, one negative and one equal to zero;
- all three limb currents equal to zero.

This will result in thirteen possible sequences for the upper and lower limbs, yielding a total of 169 combinations. If the sequence, where all three limb currents are equal to zero, is not considered, twelve sequences remain and they can be represented by three sequences for the upper half of the converter, figure 2.3 a), figure 2.4 a), Figure 2.5 a) and three sequences for the lower half of the converter, figure 2.3 b), figure 2.4 b), figure 2.5 b). To ease the representation, all the cells connected in series are replaced by a single equivalent cell with an equivalent capacitor in figure 2.3 through figure 2.5. The equivalent capacitor value is found by dividing the capacitor value from one cell by the number of cells in one limb (2.2).

$$C_{eq} = \frac{C}{n} \quad (2.2)$$

In figure 2.3 a) and b), there is one positive and two negative currents. The positive current forces the anti-parallel diode of the lower switches of the cell to be conducting; no cell capacitor of this limb is contributing to  $V_{DC \text{ virtual}}$ . The two remaining negative currents are forced through the anti-parallel diode of the upper switch, in which case the sum of all the cell capacitors voltage is contributing to  $V_{DC \text{ virtual}}$ . In figure 2.4 a) and b), two currents are positive and only one is negative, resulting in only the cell capacitor of one limb contributing to  $V_{DC \text{ virtual}}$ . For the four previous sequences, AC voltage has no influence on  $V_{DC \text{ virtual}}$  since the sum of the three phase voltage is zero. In figure 2.5 a) and b), since one of the currents is equal to

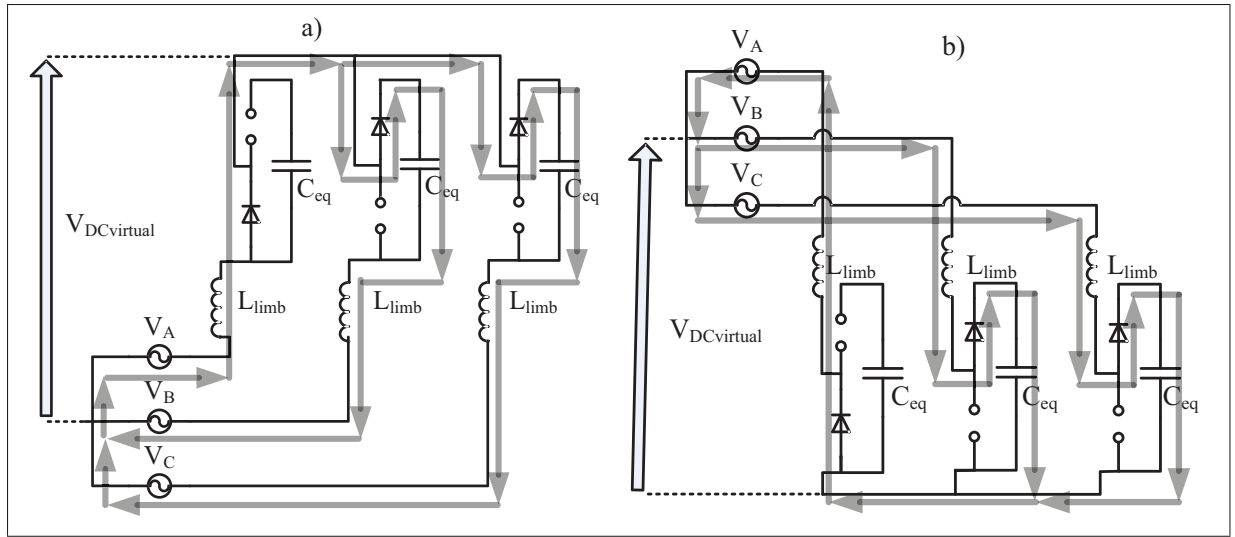


Figure 2.3 a) Sequence 1 :  $I_{AU} < 0, I_{BU} > 0$  &  $I_{CU} > 0$   
 b) Sequence 4 :  $I_{AL} < 0, I_{BL} > 0$  &  $I_{CL} > 0$

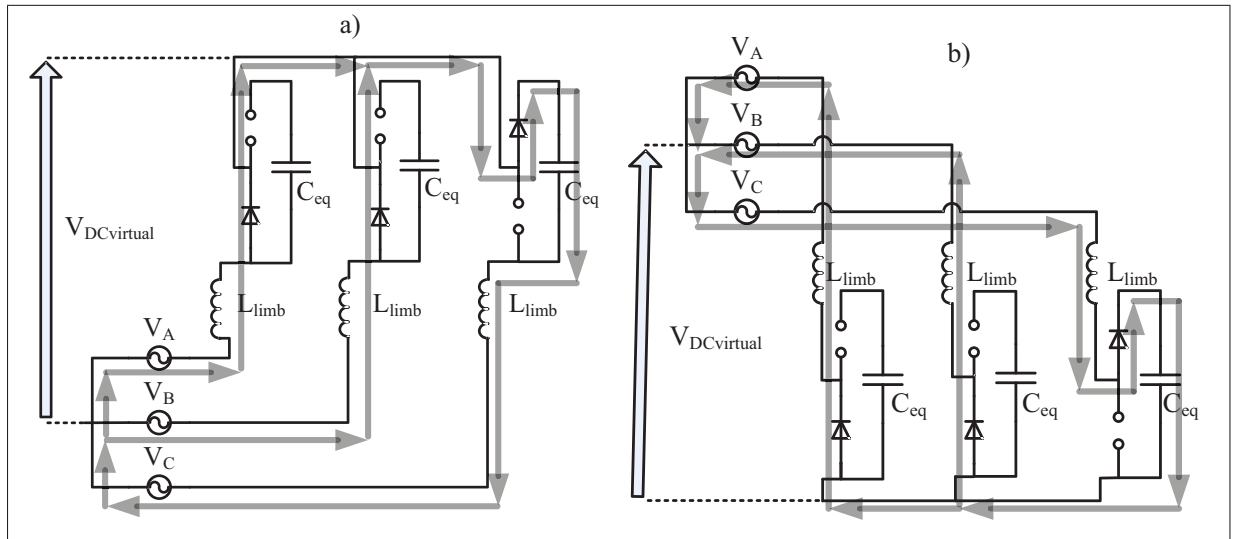


Figure 2.4 a) Sequence 2 :  $I_{AU} < 0, I_{BU} < 0$  &  $I_{CU} > 0$   
 b) Sequence 5 :  $I_{AL} < 0, I_{BL} < 0$  &  $I_{CL} > 0$

zero, only two of the three AC sources will play a role in the value of the virtual DC bus. Also once the current reaches zero it should remain zero since neither anti-parallel diodes from the cell can be polarized. If the AC voltage is positive, the lower switch anti-parallel diode cannot conduct since the virtual DC bus will always have a higher potential. As for the anti-parallel

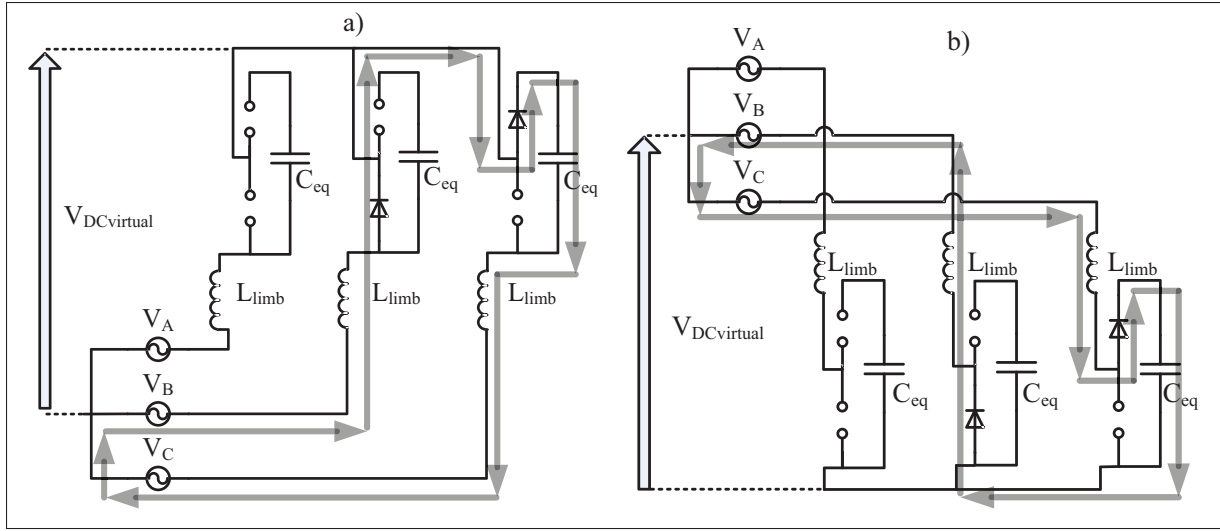


Figure 2.5 a) Sequence 3 :  $I_{AU}=0$ ,  $I_{BU}<0$  &  $I_{CU}>0$   
 b) Sequence 6 :  $I_{AL}=0$ ,  $I_{BL}<0$  &  $I_{CL}>0$

diode of the upper switch of the cell, because the sum of the cells capacitor is charged to  $V_{DC}$ , it will remain blocked.

From the six sequences shown in figure 2.3 to 2.5, three Laplace equivalent circuits can be obtained. In frequency domain, inductance and capacitor, each with its own initial value in the time domain, becomes inductance and capacitor in series with a voltage source. Inductance value is now multiplied by the Laplace variable  $s$  and the voltage source is equal to the inductance value multiplied by the initial current. Capacitor value becomes one over  $s$  over the capacitor value of the time domain. The voltage source in series with the capacitor has the same initial value as the capacitor in time domain divided by  $s$  (Nilsson and Riedel, 2008). Figure 2.6 is the frequency domain when two of the limb currents are positive and one is negative, equivalent to figure 2.3. Since  $V_{DC \text{ virtual}}$  is determined by the cell capacitor voltage, the same equivalent circuit can be used for the upper or lower half of the converter. Likewise, figure 2.7 and 2.8 are the equivalent circuit of figure 2.4, and figure 2.5 respectively.

The equivalent circuit depicted in figure 2.6 yields (2.3), when Kirchhoff's current law is applied.  $V_A$ ,  $V_B$  and  $V_C$  refer to the AC voltage. Since the over-voltage occurs over a very short period,  $V_A$ ,  $V_B$  and  $V_C$  are assumed to be constant.  $V_{CB}$  and  $V_{CC}$  are the initial voltages for the

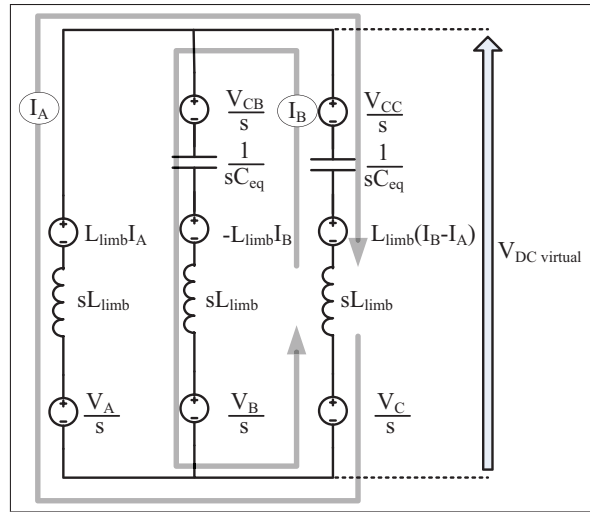


Figure 2.6 Laplace equivalent circuit of sequence 1 and 4

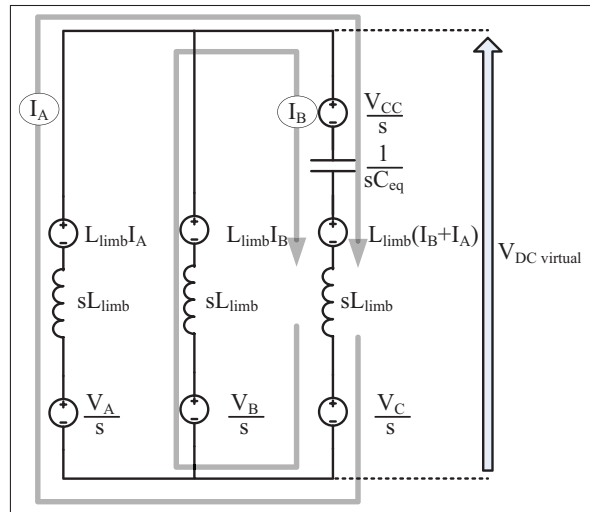


Figure 2.7 Laplace equivalent circuit of sequence 2 and 5

inserted capacitors just after blocking with all the cells in series.  $I_A$  and  $I_B$  are the initial limb currents. These equations are valid while the anti-parallel diodes are conducting which only lasts for a very short period of time and for this reason the different voltages are assumed constant.

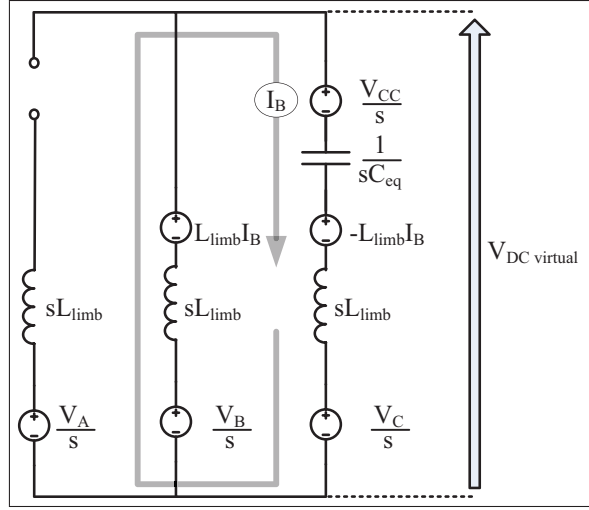


Figure 2.8 Laplace equivalent circuit of sequence 3 and 6

$$\begin{aligned}
 0 = & \frac{V_{\text{DC virtual}} - L_{\text{limb}} \cdot I_A - \frac{V_A}{s}}{sL_{\text{limb}}} + \frac{V_{\text{DC virtual}} - \frac{V_{C_B}}{s} + L_{\text{limb}} \cdot I_B - \frac{V_B}{s}}{\frac{1}{sC_{eq}} + sL_{\text{limb}}} \\
 & + \frac{V_{\text{DC virtual}} - \frac{V_{C_C}}{s} - L_{\text{limb}} \cdot (I_B - I_A) - \frac{V_C}{s}}{\frac{1}{sC_{eq}} + sL_{\text{limb}}} \quad (2.3)
 \end{aligned}$$

Equations (2.4) and (2.5) are obtained from figure 2.7 and figure 2.8.

$$\begin{aligned}
 0 = & \frac{V_{\text{DC virtual}} - L_{\text{limb}} \cdot I_A - \frac{V_A}{s}}{sL_{\text{limb}}} + \frac{V_{\text{DC virtual}} - L_{\text{limb}} \cdot I_B - \frac{V_B}{s}}{sL_{\text{limb}}} \\
 & + \frac{V_{\text{DC virtual}} - \frac{V_{C_C}}{s} + L_{\text{limb}} \cdot (I_B + I_A) - \frac{V_C}{s}}{\frac{1}{sC_{eq}} + sL_{\text{limb}}} \quad (2.4)
 \end{aligned}$$

$$\begin{aligned}
 0 = & \frac{V_{\text{DC virtual}} - L_{\text{limb}} \cdot I_B - \frac{V_B}{s}}{\frac{1}{sC_{eq}} + sL_{\text{limb}}} + \frac{V_{\text{DC virtual}} - \frac{V_{C_C}}{s} + L_{\text{limb}} \cdot (I_B) - \frac{V_C}{s}}{\frac{1}{sC_{eq}} + sL_{\text{limb}}} \quad (2.5)
 \end{aligned}$$

Solving eq. (2.3) to (2.5) for  $V_{\text{DC virtual}}$  and transforming them into time-domain yield equations (2.6) to (2.8). Voltage value of the virtual DC bus is then obtained based on  $V_{A,B,C}$ ,  $V_{C_B,C_C}$  and  $I_{A,B}$

$$V_{\text{DC virtual}}(t) = V_A + \cosh\left(\frac{\hat{i}t}{\sqrt{3}\sqrt{L_{\text{limb}}C_{eq}}}\right) \frac{(V_B + V_C + V_{C_B} - 2V_A + V_{C_C})}{3} + \sinh\left(\frac{\hat{i}t}{\sqrt{3}\sqrt{L_{\text{limb}}C_{eq}}}\right) \frac{\hat{i}I_A \sqrt{L_{\text{limb}}C_{eq}}\sqrt{3}}{3C_{eq}} \quad (2.6)$$

$$V_{\text{DC virtual}}(t) = \frac{V_A}{2} + \frac{V_B}{2} + \cosh\left(\frac{\hat{i}\sqrt{2}t}{\sqrt{3}\sqrt{L_{\text{limb}}C_{eq}}}\right) \frac{2V_{C_C} + 2V_C - V_A - V_B}{6} + \sinh\left(\frac{\hat{i}\sqrt{2}t}{\sqrt{3}\sqrt{L_{\text{limb}}C_{eq}}}\right) \frac{\hat{i}\sqrt{L_{\text{limb}}C_{eq}}(I_A + I_B)}{\sqrt{6}C_{eq}} \quad (2.7)$$

$$V_{\text{DC virtual}}(t) = V_B + \cosh\left(\frac{\hat{i}t}{\sqrt{2L_{\text{limb}}C_{eq}}}\right) \left(\frac{V_{C_C} - V_B + V_C}{2}\right) + \sinh\left(\frac{\hat{i}t}{\sqrt{2L_{\text{limb}}C_{eq}}}\right) \hat{i}\sqrt{2L_{\text{limb}}C_{eq}}I_B \quad (2.8)$$

Again, keeping in mind that this phenomenon lasts for a very short period of time ( $t \approx 0$ ) and by considering  $V_{C_B} = V_{C_C} = V_{DC}$ , (2.6), (2.7) and (2.8) become

$$V_{\text{DC virtual}}(t) = \frac{2V_{DC}}{3} + \underbrace{\frac{(V_A + V_B + V_C)}{3}}_0 \quad (2.9)$$

$$V_{\text{DC virtual}}(t) = \frac{V_{\text{DC}}}{3} + \underbrace{\frac{(V_A + V_B + V_C)}{3}}_0 \quad (2.10)$$

$$V_{\text{DC virtual}}(t) = \frac{V_{\text{DC}}}{2} + \underbrace{\left(\frac{V_B + V_C}{2}\right)}_{\text{max}=\frac{V_{\text{AC}}}{2}} \quad (2.11)$$

According to equations (2.9) to (2.11), one can observe that the voltage of the virtual DC bus, for the upper or lower half of the converter, is between  $\frac{1}{3}$  and  $\frac{2}{3}$  of total pole-to-pole DC bus which is equal to  $V_{\text{DC}}$  just before blocking. If both, upper and lower half of the converter are considered, the total pole-to-pole voltage of the virtual DC bus should be between  $\frac{2}{3}$  and  $\frac{4}{3}$  of its regulated value depending on the limbs current polarity. This shows that depending on the limb currents polarity at blocking, voltage on DC bus can drop to an under-voltage or spike to an over-voltage. This will be proven by using simulation and experimental results.

## 2.4 Simulation results

Simulation of MMC has been proven to be challenging because of the high number of power components. This has lead to new modelling techniques to obtain swiftly accurate simulation results (Gnanarathna *et al.*, 2011; Ahmed *et al.*, 2012; Grégoire *et al.*, 2014a, 2011b). The over-voltage transient observed in this paper lasts for a very short period of time, from a few tenths to several  $\mu\text{s}$ . These performances were obtained using OPAL-RT developed MMC models on FPGA chips (Grégoire *et al.*, 2011a), where a time-step as low as 500 ns can be obtained. In this simulation, a disturbance is caused by miss-firing cells which results in a circulating current and blocking of the converter gating signals at 0 ms. Figure 2.9 shows the converter behaviour from the moment when the pulses are blocked until all currents reached zero. Figure 2.9 a) shows the different voltage fluctuations, over/under-voltage, occurring on the virtual DC bus for both the positive and negative pole. Currents for the upper and lower half of the converter can be observed in figure 2.9 b) and figure 2.9 c).

During  $T_1$ , a pole-to-pole voltage of  $\frac{4}{3} V_{\text{DC}}$  is obtained because  $I_{\text{AU}}$  and  $I_{\text{BU}}$  of the upper limb and  $I_{\text{AL}}$  and  $I_{\text{CL}}$  of the lower limb are positive, which correspond to the operating sequence



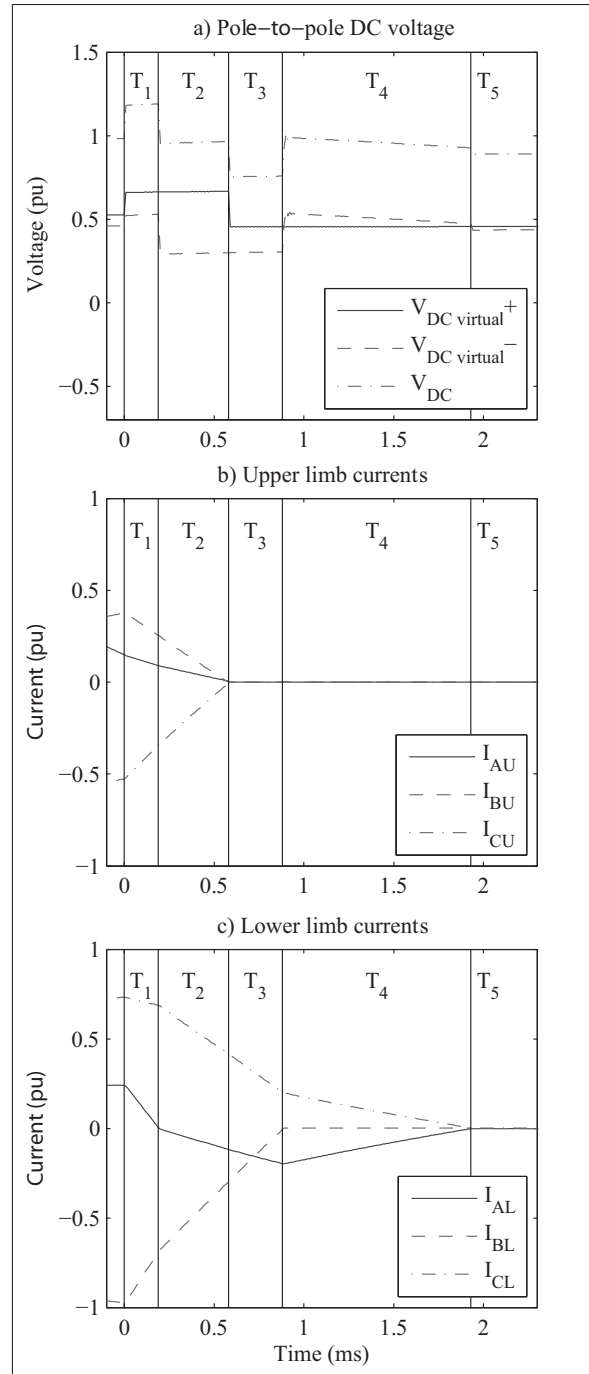


Figure 2.9 Simulation results of a blocking sequence

2 (figure 2.3 a)) and 4 (figure 2.3 b)). During the time interval  $T_2$ ,  $I_{AL}$  becomes negative and the sequence then becomes 1 (figure 2.3 a)) and 5 (figure 2.4 b)).  $V_{DC \text{ virtual}+}$  remains  $\frac{2}{3}$  while  $V_{DC \text{ virtual}-}$  becomes  $\frac{1}{3}$  making pole-to-pole voltage equal to  $V_{DC}$ . At time interval  $T_3$ , the three

currents from the upper limbs reach zero and  $V_{DC \text{ virtual}+}$  is now equal to the AC voltage as seen from the DC side where as  $V_{DC \text{ virtual}-}$  stays at  $\frac{1}{3}$  of  $V_{DC}$ . During  $T_4$ ,  $I_{BL}$  from the lower limb reaches zero,  $I_{AL}$  and  $I_{CL}$  have the same value with opposite polarity as depicted in figure 2.5 b). Finally, at  $T_5$ , all currents are equal to zero and the pole-to-pole voltage is equal to the rectified AC voltage.

Simulation results clearly demonstrate that uncontrolled blocking of the gate signals can result in an over-voltage on the virtual DC bus. In this case, all six limb currents reach zero in 2 ms but the over-voltage itself lasted only 190  $\mu$ s. Its duration could even be shorter if blocking is initiated with smaller currents. If the converter is not working at nominal power, it can be hard to notice this phenomena in simulation, unless a very small time-step is used.

## 2.5 Experimental Results

A down-scaled MMC prototype has been developed and used at KTH Royal Institute of Technology in Stockholm, Sweden to validate the operation of the MMC during normal and disturbed regimes. The laboratory prototype is shown in figure 2.10, and the parameters used for this experiment are summarized in Table 2.1. The converter operates as a STATCOM, without the use of DC bus capacitor, as mentioned in section 2.3. The converter is connected to the AC network via a line inductance of 0.3 p.u. and synchronized to the grid voltage using a PLL. The converter is controlled using the so-called "direct" modulation principle, i.e. purely sinusoidal insertion indices (Antonopoulos *et al.*, 2009).

After blocking all the switches, an over-voltage is observed at the DC-bus, due to the two charging currents in phase A. When all the limb currents drop to zero, and the energy stored in all the limb inductors is dissipated, the DC-side voltage drops to the rectified AC-side voltage; just as it can be observed when having a six-pulse diode rectifier. These results confirmed those previously obtained in section 2.3 and 2.4.

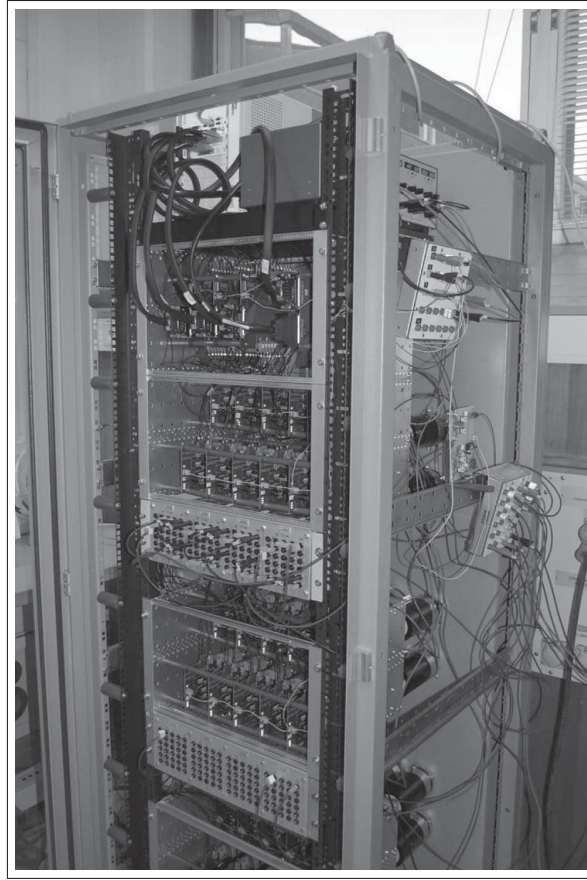


Figure 2.10 Photograph of the 3.25kVA experimental prototype of a five-level MMC

## 2.6 Remedial Strategy

The solution for an over-voltage-free blocking sequence is to block the gating signal at zero-crossing of each limb current. It is assumed that the converter is no longer following the reference set by the controller and therefore limb current cannot be safely controlled to zero. Figure 2.12 shows the logic to be applied for each limb gating signal. "Pulse enable" comes from the main controller. A value of 1 means that the cells from that limb are actively controlled. A value of 0 indicates that the controller is blocking the gating signals, forcing all gates to zero. If the logic detecting the zero-crossing of the limb current is inactive, the output "Pulse blocking" would be equal to "Pulse enable". By adding the logic of zero-crossing, when

Table 2.1 Experimental Setup Parameters

<b>Grid parameters</b>	
Grid line-to-line voltage	125 V
Grid frequency	50 Hz
Line inductance	10 mH
<b>Converter parameters</b>	
Apparent power	3.25 kVA
Line-to-line voltage	180 V
Line current	10 A
Number of cells per limb	5
Cell average voltage	80 V
Pole-to-pole rated dc-bus voltage	400 V
Cell capacitance	3.3 mF
Limb inductance	4.67 mH
<b>Controller parameters</b>	
Control-loop step time	200 $\mu$ s
Measurement sampling time	20 $\mu$ s

"Pulse enable" becomes low, "Pulse blocking" will stay high until the limb current crosses zero. Figure 2.13 shows the timing diagram of the scheme.

As it is shown in figure 2.13, when "Pulse enable" becomes low, "Pulse blocking" stays high and the converter is still being controlled, although limb currents are not completely responding until "limb current" reaches zero. Doing so, each limb voltage is controlled until there is no more energy stored in the limb inductance and therefore no DC over-voltage can result from uncontrolled sub-module. When this scheme is applied to each limb, the blocking sequence obtained is shown in figure 2.14. "Pulse enable" becomes low at 1.4545 s. The first current reaching zero is the one from the upper limb phase A within 1 ms after the blocking command, followed by the lower limb phase C, 3 ms later. When one of the currents from the upper or lower half of the converter reaches zero, the two remaining currents have the same value with opposite polarity. Therefore the two remaining currents in the upper or in the lower half of

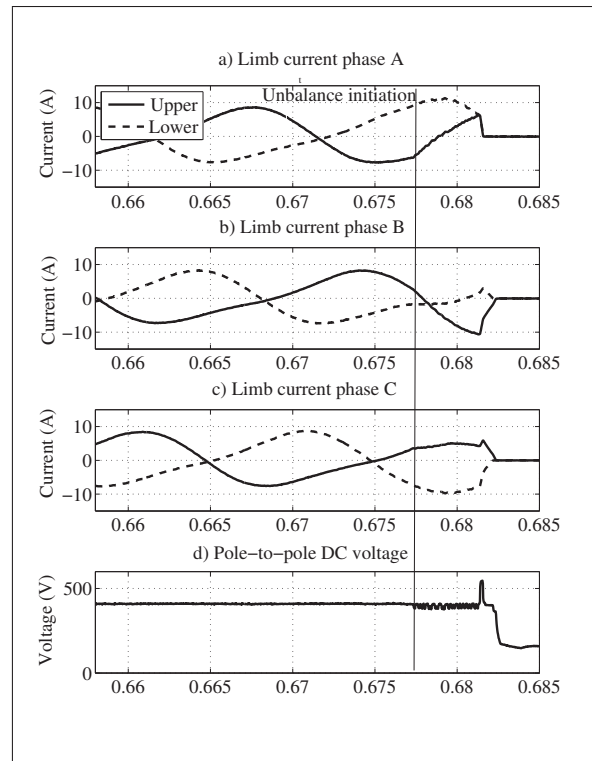


Figure 2.11 Limb currents and DC bus voltage transients at blocking as obtained from experimental setup

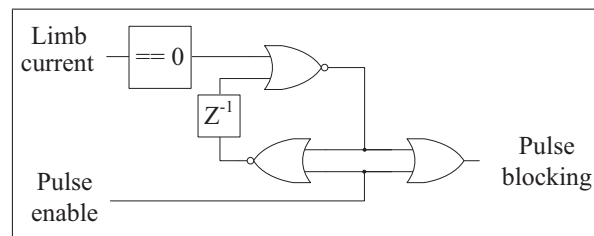


Figure 2.12 Control scheme for a single limb to block at zero-crossing

the converter will cross zero at the same time. A total of 12 ms have elapsed for all currents to reach zero and all the switches are turned OFF. During this time the DC bus voltage never exceeds its rated value. It has been demonstrated that the converter comes to a complete stop in a few milliseconds without any over-voltage that could be damaging to the converter.

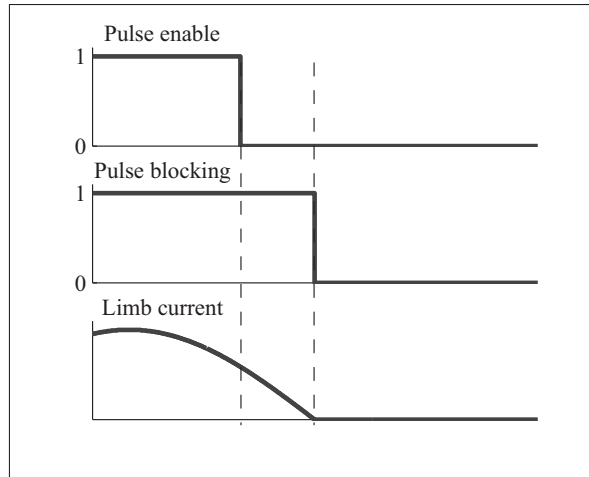


Figure 2.13 Timing diagram to block at zero-crossing

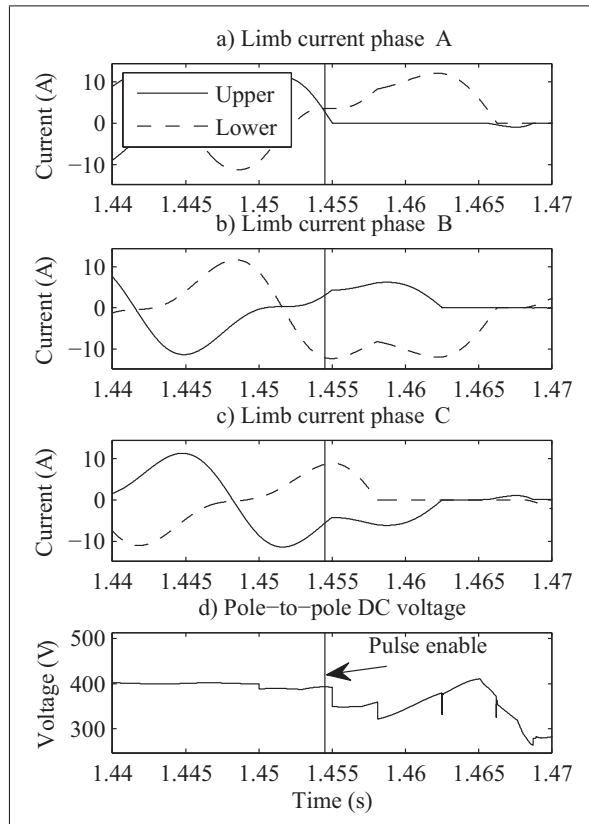


Figure 2.14 Limb currents and DC bus voltage for blocking at zero-crossing for each limb

## **2.7 Conclusion**

In the first part of the paper, a potentially harmful over-voltage transient that could occur on MMC using HB-SM structure was identified. This transient has been studied by means of analytical equations, real-time simulation and experimental validation. Following this analysis, a remedial strategy based on zero-crossing detection of each limb current has been proposed and tested to overcome this unwanted phenomenon. This novel approach allows to refine the design of MMC converter devices rating as well as the inductance inner layer insulation, due to the better understanding of the transient behaviour of the converter. Such transients should also be taken into account in the protection coordination of a DC network.

## **2.8 Acknowledgement**

The authors would like to show their gratitude to OPAL-RT Technologies for their support. They also wish to thank FQRNT, CRSNG, ELEKTRA AB, and Canada Research Chairs for supporting this work.





## CHAPTER 3

### REAL-TIME SIMULATION BASED MULTI-SOLVER DECOUPLING TECHNIQUE FOR COMPLEX POWER ELECTRONICS CIRCUITS

Luc-André Grégoire<sup>1</sup>, Handy Fortin Blanchette<sup>1</sup>, Jean Bélanger<sup>2</sup>, Kamal Al-Haddad<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering , École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

<sup>2</sup> CEO and CTO, OPAL-RT technologies Inc.,  
1751 Richardson, Suite 2525, Montréal, Québec, Canada H3K 1G6

Article submitted to «IEEE Transactions on Power Delivery» in June 2015.

#### **abstract**

This paper proposes a new method to decouple and subdivide electrical circuits, containing power electronics devices, in order to achieve fast and accurate real-time simulation. In this method, each state variable can be discretized using different discretization methods. Combining implicit and explicit ODE solvers, state-space equations are decoupled while remaining accurate and stable. Unlike most traditional decoupling technique previously proposed, this one does not require artificial delay or supplementary states to be added in order to decouple the system. Furthermore, this technique is meant to be implemented with commercially available simulation software. Doing so, a large and complex circuit containing several hundreds of state variables can be easily and accurately simulated with minor modification to the existing models. Finally, stability and accuracy of the proposed technique is thoroughly demonstrated in a numerical example during steady state and under fault conditions.

#### **3.1 Introduction**

Nowadays, simulation is an essential development tool for researchers, engineers and practitioners. The ever increases of time varying discrete system, containing disperse time constants

and discontinuities, combined with system complexity has motivated and lead researchers toward the development of fast and accurate real-time simulators. To achieve such a goal, considering complex power electronics topologies, accurate decoupling, resolution accuracy and algorithm stability have become the ultimate goal to reach, especially in real-time simulation applications. In the case of large power systems, propagation delays are often used for decoupling purposes using distributed parameters lines, also known as Bergeron's line model (Dommel, 1969; Jalili-Marandi *et al.*, 2010; Watson and Arrillaga, 2003). This method cannot be applied for shorter lines where propagation delays are smaller than simulation time-step, as it is often the case when dealing with power system integrating power electronics devices in HVDC, microgrids, renewable energy integration etc. In such case, Bergeron's distributed parameters line can still be used by forcing some of the parameters to obtain exactly one-time step propagation delay. This results in adding shunt parasitic capacitors to an otherwise purely inductive line; such a line is then referred as a stubline. (Hong *et al.*, 2009; Watson and Arrillaga, 2003; Wang *et al.*, 2010). Another method for decoupling system is to add a delay on slow varying states, such as a capacitor on a DC-BUS (Dufour and Belanger, 2004). However, adding such unnatural delay can result in numerical instability of the system (Dufour and Belanger, 2004). In (Kato *et al.*, 2014, 2013), a combination of implicit and explicit solvers are used to eliminate these unnatural delays and reduce instability issues. Large circuits are subdivided into smaller sub-circuits and discretized with Backward Euler while state between each sub-circuits uses Forward Euler. At each simulation step, an iterative method is therefore used, where the time-step may be reduced to ensure convergence of the decoupled systems. Though, this technique gives acceptable results, such iterative approach cannot be applied to real-time simulation because of its hard-time constraints. In (Benigni *et al.*, 2014b), a combination of a few state variables is used to decouple and subdivide the circuit into numerous subsystems. Each major subsystem is solved using its own locally assigned solving method, like Euler, Trapezoidal or even Runge-kutta. The coupling state-variables, used between the sub-systems, are solved strictly with the trapezoidal method; they are also coupled to other different sub-circuits with controlled sources and impedances. Once decoupled, a method to study the complete system is given. The only drawback of the latter is the necessity of up-

dating its equations at intermediate time-step. This naturally leads to decreasing by a factor of two the effective time-step. Authors in (Tomim *et al.*, 2010; Dufour *et al.*, 2011b) propose a similar two-step approach, where solutions for each sub-circuit is done individually using either nodal approach or state-space modeling approach. Once solved, the Norton equivalent of each sub-circuit is found, and a global solution is obtained through a nodal method. But like the previous method, where intermediate steps were required, combined state-space, and nodal solutions must be computed for each time-step. The new multi-solver method that is proposed in this paper offers a single-step solution, where the coupling is achieved using multiple components or state-variables, as reported in (Benigni *et al.*, 2014b). Also, instead of using solely an implicit solver, a combination of implicit and explicit solvers is used. Since the approach is limited to a group of coupling states, the overall stability of the circuit can be verified easily, as it is demonstrated with the proposed pole analysis method also presented. Furthermore, this proposed method can be implemented in real-time simulation commercially available software to enhance their performance.

This paper is divided as follow. In section 3.2, time and solvers constraints for real-time simulation are presented. The proposed multi-solver technique, its implementation, and its stability analysis are presented in section 3.3 ;it is then followed in section 3.4 by a numerical example comparing the proposed method with classical ones. Finally, conclusions are drawn in section 3.5.

### **3.2 Time and solvers constraints of real-time simulation**

Real-time simulation should not be mistaken with regular simulation. The main difference is that real-time simulation needs to be synchronized with external-hardware; its execution time must be deterministic. For this reason, fixed-step solvers are used in real-time simulation where no iterations are made and execution time remains the same for every time-step. In this section, both simulation cycle and solvers of real-time application are presented.

### 3.2.1 Deterministic simulation

In real-time simulation, each simulation step can be divided into three sections. First, signals required for the simulation are sampled through analog or digital inputs. Secondly, one step of simulation is executed, and new values are computed. Finally, newly computed values are sent to the simulator's outputs, where they are applied to external hardware. Figure 3.1 a) shows the different actions needed for every time-step.

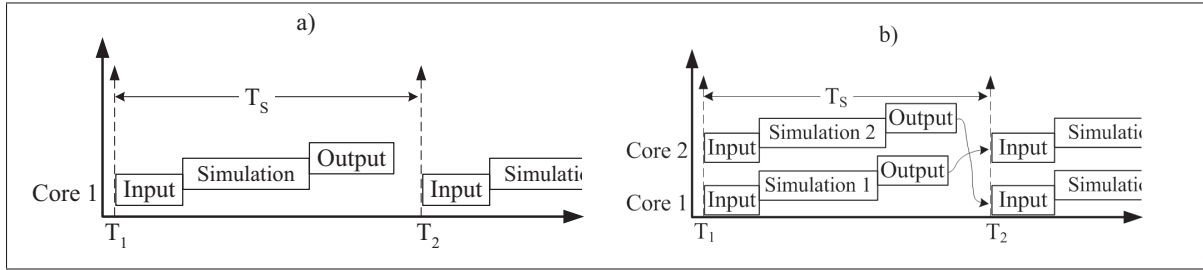


Figure 3.1 Execution timeline of real-time simulation  
a) using a single core b) using multi-core

A discretized system with a sampling time  $T_s$  of  $50 \mu s$  needs to execute the following three steps: reading inputs, computing the model, and sending the outputs, within  $50 \mu s$ . If this timing requirement cannot be met, then  $T_s$  should be increased, reducing, therefore, the accuracy and possibly the stability of the model. If  $T_s$  cannot be increased, it is possible to divide the computing of the model between several central processing units (CPU), as shown in figure 3.1 b). The same, above mentioned, three steps are applied to multi-core microprocessors. First signals from different cores are acquired from shared-memory, simulation is computed, and newly computed values are sent back to shared-memory. In figure 3.1 b), the process "Simulation 2" can only use values from the previous step of "Simulation 1". This is only possible if both processes are decoupled as it can be done with the method proposed in this paper.

### 3.2.2 Numerical integration method

When looking at discretization of a continuous system, numerous integration methods are nowadays available. Whichever family of solver is considered, one should remember that numerical integration remains an approximation of (3.1).

$$s = \frac{\ln(z)}{T} \quad (3.1)$$

Where  $s$  is the Laplace operator,  $T$  is the integration time-step, and  $z$  is the discrete operator. Equation (3.1) can be approximated with different methods such as Taylor's series or Padé's approximation (Hartley *et al.*, 1994; Wanner and Hairer, 1991). The more terms are kept from the series, the smaller is the local truncation error (LTE). Keeping only one or two terms in the series gives three of the most popular approximations in power system simulation, which are Forward Euler (FE), trapezoidal (TR) and Backward Euler (BE) as shown in (3.2), (3.3) and (3.4).

$$s_{FE} = \frac{1}{T} (z - 1) \quad (3.2) \quad s_{TR} = \frac{2}{T} \left( \frac{z - 1}{z + 1} \right) \quad (3.3)$$

$$s_{BE} = \frac{1}{T} \left( \frac{z - 1}{z} \right) \quad (3.4)$$

Equations (3.5), (3.6) and (3.7) give LTE for each different solver.

$$LTE_{FE} = \frac{1}{2} T^2 \ddot{x}_{FE} + O(T^3) \quad (3.5)$$

$$LTE_{TR} = \frac{1}{6} T^3 \ddot{\ddot{x}}_{TR} + O(T^4) \quad (3.6)$$

$$LTE_{BE} = -\frac{1}{2} T^2 \ddot{x}_{BE} + O(T^3) \quad (3.7)$$

Further discussions and numerical analysis can be found in (Najm, 2010); also special attention should be drawn to the following observations.  $LTE_{FE}$  has a positive value and FE is an under-damped solver; meaning that steady-state regime takes more time to be reached. BE is known

to be an over-damped solver, increasing its stability, and  $LTE_{BE}$  has a negative value. Both (3.5) and (3.7) have truncation errors of  $T^2$  magnitude, with opposite sign.  $LTE_{TR}$  has a factor of magnitude  $T^3$ , making TR the most accurate solver between those three as it has the smallest LTE.

To discretize continuous state-space equations, the Laplace operator  $s$ , is replaced by an expression in  $z$ , like the one from (3.2), (3.3) or (3.4), from which discrete equations are obtained. State-variable multiplied by  $z$  are isolated, and become future values or values at the next computation step. This method is called the operational substitution (OS)(Hartley *et al.*, 1994). When using OS, every  $s$  are replaced by the same expression in  $z$ . In the next section, it will be demonstrated that combining different solvers within one system can allow decoupling of its state-variables.

### 3.3 Multi-solver method

The proposed method relies on two key elements. One is the operational substitution method (OS); which allows for flexible discretization of continuous system; and the other is the complementarity of Backward and Forward Euler integration method, including their advantages and flaws. An introduction to multi-solver methods is first presented followed by multi-solver application to a larger system.

#### 3.3.1 Introduction to multi-solver

In this paper, the term solver refers to the integration method used for discretization. The term multi-solver (MS) refers to using different approximations for different state variables within one system. This may seem counter-intuitive but by combining an explicit method, like FE, to an implicit one, like BE, allows decoupling states from a system and resolving them in parallel. Taking, for example, the second order state-space system with initial condition equal to zero in (3.8). The continuous matrices  $A^c$  and  $B^c$  can be discretized using a multi-solver method. This is done by first rewriting each state-equation as an individual equation for  $x_1$  and  $x_2$ . Then,

Laplace operator  $s$  multiplying  $x_1$  and  $x_2$  is replaced by the expression in  $z$  from equation (3.2), and (3.4), respectively; this will then be referred as an FEBE solver yielding (3.9) and (3.10)

$$\begin{bmatrix} x_1 s \\ x_2 s \end{bmatrix} = \overbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}^{A^c} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \overbrace{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}^{B^c} u \quad (3.8)$$

$$x_1 \frac{z-1}{T} = a_{11}x_1 + a_{12}x_2 + b_1u \quad (3.9)$$

$$x_2 \frac{z-1}{Tz} = a_{21}x_1 + a_{22}x_2 + b_2u \quad (3.10)$$

Using algebra, and both (3.9) and (3.10), state-variables multiplied by  $z$  are isolated, and equations (3.11) and (3.12) are obtained.

$$x_1 z = (Ta_{11} + 1)x_1 + (Ta_{12})x_2 + (Tb_1)u \quad (3.11)$$

$$\begin{aligned} x_2 z = & - \left( \frac{T^2 a_{21} a_{11} + Ta_{21}}{Ta_{22} - 1} \right) x_1 - \left( \frac{T^2 a_{21} a_{12} + 1}{Ta_{22} - 1} \right) x_2 \\ & - \left( \frac{T^2 a_{21} b_1}{Ta_{22} - 1} \right) u - \left( \frac{Tb_2}{Ta_{22} - 1} \right) uz \end{aligned} \quad (3.12)$$

Equation (3.11), being an explicit solution, only needs the previous values of the system; therefore no  $z$  is found on the right side of the equation. In (3.12), there is now an input  $u$  multiplied by  $z$ , which is inherent to implicit methods; outputs of the present step is computed based on the input of the previous and present step. Equations (3.11) and (3.12) can now be simulated in parallel, as shown in figure 3.1. Input  $u$  and (3.12) would need to be simulated on the same core since the value of  $u$  at the present step is required to compute  $x_2$ . As for (3.11), it can be computed on a different core as it only needs values of  $x_2$  and  $u$  from the previous step. In order to test stability through poles location, both equations need to be in the same state-space matrix system. Equation (3.13) gives such a system where (3.11) and (3.12) are translated in a

discrete state-space system using recurrent equations.

$$\begin{aligned}
 \begin{bmatrix} x_{1n} \\ x_{2n} \end{bmatrix} &= \overbrace{\begin{bmatrix} Ta_{11} + 1 & Ta_{12} \\ -\frac{T^2 a_{21} a_{11} + Ta_{21}}{Ta_{22} - 1} & -\frac{T^2 a_{21} a_{12} + 1}{Ta_{22} - 1} \end{bmatrix}}^{A^d} \begin{bmatrix} x_{1_{n-1}} \\ x_{2_{n-1}} \end{bmatrix} \\
 &+ \overbrace{\begin{bmatrix} Tb_1 \\ -\frac{T^2 a_{21} b_1}{Ta_{22} - 1} \end{bmatrix}}^{B^{d1}} u_{n-1} + \overbrace{\begin{bmatrix} 0 \\ -\frac{Tb_2}{Ta_{22} - 1} \end{bmatrix}}^{B^{d2}} u_n
 \end{aligned} \tag{3.13}$$

Note that there is now two distinct input matrices in (3.13),  $B^{d1}$  and  $B^{d2}$ . Throughout this paper, matrices with subscript  $d1$  indicate a discrete matrix which is multiplied by a vector of values from the previous step. Likewise,  $d2$  refers to a discrete matrix which is multiplied by a vector of values available at the current step. In this example, the choice of the solver for  $x_1$  and  $x_2$  was done arbitrarily. In order to guide this choice of solvers, the LTE of the different solver method can be used. LTE are based on the second or third derivative of a system. Those derivatives are obtained by deriving state-space equation, as shown in (3.14).

$$\begin{aligned}
 \dot{X} &= AX + BU \rightarrow \ddot{X} = A\dot{X} + \overbrace{B\dot{U}}^0 \\
 \ddot{X} &= A(AX + BU) = AAX + ABU
 \end{aligned} \tag{3.14}$$

The term  $\dot{U}$  is equal to zero, assuming that the dynamic of the input is relatively slow. Applying (3.14) to (3.13), expressions for  $\ddot{x}_1$  and  $\ddot{x}_2$  are found. These expressions are functions of  $x_1$ ,  $x_2$ , and  $u$ . Although they cannot be fully evaluated, because  $x_1$ ,  $x_2$ , and  $u$  are time variant, therefore a general idea of how they evolve is obtained. When an FEBE solver is used, the resulting LTE would be a combination of (3.5) and (3.7) yielding (3.15).

$$LTE_{FEBE} = \frac{1}{2}T^2(\ddot{x}_{FE} - \ddot{x}_{BE}) + O(T^3) \tag{3.15}$$

In such case, if  $\ddot{x}_{FE}$  is in the same range than  $\ddot{x}_{BE}$ , they are canceling each other resulting in a smaller LTE. In section 3.2, it was mentioned that BE is an over-damp solver and its LTE



has a negative term. Therefore, if  $\ddot{x}_1$  and  $\ddot{x}_2$  are not in the same range, the state variable with the largest derivative should discretize with BE. Naturally, unless simulations are made,  $\ddot{x}_1$  and  $\ddot{x}_2$  remain approximation. The latter gives sufficient information to make an inform decision on the most appropriate solvers to be applied. In the following section, the most appropriate state-variables used to apply the proposed decoupling technic is discussed.

### 3.3.2 Testing stability of multi-solver method applied to large network

This method can easily be applied for small system, but applying it to very large one would take lots of effort, and most likely lead to numerous mistakes. Using commercial simulation software (Ourari *et al.*, 2007; Paré *et al.*, 2003; Mahseredjian *et al.*, 2007), very large circuits can be implemented using larger component libraries or even solver with higher accuracy (Dufour *et al.*, 2003a) while minimizing the possibility of errors. The proposed method is meant to be applied only to a few state variables from a larger system. The point chosen for decoupling should have as little interaction as possible with other state variables; also referred to as lightly coupled state-variables. Transmission lines are coupled to network only at their ends, and state variables within the line have little to no interaction with the aforementioned network. Therefore, transmission lines, or other components alike, are often the best location to decouple a system; allowing large networks to be subdivided in smaller one. For instance, network 1 and network 2 shown in figure 3.2 can be two very large systems coupled through transmission lines, with the state-space equations given by (3.16).

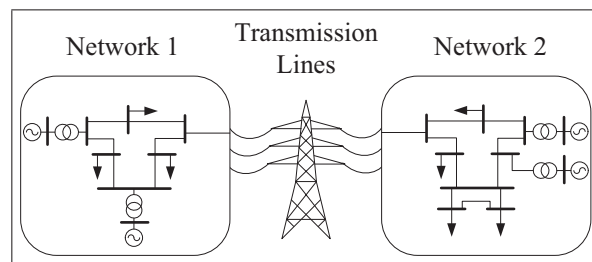


Figure 3.2 Example of two networks coupled with a transmission lines

$$\begin{bmatrix} X_{1s} \\ X_{Ts} \\ X_{2s} \end{bmatrix} = \overbrace{\begin{bmatrix} A_1^c & A_{1T}^c & 0 \\ A_{T1}^c & A_T^c & A_{T2}^c \\ 0 & A_{2T}^c & A_2^c \end{bmatrix}}^{A^c} \begin{bmatrix} X_1 \\ X_T \\ X_2 \end{bmatrix} + \overbrace{\begin{bmatrix} B_1^c & 0 \\ 0 & 0 \\ 0 & B_2^c \end{bmatrix}}^{B^c} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad (3.16)$$

In (3.16), state-space matrix  $A^c$  contains both networks and the transmission lines as shown figure 3.2.  $X_1$ ,  $X_T$ , and  $X_2$  are vectors containing state-variables belong exclusively to the network 1, transmission lines and network 2 receptively. States from  $X_1$  are only coupled to  $X_2$  through the transmission lines, therefore states from  $X_1$  are not directly dependent from the one of  $X_2$ , and vice and versa. Matrix  $A^c$  can be subdivided in smaller matrices where  $A_1^c$  and  $A_2^c$  are the state-space matrices of network 1 and network 2 and  $A_T^c$  is the one of the transmission lines.  $B_1^c$  and  $B_2^c$  in matrix  $B^c$  are the inputs applied to the network 1 and network 2. Sub-matrices  $A_{1T}^c$ ,  $A_{2T}^c$ ,  $A_{T1}^c$  and  $A_{T2}^c$  are coupling  $A_1^c$ ,  $A_T^c$  and  $A_2^c$  together. Now (3.16) can be divided in three sets of equations that can be discretized using various solvers. Values in (3.17) and (3.19) are obtained from commercially available software using their own solvers, and the proposed method is used to obtain (3.18). Equations (3.17) to (3.19) are under a generalized form, according to the choice of solver, some of their matrices might be sparse.

$$X_{1n} = A_1^{d1} X_{1n-1} + A_{1T}^{d1} X_{Tn-1} + A_{1T}^{d2} X_{Tn} + B_1^{d1} U_{1n-1} + B_1^{d2} U_{1n} \quad (3.17)$$

$$X_{Tn} = A_T^{d1} X_{Tn-1} + \begin{bmatrix} A_{T1}^{d1} & A_{T2}^{d1} \end{bmatrix} \begin{bmatrix} X_{1n-1} \\ X_{2n-1} \end{bmatrix} + \begin{bmatrix} A_{T1}^{d2} & A_{T2}^{d2} \end{bmatrix} \begin{bmatrix} X_{1n} \\ X_{2n} \end{bmatrix} \quad (3.18)$$

$$X_{2n} = A_2^{d1} X_{2n-1} + A_{2T}^{d1} X_{Tn-1} + A_{2T}^{d2} X_{Tn} + B_2^{d1} U_{2n-1} + B_2^{d2} U_{2n} \quad (3.19)$$

Since the system has been divided in three sub-systems, state-variables belonging to external sub-systems are now seen as inputs. When implicit solvers are used, those states not only require values of the previous step but also values at the present one. This results in yielding matrices  $A_{1T}^{d2}$ ,  $A_{T1}^{d2}$ ,  $A_{T2}^{d2}$ ,  $A_{2T}^{d2}$ ,  $B_1^{d2}$ , and  $B_2^{d2}$ . The whole discrete system is now given by (3.20) which becomes (3.21) once  $X_n$  has been isolated. Stability of the discretized and decoupled

system can now be studied using poles location representation obtained from (3.21).

$$\begin{aligned}
 \underbrace{\begin{bmatrix} X_n \\ X_{1_n} \\ X_{T_n} \\ X_{2_n} \end{bmatrix}}_{X_n} &= \underbrace{\begin{bmatrix} A_1^{d1} & A_{1T}^{d1} & 0 \\ A_{T1}^{d1} & A_T^{d1} & A_{T2}^{d1} \\ 0 & A_{2T}^{d1} & A_2^{d1} \end{bmatrix}}_{A^{d1}} \underbrace{\begin{bmatrix} X_{1_{n-1}} \\ X_{T_{n-1}} \\ X_{2_{n-1}} \end{bmatrix}}_{X_{n-1}} + \underbrace{\begin{bmatrix} 0 & A_{1T}^{d2} & 0 \\ A_{T1}^{d2} & 0 & A_{T2}^{d2} \\ 0 & A_{2T}^{d2} & 0 \end{bmatrix}}_{A^{d2}} \underbrace{\begin{bmatrix} X_{1_n} \\ X_{T_n} \\ X_{2_n} \end{bmatrix}}_{X_n} \\
 &+ \underbrace{\begin{bmatrix} B_1^{d1} & 0 \\ 0 & 0 \\ 0 & B_2^{d1} \end{bmatrix}}_{B^{d1}} \begin{bmatrix} U_{1_{n-1}} \\ U_{2_{n-1}} \end{bmatrix} + \underbrace{\begin{bmatrix} B_1^{d2} & 0 \\ 0 & 0 \\ 0 & B_2^{d2} \end{bmatrix}}_{B^{d2}} \begin{bmatrix} U_{1_n} \\ U_{2_n} \end{bmatrix}
 \end{aligned} \tag{3.20}$$

$$\begin{aligned}
 X_n &= A^{d1} X_{n-1} + A^{d2} X_n + B^{d1} U_{n-1} + B^{d2} U_n \\
 &= \underbrace{\left( I - A^{d2} \right)^{-1} A^{d1}}_{\text{Poles of the system}} X_{n-1} + \left( I - A^{d2} \right)^{-1} B^{d1} U_{n-1} + \left( I - A^{d2} \right)^{-1} B^{d2} U_n
 \end{aligned} \tag{3.21}$$

Poles from the discretized decoupled system can be compared with the one of the continuous coupled system in (3.16). To do so,  $z$  in (3.1) needs to be isolated, and  $s$  needs to be substituted by the continuous poles of the reference system,  $\lambda_s^{ref}$ , to obtained the reference poles,  $\lambda_z^{ref}$ , as shown in (3.22).

$$\lambda_z^{ref} = e^{\lambda_s^{ref} T} \tag{3.22}$$

If the poles of the decoupled system are within the unity circle, the model is stable. If they have the same value as the poles  $\lambda_z^{ref}$ , found with (3.22), then the decoupled system is also accurate. By referring to (Kato *et al.*, 2014, 2013; Hong *et al.*, 2009) only the stability of each individual sub-system could be tested with regards to the global system, whereas using (3.21), the stability of the whole system can be studied. Moreover, values from (3.17) and (3.19) are only required to test the stability of the decoupled model. Finally, in order to implement the proposed method, only the values from (3.18) are required.

### 3.3.3 Applying multi-solver method to large network

State-variables from (3.18) are integrated to dedicated simulation tools using companion model (Johnson, 2003b,a). This method consists in representing a circuit by a controlled source and equivalent impedance, as shown in figure 3.3. For explanation purposes, let (3.18) takes the

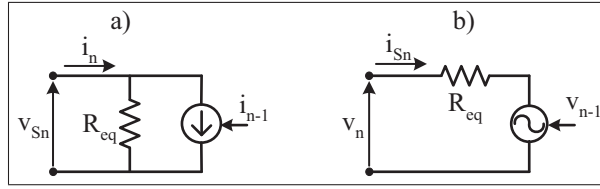


Figure 3.3 Companion model a) using equivalent current source  
b) using equivalent voltage source

following form (3.23). Vector  $X_T$  has a minimum of two state variables, but could be wider. Vectors  $X_{1_{n-1}}$ ,  $X_{T_{n-1}}$  and  $X_{2_{n-1}}$  are known at the beginning of the time-step, while  $X_{1_n}$ ,  $X_{T_n}$  and  $X_{2_n}$  need to be solved simultaneously.

$$\begin{aligned}
 \underbrace{\begin{bmatrix} i_n \\ \vdots \\ v_n \end{bmatrix}}_{X_{T_n}} &= A_T^{d1} \underbrace{\begin{bmatrix} i_{n-1} \\ \vdots \\ v_{n-1} \end{bmatrix}}_{X_{T_{n-1}}} + \underbrace{\begin{bmatrix} A_{T1}^{d2} & A_{T2}^{d2} \\ 0 & a_{11}^{d2} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{22}^{d2} & 0 \end{bmatrix}}_{\begin{bmatrix} A_{T1}^{d2} & A_{T2}^{d2} \end{bmatrix}} \underbrace{\begin{bmatrix} X_{1_n} & X_{2_n} \end{bmatrix}^T}_{\begin{bmatrix} \vdots \\ v_{Sn} \\ i_{Sn} \\ \vdots \end{bmatrix}} \\
 &+ \begin{bmatrix} A_{T1}^{d1} & A_{T2}^{d1} \end{bmatrix} \begin{bmatrix} X_{1_{n-1}} \\ X_{2_{n-1}} \end{bmatrix}
 \end{aligned} \tag{3.23}$$

It is important to note that the sparsity of the combine submatrices  $A_{T1}^{d2}$  and  $A_{T2}^{d2}$  is obtained by choosing an appropriate decoupling point. The equation for state-variable  $i_n$  can be rewritten

in (3.24) to be applied to the companion model of figure 3.3 a).

$$i_n = \overbrace{A^{d1}(i_n, :) X_{n-1}}^{i_{1n-1}} + \overbrace{a11 \frac{1}{Req} v_{S_n}}^{\frac{1}{Req}} \quad (3.24)$$

The same scheme is applied to  $v_n$  in (3.25) which applies to companion model of figure 3.3 b).

$$v_n = \overbrace{A^{d1}(v_n, :) X_{n-1}}^{v_{1n-1}} + \overbrace{a22 \frac{Req}{T2} i_{S_n}}^{Req} \quad (3.25)$$

Values found in matrices  $A_{T1}^{d2}$  and  $A_{T2}^{d2}$  are either resistance or conductance according to the companion model used. In regards to parallel computing,  $i_n$  can be solved simultaneously with  $v_{S_n}$  on one core, while  $v_n$  is computed on a different core with  $i_{S_n}$ . This is possible since only values from the previous step (n-1) are needed from state-variables of different cores. Application of this method will be further demonstrated by the example presented in next the section.

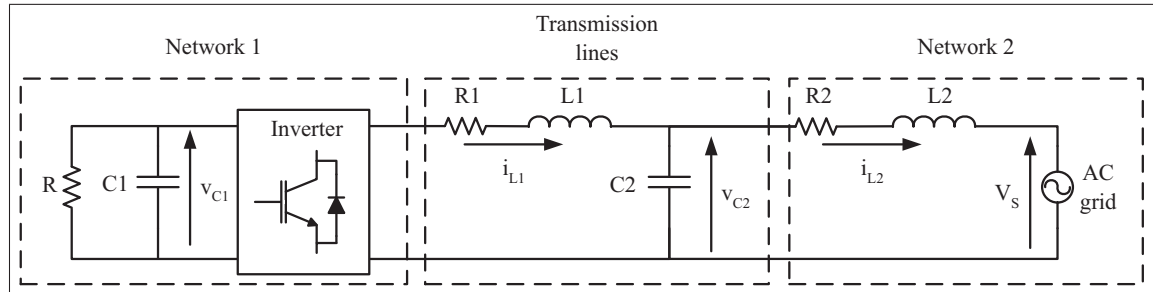


Figure 3.4 Grid connected inverter circuit with LCL filter

### 3.4 Numerical example

In this section, an example shows that the proposed method remains accurate even when the coupling is done with a non-linear time-varying system having power electronics converter. Using poles location analysis, the performance of the decoupling technique can be forecast

before running any simulation. Simulation results are obtained using Matlab/Simulink and SimPowerSystems (SPS) toolbox. Figure 3.4 shows an inverter feeding an RC load is decoupled from the grid using an LCL filter. Parameters for simulation of the filter were taken from (Twining and Holmes, 2003) and are given in table 3.1. State-space equations of the coupled system are given by (3.26).

Table 3.1 Simulation parameters of the circuit

Nominal voltage	100 V
Rated power	5 kVA
C1	2 mF
R	5 $\Omega$
L1	3.5 mH
R1	0.10 $\Omega$
C2	15 $\mu F$
L2	1.5 mH
R2	0.05 $\Omega$
Switching frequency	5 kHz
Sampling time	10 $\mu s$

$$\begin{bmatrix} v_{C1}s \\ i_{L1}s \\ v_{C2}s \\ i_{L2}s \end{bmatrix} = \begin{bmatrix} \frac{-1}{RC1} & \frac{-d}{C1} & 0 & 0 \\ \frac{d}{L1} & \frac{-R1}{L1} & \frac{1}{L1} & 0 \\ 0 & \frac{1}{C2} & 0 & \frac{-1}{C2} \\ 0 & 0 & \frac{1}{L2} & \frac{-R2}{L2} \end{bmatrix} \begin{bmatrix} v_{C1} \\ i_{L1} \\ v_{C2} \\ i_{L2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{-1}{L2} \end{bmatrix} V_S \quad (3.26)$$

According to the modulation technique applied to the inverter, the variable  $d$  can either be  $-1$ ,  $0$ , or  $1$ . In this example, state variables obtained from  $L1$  and  $C2$  are used for decoupling the inverter from the AC network. To choose the appropriate solver, (3.14) is applied to (3.26), using nominal RMS values for the state variables and the input  $V_S$ . This test has shown that  $\ddot{v}_{C2}$  has the largest values; therefore BE and FE should be used to discretize  $v_{C2}$  and  $i_{L1}$  respectively. The trapezoidal method is used for the remaining state variables,  $v_{C1}$ , and  $i_{L2}$ . As it can be seen in figure 3.4, the circuit has been divided in three sub-circuits like it was done in figure 3.2.

Network 1 is represented by  $R$ ,  $C1$ , and the inverter, network 2 is  $R2$ ,  $L2$ , and the AC grid, and the transmission lines are the components  $R1$ ,  $L1$ , and  $C2$ . Once discretized, equation (3.26) gives (3.27), (3.28), and (3.29), corresponding to discrete equations of network 1, transmission lines, and network 2. The corresponding matrices from (3.20) are identified in equations (3.27) to (3.29), and will be used to compute for poles location of the newly decoupled system.

$$v_{C1_n} = \overbrace{0.9990}^{A_1^{d1}} v_{C1_{n-1}} + \overbrace{\begin{bmatrix} -0.0025d & 0 \end{bmatrix}}^{A_{1T}^{d1}} \begin{bmatrix} i_{L1_{n-1}} \\ v_{C2_{n-1}} \end{bmatrix} + \overbrace{\begin{bmatrix} -0.0025d & 0 \end{bmatrix}}^{A_{1T}^{d2}} \begin{bmatrix} i_{L1_n} \\ v_{C2_n} \end{bmatrix} \quad (3.27)$$

$$\begin{bmatrix} i_{L1_n} \\ v_{C2_n} \end{bmatrix} = \overbrace{\begin{bmatrix} 0.9997 & -0.0029 \\ 0.6665 & 0.9981 \end{bmatrix}}^{A_T^{d1}} \begin{bmatrix} i_{L1_{n-1}} \\ v_{C2_{n-1}} \end{bmatrix} + \overbrace{\begin{bmatrix} 0.0029d \\ 0.0019d \end{bmatrix}}^{A_{T1}^{d1}} v_{C1_{n-1}} + \overbrace{\begin{bmatrix} 0 \\ -0.6667 \end{bmatrix}}^{A_{T2}^{d2}} i_{L2_n} \quad (3.28)$$

$$\begin{aligned} i_{L2_n} &= \overbrace{0.9997}^{A_2^{d1}} i_{L2_{n-1}} + \overbrace{\begin{bmatrix} 0 & 0.0033 \end{bmatrix}}^{A_{2T}^{d1}} \begin{bmatrix} i_{L1_{n-1}} \\ v_{C2_{n-1}} \end{bmatrix} + \\ &\overbrace{\begin{bmatrix} 0 & 0.0033 \end{bmatrix}}^{A_{2T}^{d2}} \begin{bmatrix} i_{L1_n} \\ v_{C2_n} \end{bmatrix} - \overbrace{0.0033}^{B_2^{d1}} V_{S_{n-1}} - \overbrace{0.0033}^{B_2^{d2}} V_{S_n} \end{aligned} \quad (3.29)$$

Equations (3.27) and (3.29) are obtained using TR, an implicit solver, which explains the need for not only inputs from the previous step ( $n-1$ ) but also the current step ( $n$ ). Nevertheless, both equations can be executed in parallel since one of the state variable from (3.28),  $i_{L1}$ , only needs values of the previous step to be computed. Equations from network 1 and  $i_{L1}$  are simulated on one core, and network 2 and  $v_{C2}$  are simulated on a different core. Figure 3.5 a) shows the companion model used to couple  $i_{L1}$  to the inverter, and figure 3.5 b) the one coupling  $v_{C2}$  to  $i_{L2}$ . There is no  $R_{eq}$  in figure 3.5 a) since  $i_{L1}$  only depends on previous values. The value

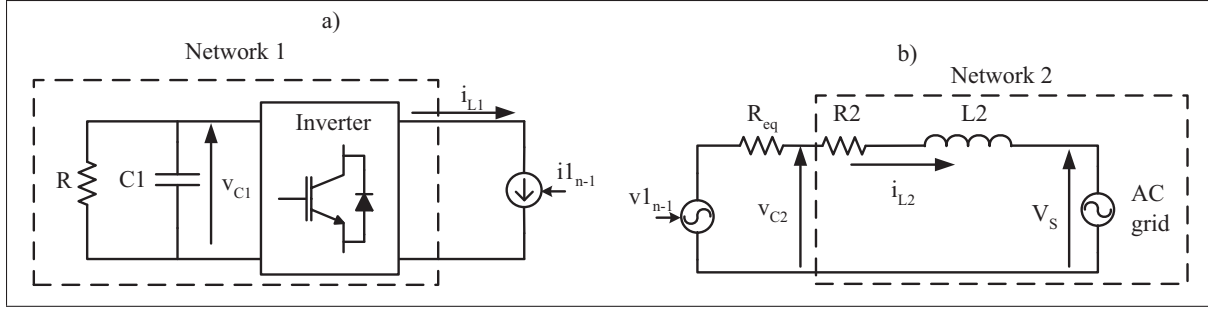


Figure 3.5 a) Companion model coupling  $i_{L1}$  to the inverter  
b) companion model coupling  $v_{C2}$  to  $i_{L2}$

of  $R_{eq}$  in figure 3.5 b) can be found in matrix  $A_{T2}^{d2}$ . Network 1 and network 2 are now fully decoupled, and can be simulated in parallel.

Furthermore, by applying (3.21) to the results obtained from (3.27) to (3.29) state-space matrix of the decoupled system is obtained in (3.30).

$$A^d = (I - A^{d2})^{-1} A^{d1}$$

$$A^d = \begin{bmatrix} 0.9990 & 0.0050 & -0.0000 & 0 \\ -0.0029 & 0.9997 & -0.0029 & 0 \\ 0 & 0.3330 & 0.9978 & -0.6658 \\ 0 & 0.0011 & 0.0067 & 0.9974 \end{bmatrix} \quad (3.30)$$

To demonstrate advantages from this new method, it is compared with two different approaches. For all cases, the objective remains to isolate the inverter from the AC grid. Values of the four state-variable,  $v_{C1}$ ,  $i_{L1}$ ,  $v_{C2}$  and  $i_{L2}$  are used to determine the accuracy of the different methods under test. An open-loop control is used to ensure that errors inherent to the discretization methods are not compensated by a controller. The three following implementations are studied and compared to a reference obtained without any decoupling:



- Case 1: The proposed method;
- Case 2: Replacing L1 by a stubline;
- Case 3: Replacing C2 by a stubline.

Before running simulations, poles of the different methods are obtained and shown in table 3.2. Reference poles are obtained using (3.22) on (3.26). Cases 2 and 3 have an extra pole because of the stubline. As it was mentioned in the introduction, stubline adds parasitic state-variable to the circuit. This parasitic state-variable usually yields a pole near -1 on the unitary circle. Fast oscillations are expected in case 2 and 3 because  $\lambda_5$  is located near -1. Nonetheless, all the poles are within the unity circle ensuring, therefore, global system stability. Although, poles location gives information on system stability, it gives no information on the accuracy of the model. By comparing the poles from the different cases and the ones from the reference informs the user on the accuracy of the system. In this example, all poles are almost identical, and therefore very similar results are expected. Besides the extra pole from case 2 and 3, the only noticeable difference is for  $\lambda_{1,2}$  of case 1. Its real part is slightly smaller than the reference pole, which means that this pole is more damped for case 1; due to the use of BE solver. Based

Table 3.2 Poles for different methods

	$\lambda_{1,2}$	$\lambda_{3,4}$	$\lambda_5$
Ref	$0.9967 \pm i0.0796$	$0.9993 \pm i0.0031$	-
Case 1	$0.9956 \pm i0.0795$	$0.9993 \pm i0.0031$	-
Case 2	$0.9967 \pm i0.0796$	$0.9993 \pm i0.0031$	-0.9997
Case 3	$0.9967 \pm i0.0796$	$0.9993 \pm i0.0031$	-0.9999

on poles location analysis, simulation results for all three cases should be very similar, except for the possibility of fast oscillation in cases 2 and 3. Also steady-state should be reached faster in case 1. In the simulation, phase and amplitude of the modulating signal are adjusted

to achieve nominal power injected to the DC load. For each state variable observed,  $v_{C1}$ ,  $i_{L1}$ ,  $v_{C2}$  and  $i_{L2}$ , the relative error between the reference simulation and each case is calculated. Figure 3.6 a) and figure 3.7 a) show results for  $v_{C1}$  and  $i_{L2}$ ; curves for the different cases are superimposed. Their relative error for all three cases is less than 1%, as shown in figure 3.6 b) and figure 3.7 b). This can be explained by the fact that the same solver, TR, was used for all three cases.

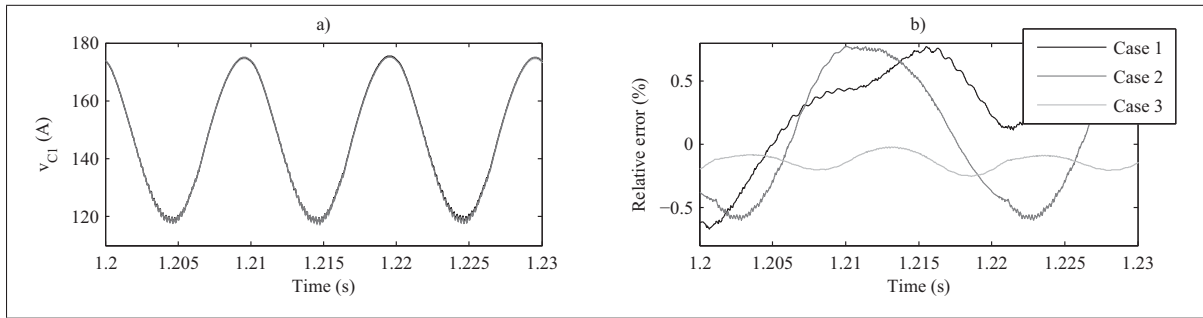


Figure 3.6 a) Current  $v_{C1}$  for the reference and the three methods  
b) relative errors of  $v_{C1}$  for cases 1 to 3

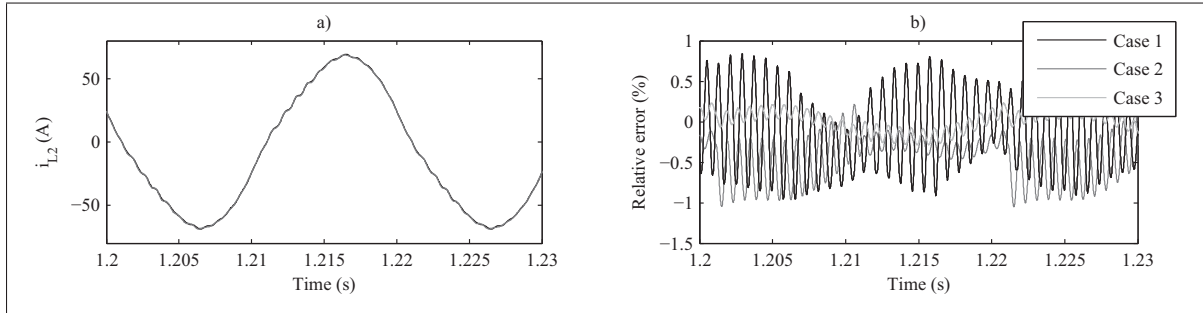


Figure 3.7 a) Current  $i_{L2}$  for the reference and the three methods  
b) relative errors of  $i_{L2}$  for cases 1 to 3

Figure 3.8 a) shows  $i_{L1}$  for the different method and figure 3.8 b) the relative error for each method. The maximum relative error is obtained with the method from case 2 with a peak value of 3%, as for case 1 and 3 a peak value of 1% and 0.1% are respectively obtained. Furthermore, results from case 2 are highly oscillating, which is due to the stubline. Numerical oscillation

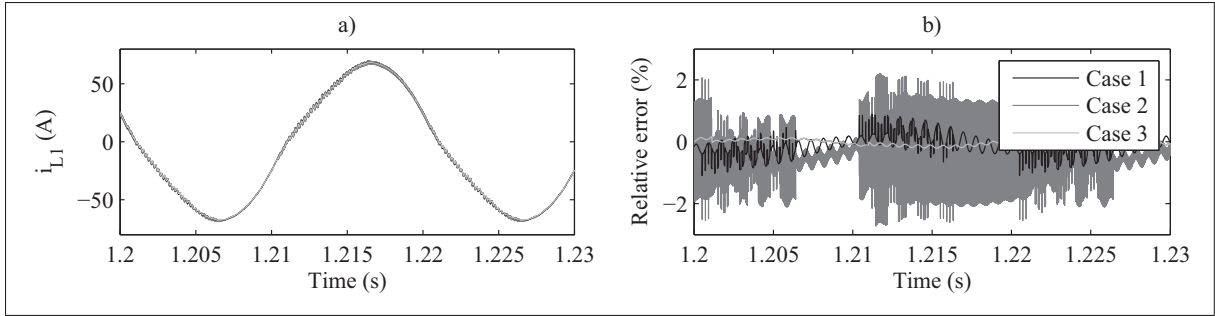


Figure 3.8 a) Current  $i_{L1}$  for the reference and the three methods  
b) relative errors of  $i_{L1}$  for cases 1 to 3

observed for case 2 on  $i_{L1}$  can now be observed for case 3 on  $v_{C2}$  in figure 3.9. Because of

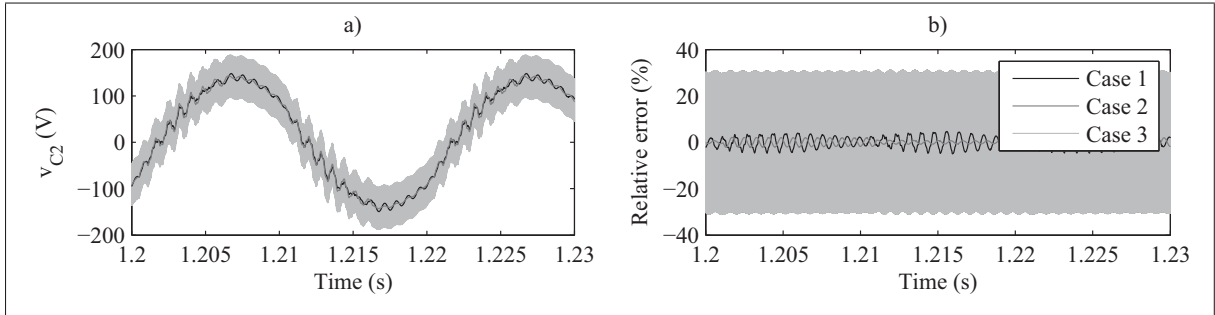


Figure 3.9 a) Voltage  $v_{C2}$  for the reference and the three methods  
b) relative errors of  $v_{C2}$  for cases 1 to 3

those numerical oscillations, the relative error now reaches 35% for case 3, when peak relative errors for case 1 and 2 are only 5% and 2%. From the three implementation methods used, one can observe that the proposed method offers the smallest relative errors.

Now simulation results during transient will be studied by applying a short-circuit at  $C_2$  which last half a cycle. Figure 3.10 a) shows  $i_{L1}$  for all three cases and figure 3.10 b) shows their relative error to the reference. When looking at the relative error, the relative error, the latter seems much larger for the proposed method; because the BE solver was used. Since the system is more damped, steady state is reached faster for the proposed method. This can easily be observed on  $v_{C2}$  in figure 3.11 a). After the fault,  $v_{C2}$  of case 1 returns to a steady state in one cycle. After a few cycles, steady state is reached by the other methods and relative errors return

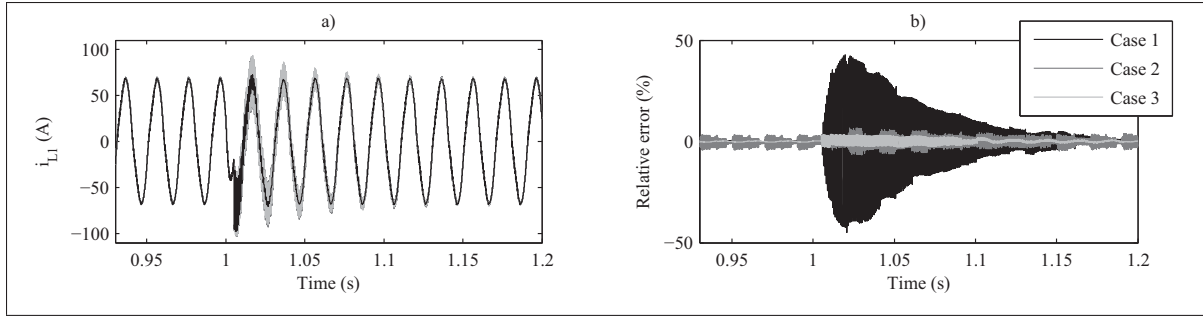


Figure 3.10 a) Current  $i_{L1}$  during fault for the reference and the three methods  
b) relative errors of  $i_{L1}$  during fault for cases 1 to 3

to their value, as shown in figure 3.11 b). This example has demonstrated the accuracy of the

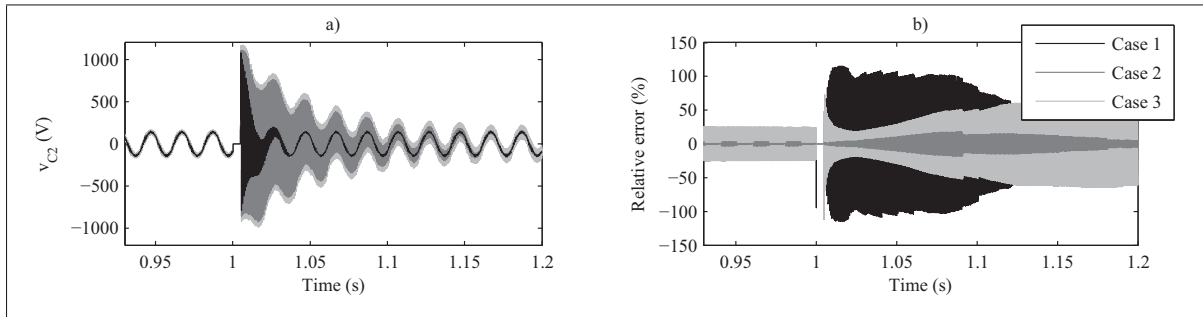


Figure 3.11 a) Voltage  $v_{C2}$  during fault for the reference and the three methods  
b) relative errors of  $v_{C2}$  during fault for cases 1 to 3

proposed method even for a non-linear system. Unlike the stubline, no parasitic states are added and, therefore, the proposed method does not suffer from numerical oscillation. Stability and accuracy of the proposed method have been demonstrated through simulation results and poles location analysis. Studying the poles of the system allows identifying every possible pole, even those who might not be excited in every simulation scenarios. Furthermore, knowing the exact reference pole location, using (3.22), accuracy of the different methods can be compared. Also, one must keep in mind that when the main goal of decoupling the circuit is to achieve real-time simulation capability, a relative error of 5% is considered well within acceptable range for such application (Blanchette *et al.*, 2012; Grégoire *et al.*, 2014a).

### 3.5 Conclusion

In this paper, a decoupling technique suitable for circuit simulation containing power electronic switches was presented. The proposed technique is proven most useful in real-time simulation application, where faster-parallel computing with high precision and stable simulation is required. It can also be applied to commercially available software. The impact of the decoupling technique can be evaluated through poles analysis even though part of the circuit is discretized by third party software. Although the choice of solvers and the location of decoupling still needs human intervention, using the clear rules given on how to apply the decoupling technique, automatization of the method can be easily achieved removing any human interaction. This method could also be used in hardware-in-the-loop (HIL) and power-HIL (PHIL) simulation where decoupling is needed between the simulated model and the physical hardware. Future work will aim at applying this method for multi-rate systems, where the integration time-step may vary between different states which constitute great challenges in nowadays real-time simulations of complex systems.

### 3.6 Acknowledgement

The authors wish to thank FQRNT, CRSNG, OPAL-RT technologies and Canada Research Chairs (CRC) for supporting this work.



## CHAPTER 4

### GENERALIZED VALIDATION METHOD FOR MULTI-RATE DIGITAL SIMULATION

Luc-André Grégoire<sup>1</sup>, Handy Fortin Blanchette<sup>1</sup>, Jean Bélanger<sup>2</sup>, Kamal Al-Haddad<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering , École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

<sup>2</sup> CEO and CTO, OPAL-RT technologies Inc.,  
1751 Richardson, Suite 2525, Montréal, Québec, Canada H3K 1G6

Article submitted to « IEEE Transactions on Industrial Informatics » in January 2016.

#### **abstract**

This paper presents a new real time simulation method to demonstrate the stability and accuracy of a discretized power circuits containing power electronics devices with multiple sampling rates. In single-rate simulation, the discretized system is stable when its discrete poles are within the unitary circle. When using multi-rate solvers, one or many state-variables are discretized with different sampling rates; therefore the system's equations cannot be unified in a single state-space matrix where pole analysis is applied. Therefore, a formal mathematical analysis is introduced to demonstrate the stability of multi-rate real time simulation is proposed. Every state variables of a system, regardless of their discretization time step are found in a single matrix. Classical pole analyses are thereafter used to test stability. The method is generalized, and can be applied to any multi-rate simulation circuits. Finally, the proposed method is demonstrated and supported with a numerical example based on a micro grid device using static compensator. The proposed method was found accurate and reliable.

#### 4.1 Introduction

Over the last decade, simulation has become an essential development tool for engineers and researchers. More so, real-time simulation can now replace expensive downscale prototype that had to be build for every new test configuration. Driven by this keen interest, real-time simulator technology has become more powerful, and their simulation time-step have been reduced drastically ensuring higher accuracy. Dealing with complex systems containing very large and very small time constants, the use of multi-rate (MR) simulation solvers have been proposed (Benigni *et al.*, 2014a; Benigni and Monti, 2014a; Matar *et al.*, 2004; Inaba *et al.*, 2011). MR solvers can reduce computation burden of stiff systems, by using the most appropriate step-size; usually small for fast system dynamics whereas large time-step is chosen for slow dynamics. When it comes to simulation of power electronic converters, one can also choose a very small time-step to achieve higher accuracy on gate signals of power electronic switches. During such implementation, MR solvers can be used for simulation over different platforms, like a combination of microprocessor (CPU) and Field Programmable Gate Array (FPGA)(Saad *et al.*, 2015a; Grégoire *et al.*, 2014b). In this case, slower dynamics are simulated on CPU with a larger time-step, from 10 to 50 microseconds ( $\mu s$ ); Whereas faster dynamics and gating signals are done on FPGA with time-step between 100 to 500 nanoseconds (mn). Although obtained results may be convincing (Saad *et al.*, 2015a; Grégoire *et al.*, 2014b; Belanger *et al.*, 2013; Li *et al.*, 2014a; Inaba *et al.*, 2011), they often lack of numerical demonstrations, especially when it comes to numerical stability. Without such proof, various simulations scenario must be run to validate the implementation. Those numerous tests are meant to ensure system stability for all operating conditions, and that every single poles, which are unknown, can be excited without resulting into numerical instability.

MR solver and demonstration of their stability can be found in the literature. For example, to some extend, Runge-Kutta (Rice, 1960) can be consider a MR solver. Sub-step, inherent to Runge-Kutta, allows to solve parts of the circuit with smaller step-size. Furthermore, stability and local truncation error of those technique are well known (Butcher, 1987). These methods indeed increase accuracy of faster states, but would not take into consideration fast switching



event on power electronic devices. Since inputs are only updated at large time-step, fast inputs, such as gating signals of power switches, remain constant over one large time-step in Runge-Kutta solver, which does not increase accuracy. In (Pekarek *et al.*, 2004), authors proposed a MR solver where fast inputs can be sampled with accuracy. Long term stability from (Pekarek *et al.*, 2004) was demonstrated using the method proposed in (Gautschi, 1997), but no explicit information on simulation accuracy was obtained. Therefore, authors used a variable-step solver where accuracy is verified at the end of each time-step, and sampling time is modified when required. Such implementation is naturally not suitable for real-time simulation as the execution time is a non-deterministic one. In (McLaren *et al.*, 1992; Marti and Linares, 1994; Moreira *et al.*, 2006), authors raise the same concerns in regards to the uselessness of iterative solver for real-time simulation. A good contribution on the subject has been made in (Moreira *et al.*, 2006), where extensive literature review of MR simulation is presented. Detailed implementation for MR technique was proposed, accuracy of the method is shown through simulation results, but once again it lacks numerical demonstration. Demonstrating stability based on simulation results raise two problems. The first one deals with generating reference simulations of the circuit validating the model; those references are often obtained using either a very small time-step or various-step solver, which in both cases is very time consuming. Secondly, despite having reference simulations made, it is nearly impossible to ensure that for some cases, not considered by the reference model, accuracy and stability is preserved. In this paper a generic method to test stability based on poles location of MR system is proposed. State-variable equations, discretized with various sampling time, are linearized around the smallest sampling rate. The different state-variables are then unified in a single state-space matrix system. Eigenvalues of the MR system are identified and are compared to the one of a continuous system, giving information on both numerical stability and accuracy achieved. Moreover, by comparing poles location of the MR system to the continuous one, the most appropriate time-step for each state-variables can be found without any simulation. Presented in its general form, the proposed method can be applied to any MR implementation, and its effectiveness is demonstrated through numerical example.

This paper is organised as follows. Section 4.2 defines MR system and proposes a linearization method. Once linearized, poles location analysis is applied. Numerical implementation of the method is given in section 4.3. Accuracy for different MR discretization methods are studied, from which the most appropriate sampling time have been chosen, in section 4.3. Simulation results confirming optimal time-step selection are shown in section 4.3.1, followed by a conclusion in section 4.4.

## 4.2 Mutli-rate simulation

In this paper, the term multi-rate (MR) simulation refers to using different time-steps/sampling times to compute different state-variables within a system. Time-steps chosen in MR solvers remain constant over the whole simulation duration. Variable-step solvers, unlike MR solves, only use one time-step for every state-variables, and the latter can vary during the simulation, MR solvers are proven most useful for stiff systems, which are characterized by very large and very small time constants; each state-variable uses the most appropriate time-step to avoid unnecessary computation time. Another good candidate for MR simulations are power electronic circuits found in larger power system applications. Smaller time-steps are used to ensure accuracy on gating signals as well as snubber circuits of the power switches. Using the same sampling time for slow varying states leads to unnecessary large computation time. It is therefore common practice to use large time-steps for the slow-varying states, and small time-steps for fast one. The state-variables, who have the same sampling time, are regrouped in sub-system according to their dynamics. The slow dynamic sub-systems would have a time-step of  $\Delta T$  while the fast one have a time-step of  $\Delta t$ . It should be noted that  $\Delta T$  does not have to be an integer of  $\Delta t$ . Also, there is no mathematical limitation to the number of different sampling time to use, other than the complexity which is inherent to its implementation.

In MR simulation, from the point of view of the fast dynamic sub-system, state-variables are updated at every computational steps, while state-variables from the slow dynamic sub-system remain constant for a duration equal to the large time-step (Inaba *et al.*, 2011; Li *et al.*, 2014a). Figure 4.1 shows a function integrated with two different sampling rates, where  $\Delta T$  is equal

to  $2.5\Delta t$ . From the fastest sub-system point of view, it appears that the slow variable has discontinuity; it changes only once every two  $\Delta t$  steps. If such a behaviour can be obtained

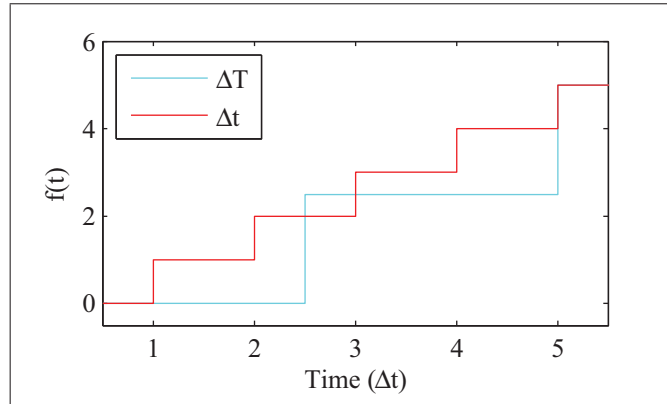


Figure 4.1 Signal integrated with two different sampling rates

with a state-space matrix representation, then classic poles analysis could be used.

#### 4.2.1 Proposed state-space matrix representation

The proposed state-space matrix representation is used to combine two, or more, sampling time from one system in a unified matrix representation. In order to achieve such behaviour, a nonlinear system, where state-variables may remain constant is required. First, let's define a system where every state-variables are updated as shown in equation (4.1).

$$\begin{bmatrix} X_{S_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} A_S & A_{SF} \\ A_{FS} & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} B_S \\ B_F \end{bmatrix} U_n \quad (4.1)$$

In (4.1), the state vector is divided in two vectors,  $X_S$  and  $X_F$ , containing respectively the slow and the fast dynamic state-variables. The A matrix is divided in sub-matrices  $A_S$ ,  $A_F$ ,  $A_{SF}$  and  $A_{FS}$ .  $A_S$  and  $A_F$  only contain values from diagonal of the slow and fast system, while  $A_{SF}$  and  $A_{FS}$  contain values coupling the different state-variables.

In the case where the slow dynamic state-variables  $X_S$  remain constant, the system is defined by (4.2)

$$\begin{bmatrix} X_{S_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ A_{FS} & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} 0 \\ B_F \end{bmatrix} U_n \quad (4.2)$$

Using (4.2), states from vector  $X_F$  are evolving, while the one from  $X_S$  are equal to their previous values. During simulation, (4.2) is used for every small time-step of  $\Delta t$ , and it is replaced by (4.1) once every  $\Delta T$ . By combining (4.1) and (4.2), the discrete nonlinear system (4.3) is obtained. In (4.3), the variable  $\rho$  is introduced so that (4.1) or (4.2) can be obtained. When  $\rho$  equals 1, the matrices are equal the one of (4.1), and both  $X_{S_n}$  and  $X_{F_n}$  are updated. When  $\rho$  is equal to 0, the matrices are equal to (4.2); therefore,  $X_{S_n}$  remains constant and only  $X_{F_n}$  is updated.

$$\begin{bmatrix} X_{S_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} \rho A_S + (I - \rho) & \rho A_{SF} \\ A_{FS} & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} \rho B_S \\ B_F \end{bmatrix} U_n \quad (4.3)$$

Equation (4.3) can be qualified as nonlinear since its outputs are not only proportional to its inputs but also to the time-varying variable  $\rho$ . Methods to verify the stability and poles location of circuit containing nonlinearity have been used for a long time. In the case of power converter, a small signal equivalent circuit (Erickson and Maksimovic, 2001) can be obtained to identify the dynamic of the system around its operating points. This method requires the continuous system and cannot be applied to (4.3) as it is a discrete system. In (Shortt and Lee, 1983), authors have linearized discrete system with power switches for a specific duty cycle, or operating point. This method can be applied to (4.3) where  $\rho$  is used in order to linearize the system around one operating point. In (4.3),  $\rho$  will have a value of 1 during  $\Delta t$  and is equal to 0 otherwise, for a period of  $\Delta T$ . The ratio between  $\Delta t$  and  $\Delta T$  remains constant for the duration of the simulation. The operating point used to linearize  $\rho$  is given by (4.4).

$$\bar{\rho} = \frac{\Delta t}{\Delta T} \quad (4.4)$$

Without loss of generality, transformation matrix  $M_\rho$  of (4.5) can be used to linearize a multi-rate matrix (MR) to a single-rate (SR) one, as shown in (4.6).

$$M_\rho = \begin{bmatrix} \bar{\rho} & 0 \\ 0 & I \end{bmatrix} \quad (4.5)$$

$$\begin{aligned} A_{\text{Single-Rate}} &= M_\rho \overbrace{\begin{bmatrix} A_S & A_{SF} \\ A_{FS} & A_F \end{bmatrix}}^{A_{\text{Multi-Rate}}} + I - M_\rho \\ B_{\text{Single-Rate}} &= M_\rho \overbrace{\begin{bmatrix} B_S \\ B_F \end{bmatrix}}^{B_{\text{Multi-Rate}}} \end{aligned} \quad (4.6)$$

With the matrix  $A_{\text{Single-Rate}}$  now linearized, as if only  $\Delta t$  was used, classic poles location analysis can be performed.

In (4.3), state-variables are used directly between fast and slow sub-system. In some MR algorithm, average and extrapolated values of the different state-variables are used (Benigni *et al.*, 2014a); in such a case, state-variables used between fast and slow systems need to be altered. For example, the average value of a fast state to be used in a slower one. The average of  $X_F$  over  $\Delta T$  can be calculated using (4.7).

$$\tilde{X}_F(t + \Delta T) = \overbrace{\frac{\Delta t}{\Delta T}}^{\bar{\rho}} \sum_{i=0, \Delta t, 2\Delta t, \dots}^{\Delta T} X_F(t + i) \quad (4.7)$$

$\tilde{X}_F$  is then added to the state-space of the system. As it was previously explained, two distinct equations are therefore required, whether only state-variables of  $X_F$ , or the one of both  $X_F$

and  $X_S$  are being updated. Equation (4.8) is used when all state-variables of the system are evolving, and (4.9) is used when only the fast one are being updated.

$$\begin{bmatrix} X_{S_n} \\ \tilde{X}_{F_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} A_S & A_{SF} & 0 \\ \bar{\rho}A_{FS} & 0 & \bar{\rho}A_F \\ A_{FS} & 0 & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ \tilde{X}_{F_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} B_S \\ \bar{\rho}B_F \\ B_F \end{bmatrix} U_n \quad (4.8)$$

$$\begin{bmatrix} X_{S_n} \\ \tilde{X}_{F_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \bar{\rho}A_{FS} & 1 & \bar{\rho}A_F \\ A_{FS} & 0 & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ \tilde{X}_{F_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{\rho}B_F \\ B_F \end{bmatrix} U_n \quad (4.9)$$

Combining (4.8) and (4.9) yields equation (4.10). This equation can be linearized and used for poles location analysis while taking into account the averaging of the fastest state-variables.

$$\begin{bmatrix} X_{S_n} \\ \tilde{X}_{F_n} \\ X_{F_n} \end{bmatrix} = \begin{bmatrix} \bar{\rho}A_S + (I - \bar{\rho}) & \bar{\rho}A_{SF} & 0 \\ \bar{\rho}A_{FS} & I - \bar{\rho} & \bar{\rho}A_F \\ A_{FS} & 0 & A_F \end{bmatrix} \begin{bmatrix} X_{S_{n-1}} \\ \tilde{X}_{F_{n-1}} \\ X_{F_{n-1}} \end{bmatrix} + \begin{bmatrix} \bar{\rho}B_S \\ \bar{\rho}B_F \\ B_F \end{bmatrix} U_n \quad (4.10)$$

This demonstrates how fast state-variables average value is computed and communicated to the slower sub-system. Similar schemes can be implemented for different communication approach, or MR algorithm. Consequently, once linearized with  $\bar{\rho}$ , system stability can be verified as well as the dynamic of the multi-rate system, including the coupling method between the slow and the fast sub-system. A numerical example of the proposed method is presented in the following section.

### 4.3 Numerical example

In this section, the study of a three-buses microgrid power system, with a STATCOM connected through an LCL filter, is proposed. Schematic of the circuit is shown in figure 4.2, and param-

eters used for simulation are given in table 4.1. Voltage source  $V_A$  and  $V_B$  and their impedances represent simplified equivalent networks. Simplified equivalent networks were chosen to keep the equations concise. Nominal power is transferred from  $V_A$  to  $V_B$ , and the STATCOM reference is set to maintain nominal voltage at point of common coupling (PCC). Such a network is well suited for MR application; large time-step is used for transmission networks, since it has a slower dynamic, and small time-step is used for the STATCOM and its filter, where dynamic is much faster.

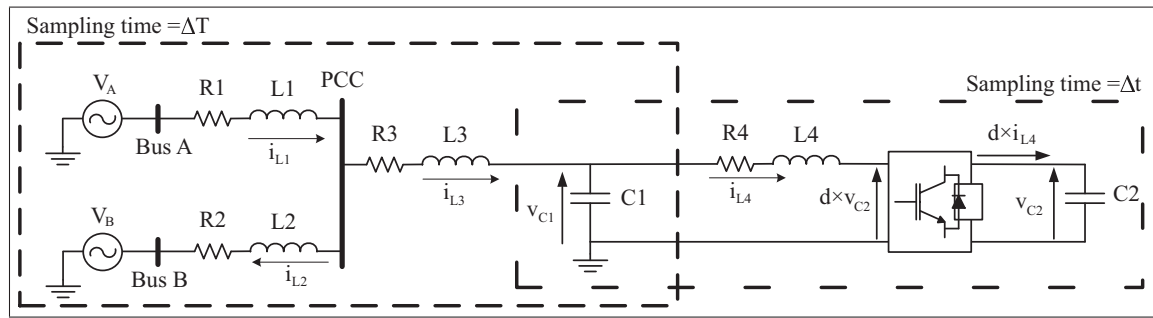


Figure 4.2 Single phase 3-buses system with STATCOM

Table 4.1 Simulation parameters

Nominal power	100	kVA
Nominal voltage	1	kV
L1 & L2	2.7	mH
R1 & R2	0.1	$\Omega$
L3 & L4	6.6	mH
R3 & R4	0.125	$\Omega$
C1	6	$\mu$ F
C2	130	mF
PWM	5	kHz
$\Delta T$	50	$\mu$ s
$\Delta t$	600	ns

The continuous state-space equations of the microgrid are given in (4.11), where the variable  $d$  represents the duty cycle applied to the STATCOM. The STATCOM is made of four power switches, and can therefore achieves three levels;  $d$  can then be equal to either 1, 0, or -1.

From the system of figure 4.2, five state-variables are identified,  $i_{L2}$ ,  $i_{L3}$ ,  $i_{L4}$ ,  $v_{C1}$ , and  $v_{C2}$ ; its state-space equation is given by (4.11). Using those state-variables, two different cases are studied and compared for the circuit of figure 4.2. Case 1,  $i_{L2}$ ,  $i_{L3}$ , and  $v_{C1}$  are discretized using a large time-step ( $\Delta T$ ), and  $i_{L4}$ ,  $v_{C2}$ , and the power inverter use a small time-step ( $\Delta t$ ). In the second case,  $i_{L2}$  and  $i_{L3}$  are discretized using a large time-step ( $\Delta T$ ), and  $i_{L4}$ ,  $v_{C1}$ ,  $v_{C2}$ , and the power inverter use a small time-step ( $\Delta t$ ). A pole placement analysis is done before running the simulation to verify the stability and the accuracy of the MR models. The discrete MR systems are obtained using the method proposed in (Grégoire *et al.*, 2015b). Each state-variable is discretized with the desired time-step using operational substitution. The resulting state-space matrices for case 1 and case 2 are given in (4.12) and (4.13) respectively.

$$\begin{bmatrix} i_{L3s} \\ i_{L2s} \\ i_{L4s} \\ v_{C1s} \\ v_{C2s} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 \\ a_{21} & a_{22} & 0 & a_{24} & 0 \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ a_{41} & 0 & a_{43} & 0 & 0 \\ 0 & 0 & a_{53} & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{L3} \\ i_{L2} \\ i_{L4} \\ v_{C1} \\ v_{C2} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_A \\ V_B \end{bmatrix} \quad (4.11)$$

$$\begin{aligned}
 a_{11} &= -\frac{L2R1+L1R3+L2R3}{L1L2+L1L3+L2L3} & a_{12} &= \frac{L1R2-L2R1}{L1L2+L1L3+L2L3} & a_{14} &= -\frac{L1R3+L2}{L1L2+L1L3+L2L3} \\
 a_{21} &= \frac{L1R3-L3R1}{L1L2+L1L3+L2L3} & a_{22} &= -\frac{L1R2+L3R1+L3R2}{L1L2+L1L3+L2L3} & a_{24} &= \frac{L1}{L1L2+L1L3+L2L3} \\
 a_{33} &= \frac{-R4}{L4} & a_{34} &= \frac{1}{L4} & a_{35} &= d\frac{-R4}{L4} & a_{41} &= \frac{1}{C1} \\
 a_{43} &= \frac{-1}{C1} & a_{53} &= \frac{d}{C2} \\
 b_{11} &= \frac{L2}{L1L2+L1L3+L2L3} & b_{12} &= \frac{L1}{L1L2+L1L3+L2L3} \\
 b_{21} &= \frac{L3}{L1L2+L1L3+L2L3} & b_{22} &= -\frac{L1+L3}{L1L2+L1L3+L2L3}
 \end{aligned}$$



$$\begin{bmatrix} i_{L3_N} \\ i_{L2_N} \\ i_{L4_n} \\ v_{C1_N} \\ v_{C2_n} \end{bmatrix} = \overbrace{\begin{bmatrix} 0.9757 & 0 & 0.0206 & -0.0124 & 0 \\ 0.0112 & 0.9981 & -0.0103 & 0.0062 & 0 \\ 0.0003 & 0 & 0.9996 & 0.0002 & -0.0002 \\ 3.2749 & 0 & -3.2810 & 0.9791 & 0.0003 \\ 0 & 0 & 0.0038 & 0 & 1 \end{bmatrix}}^{A_{\text{Multi-Rate}}^{\text{Case 1}}} \begin{bmatrix} i_{L3_{N-1}} \\ i_{L2_{N-1}} \\ i_{L4_{n-1}} \\ v_{C1_{N-1}} \\ v_{C2_{n-1}} \end{bmatrix} + \overbrace{\begin{bmatrix} 0.0031 & 0.0031 \\ 0.0032 & -0.0062 \\ 0 & 0 \\ 0.0051 & 0.0051 \\ 0 & 0 \end{bmatrix}}^{B_{\text{Multi-Rate}}^{\text{Case 1}}} \begin{bmatrix} V_{A_N} \\ V_{B_N} \end{bmatrix} \quad (4.12)$$

$$\begin{bmatrix} i_{L3_N} \\ i_{L2_N} \\ i_{L4_n} \\ v_{C1_n} \\ v_{C2_n} \end{bmatrix} = \overbrace{\begin{bmatrix} 0.9960 & 0 & 0.0002 & -0.0125 & 0 \\ 0.0010 & 0.9981 & -0.0001 & 0.0063 & 0 \\ 0 & 0 & 0.9999 & 0.0002 & -0.0002 \\ 0.0331 & 0 & -0.0332 & 0.9998 & 0 \\ 0 & 0 & 0.0038 & 0 & 1 \end{bmatrix}}^{A_{\text{Multi-Rate}}^{\text{Case 2}}} \begin{bmatrix} i_{L3_{N-1}} \\ i_{L2_{N-1}} \\ i_{L4_{n-1}} \\ v_{C1_{n-1}} \\ v_{C2_{n-1}} \end{bmatrix} + \overbrace{\begin{bmatrix} 0.0031 & 0.0031 \\ 0.0031 & -0.0063 \\ 0 & 0 \\ 0.0001 & 0.0001 \\ 0 & 0 \end{bmatrix}}^{B_{\text{Multi-Rate}}^{\text{Case 2}}} \begin{bmatrix} V_{A_N} \\ V_{B_N} \end{bmatrix} \quad (4.13)$$

Although they are given in the form of state-space equations, this is erroneous since more than one time-step is used; this will be corrected when the system is linearized. Since the

simulation contain different sampling rate, let's denoted by a  $n$  for variables updated every  $\Delta t$  and  $N$  for the ones updated every  $\Delta T$ . If eigenvalues are found for  $A_{\text{Multi-Rate}}^{\text{Case 1}}$  and  $A_{\text{Multi-Rate}}^{\text{Case 2}}$ , their values would not exactly give the real behavior of the system. Therefore, they need to be linearized into single-rate (SR) matrices using (4.5) and (4.6), yielding to (4.14) and (4.15).

$$A_{\text{Single-Rate}}^{\text{Case 1}} = M^{\text{Case 1}} A_{\text{Multi-Rate}}^{\text{Case 1}} + I - M^{\text{Case 1}} =$$

$$\begin{bmatrix} 0.9998 & 0 & 0.0002 & -0.0001 & 0 \\ 0.0001 & 1.0000 & -0.0001 & 0.0001 & 0 \\ 0.0003 & 0 & 0.9999 & 0.0002 & -0.0002 \\ 0.0331 & 0 & -0.0332 & 0.9998 & 0 \\ 0 & 0 & 0.0038 & 0 & 1 \end{bmatrix} \quad (4.14)$$

$$A_{\text{Single-Rate}}^{\text{Case 2}} = \begin{bmatrix} 1 & 0 & 0.0001 & -0.0001 & 0 \\ 0.0001 & 1.0000 & -0.0001 & 0.0001 & 0 \\ 0.0001 & 0 & 0.9999 & 0.0002 & -0.0002 \\ 0.0331 & 0 & -0.0332 & 0.9998 & 0 \\ 0 & 0 & 0.0038 & 0 & 1 \end{bmatrix} \quad (4.15)$$

Eigenvalues from the two cases can now be obtained and compared to the reference ones. Discrete eigenvalues of the reference model are obtained using (4.16) (Hartley *et al.*, 1994) and the eigenvalues from the continuous system in (4.11).

$$\lambda_z = e^{\lambda_s T} \quad (4.16)$$

Table 4.2 shows poles obtained for the different matrices. Note that matrices  $A_{\text{Multi-Rate}}^{\text{Case 1}}$  and  $A_{\text{Multi-Rate}}^{\text{Case 2}}$  are included to demonstrate the error yields by such matrices. Relative errors are computed for each poles when compared to the reference poles.

Looking at table 4.2,  $A_{\text{Single-Rate}}^{\text{Case 2}}$  offers the smallest relative errors after applying the proposed method, and should therefore be the most accurate.  $A_{\text{Multi-Rate}}^{\text{Case 1}}$  and  $A_{\text{Multi-Rate}}^{\text{Case 2}}$  have the largest

Table 4.2 Poles for different methods

	Ref	$A_{\text{Multi-Rate}}^{\text{Case 1}}$	$A_{\text{Multi-Rate}}^{\text{Case 2}}$	$A_{\text{Single-Rate}}^{\text{Case 1}}$	$A_{\text{Single-Rate}}^{\text{Case 2}}$
$\lambda_1$	0.9999	0.9981	0.9981	0.9999	0.9999
$\ \text{Rel. err.}\ $ (%)	-	0.18	0.18	0	0
$\lambda_{2,3}$	0.9999 $\pm i0.0032$	0.9772 $\pm i0.2031$	0.9979 $\pm i0.0204$	0.9999 $\pm i0.0032$	0.9999 $\pm i0.0032$
$\ \text{Rel. err.}\ $ (%)	-	20.11	1.72	0.0354	0.0099
$\lambda_{4,5}$	0.9999 $\pm i0.0005$	0.9999 $\pm i0.0008$	0.9999 $\pm i0.0008$	0.9999 $\pm i0.0005$	0.9999 $\pm i0.0005$
$\ \text{Rel. err.}\ $ (%)	-	0.0315	0.0315	0.0004	0.0004

relative errors, since only one sampling rate is considered to find the eigenvalues and not the MR nature of the system.  $A_{\text{Single-Rate}}^{\text{Case 1}}$  and  $A_{\text{Single-Rate}}^{\text{Case 2}}$  relative errors are rather small, and therefore both cases yield to good results. This is now demonstrated through simulation for the reference, case 1, and case 2.

#### 4.3.1 Multi-rate simulation results

Simulation results are obtained using OPAL-RT technologies real-time simulator. The slow subsystem is simulated on CPU with a time-step of  $50 \mu\text{s}$ , and the fast subsystem uses a time-step of 600 ns on FPGA.  $\Delta t$  was consciously chosen not be an integer of  $\Delta T$  to demonstrate the flexibility of the proposed method. This is possible since simulation on the FPGA is done at 600 ns, the FPGA internal clock runs at 5 ns allowing synchronization with the CPU running at  $50 \mu\text{s}$ . Figure 4.3 a) shows the current in L1 for the reference, case 1 and case 2. All three methods give very similar results which are superimposed. In figure 4.3 b), relative errors are presented.

Case 2 is the most accurate with a relative error smaller than 2%, whereas case 1 yields to relative error reaching up to 4%. For this state-variable, case 1 and 2 are using an integration time-step of  $50 \mu\text{s}$ , and the reference uses a sampling time of 600 ns. When calculating relative error, high frequency oscillation can be observed since the reference model vary much faster than the MR model, as shown in figure 4.4.

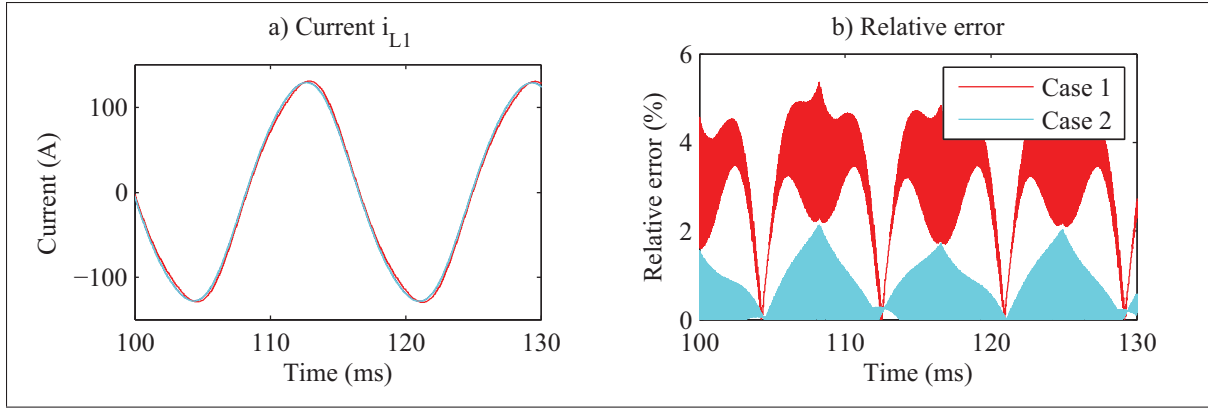


Figure 4.3 a) Simulation results for  $i_{L1}$  b) relative errors for Case 1 and 2

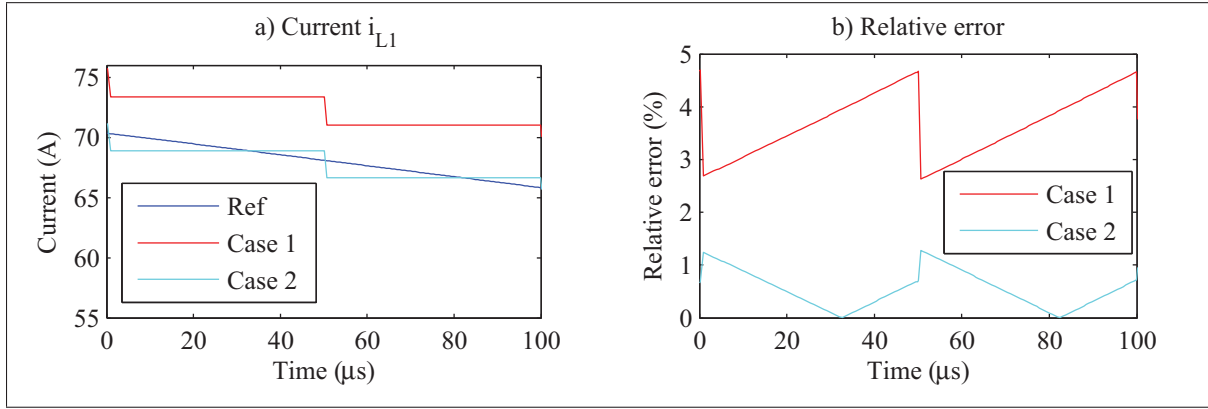


Figure 4.4 Zoom on  $i_{L1}$  for the reference and the MR model

Simulation results of  $v_{C1}$  are shown in figure 4.5. Relative error now reach 3% and around 0.2% for case 1 and 2 respectively. For both cases,  $v_{C1}$  does the coupling between the fast and the slow subsystems. For case 1, MR coupling is done between a slow varying state-variable,  $v_{C1}$  and the fast varying current in L4. For case 2, MR coupling is done between a slow varying state-variable,  $v_{C1}$  and the slow varying current in L3. The impact of such coupling can clearly be observed on the current in L4 of figure 4.6.

Simulation results for  $i_{L4}$  and their relative errors are shown in figure 4.6 a) and figure 4.6 b). For this state-variable, a sampling rate of 600 ns is used for case 1, case 2, and the reference. Case 1 relative error reaches 10%, and the one for case 2 is only 0.5%, as shown in figure 4.6 b). The rather large relative error for case 1 can be explained by the slow sampling period of

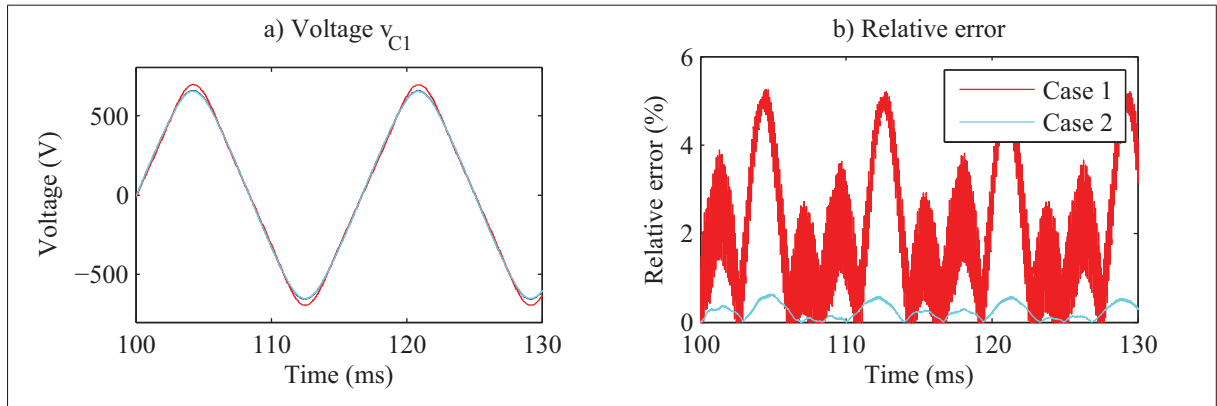


Figure 4.5 a) Simulation results for  $v_{C1}$  b) relative errors for Case 1 and 2

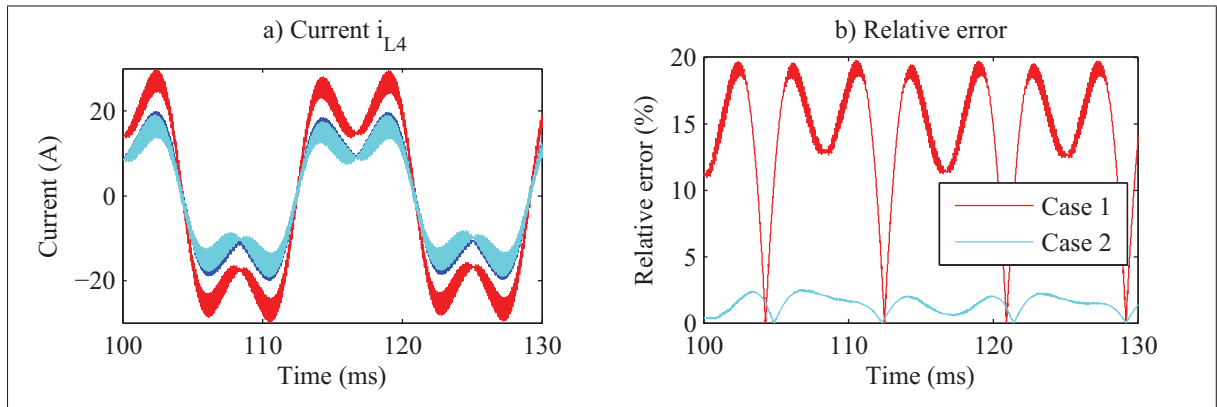


Figure 4.6 a) Simulation results for  $i_{L4}$  b) relative errors for Case 1 and 2

the state-variable  $C1$ . Although  $i_{L4}$  is computed with a time-step of 600 ns for all models, since the capacitor  $C1$  for case 1 uses a time-step of 50  $\mu$ s, this affects accuracy of  $i_{L4}$  for case 1. Finally, when the main goal is to achieve real-time simulation capability, a relative error of 5% is considered within acceptable range for such application (Blanchette *et al.*, 2012; Grégoire *et al.*, 2014a).

Simulation results confirmed that case 2 gives the most accurate results, as it had been predicted from the poles location analysis. Using the proposed method, poles location analysis can be done, and be used as guideline in choosing the most appropriate time-step for the different state-variables of a system. Accuracy and stability of the MR system can be obtained prior to

any simulation, as shown in table 4.2. Obtaining these results without any simulation is very interesting for real-time simulation application as it requires dedicated hardware.

#### **4.4 Conclusion**

In this paper, a generic method to test stability and accuracy of multi-rate solvers has been proposed. The method is very flexible and can be extended to various implementations. Although, only two sampling rates were used, the method can be extended to any number of time-step, even for time-steps that don't have common integer. The proposed method overcomes the drawback of traditional methods when it comes to poles location analysis. Using the proposed method, system stability and accuracy can be tested through poles location analysis without running any simulations. Finally, a better choice of appropriate sampling rate can be done without the use of trial and error time consuming method.

#### **4.5 Acknowledgement**

The authors wish to thank FQRNT, CRSNG, OPAL-RT technologies and Canada Research Chairs (CRC) for supporting this work.

## CHAPTER 5

### CONVERTER MODELING FOR MULTI-RATE/REAL-TIME

In this chapter, detail implementation of a MMC is given for real-time simulation application. Using the simulation techniques and stability analysis as proposed in the previous chapters. Circuit shown in figure 5.1 is then divided in three sections; the power system with transmission lines, the power converter, and the sub-modules of the converter.

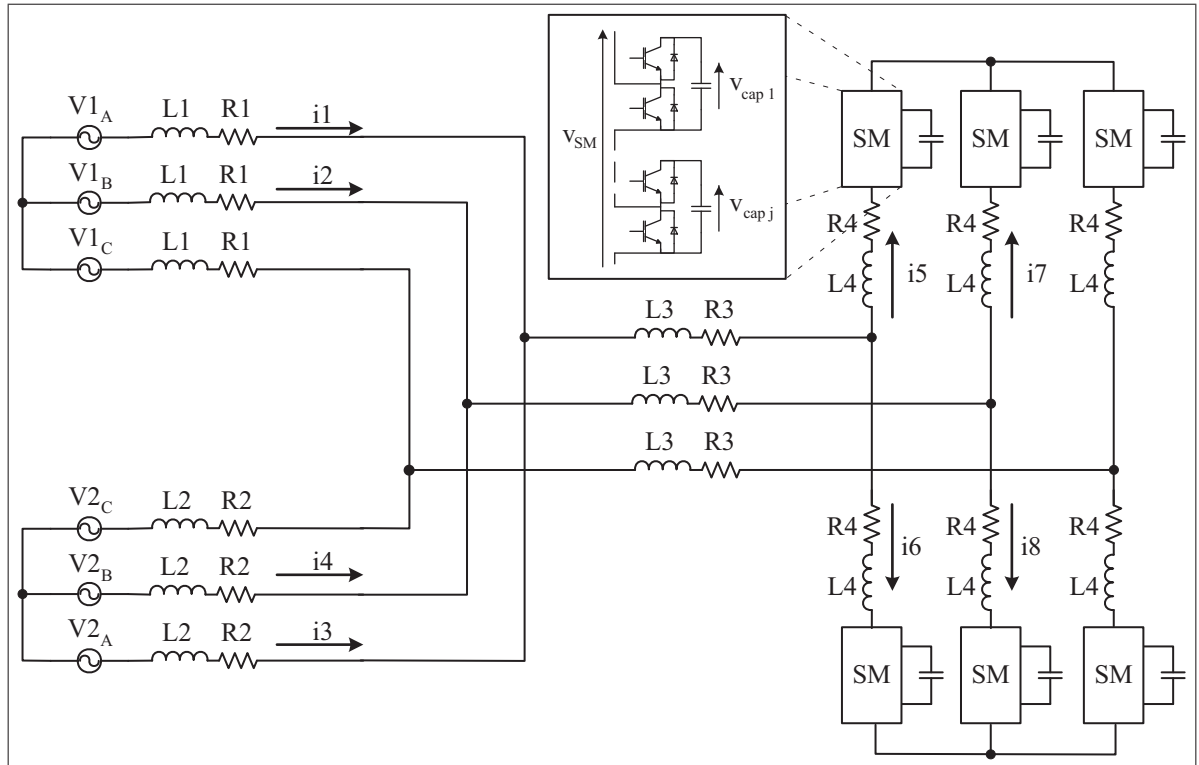


Figure 5.1 Complete model to be simulated

Schematic of figure 5.1 represents a power system where power is transferred from V1 toward V2 over a transmission lines. The MMC is connected at midway on the transmission line and it is used as a STATCOM. The AC network with its slow dynamic is simulated on CPU using a large times-step. The converter and its surrounding components is simulated on FPGA with a small time-step. Finally the sub-module (SM) itself is simulated using a switching function.

State-space equations of the continuous system are identified in (5.1).

$$\begin{aligned}
 \begin{bmatrix} X_{1s} \\ X_{2s} \\ X_{3s} \end{bmatrix} &= \overbrace{\begin{bmatrix} A_1^c & A_{12}^c & A_{13}^c \\ A_{21}^c & A_2^c & A_{23}^c \\ A_{31}^c & A_{32}^c & A_3^c \end{bmatrix}}^{A^c} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \overbrace{\begin{bmatrix} B_1^c \\ 0 \\ 0 \end{bmatrix}}^{B^c} U \\
 X1 &= \begin{bmatrix} i1 & \dots & i4 \end{bmatrix}^T \quad X2 = \begin{bmatrix} i6 & i8 \end{bmatrix}^T \quad X3 = \begin{bmatrix} v_{cap1} & \dots & v_{cap6j} \end{bmatrix}^T \\
 U &= \begin{bmatrix} V1_A & V1_B & V1_C & V2_A & V2_B & V2_C \end{bmatrix}^T
 \end{aligned} \tag{5.1}$$

In (5.1), state-vectors and state-space matrices are separated by lines; those represent points where extra state-variables will be added during discretization and decoupling of the model. State-variable  $i1$  to  $i4$ , in vector  $X1$ , are the different line currents from the AC network; lines are purely inductive with only resistive losses. More complex line models could have been used but it would only have made the resulting equations larger. State-variable  $i5$  to  $i8$ , in vector  $X2$ , are the different limb currents from the MMC converter. Since the MMC is used as a STATCOM only four state-variables are required for the converter. When power is transferred over a DC link, a fifth state-variable is required. Furthermore, in this example the MMC is directly connected to the AC network when it should be done through a power transformer; again power transformer was omitted to keep state-space equations to a manageable size. Finally,  $v_{cap1}$  to  $v_{cap6j}$ , in vector  $X3$ , are the state-variables for capacitor voltage for all the SM of the MMC, where  $j$  is the number of SM per limb. Values for matrices  $A^c$  and  $B^c$  can be found in appendix I.

This chapter is divided as follow. In section 5.1 decoupling methods for parallel and multi-rate simulation are presented. Then, the different parts of the simulation like the SM, the power system, and the converter are explained in section 5.2 to 5.4. Numerical stability and accuracy of the multi-rate real-time simulation model are discussed in section 5.5. Numerical accuracy of the model is also validated through simulation in section 5.6. Finally, conclusion is presented in section 5.7.



### 5.1 Parallel and multi-rate simulation

Real-time (RT) simulation is often characterized by hard time constraints difficult to achieve. These timing constraints can be resolved by using parallel and multi-rate (MR) simulation. For both solutions, state-space equations need to be decoupled. Using transmission line, model can be decoupled using the natural transmission delay inherent to the line. For short-line, it can be replaced by a lossless traveling line, which is the equivalent of an inductance with shunt parasitic capacitance Hong *et al.* (2009); Watson and Arrillaga (2003); Wang *et al.* (2010), also known as stubline. In Watson and Arrillaga (2003), the method uses single-rate simulation, and a traveling time equal to the simulation time-step is required. Value of the shunt parasitic capacitor added to the inductance is given by (5.2) (Dommel, 1969).

$$C_{shunt} = \frac{T_S^2}{L} \quad (5.2)$$

Where  $T_S$  is the simulation time-step and  $L$  is the inductance used for decoupling. In order to minimize the parasitic capacitor, this method must be used with a small sampling time and a large inductance.

Figure 5.2 shows the implementation of the method where  $V_A$ ,  $V_B$ ,  $Z_A$  and  $Z_B$  are Thevenin's equivalent circuit of larger networks, and the remaining components make the stubline. Volt-

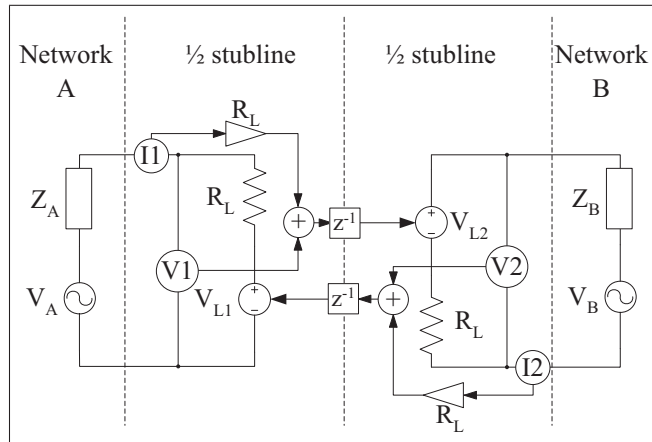


Figure 5.2 Stubline example

age sources  $V_{L1}$  and  $V_{L2}$  are controlled by the measurements with one-step delay, making it possible to simulate network A and network B in parallel, including half of the stubline with each network. Aside from the state-variables from the two networks, the stubline has four state-variables, given in (5.3).

$$\begin{bmatrix} I_{1n} \\ I_{2n} \\ V_{L1n} \\ V_{L2n} \end{bmatrix} = \begin{bmatrix} a11 & 0 & a13 & a14 \\ 0 & a22 & a23 & a24 \\ a31 & 0 & a33 & a34 \\ 0 & a42 & a43 & a44 \end{bmatrix} \begin{bmatrix} I_{1n-1} \\ I_{2n-1} \\ V_{L1n-1} \\ V_{L2n-1} \end{bmatrix} + \begin{bmatrix} b11 & 0 \\ 0 & b22 \\ b31 & 0 \\ 0 & b42 \end{bmatrix} \begin{bmatrix} V_{A_n} \\ V_{B_n} \end{bmatrix} \quad (5.3)$$

Simulation of (5.3) over two CPU is represented by the chronogram in figure 5.3 a). For each time-step, values from the different CPU are received, solutions for network A and B are computed, and results are finally exchanged between the CPU. Real-time simulation can be achieved if those three steps are executed within the simulation time-step used for the discretization of the system. In figure 5.3 a), CPU1 and CPU2 requires  $15 \mu s$  and  $10 \mu s$  to solve their respective system, leaving  $10 \mu s$  for the communication between the CPU. The traveling time of the waveform from network A to network B is then one time-step or  $25 \mu s$ . If an FPGA is used to solve network B, the required time to solve the system can be reduced from  $10 \mu s$  to only  $2 \mu s$ , as shown in figure 5.3 b). The reason why computational time can be reduced when using FPGA is because FPGA requires an application specific processor (ASP) Saad *et al.* (2015b). Unlike CPU, meant to handle a width range of instructions, processors coded on an FPGA are dedicated for specific application; making them extremely efficient. In figure 5.3 b), the model still uses single-rate simulation since both networks are discretized using a  $25 \mu s$  time-step. In this case, it is no longer parallel computing but rather serial computing. Network A still takes  $15 \mu s$  to compute, but the communication with the FPGA and solving network B on FPGA can be done before the next step. In figure 5.3 b), the traveling time of the waveform is reduced to half a time-step, or  $12.5 \mu s$ , since results from the FPGA is obtained before the next CPU step. This reduces by four the value of the shunt parasitic capacitor of the stubline. In figure 5.3 c) a smaller simulation time-step is used on FPGA, multi-rate simulation

is then achieved. In figure 5.3 c), network A is solved in  $15 \mu s$  with a large time-step ( $T_S$ ) of  $25 \mu s$ . Once solved, results are sent to the FPGA, where network B is solved using a smaller integration time-step ( $t_s$ ) of  $5 \mu s$ . After two  $t_s$ , values are sent to CPU for the next  $T_S$ , and solution for network B makes three more iterations before receiving the updated value from the CPU. Using multi-rate allows to observe faster dynamics, as well as reducing latency from

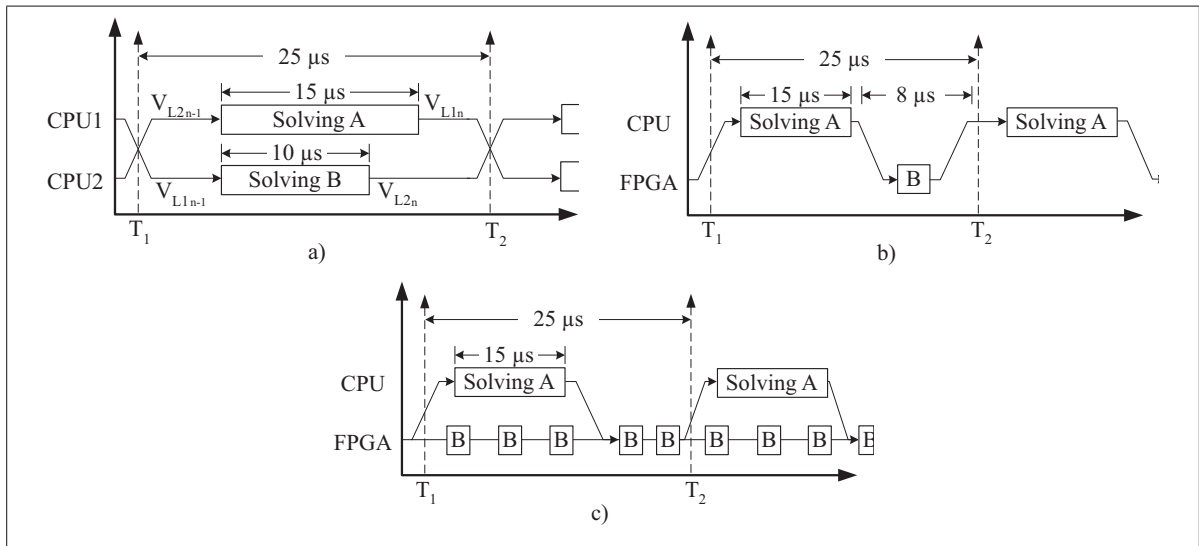


Figure 5.3 Chronogram for parallel computing using  
 a) two CPU with single-rate b) CPU/FPGA with single-rate  
 c) CPU/FPGA with multi-rate

external inputs and outputs.

Now using the method proposed in chapter 4, stability and accuracy of the multi-rate model can be studied. Equation (5.3) becomes (5.4) where  $\bar{\rho}$  is the ratio between the  $t_s$  and  $T_S$ . Equation

(5.4) will be used in section 5.5 to do the coupling between the state-vectors X1 and X2.

$$\begin{aligned}
 \begin{bmatrix} I_{1_n} \\ I_{2_n} \\ V_{L1_n} \\ V_{L2_n} \end{bmatrix} &= \overbrace{\begin{bmatrix} a_{11}\bar{\rho} + (1 - \bar{\rho}) & 0 & a_{13}\bar{\rho} & a_{14}\bar{\rho} \\ 0 & a_{22} & a_{23} & a_{24} \\ a_{31}\bar{\rho} & 0 & a_{33}\bar{\rho} + (1 - \bar{\rho}) & a_{34}\bar{\rho} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}}^{K_{1,d1}} \begin{bmatrix} I_{1_{n-1}} \\ I_{2_{n-1}} \\ V_{L1_{n-1}} \\ V_{L2_{n-1}} \end{bmatrix} \\
 &+ \overbrace{\begin{bmatrix} K_{11}^{d2} & K_{12}^{d2} \\ b_{11}\bar{\rho} & 0 \\ 0 & b_{22} \\ b_{31}\bar{\rho} & 0 \\ 0 & b_{42} \end{bmatrix}}^{[K_{11}^{d2} \ K_{12}^{d2}]} \begin{bmatrix} V_{A_n} \\ V_{B_n} \end{bmatrix} \quad (5.4)
 \end{aligned}$$

Another method of decoupling system is the use of explicit solver like it is done for the modeling the SM in the next section.

## 5.2 Sub-Module

The SM of the converter are simulated using a switching function and the explicit solver forward Euler (FE). Every SM are then decoupled and can be all simulated in parallel. When pulses are applied to the SM, the proposed approach is very similar to the one seen in literature. The behavior of the switching function when no pulse is applied to the SM (Blocking mode) is part of the contributions in this thesis. It has been often proposed to replace a blocking power switch by a R-C snubber, which generate large losses during normal operation. In the proposed method, losses due to this snubber are only present during blocking, and are completely removed during normal operation of the converter. Figure 5.4 shows the half-bridge SM to be simulated. The switching function to be simulated has three inputs,  $I_{limb}$ , S1, and S2 and returns  $V_{SM}$  and  $v_{cap}$  as outputs. In controlled mode, S1 and S2 are always complementary. The option where both switches are closed results in short-circuiting the capacitor, and is therefore

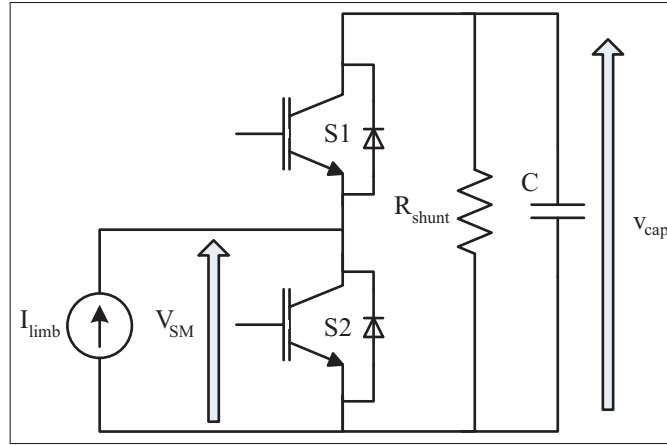


Figure 5.4 Half-bridge sub-module

treated as an error. If both switches are open, the SM is no longer in controlled in mode (CM) and enter natural rectifying mode (NRM). Equation (5.5) gives the output voltage of the SM and (5.6) the state equation for the capacitor.

$$V_{SM} = \begin{cases} \text{Error} & \text{when } S1 = 1 \text{ \& } S2 = 1 \\ v_{cap} & \text{when } S1 = 1 \text{ \& } S2 = 0 \\ 0 & \text{when } S1 = 0 \text{ \& } S2 = 1 \\ \text{NRM} & \text{when } S1 = 0 \text{ \& } S2 = 0 \end{cases} \quad (5.5)$$

$$v_{cap} = \begin{cases} \text{Error} & \text{when } S1 = 1 \text{ \& } S2 = 1 \\ v_{cap} + \frac{I_{limb}}{Cs} & \text{when } S1 = 1 \text{ \& } S2 = 0 \\ v_{cap} & \text{when } S1 = 0 \text{ \& } S2 = 1 \\ \text{NRM} & \text{when } S1 = 0 \text{ \& } S2 = 0 \end{cases} \quad (5.6)$$

In NRM, the output  $V_{SM}$  is function of the anti-parallel diodes of the IGBT and  $I_{limb}$ . According to figure 5.4, a positive current polarizes  $S1$ , charge the capacitor, and  $V_{SM}$  is equal to  $V_{cap}$ . A negative current polarizes  $S2$ , does not charge the capacitor, and  $V_{SM}$  is equal to zero. In the case where the diodes are no longer polarized, the input current  $I_{limb}$  should be equal to zero. This means that during blocking, the values of  $V_{SM}$  must varies to regulated  $I_{limb}$  to zero;

which can be done using a PI regulator. Equations (5.7) and (5.8) give the switching function behaviour in NRM.

$$V_{SM} = \begin{cases} v_{cap} & \text{when } I_{limb} > 0 \\ 0 & \text{when } I_{limb} < 0 \\ \left(K_p + \frac{K_i}{s}\right) \cdot I_{limb} + V_0 & \text{when blocking} \end{cases} \quad (5.7)$$

$$v_{cap} = \begin{cases} \frac{I_{limb}}{C_s} + v_{cap} & \text{when } I_{limb} > 0 \\ v_{cap} & \text{otherwise} \end{cases} \quad (5.8)$$

Two conditions are required to initialize blocking;  $v_{cap}$  has to be greater than  $V_{SM}$ , and  $I_{limb}$  must cross zero.

The next step is the discretization and implementation of the different equations. Two numerical integrators are required for each SM, one for the capacitor and one for the PI controller. In both cases, FE integration method is used, allowing the decoupling and parallel processing of the SM. For the capacitor, (5.6) becomes (5.9) once discretized.

$$v_{cap_n} = \left(1 - \frac{T}{C \cdot R_{shunt}}\right) v_{cap_{n-1}} + \left(\frac{T \cdot d}{C}\right) I_{limb_{n-1}} \quad (5.9)$$

FE method being explicit, only values from the previous step are required. A discharge resistance,  $R_{shunt}$  is added allowing a slow discharge of the capacitor, as it would be in a real prototype. In controlled mode, the variable  $d$  is equal to the duty cycle of  $S1$ . In NRM,  $d$  is equal to 1 for a positive current, and 0 otherwise. The value of  $d$  is then defined accordingly in (5.10).

$$d = \max(S1, (I_{limb} > 0)) \quad (5.10)$$

Where  $S1$  can be any value between 0 and 1; when  $S1$  is a fraction, it means that  $S1$  is only conducting during a fraction of the current time-step. Such interpolation is used for  $d$  and the same can be done when testing if  $I_{limb}$  is greater than zero; which would return a fraction if

$I_{limb}$  crosses zero during the current time-step. Finally,  $d$  is equal to the largest value between those two conditions as shown in (5.10).

As for the value of  $V_{SM}$ , it is obtained differently whether it is in conducting or blocking mode. Equation (5.11) gives this relation for blocking and conducting mode.

$$V_{SM_n} = \begin{cases} d \cdot v_{cap_n} & \text{when conducting} \\ R_{snub} \cdot I_{limb_{n-1}} + v_{snub_n} & \text{when blocking} \end{cases} \quad (5.11)$$

$$v_{snub_n} = v_{snub_{n-1}} + \frac{T}{C_{snub}} \cdot I_{limb_{n-1}} \quad (5.12)$$

When conducting,  $V_{SM}$  is function of  $d$ , like  $V_{cap}$ . When blocking, current is regulated to zero using a PI controller, like in (5.7). Gains  $K_p$  and  $K_i$  of PI are replaced by  $R_{snub}$  and  $C_{snub}$ , since a PI controller behave exactly like a RC snubber. It is important to keep in mind that this is a numerical snubber, which is highly dependent of the simulation time-step. By simulating the SM on FPGA allows very little losses achieving accurate simulation results in real-time. SM can represented by state-space equation given in (5.13), based on capacitor, and snubber

voltages.

$$\begin{aligned}
 \overbrace{\begin{bmatrix} v_{\text{cap } 1_n} \\ \vdots \\ v_{\text{cap } 6j_n} \end{bmatrix}}^{X_{3n}} &= \overbrace{\begin{bmatrix} 1 - \frac{T}{C \cdot R_{\text{shunt}}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 - \frac{T}{C \cdot R_{\text{shunt}}} \end{bmatrix}}^{A_3^{d1}} X_{3n-1} + \overbrace{\begin{bmatrix} \frac{T \cdot d_1}{C} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{T \cdot d_{6j}}{C} \end{bmatrix}}^{K_{32}^{d1}} I_{n-1} \\
 \overbrace{\begin{bmatrix} v_{\text{snub } 1_n} \\ \vdots \\ v_{\text{snub } 6j_n} \end{bmatrix}}^{X_{23n}} &= \overbrace{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}^{A_{\text{snub}}^{d1}} X_{23n-1} + \overbrace{\begin{bmatrix} \frac{T}{C_{\text{snubb}}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{T}{C_{\text{snubb}}} \end{bmatrix}}^{K_{\text{snub}}^{d1}} I_{n-1} \\
 \overbrace{\begin{bmatrix} v_{SM} \\ v_{SM1_n} \\ \vdots \\ v_{SM6_n} \end{bmatrix}}^{V_{SM}} &= \begin{cases} \overbrace{\begin{bmatrix} d_1 : d_j & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{5j+1} : d_{6j} \end{bmatrix}}^C X_{3n} & \text{When conducting} \\ \overbrace{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}^C X_{23n} + \overbrace{[R_{\text{snub}}]}^D I_{n-1} & \text{When blocking} \end{cases}
 \end{aligned} \tag{5.13}$$

In equation (5.13),  $j$  denotes the number of SM in one limb; there is then a total of  $6j$  SM in the whole converter. The size of  $X_3$ , containing the SM capacitor voltages, is the equal to the total number of SM in the converter, and the same goes for  $X_{23}$  containing snubber voltages. The six voltages required by each arm is found in  $V_{SM}$ , and is obtained by either  $X_3$ , when conducting, or a combination of  $X_{23}$  and the arm current when blocking. State-space matrices  $A_3$  and  $A_{\text{snub}}$  only have elements on the diagonal; demonstrating that it can be in solved in parallel. Using



the two decoupling technique presented in this section, the system from figure 5.1 can now be decoupled in three sub-systems presented in the following sub-section.

### 5.3 Discrete power system model

The first part part of the system to be decoupled is the two network with their transmission lines. This part of the system is simulated using Matlab/Simulink and SimPowerSystems toolbox. It is decoupled from the MMC converter by replacing the lines inductance  $L3$ , from figure 5.1, by MR stublines. The new model to discretize is shown in figure 5.5. Inductors  $L3$  have been

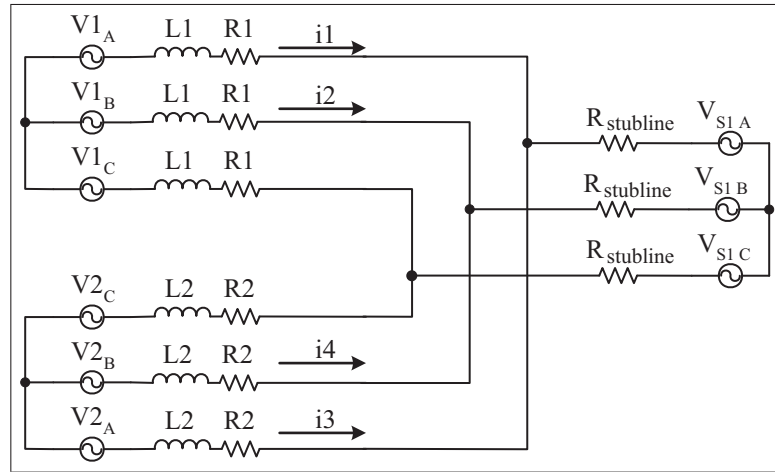


Figure 5.5 Power system decoupled with stubline

replaced by voltage source with a resistance  $R_{stubline}$ . Using trapezoidal (TR) method to solve the solve the system  $R_{stubline}$  is given by (5.14)

$$R_{stubline} = \frac{L}{T_s} + \frac{R}{2} \quad (5.14)$$

Where  $L$  is the inductor value used for the stubline,  $R$  is the conducting losses of the inductor, and  $T_s$  is the sampling time of the simulation.

Part of the state-space system from (5.1) can now be decoupled and becomes the one presented in (5.15). Voltage sources,  $V_{S1A}$ ,  $V_{S1B}$ ,  $V_{S1C}$ , inherent to stublines are added to the system,

and they can be considered as inputs to the system.

$$\begin{aligned}
 X_1 s &= A_1^c X_1 + K_{12}^c X_{12} + B_1^c U \\
 X_1 &= \begin{bmatrix} i1 & \dots & i4 \end{bmatrix}^T \quad X_{12} = \begin{bmatrix} V_{S1A} & V_{S1B} & V_{S1C} \end{bmatrix}^T \\
 U &= \begin{bmatrix} V1_A & V1_B & V1_C & V2_A & V2_B & V2_C \end{bmatrix}^T
 \end{aligned} \tag{5.15}$$

Equation (5.15) is discretized using TR method yielding (5.16).

$$X_{1n} = A_1^{d1} X_{1n-1} + K_{12}^{d1} X_{12n-1} + B_1^{d1} U_{n-1} + K_{12}^{d2} X_{12n} + B_1^{d2} U_n \tag{5.16}$$

TR method being an implicit solver, inputs of both the present and previous step are required to find the output of the current step. Matrices multiplied by values from the previous step are followed by the subscript d1, and matrices multiplied by values from the current step are followed by the subscript d2. Detail value for the different matrices can be found in appendix I. The same scheme is applied to the other part of the system and is detailed in the following section.

#### 5.4 Discrete converter

This part of the model is simulated using a small sampling time. This is achieved using the eHS solver (Belanger *et al.*, 2013) from OPAL-RT Technologies. MMC has many inductance connected together, which yields state-variable highly coupled. It was demonstrated in (Grégoire *et al.*, 2015a, 2014b) that a one-step solution, containing all the inductive elements is required. Figure 5.6 shows the circuit to be simulated using eHS on the FPGA, which uses a backward Euler (BE) solver. In figure 5.6, the voltages  $V_{SM1}$ ,  $V_{SM2}$ ,  $V_{SM3}$ ,  $V_{SM4}$ ,  $V_{SM5}$  and  $V_{SM6}$ , are obtained by summing the cell's voltage from each limb, as shown in (5.13).

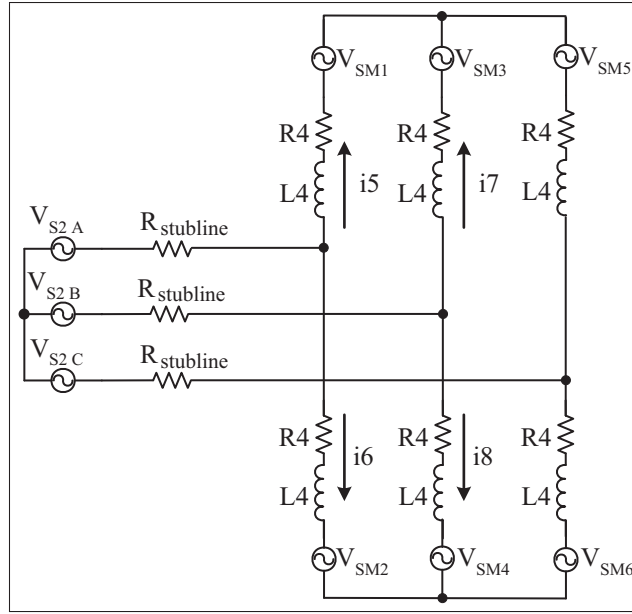


Figure 5.6 Decoupled equivalent modular multilevel converter

Using Kirkoof's current law (KCL), four equations are found and yield (5.17). Values for matrices  $A_2^c$ ,  $K_{21}^c$  and  $K_{23}^c$  are given in appendix I.

$$X_2 s = A_2^c X_2 + K_{21}^c X_{12} + K_{23}^c V_{SM}$$

$$X_2 = \begin{bmatrix} i5s & \cdots & i8s \end{bmatrix}^T \quad X_{12} = \begin{bmatrix} v_{S2\ A} & v_{S2\ B} & v_{S2\ C} \end{bmatrix}^T \quad V_{SM} = \begin{bmatrix} v_{SM1} & \cdots & v_{SM6} \end{bmatrix}^T \quad (5.17)$$

Equation (5.17) is then discretized using (BE) and (5.18) is obtained. Where  $X_2$ ,  $X_{21}$ ,  $V_{SM}$  are the vectors containing state-variables of the current, voltages from the stubline, and the sum of voltages from the SM.

$$X_{2n} = A_2^{d1} X_{2n-1} + K_{21}^{d2} X_{12n} + K_{23}^{d2} V_{SMn} \quad (5.18)$$

Unlike TR, BE only requires input values of the current step to find the solution of the current step. In (5.18) the subscript  $d1$  and  $d2$  indicate if the vector multiplying the matrix is obtained at the previous step,  $n-1$  in the case of  $d1$ , or at the present step,  $n$  in the case of  $d2$ .

The last state vector to add to the system is the one computing stubline's voltages. State-space equations to compute the stubline voltages of vectors  $X_{12}$  are obtained using (5.19).

$$X_{12n} = A_{stub}^{d1} X_{12n-1} + \begin{bmatrix} 0 & K_{stub1}^{d1} \\ K_{stub2}^{d1} & 0 \end{bmatrix} \begin{bmatrix} X_{1n-1} \\ X_{2n-1} \end{bmatrix} \quad (5.19)$$

Although stubline only required value from the previous step to compute its voltage, it is still implemented using an implicit method. Contribution from state-variable of the current step are taken into account by  $R_{snubber}$  added to model; as it was done with the companion model in chapter 3.

Using all the different discretized matrices, the complete decoupled system of (5.20) is obtained. Further more, since different sampling rate are used throughout the model, the system

must be linearized using the proposed method in in chapter 4.

$$\begin{aligned}
 & \begin{bmatrix} X_{1_n} \\ X_{12_n} \\ X_{2_n} \\ X_{23_n} \\ X_{3_n} \end{bmatrix} = \overbrace{\begin{bmatrix} \bar{\rho} B_1^{d1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{B^{d1}} U_{n-1} + \overbrace{\begin{bmatrix} \bar{\rho} B_1^{d2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{B^{d2}} U_n \\
 & + \overbrace{\begin{bmatrix} \bar{\rho} A_1^{d1} + 1 - \bar{\rho} & \bar{\rho} K_{12}^{d1} & 0 & 0 & 0 \\ \bar{\rho} K_{stub1}^{d1} & \bar{\rho} A_{stub}^{d1} + 1 - \bar{\rho} & \bar{\rho} K_{stub2}^{d1} & 0 & 0 \\ 0 & 0 & A_2^{d1} + K_{23}^{d2} V_{SM} & 0 & 0 \\ 0 & 0 & K_{snub}^{d1} & A_{snub}^{d1} & 0 \\ 0 & 0 & K_{32}^{d1} & 0 & A_3^{d1} \end{bmatrix}}^{A^{d1}} \overbrace{\begin{bmatrix} X_{1_{n-1}} \\ X_{12_{n-1}} \\ X_{2_{n-1}} \\ X_{23_{n-1}} \\ X_{3_{n-1}} \end{bmatrix}}^X \quad (5.20) \\
 & + \overbrace{\begin{bmatrix} 0 & \bar{\rho} K_{12}^{d2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & K_{21}^{d2} & 0 & K_{23}^{d2} C_{snub} & K_{23}^{d2} C_{SM} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}^{A^{d2}} \begin{bmatrix} X_{1_n} \\ X_{12_n} \\ X_{2_n} \\ X_{23_n} \\ X_{3_n} \end{bmatrix}
 \end{aligned}$$

The system of figure 5.1 is now decoupled and discretized using MR simulation. Equation (5.21), from which poles location analysis can be applied, is obtained by isolating state-vector  $X_n$ .

$$\begin{aligned}
 X_n &= A^{d1} X_{n-1} + A^{d2} X_n + B^{d1} U_{n-1} + B^{d2} U_n \\
 &= \underbrace{\left( I - A^{d2} \right)^{-1} A^{d1} X_{n-1}}_{\text{Poles of the system}} + \left( I - A^{d2} \right)^{-1} B^{d1} U_{n-1} + \left( I - A^{d2} \right)^{-1} B^{d2} U_n \quad (5.21)
 \end{aligned}$$

Pole location analysis can now be applied to the matrix in (5.21) to validate the stability of the decoupled MR system.

### 5.5 Numerical stability and accuracy

There is a very important difference between numerical stability and accuracy. Using pole location analysis, model stability is achieved if every poles are within unity circle. If a model is stable, it eventually converges and reaches steady-state. How it reaches steady-state determines whether it is accurate or not. Information about system dynamics is found in pole location of the state-space equations. By comparing the poles of the reference model from (5.1) with the one obtained for the discrete decoupled system of (5.21) accuracy of the model can be verified. The slow sub-system uses a simulation time-step of  $50 \mu\text{s}$  and TR solver. The fast sub-system uses a simulation time-step of  $500 \text{ ns}$  and a combination of BE and FE solver. Simulation parameters for the model are given in SI and in pu in table 5.1.

Table 5.1 Model parameters of figure 5.1

Nominal power	200 MVA	1 pu
Nominal voltage	230 kV	1 pu
Nominal frequency	50 Hz	1 pu
L1, L2	420 mH	0.49 pu
R1, R2	2.6 $\Omega$	0.01 pu
L3	168 mH	0.2 pu
R3	0.5 $\Omega$	0.002 pu
L4	24 mH	0.0285 pu
R4	75 m $\Omega$	0.0003 pu
Number of SM per limb	250	250
C	2 mF	0.006 pu

Since the discretized system has two modes of operation, conducting and blocking, stability and accuracy is tested for both mode. Also, because of the decoupling state-variable,  $X_{12}$  and  $X_{23}$ , the decoupled system is expected to have more poled than the reference model.

### 5.5.1 Pole location analysis

When observing poles of a discrete system, the dynamic of the pole is given by (5.22) to (5.25)

$$\omega_n = \left| \frac{\ln(z)}{T_S} \right| \quad (5.22) \quad \zeta = -\cos(\angle \ln(z)) \quad (5.23)$$

$$\tau = \frac{1}{\omega_n \zeta} \quad (5.24) \quad \text{Sampling of } \omega_n = \frac{2\pi}{\angle z} \quad (5.25)$$

In (5.22) and (5.23), the discrete pole  $z$  is transferred to the continuous plane, where classical method may be used. The same can be observed for (5.24) where the time constant of the system is obtained from (5.22) and (5.23). Finally, (5.25) gives information on the number of samples that are used to represent the natural frequency of the pole. It means that a pole located on the real axes, near -1, has an angle of  $\pi$  and would only have to sample per cycle of  $\omega_n$ . Such poles were observed in in chapter 3 when stubline was used.

### 5.5.2 Conducting mode

Over the course of operation, poles of the system varies as different switching patterns are applied. Equivalent capacitor value from one limb varies by a factor equal to the number of cells in limbs; equal to  $C$  when only one SM is conducting and  $\frac{C}{j}$  when all the SM are conducting. Over normal operation, modulating signal for each limb is a sinusoidal signal varying between 0 and 1 at the natural frequency. Modulating signal are complementary between the upper and lower limb, and there is an offset of  $2\pi/3$  between each phases. This modulating signal is applied to the system over two cycles during which poles from (5.21) are obtained every  $50 \mu s$ , and are shown in figure 5.7 a). In figure 5.7 a), poles are always within the unity circle, distributed mostly along the real-axis, very close to the limit of the unity circle. This can be explained by the choice of the sampling time used for the pole location analysis. The closer the pole is from the edge of the circle, the longer it takes to reach its steady-state, in regard to the sampling time of the model. If the pole analysis was done using a larger sampling time, steady-state would be reached in less simulation steps, and therefore its pole would be closer to the center of the circle. Figure 5.7 b) zoom on the poles, and poles movement for different modulation index can be observed by the line linking the moving poles. Most poles have little

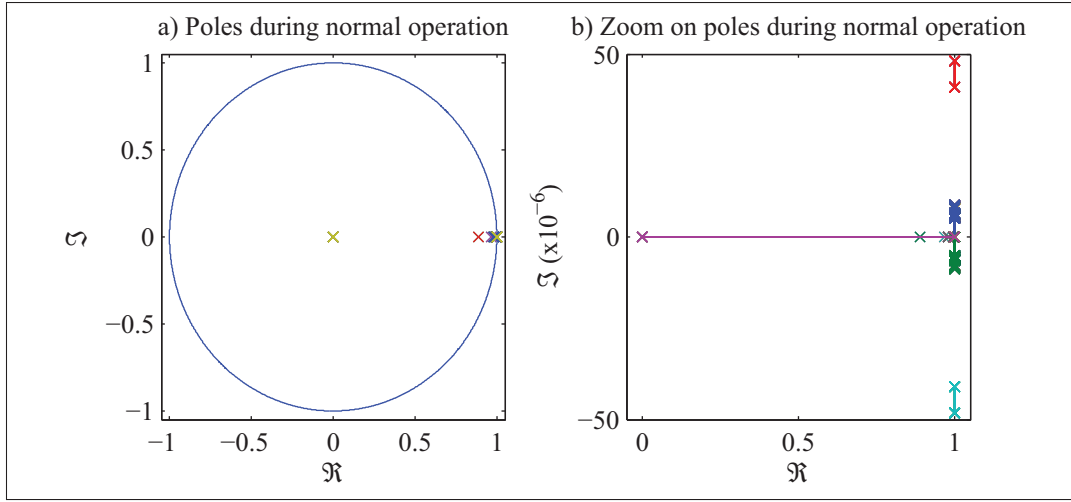


Figure 5.7 Poles location during steady-state a) for the unity circle b) zoom in

variation, except for the ones going from the edge of the circle to its origin. When modulation index reaches 0, there is no more capacitor in the limb, and therefore no more pole, which is then appearing at the origin. Finally, figure 5.7 shows that during normal operation, every poles stay within the circle and the system is stable.

Abnormal operation of the converter should also be considered, like the cases presented in chapter 2 and (Grégoire *et al.*, 2014b). Those cases were obtained in NRM and were responsible for glitches on the DC bus. For those cases, either all the SM or none of them are connected in series, resulting in modulation index equal to either 1 or 0. Figure 5.8 shows four different cases, where  $M_1$  to  $M_6$  indicated if the SM of each limb are bypassed, for  $M=0$ , or if they are conducting,  $M=1$ . For all those cases, model remains stable.

Accuracy of the proposed method can also be tested by comparing them to the poles obtained from the reference model, results are presented in table 5.2. There is very little difference between the different reference and the proposed method. The largest absolute error observed on  $\lambda_{1,2}$  and it would results in a variation of 0.5 Hz in the natural frequency of the pole, and a time constant of 600 ms instead of a time constant of 586 ms. These results are confirmed with simulation results in section 5.6.



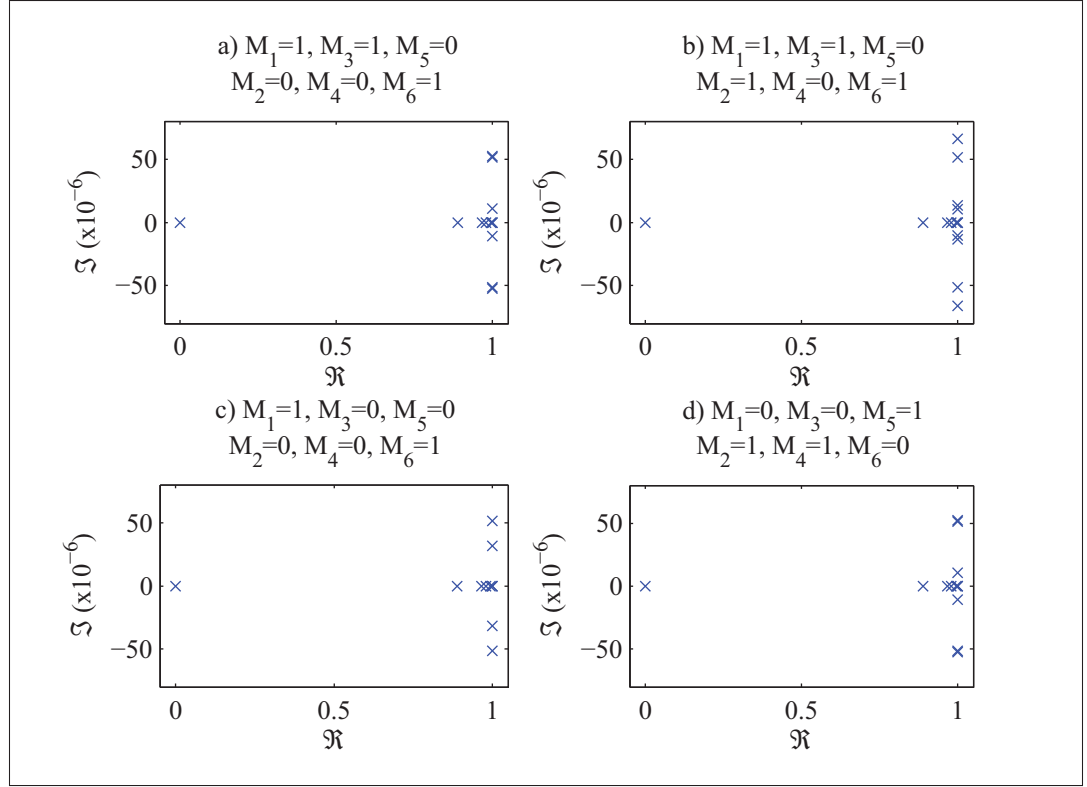


Figure 5.8 Poles location during NRM for positive current in  
a) 2 upper limbs & 1 lower limb b) 2 upper limbs & 2 lower limbs  
c) 1 upper limb & 1 lower limb d) 1 upper limb & 2 lower limbs

Table 5.2 Poles of the system during conducting mode

	Reference	Multi-rate model
$\lambda_{1,2}$	$0.9999 \pm j52.4148 \times 10^{-6}$	$0.9999 \pm j47.3815 \times 10^{-6}$
$\lambda_{3,4}$	$0.9999 \pm j32.9655 \times 10^{-6}$	$0.9999 \pm j36.0759 \times 10^{-6}$
$\lambda_{5,6}$	$0.9999 \pm j6.4675 \times 10^{-6}$	$0.9999 \pm j8.0816 \times 10^{-6}$
$\lambda_{7,8}$	$0.9999 \pm j3.2524 \times 10^{-6}$	$0.9999 \pm j6.0167 \times 10^{-6}$
$\lambda_{9,10}$	0.9999969	0.9999969
$\lambda_{11,12}$	0.9999999	0.9999999

### 5.5.3 Blocking mode

On a real MMC, blocking of the power components occurs exactly at zero-crossing of the current; power switch impedance becomes very large at zero current. In simulation, because of discretization, current might not be exactly zero at blocking. Therefore, when the low impedance

of the power switch or diode is replaced by a high impedance, it can result in high voltage spikes at the power switches, since current is not exactly zero. To avoid over voltage, dynamic of the limbs current should be controlled and regulated to zero. This is done by introducing  $R_{snub}$  and  $C_{snub}$  which play the role of a PI controller. Values for  $R_{snub}$  and  $C_{snub}$  are obtained according to the simulation time-step and the limb inductance of the converter.  $C_{snub}$  is given by (5.26), its oscillating frequency should be 10 to 20 times smaller than the sampling frequency.

$$C_{snub} = \left( \frac{k_1 T_S}{2\pi} \right)^2 \frac{1}{L_{limb}} \quad (5.26)$$

Where  $T_S$  is the sampling time of the simulation,  $k_1$  is a value between 10 and 20, and  $L_{limb}$  is the arm inductance. The value of  $R_{snub}$  is then obtained using (5.27) so that the system has a nearly critical damping factor.

$$R_{snub} = \frac{k_2 4\pi L_{limb}}{k_1 T_S} \quad (5.27)$$

Value of  $k_2$  is the damping factor of the circuit and should be between 0.8 and 1.2.  $L_{limb}$  is present in both (5.26) and (5.27), and although it is the main inductance of the current that need to be regulated, other inductance in the circuit also influence the current dynamic. Therefore gain  $k_1$  and  $k_2$  might need some adjusting, and poles of the system should be verified using (5.21). Impact on poles location for different  $k_1$  and  $k_2$  is shown in figure 5.9. Values for  $k_1$  and  $k_2$  are outside the proposed boundary to highlight their roles on poles location. In figure 5.9 a) and b), the damping ratio is small,  $k_2=0.5$ , and oscillating poles can be observed. In a discrete pole location analysis, the absolute value of the pole determines its natural frequency, while its angle determines the number of sampling over one cycle of its natural frequency. When  $k_1$  is small, the natural frequency of the pole increases, and since the sampling time remains the same, poles naturally move toward the left-side of the circle, as demonstrated by (5.25). In figure 5.9 c) the damping ratio is small, and the natural frequency is high, which results in a numerical instability. In figure 5.9 d) the natural frequency is reduced, and now although no poles are outside the unity circle, some are exactly equal to 1.

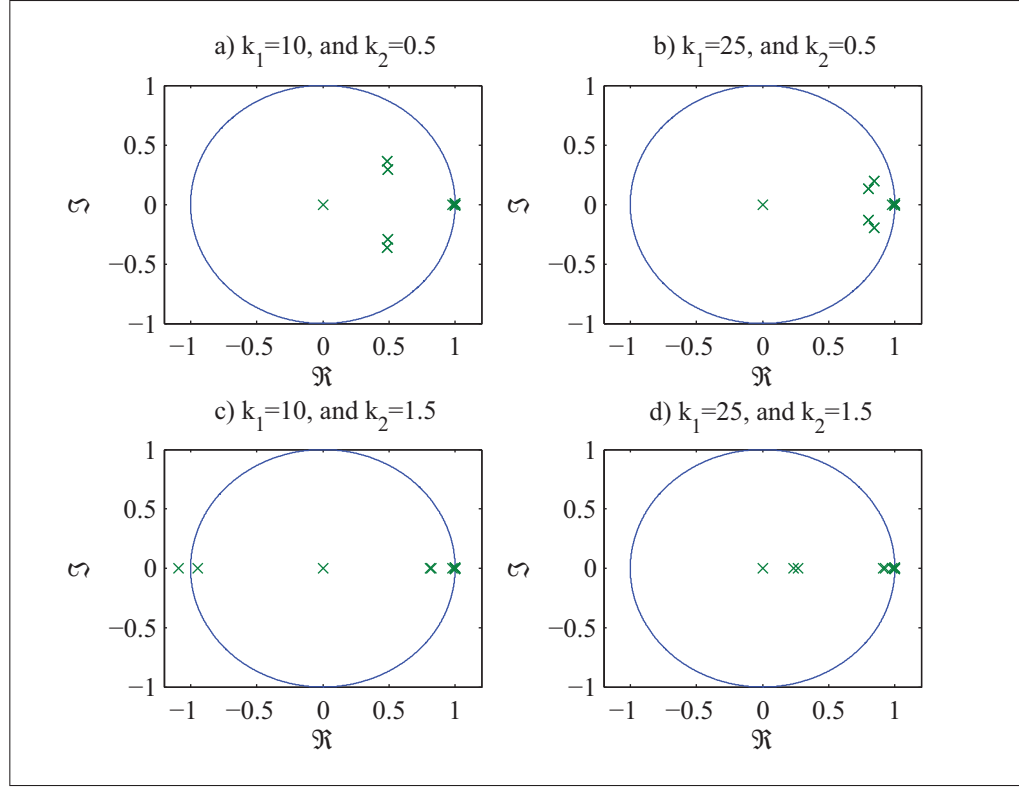


Figure 5.9 Poles location during blocking for  
a)  $k_1=10$  &  $k_2=0.5$  b)  $k_1=25$  &  $k_2=0.5$   
c)  $k_1=10$  &  $k_2=1.5$  d)  $k_1=25$  &  $k_2=1.5$

To avoid critically stable system, values for  $k_1$  and  $k_2$  are set to 10 and 1.1 respectively. During blocking, reference model switches are replaced by resistance of  $120 \text{ M}\Omega$  or  $450 \times 10^3 \text{ pu}$ , resulting in leakage current of  $2 \times 10^{-6} \text{ pu}$ .  $R_{snub}$  and  $C_{snub}$  have a combined impedance of  $120 \text{ M}\Omega$  at 50 Hz, and therefore similar leakage currents are expected. This is verified in the next section through simulation results.

## 5.6 Simulation results

The power system electrical schematic presented in figure 5.1 is simulated for converter initialization, steady-state operation, and emergency stop caused by a fault. Simulation results are presented for the proposed model running in real-time, and they are compared with a reference model obtained using offline simulation with variable-step solver. Simulation parameters used

are the one from table 5.1. The model is implemented using OPAL-RT technologies real-time simulator OP4500 (OPAL-RT Technologies Inc., 2014). As it was previously mentioned in section 5.5, the slower sub-systems uses a simulation time-step of  $50 \mu\text{s}$ , and TR solver. This sub-system is run on CPU using Matlab/Simulink SimPowerSystem toolbox. The converter part is simulated using OPAL-RT eHS solver on FPGA with a time-step of 500 ns. Finally, SM have been implemented on the FPGA using pipe lining and parallel implementation method with a sampling rate of 500 ns. Using this implementation, a total of 256 SM per limb can be simulated, or a total of 1536 SM per converter. Additional tools are added to the FPGA configuration, allowing signals monitoring, or controlling the converter using IOs and an external controller. In the simulation results, only Voltage sources from network 1 have an amplitude of 1 pu and are lagging of 40 degrees. Voltage sources from network 2 have an amplitude of 1 pu and are leading of 40 degrees. Voltage from the MMC, operating as a STATCOM, is then adjusted to compensate reactive power at each source to ensure maximum power transfer.

### 5.6.1 Steady-state

Two steady-state operation points are presented in this section; simulation when the STATCOM is disconnected and when is compensating the networks. Figure 5.10 shows simulation results when the STATCOM is disconnected. In this case, results from the proposed model are superimposed on the reference model. Naturally, there is no power flowing in bus 3 since the STATCOM is disconnected. Power is transferred from bus 1 toward bus 2 with power factor of 0.79. Simulation accuracy remains the same when the STATCOM is activated, as shown in figure 5.11.

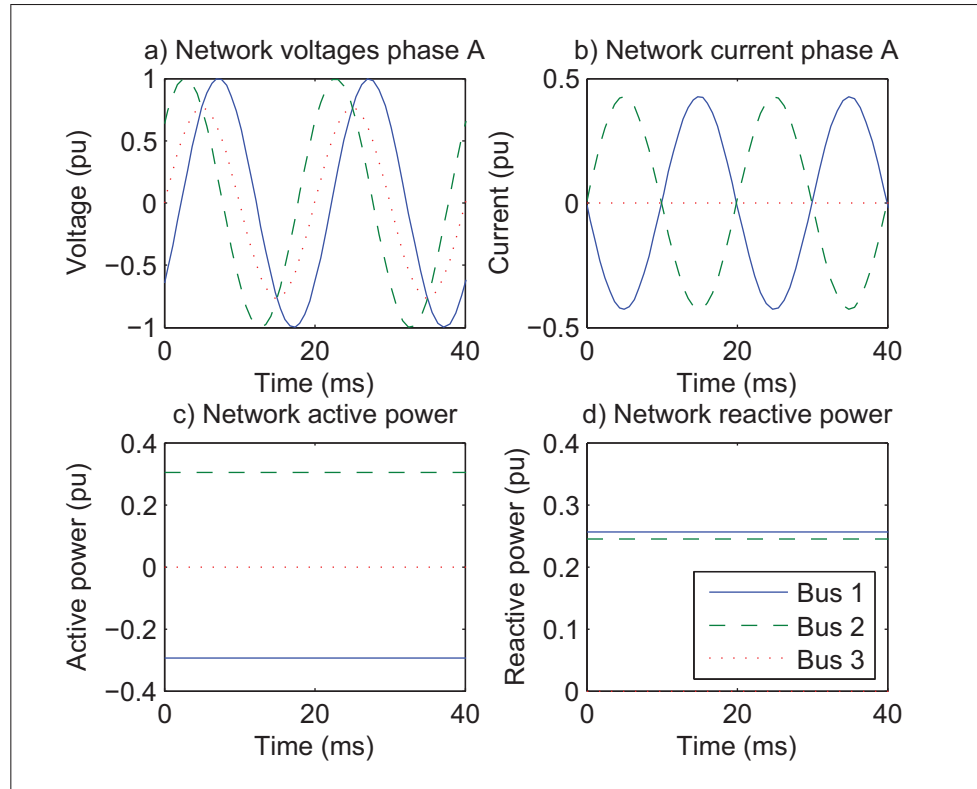


Figure 5.10 Simulation results without STATCOM for  
a) network voltages phase A b) network currents phase A  
c) network active power d) network reactive power

In figure 5.11 a), voltage at bus 3 now reaches 1.43 pu, the STATCOM is generating nearly 1 pu of reactive power, allowing power transfer at unitary power factor. When gating signals are applied to the converter, in control mode, real-time simulation of the MMC offers little challenges. This is validated with the simulation results in steady-state as results from the proposed model are superimposed with the reference model.

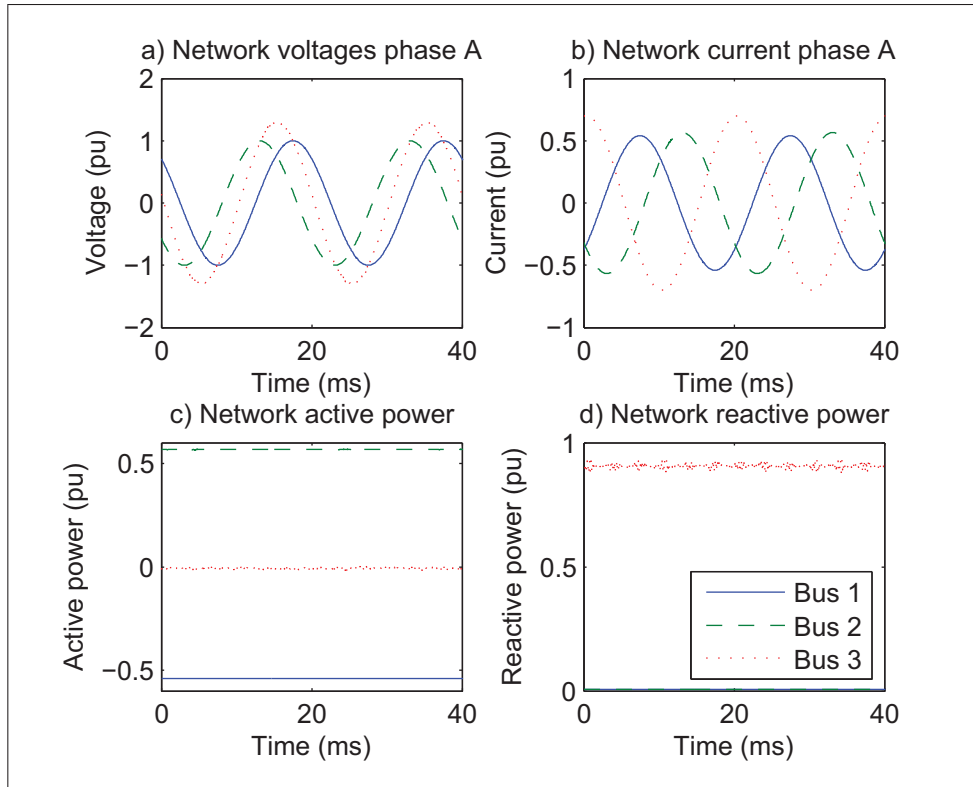


Figure 5.11 Simulation results with STATCOM for  
a) network voltages phase A b) network currents phase A  
c) network active power d) network reactive power

### 5.6.2 Converter initialization

During initialization, no energy is stored in the converter. It results an inrush current only limited by the line impedance of the converter. During this transient, differences appear between the reference model and the proposed model. Figure 5.12 shows the current in phase A for each bus.

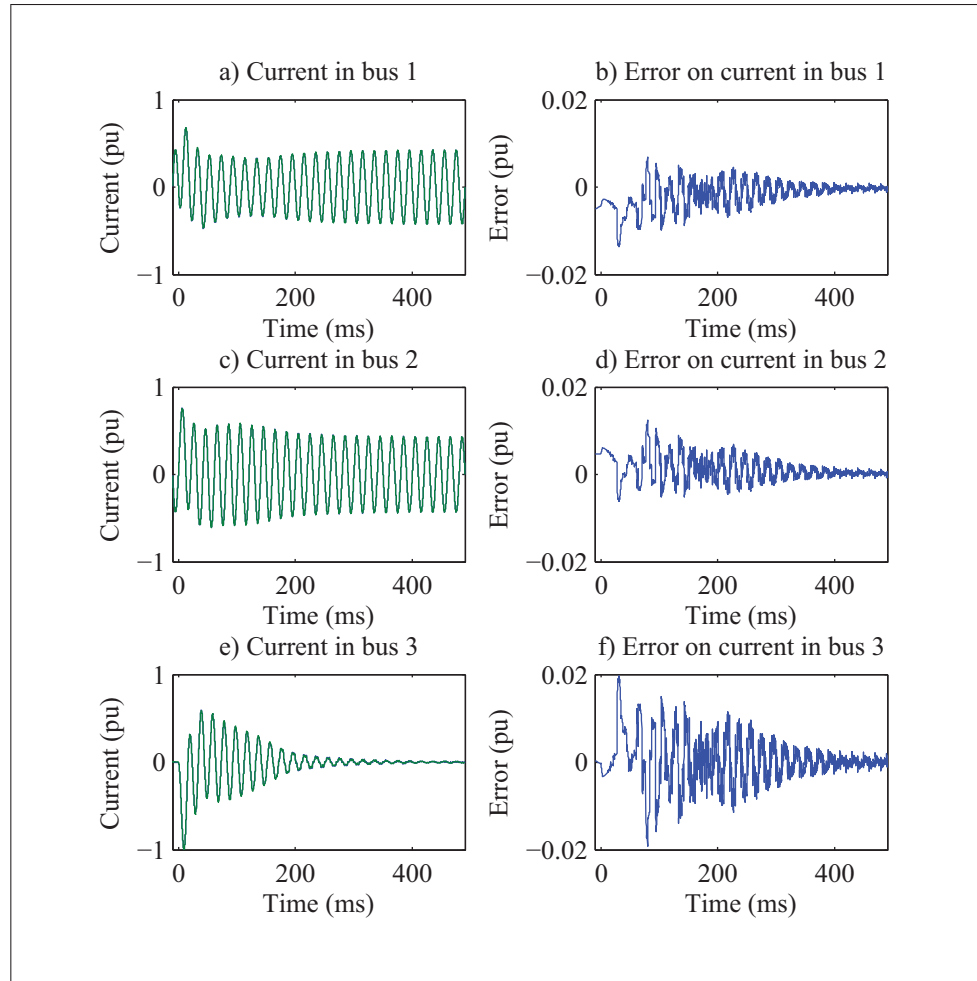


Figure 5.12 Inrush current in phase A during initialization  
a) bus 1 b) absolute error at bus 1 c) bus 2  
d) absolute error at bus 2 e) bus 3 f) absolute error at bus 3

Error remains below 0.02 pu, which is within acceptable margin for real-time simulation, and returns to 0.005 pu once the initialization is over. Furthermore, when observing the results closely, error is maximum when the current should be zero, as shown in figure 5.13.

Those oscillations are due to the multi-rate decoupling of the circuit. Reducing simulation time-step of the slow sub-system would also reduce error. Results for limb voltage are presented in figure 5.14.

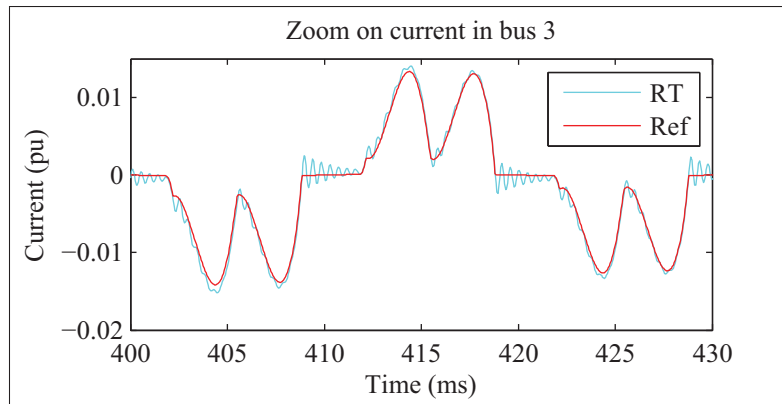


Figure 5.13 Simulation results with STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power

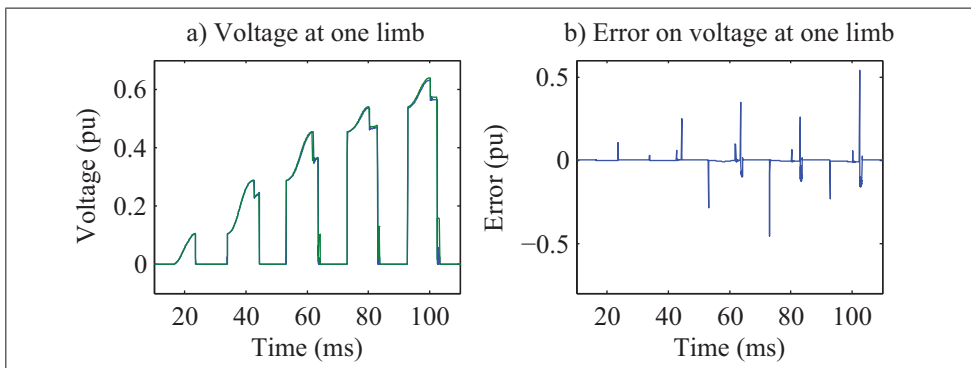


Figure 5.14 Simulation results with STATCOM for a) network voltages phase A b) network currents phase A c) network active power d) network reactive power

In figure 5.14 b), error might seem higher as it reaches 0.5 pu, but they only last 1 time-step, and can therefore be neglected. Error remains smaller than 0.005 pu the remaining time. Although small discrepancies were observed during initialization, results are very good. Errors were expected because of the numerical snubber and the decoupling applied to the circuit. Nonetheless, the model remained stable and accurate, while being executed in real-time.



### 5.6.3 DC glitch

In this last test, currents in the limbs become unbalance and the pulse are abruptly stopped. It should result in an overvoltage on the DC bus, as it was presented in (Grégoire *et al.*, 2015a). Figure 5.15 shows the real-time simulation results. In figure 5.15 a), virtual DC bus are

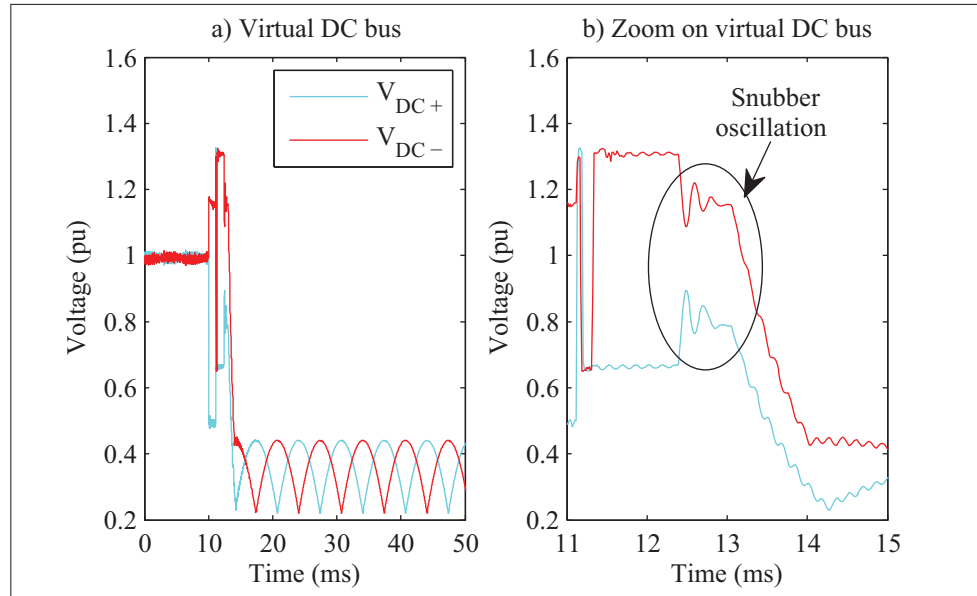


Figure 5.15 a) Virtual DC bus during DC glitch  
b) zoom on the effect of numerical snubbers

controlled during the first 10 ms, and then  $V_{DC+}$  has a major drop 0.5 pu, due to lost of SM. After 1 ms the pulse are blocked and the converter is no longer controlled; stored energy from the inductance is discharged through the anti-parallel diode. This results in an overvoltage lasting 20  $\mu$ s at 11.14 ms. During this process, the only noticeable difference is in the oscillation caused by the numerical snubbers, shown in figure 5.15 b). Results obtained are stable with marginal error bellow 0.05 pu while having an acceleration factor of at least a 100 if not a 1000 times.

## 5.7 Conclusion

In this chapter, numerical implementation of a MMC for real-time application was presented. The system to simulate was decoupled in three sub-systems, allowing the use of different simulation platforms, solvers, and sampling rates. Multi-rate stublines were introduced, to couple the slow sub-system on CPU with the fast sub-system on FPGA. SM were simulated using switching function implemented in parallel on FPGA. Through an numerical example, poles location analysis for multi-rate system was used. Choice of numerical snubber was validated before simulation using the poles of the system. The proposed model has been validate using a reference model with simulation for different cases study. The same method can also be applied to larger circuit or other MMC topologies.

## GENERAL CONCLUSION

This thesis presents a thorough implementation and validation of a modular multilevel converter simulated in real-time, and using multi-rate simulation. In 1, the challenge of simulated MMC was presented. Literature review on modeling and simulation of MMC has been completed. Constrains inherent to real-time simulation, advantages of different solvers, and the choice of simulation platforms were discussed. Finally, current model limitations and state of the art from commercially available solutions were introduced.

Due to the very high number of components of MMC topology, identifying the most appropriate decoupling approach, or sampling time was a laborious task. Chapter 2 narrowed down these requirements by identifying one of the most difficult phenomena to reproduce via simulation. The first contribution of this research work was the existence of an over voltage which can be observed for a particular operation of the MMC; this case was never reported in the literature before. It is identified and a mitigation technique is proposed. This was done thanks to the rigorous mathematical analysis allowing to forecast amplitude and duration of the over voltage. The mathematical analysis was then verified using simulation, and confirmed using an experimental setup. It showed that coupling among all six MMC arms was critical, and that sub-microsecond discretization was required in order to observe the phenomenon in Real-Time simulation.

In chapter 3, multi-solver simulation was introduced; this was the second contribution of this thesis. The proposed method allows the decoupling of very large state-space system, and tools to verify stability and accuracy using poles location analysis. The method was then demonstrated through a simple example, and obtained results were compared to the ones obtained using classical decoupling method. Using a very similar approach, a new method to achieve multi-rate simulation was proposed in chapter 4; which constitute the third contribution of the thesis. Therefore, by choosing the most appropriate sampling time for each state-variable of a system, computational burden can be reduced, and timing requirement for real-time simulation were met.

Finally, using the new methods that was proposed in this thesis, a multi-rate/real-time MMC was used as a STATCOM for a power system device. The complete circuit was divided into three sections, to ease the understanding of the different step required for validation of the model. Each section was ultimately regrouped in a single state-space equations system where, stability and accuracy was validated. When multi-rate real-time simulation results were compared to a reference model using a variable-step solver, relative error less or equal to 5% was obtained.

The different validation tools presented in this thesis can be applied to any converter topologies and not just for limited to the MMC family. They could also be used to enhance simulation speed even for offline simulation and could be part of future work. Furthermore, research on wideband or frequency dependent lines using multi-rate approach could enhance the proposed model. Finally, these multi-rate and decoupling methods could also be applied for power-hardware-in-the-loop (PHIL) application.

## APPENDIX I

This appendix contains the detailed equations from the system presented in chapter 5

$$A_1^c = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & 0 & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & 0 & a_{44} \end{bmatrix} \quad (\text{A I-1})$$

$$A_{13}^c = \begin{bmatrix} a_{17} & a_{18} & a_{19} & a_{110} & a_{111} & a_{112} \\ 0 & 0 & a_{29} & a_{210} & a_{211} & a_{212} \\ a_{37} & a_{38} & a_{39} & a_{310} & a_{311} & a_{312} \\ 0 & 0 & a_{49} & a_{410} & a_{411} & a_{412} \end{bmatrix} \quad (\text{A I-2})$$

$$A_{21}^c = \begin{bmatrix} a_{51} & a_{52} & a_{53} & a_{54} \\ 0 & a_{62} & 0 & a_{64} \end{bmatrix} \quad (\text{A I-3})$$

$$A_2^c = \begin{bmatrix} a_{55} & 0 \\ 0 & a_{66} \end{bmatrix} \quad (\text{A I-4})$$

$$A_{23}^c = \begin{bmatrix} a_{57} & a_{58} & a_{59} & a_{510} & a_{511} & a_{512} \\ 0 & 0 & a_{69} & a_{610} & a_{611} & a_{612} \end{bmatrix} \quad (\text{A I-5})$$

Equation (A I-6) gives the elements of the matrix of (A I-1).

$$\begin{aligned}
 & \text{denom} = 8(L1 + 2L3 + L4)(2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 2 \cdot L2 \cdot L3 + L2 \cdot L4) \\
 a_{11} = & - \frac{(32 \cdot R1 \cdot L3^2 + 8 \cdot R1 \cdot L4^2 + 16 \cdot L1 \cdot L2 \cdot R1 + 16 \cdot L1 \cdot L3 \cdot R1 + 16 \cdot L1 \cdot L2 \cdot R3 + 8 \cdot L1 \cdot L4 \cdot R1 + 32 \cdot L2 \cdot L3 \cdot R1)}{\text{denom}} \\
 & - \frac{(8 \cdot L1 \cdot L2 \cdot R4 + 16 \cdot L2 \cdot L4 \cdot R1 + 32 \cdot L2 \cdot L3 \cdot R3 + 32 \cdot L3 \cdot L4 \cdot R1 + 16 \cdot L2 \cdot L3 \cdot R4 + 16 \cdot L2 \cdot L4 \cdot R3 + 8 \cdot L2 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{12} = & - \frac{(2 \cdot R1 \cdot L4^2 - 8 \cdot R4 \cdot L3^2 + 4 \cdot R3 \cdot L4^2 - 4 \cdot L1 \cdot L2 \cdot R4 + 4 \cdot L2 \cdot L4 \cdot R1 - 4 \cdot L1 \cdot L3 \cdot R4 + 4 \cdot L3 \cdot L4 \cdot R1)}{\text{denom}} \\
 & - \frac{(-2 \cdot L1 \cdot L4 \cdot R4 - 8 \cdot L2 \cdot L3 \cdot R4 + 8 \cdot L2 \cdot L4 \cdot R3 + 8 \cdot L3 \cdot L4 \cdot R3 - 4 \cdot L3 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{13} = & - \frac{(16 \cdot L1 \cdot L2 \cdot R3 - 8 \cdot R2 \cdot L4^2 - 32 \cdot R2 \cdot L3^2 - 16 \cdot L1 \cdot L3 \cdot R2 + 8 \cdot L1 \cdot L2 \cdot R4 - 8 \cdot L1 \cdot L4 \cdot R2)}{\text{denom}} \\
 & - \frac{(32 \cdot L2 \cdot L3 \cdot R3 + 16 \cdot L2 \cdot L3 \cdot R4 + 16 \cdot L2 \cdot L4 \cdot R3 - 32 \cdot L3 \cdot L4 \cdot R2 + 8 \cdot L2 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{14} = & - \frac{(2 \cdot R1 \cdot L4^2 - 8 \cdot R4 \cdot L3^2 + 4 \cdot R3 \cdot L4^2 - 4 \cdot L1 \cdot L2 \cdot R4 + 4 \cdot L2 \cdot L4 \cdot R1 - 4 \cdot L1 \cdot L3 \cdot R4)}{\text{denom}} \\
 & - \frac{(4 \cdot L3 \cdot L4 \cdot R1 - 2 \cdot L1 \cdot L4 \cdot R4 - 8 \cdot L2 \cdot L3 \cdot R4 + 8 \cdot L2 \cdot L4 \cdot R3 + 8 \cdot L3 \cdot L4 \cdot R3 - 4 \cdot L3 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{22} = & - \frac{4 \cdot L1 \cdot R1 + 4 \cdot L1 \cdot R3 + 4 \cdot L3 \cdot R1 + 2 \cdot L1 \cdot R4 + 2 \cdot L4 \cdot R1}{4 \cdot L1(L1 + 2L3 + L4)} \\
 a_{24} = & - \frac{4 \cdot L1 \cdot R3 - 4 \cdot L3 \cdot R1 + 2 \cdot L1 \cdot R4 - 2 \cdot L4 \cdot R1}{4 \cdot L1(L1 + 2L3 + L4)} \\
 a_{31} = & - \frac{(16 \cdot R3 \cdot L1^2 - 32 \cdot R1 \cdot L3^2 + 8 \cdot R4 \cdot L1^2 - 8 \cdot R1 \cdot L4^2 - 16 \cdot L1 \cdot L3 \cdot R1 - 8 \cdot L1 \cdot L4 \cdot R1)}{\text{denom}} \\
 & - \frac{(32 \cdot L1 \cdot L3 \cdot R3 + 16 \cdot L1 \cdot L3 \cdot R4 + 16 \cdot L1 \cdot L4 \cdot R3 - 32 \cdot L3 \cdot L4 \cdot R1 + 8 \cdot L1 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{32} = & \frac{(2 \cdot R1 \cdot L4^2 - 8 \cdot R4 \cdot L3^2 + 4 \cdot R3 \cdot L4^2 - 4 \cdot L1 \cdot L3 \cdot R4)}{\text{denom}} \\
 & - \frac{(4 \cdot L3 \cdot L4 \cdot R1 - 2 \cdot L1 \cdot L4 \cdot R4 + 8 \cdot L3 \cdot L4 \cdot R3 - 4 \cdot L3 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{33} = & - \frac{(16 \cdot R2 \cdot L1^2 + 16 \cdot R3 \cdot L1^2 + 8 \cdot R4 \cdot L1^2 + 32 \cdot R2 \cdot L3^2 + 8 \cdot R2 \cdot L4^2 + 48 \cdot L1 \cdot L3 \cdot R2)}{\text{denom}} \\
 & - \frac{(32 \cdot L1 \cdot L3 \cdot R3 + 24 \cdot L1 \cdot L4 \cdot R2 + 16 \cdot L1 \cdot L3 \cdot R4 + 16 \cdot L1 \cdot L4 \cdot R3 + 8 \cdot L1 \cdot L4 \cdot R4 + 32 \cdot L3 \cdot L4 \cdot R2)}{\text{denom}} \\
 a_{34} = & \frac{(2 \cdot R1 \cdot L4^2 - 8 \cdot R4 \cdot L3^2 + 4 \cdot R3 \cdot L4^2 - 4 \cdot L1 \cdot L3 \cdot R4)}{\text{denom}} \\
 & - \frac{(4 \cdot L3 \cdot L4 \cdot R1 - 2 \cdot L1 \cdot L4 \cdot R4 + 8 \cdot L3 \cdot L4 \cdot R3 - 4 \cdot L3 \cdot L4 \cdot R4)}{\text{denom}} \\
 a_{42} = & - \frac{4 \cdot L1 \cdot R3 - 4 \cdot L3 \cdot R1 + 2 \cdot L1 \cdot R4 - 2 \cdot L4 \cdot R1}{4 \cdot L4(L1 + 2L3 + L4)} \\
 a_{44} = & - \frac{4 \cdot L1 \cdot R1 + 4 \cdot L1 \cdot R3 + 4 \cdot L3 \cdot R1 + 2 \cdot L1 \cdot R4 + 2 \cdot L4 \cdot R1}{4 \cdot L4 \cdot (L1 + 2L3 + L4)}
 \end{aligned}
 \tag{A I-6}$$

Equation (A I-7) gives the elements of the matrix of (A I-2).

$$denom = 8(L1 + 2 \cdot L3 + L4)(2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 2 \cdot L2 \cdot L3 + L2 \cdot L4)$$

$$a_{17} = \frac{12 \cdot L1 \cdot L2 + 24 \cdot L2 \cdot L3 + 12 \cdot L2 \cdot L4}{denom}$$

$$a_{18} = \frac{4 \cdot L1 \cdot L2 + 8 \cdot L2 \cdot L3 + 4 \cdot L2 \cdot L4}{denom}$$

$$a_{19} = -\frac{12 \cdot L3^2 + 6 \cdot L1 \cdot L2 + 6 \cdot L1 \cdot L3 + 3 \cdot L1 \cdot L4 + 12 \cdot L2 \cdot L3 + 6 \cdot L3 \cdot L4}{denom}$$

$$a_{110} = -\frac{4 \cdot L3^2 + 2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 4 \cdot L2 \cdot L3 + 2 \cdot L3 \cdot L4}{denom}$$

$$a_{111} = \frac{12 \cdot L3^2 - 6 \cdot L1 \cdot L2 + 6 \cdot L1 \cdot L3 + 3 \cdot L1 \cdot L4 - 12 \cdot L2 \cdot L3 - 12 \cdot L2 \cdot L4 + 6 \cdot L3 \cdot L4}{denom}$$

$$a_{112} = \frac{4 \cdot L3^2 - 2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 - 4 \cdot L2 \cdot L3 - 4 \cdot L2 \cdot L4 + 2 \cdot L3 \cdot L4}{denom}$$

$$a_{29} = \frac{3}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{210} = \frac{1}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{211} = -\frac{3}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{212} = -\frac{1}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{37} = \frac{12 \cdot L1^2 + 24 \cdot L1 \cdot L3 + 12 \cdot L1 \cdot L4}{denom}$$

$$a_{38} = \frac{4 \cdot L1^2 + 8 \cdot L1 \cdot L3 + 4 \cdot L1 \cdot L4}{denom}$$

$$a_{39} = \frac{12 \cdot L3^2 + 6 \cdot L1 \cdot L3 + 3 \cdot L1 \cdot L4 + 6 \cdot L3 \cdot L4}{denom}$$

$$a_{310} = \frac{4 \cdot L3^2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 2 \cdot L3 \cdot L4}{denom}$$

$$a_{311} = -\frac{12 \cdot L1^2 + 30 \cdot L1 \cdot L3 + 15 \cdot L4 \cdot L1 + 12 \cdot L3^2 + 6 \cdot L4 \cdot L3}{denom}$$

$$a_{312} = -\frac{4 \cdot L1^2 + 10 \cdot L1 \cdot L3 + 5 \cdot L4 \cdot L1 + 4 \cdot L3^2 + 2 \cdot L4 \cdot L3}{denom}$$

$$a_{49} = \frac{3}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{410} = \frac{1}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{411} = -\frac{3}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{412} = -\frac{1}{4 \cdot (L1 + 2 \cdot L3 + L4)}$$

(A I-7)

Equation (A I-8) gives the elements of the matrix of (A I-3).

$$\begin{aligned}
 denom &= 8(L1 + 2 \cdot L3 + L4)(2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 2 \cdot L2 \cdot L3 + L2 \cdot L4) \\
 a_{51} &= \frac{8 \cdot L2 \cdot L4^2 \cdot R1 - 8 \cdot L1^2 \cdot L2 \cdot R4 - 16 \cdot L1 \cdot L3^2 \cdot R4 + 8 \cdot L1 \cdot L4^2 \cdot R3 - 8 \cdot L1^2 \cdot L3 \cdot R4 + 8 \cdot L1^2 \cdot L4 \cdot R3}{denom} \\
 &\quad - \frac{16 \cdot L2 \cdot L3^2 \cdot R4 + 8 \cdot L2 \cdot L4^2 \cdot R3 + 8 \cdot L1 \cdot L2 \cdot L4 \cdot R1 - 24 \cdot L1 \cdot L2 \cdot L3 \cdot R4 + 8 \cdot L1 \cdot L2 \cdot L4 \cdot R3 + 16 \cdot L2 \cdot L3 \cdot L4 \cdot R1}{denom} \\
 &\quad - \frac{8 \cdot L1 \cdot L2 \cdot L4 \cdot R4 + 16 \cdot L1 \cdot L3 \cdot L4 \cdot R3 - 8 \cdot L1 \cdot L3 \cdot L4 \cdot R4 + 16 \cdot L2 \cdot L3 \cdot L4 \cdot R3 - 8 \cdot L2 \cdot L3 \cdot L4 \cdot R4}{denom} \\
 a_{52} &= \frac{2 \cdot L2 \cdot L4^2 \cdot R1 + 4 \cdot L2 \cdot L4^2 \cdot R3 - 2 \cdot L1 \cdot L2 \cdot L4 \cdot R4 - 4 \cdot L2 \cdot L3 \cdot L4 \cdot R4}{denom} \\
 a_{53} &= \frac{8 \cdot L1 \cdot L4^2 \cdot R2 - 8 \cdot L1^2 \cdot L2 \cdot R4 + 8 \cdot L1^2 \cdot L4 \cdot R2 - 16 \cdot L1 \cdot L3^2 \cdot R4 + 8 \cdot L1 \cdot L4^2 \cdot R3 - 8 \cdot L1^2 \cdot L3 \cdot R4}{denom} \\
 &\quad + \frac{8 \cdot L1^2 \cdot L4 \cdot R3 - 16 \cdot L2 \cdot L3^2 \cdot R4 + 8 \cdot L2 \cdot L4^2 \cdot R3 - 24 \cdot L1 \cdot L2 \cdot L3 \cdot R4 + 8 \cdot L1 \cdot L2 \cdot L4 \cdot R3 + 16 \cdot L1 \cdot L3 \cdot L4 \cdot R2}{denom} \\
 &\quad - \frac{8 \cdot L1 \cdot L2 \cdot L4 \cdot R4 + 16 \cdot L1 \cdot L3 \cdot L4 \cdot R3 - 8 \cdot L1 \cdot L3 \cdot L4 \cdot R4 + 16 \cdot L2 \cdot L3 \cdot L4 \cdot R3 - 8 \cdot L2 \cdot L3 \cdot L4 \cdot R4}{denom} \\
 a_{54} &= - \frac{2 \cdot L2 \cdot L4^2 \cdot R1 + 4 \cdot L2 \cdot L4^2 \cdot R3 - 2 \cdot L1 \cdot L2 \cdot L4 \cdot R4 - 4 \cdot L2 \cdot L3 \cdot L4 \cdot R4}{denom} \\
 a_{62} &= \frac{-2 \cdot L1 \cdot R4 + 2 \cdot L4 \cdot R1 - 4 \cdot L3 \cdot R4 + 4 \cdot L4 \cdot R3}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)} \\
 a_{64} &= \frac{-2 \cdot L1 \cdot R4 + 2 \cdot L4 \cdot R1 - 4 \cdot L3 \cdot R4 + 4 \cdot L4 \cdot R3}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}
 \end{aligned} \tag{A I-8}$$

Equation (A I-9) gives the elements of the matrix of (A I-4).

$$\begin{aligned}
 a_{55} &= \frac{-2 \cdot L1 \cdot R4 + 2 \cdot L4 \cdot R1 - 4 \cdot L3 \cdot R4 + 4 \cdot L4 \cdot R3}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)} \\
 a_{66} &= \frac{-4 \cdot L1 \cdot R4 - 8 \cdot L3 \cdot R4 - 4 \cdot L4 \cdot R4}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}
 \end{aligned} \tag{A I-9}$$



Equation (A I-10) gives the elements of the matrix of (A I-5).

$$denom = 8(L1 + 2 \cdot L3 + L4)(2 \cdot L1 \cdot L2 + 2 \cdot L1 \cdot L3 + L1 \cdot L4 + 2 \cdot L2 \cdot L3 + L2 \cdot L4)$$

$$a_{57} = -\frac{4 \cdot L1^2 \cdot L3 + 8 \cdot L1^2 \cdot L4 + 4 \cdot L2 \cdot L1^2 + 8 \cdot L1 \cdot L3^2 + 20 \cdot L1 \cdot L3 \cdot L4 + 12 \cdot L2 \cdot L1 \cdot L3}{denom}$$

$$+ \frac{8 \cdot L1 \cdot L4^2 + 12 \cdot L2 \cdot L1 \cdot L4 + 8 \cdot L2 \cdot L3^2 + 20 \cdot L2 \cdot L3 \cdot L4 + 8 \cdot L2 \cdot L4^2}{denom}$$

$$a_{58} = -\frac{4 \cdot L1^2 \cdot L3 + 4 \cdot L1^2 \cdot L4 + 4 \cdot L2 \cdot L1^2 + 8 \cdot L1 \cdot L3^2 + 12 \cdot L1 \cdot L3 \cdot L4 + 12 \cdot L2 \cdot L1 \cdot L3}{denom}$$

$$+ \frac{4 \cdot L1 \cdot L4^2 + 8 \cdot L2 \cdot L1 \cdot L4 + 8 \cdot L2 \cdot L3^2 + 12 \cdot L2 \cdot L3 \cdot L4 + 4 \cdot L2 \cdot L4^2}{denom}$$

$$a_{59} = \frac{3 \cdot L1 \cdot L2 \cdot L4 + 6 \cdot L2 \cdot L3 \cdot L4}{denom}$$

$$a_{510} = \frac{L1 \cdot L2 \cdot L4 + 2 \cdot L2 \cdot L3 \cdot L4}{denom}$$

$$a_{511} = \frac{4 \cdot L1^2 \cdot L3 + 8 \cdot L1^2 \cdot L4 + 4 \cdot L2 \cdot L1^2 + 8 \cdot L1 \cdot L3^2 + 20 \cdot L1 \cdot L3 \cdot L4 + 12 \cdot L2 \cdot L1 \cdot L3}{denom}$$

$$+ \frac{8 \cdot L1 \cdot L4^2 + 9 \cdot L2 \cdot L1 \cdot L4 + 8 \cdot L2 \cdot L3^2 + 14 \cdot L2 \cdot L3 \cdot L4 + 8 \cdot L2 \cdot L4^2}{denom}$$

$$a_{512} = \frac{4 \cdot L1^2 \cdot L3 + 4 \cdot L1^2 \cdot L4 + 4 \cdot L2 \cdot L1^2 + 8 \cdot L1 \cdot L3^2 + 12 \cdot L1 \cdot L3 \cdot L4 + 12 \cdot L2 \cdot L1 \cdot L3}{denom}$$

$$+ \frac{4 \cdot L1 \cdot L4^2 + 7 \cdot L2 \cdot L1 \cdot L4 + 8 \cdot L2 \cdot L3^2 + 10 \cdot L2 \cdot L3 \cdot L4 + 4 \cdot L2 \cdot L4^2}{denom}$$

$$a_{69} = \frac{-L1 - 2 \cdot L3 - 4 \cdot L4}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{610} = \frac{-L1 - 2 \cdot L3 - 2 \cdot L4}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{611} = \frac{-L1 - 2 \cdot L3 - 4 \cdot L4}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}$$

$$a_{612} = \frac{-L1 - 2 \cdot L3 - 2 \cdot L4}{4 \cdot L4 \cdot (L1 + 2 \cdot L3 + L4)}$$

(A I-10)

$$\begin{bmatrix} A_{31}^c & A_{32}^c \end{bmatrix} = \begin{bmatrix} -\frac{1}{C} & 0 & -\frac{1}{C} & 0 & \frac{1}{C} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{C} & 0 \\ 0 & -\frac{1}{C} & 0 & -\frac{1}{C} & 0 & \frac{1}{C} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{C} \\ \frac{1}{C} & \frac{1}{C} & \frac{1}{C} & \frac{1}{C} & -\frac{1}{C} & -\frac{1}{C} \\ 0 & 0 & 0 & 0 & -\frac{1}{C} & -\frac{1}{C} \end{bmatrix} \quad (\text{A I-11})$$

$$A_3^c = \begin{bmatrix} \frac{1}{R_{shunt}C} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{R_{shunt}C} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{R_{shunt}C} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{R_{shunt}C} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{R_{shunt}C} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{R_{shunt}C} \end{bmatrix} \quad (\text{A I-12})$$

## BIBLIOGRAPHY

- Abourida, S., C. Dufour, J. Bélanger, G. Murere, N. Léchevin, and B. Yu. 2002. "Real-time PC-based simulator of electric systems and drives". In *Applied Power Electronics Conference and Exposition, 2002. APEC 2002. Seventeenth Annual IEEE*. p. 433–438. IEEE.
- Abu-Rub, H., M. Malinowski, and K. Al-Haddad, 2014. *Power Electronics for Renewable Energy Systems, Transportation and Industrial Applications*.
- Ahmadi, A. A. 2008. "Non-monotonic Lyapunov functions for stability of nonlinear and switched systems: theory and computation". PhD thesis, Massachusetts Institute of Technology.
- Ahmed, N., L. Ängquist, S. Norrga, and H.-P. Nee. 2012. "Validation of the continuous model of the modular multilevel converter with blocking/deblocking capability". In *AC and DC Power Transmission (ACDC 2012), 10th IET International Conference on*. p. 1-6.
- Akagi, H. 2011. "Classification, terminology, and application of the modular multilevel cascade converter (MMCC)". *Power Electronics, IEEE Transactions on*, vol. 26, n° 11, p. 3119–3130.
- Allebrod, S., R. Hamerski, and R. Marquardt. 2008a. "New transformerless, scalable Modular Multilevel Converters for HVDC-transmission". In *Proc. IEEE Power Electronics Specialists Conf. PESC 2008*. p. 174–179.
- Allebrod, S., R. Hamerski, and R. Marquardt. 2008b. "New transformerless, scalable Modular Multilevel Converters for HVDC-transmission". In *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE*. p. 174-179.
- Angquist, L., A. Antonopoulos, D. Siemaszko, K. Ilves, M. Vasiladiotis, and H.-P. Nee. 2011. "Open-Loop Control of Modular Multilevel Converters Using Estimation of Stored Energy". *Industry Applications, IEEE Transactions on*, vol. 47, n° 6, p. 2516-2524.
- Antonopoulos, A., L. Ängquist, and H.-P. Nee. 2009. "On dynamics and voltage control of the Modular Multilevel Converter". In *Proc. 13th European Conf. Power Electronics and Applications EPE '09*. p. 1–10.
- Arrillaga, J., Y. H. Liu, and N. R. Watson, 2007. *Flexible power transmission: the HVDC options*.
- Baracos, P., G. Murere, C. Rabbath, and W. Jin. 2001. "Enabling PC-based HIL simulation for automotive applications". In *Electric Machines and Drives Conference, 2001. IEMDC 2001. IEEE International*. p. 721–729. IEEE.

- Bartel, A. and M. Günther. 2002. "A multirate W-method for electrical networks in state-space formulation". *Journal of Computational and Applied Mathematics*, vol. 147, n° 2, p. 411–425.
- Belanger, J., A. Yamane, A. Yen, S. Cense, and P.-Y. Robert. Nov 2013. "Validation of eHS FPGA reconfigurable low-latency electric and power electronic circuit solver". In *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*. p. 5418-5423.
- Benigni, A. and A. Monti. 2014a. "A Parallel Approach to Real-Time Simulation of Power Electronics Systems". *Power Electronics, IEEE Transactions on*, vol. PP, n° 99, p. 1-1.
- Benigni, A. and A. Monti. 2014b. "A Parallel Approach to Real-Time Simulation of Power Electronics Systems". *Power Electronics, IEEE Transactions on*, vol. PP, n° 99, p. 1-1.
- Benigni, A., A. Monti, and R. Dougal. June 2014a. "Latency-Based Approach to the Simulation of Large Power Electronics Systems". *Power Electronics, IEEE Transactions on*, vol. 29, n° 6, p. 3201-3213.
- Benigni, A., A. Monti, and R. Dougal. June 2014b. "Latency-Based Approach to the Simulation of Large Power Electronics Systems". *Power Electronics, IEEE Transactions on*, vol. 29, n° 6, p. 3201-3213.
- Blanchette, H., T. Ould-Bachir, and J.-P. David. Dec 2012. "A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters". *Industrial Electronics, IEEE Transactions on*, vol. 59, n° 12, p. 4555-4567.
- Butcher, J. C., 1987. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*.
- Callavik, M., A. Blomberg, J. Häfner, and B. Jacobson. 2012. "The Hybrid HVDC Breaker-An innovation breakthrough enabling reliable HVDC grids". *ABB Grid Systems, Technical paper Nov*.
- Debnath, S., J. Qin, B. Bahrani, M. Saeedifard, and P. Barbosa. Jan 2015. "Operation, Control, and Applications of the Modular Multilevel Converter: A Review". *Power Electronics, IEEE Transactions on*, vol. 30, n° 1, p. 37-53.
- Dommel, H. W. 1969. "Digital Computer Solution of Electromagnetic Transients in Single-and Multiphase". *Networks IEEE*, vol. 88, n° 4, p. 388–399.
- Dufour, C. and J. Belanger. Sept 2004. "A PC-Based Real-Time Parallel Simulator of Electric Systems and Drives". In *Parallel Computing in Electrical Engineering, 2004. PARELEC 2004. International Conference on*. p. 105-113.
- Dufour, C., J. Mahseredjian, J. Bélanger, and J. Naredo. Nov 2010. "An Advanced Real-Time Electro-Magnetic Simulator for power systems with a simultaneous state-space nodal

- solver". In *Transmission and Distribution Conference and Exposition: Latin America (TD-LA), 2010 IEEE/PES*. p. 349-358.
- Dufour, C., S. Cense, T. O. Bachir, L. Grégoire, and J. Bélanger. 2012a. "General-purpose reconfigurable low-latency electric circuit and drive solver on FPGA". In *38th Annual Conference of the IEEE Industrial Electronics Society (IECON-2012), Montréal, Canada*.
- Dufour, C., J. Bélanger, and S. Abourida. 2003a. "Accurate simulation of a 6-pulse inverter with real-time event compensation in ARTEMiS". *Mathematics and Computers in Simulation*, vol. 63, n° 3, p. 161-172.
- Dufour, C., J. Bélanger, and S. Abourida. 2003b. "Accurate simulation of a 6-pulse inverter with real-time event compensation in ARTEMIS<sup>TM</sup>". *Mathematics and Computers in Simulation*, vol. 63, n° 3, p. 161-172.
- Dufour, C., J. Bélanger, T. Ishikawa, and K. Uemura. 2005. "Advances in Real-Time Simulation of Fuel Cell Hybrid Electric Vehicles". In *Proceedings of 21st Electric Vehicle Symposium (EVS-21), Monte Carlo, Monaco*.
- Dufour, C., J. Mahseredjian, and J. Bélanger. 2011a. "A combined state-space nodal method for the simulation of power system transients". *Power Delivery, IEEE Transactions on*, vol. 26, n° 2, p. 928-935.
- Dufour, C., J. Mahseredjian, and J. Bélanger. 2011b. "A combined state-space nodal method for the simulation of power system transients". *Power Delivery, IEEE Transactions on*, vol. 26, n° 2, p. 928-935.
- Dufour, C., S. Cense, T. Ould-Bachir, L.-A. Grégoire, and J. Bélanger. 2012b. "General-purpose reconfigurable low-latency electric circuit and motor drive solver on FPGA". *J. IECON*, p. 3073-3081.
- Dufour, C., S. Cense, and J. Belanger. 2013. "FPGA-based Switched Reluctance Motor Drive and DC-DC converter models for high-bandwidth HIL real-time simulator". In *Power Electronics and Applications (EPE), 2013 15th European Conference on*. p. 1-8. IEEE.
- Erickson, R. W. and D. Maksimovic, 2001. *Fundamentals of power electronics*.
- Ferreira, J. 2013. "The Multilevel Modular DC Converter". *Power Electronics, IEEE Transactions on*, vol. 28, n° 10, p. 4460-4465.
- Gagnon, R., M. Fecteau, P. Prud'Homme, E. Lemieux, G. Turmel, D. Paré, and F. Duong. 2012. "Hydro-Québec strategy to evaluate electrical transients following wind power plant integration in the Gaspésie transmission system". *Sustainable Energy, IEEE Transactions on*, vol. 3, n° 4, p. 880-889.
- Gautschi, W. 1997. "Numerical Analysis: An Introduction Birkhauser". *Barton, Mass, USA*.

- Gear, C. W. and D. Wells. 1984. "Multirate linear multistep methods". *BIT Numerical Mathematics*, vol. 24, n° 4, p. 484–502.
- Glinka, M. and R. Marquardt. Jun. 2005. "A new AC/AC multilevel converter family". *IEEE Trans. on Ind. Electron.*, vol. 52, n° 3, p. 662–669.
- Gnanarathna, U., A. Gole, and R. Jayasinghe. 2011. "Efficient Modeling of Modular Multilevel HVDC Converters (MMC) on Electromagnetic Transient Simulation Programs". *Power Delivery, IEEE Transactions on*, vol. 26, n° 1, p. 316–324.
- Grégoire, L.-A., H. Blanchette, W. Li, A. Antonopoulos, L. Angquist, and K. Al-Haddad. May 2015a. "Modular Multilevel Converters Overvoltage Diagnosis and Remedial Strategy During Blocking Sequences". *Power Electronics, IEEE Transactions on*, vol. 30, n° 5, p. 2777–2785.
- Grégoire, L., J. Bélanger, and W. Li. 2011a. "FPGA-BASED REAL-TIME SIMULATION OF MULTILEVEL MODULAR CONVERTER HVDC SYSTEMS". In *Proc. ELECTRI-MACS 2011*.
- Grégoire, L. A., J. Bélanger, C. Dufour, H. Blanchette, K. Al-Haddad, et al. 2014a. "Real-Time Simulation of Modular Multilevel Converters (MMCs)". In *Power electronics for renewable energy systems, transportation and industrial applications*. p. 591–607. John Wiley & Sons, Ltd.
- Grégoire, L.-A., W. Li, J. Belanger, and L. Snider. 2011b. "Validation of a 60-Level Modular Multilevel Converter Model-Overview of Offline and Real-Time HIL Testing and Results". *IPST 2011, International Power System Transient Conference*.
- Grégoire, L.-A., J. Belanger, and W. Li. 2012. "FPGA-Based Real-Time Simulation of Modular Multilevel Converter HVDC Systems". *Journal of Energy and Power Engineering*, vol. 6, n° 7, p. 1119–1125.
- Grégoire, L.-A., H. Fortin-Blanchette, K. Al-Haddad, W. Li, and J. Bélanger. 2014b. "Real-Time Simulation of Modular Multilevel Converter on FPGA with sub-microsecond Time-Step". In *40th Annual Conference of the IEEE Industrial Electronics Society (IECON-2014)*.
- Grégoire, L.-A., M. Sleiman, H. Fortin-Blanchette, and K. Al-Haddad. March 2015b. "Numerical Stability of Multi-Rate System Using Lyapunov's Theorem: Applied to Real-Time Simulation". In *Industrial Technology (ICIT), 2015 IEEE International Conference on*.
- Gu, Y. L., X. M. Huang, P. Qiu, W. Hua, Z. R. Zhang, and Z. Xu. 2013. "Study of Overvoltage Protection and Insulation Coordination for MMC based HVDC". *Applied Mechanics and Materials*, vol. 347, p. 1812–1817.
- Guan, M. and Z. Xu. 2012. "Modeling and control of a modular multilevel converter-based HVDC system under unbalanced grid conditions". *Power Electronics, IEEE Transactions on*, vol. 27, n° 12, p. 4858–4867.

- Haddad, W. M. and V. Chellaboina, 2008. *Nonlinear dynamical systems and control: A Lyapunov-based approach*.
- Hagiwara, M. and H. Akagi. 2009. "Control and Experiment of Pulsewidth-Modulated Modular Multilevel Converters". *Power Electronics, IEEE Transactions on*, vol. 24, n° 7, p. 1737-1746.
- Hartley, T. T., G. O. Beale, and S. P. Chiacelli, 1994. *Digital Simulation of Dynamic Systems: A Control Theory Approach*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc.
- Hiller, M., D. Krug, R. Sommer, and S. Rohner. 2009. "A new highly modular medium voltage converter topology for industrial drive applications". In *Power Electronics and Applications, 2009. EPE'09. 13th European Conference on*. p. 1–10. IEEE.
- Hong, M., S. Horie, Y. Miura, T. Ise, and C. Dufour. 2009. "A method to stabilize a power hardware-in-the-loop simulation of inductor coupled systems". In *Int. Conf. on Power Systems Transients (IPST2009) in Kyoto, Japan*.
- Hudson, J., E. Hunter, and D. Wilson. 1966. "EHV-DC Simulator". *Power Apparatus and Systems, IEEE Transactions on*, , p. 1101–1107.
- Hui, S. and K. Fung. 1997. "Fast decoupled simulation of large power electronic systems using new two-port companion link models". *Power Electronics, IEEE Transactions on*, vol. 12, n° 3, p. 462–473.
- Ilves, K., A. Antonopoulos, S. Norrga, and H.-P. Nee. 2012a. "A New Modulation Method for the Modular Multilevel Converter Allowing Fundamental Switching Frequency". *Power Electronics, IEEE Transactions on*, vol. 27, n° 8, p. 3482-3494.
- Ilves, K., A. Antonopoulos, S. Norrga, and H.-P. Nee. 2012b. "Steady-State Analysis of Interaction Between Harmonic Components of Arm and Line Quantities of Modular Multilevel Converters". *Power Electronics, IEEE Transactions on*, vol. 27, n° 1, p. 57-68.
- Ilves, K., A. Antonopoulos, L. Harnefors, S. Norrga, and H.-P. Nee. 2012c. "Circulating current control in modular multilevel converters with fundamental switching frequency". In *Power Electronics and Motion Control Conference (IPEMC), 2012 7th International*. p. 249-256.
- Inaba, Y., S. Cense, T. Bachir, H. Yamashita, and C. Dufour. Aug 2011. "A dual high-speed PMSM motor drive emulator with Finite Element Analysis on FPGA chip with full fault testing capability". In *Power Electronics and Applications (EPE 2011), Proceedings of the 2011-14th European Conference on*. p. 1-10.
- J. DORN, H. HUANG, D. R. 2008. "A new Multilevel Voltage-Sourced Converter Topology for HVDC Applications". In *CIGRE 2008 Paris*.



- Jalili-Marandi, V., L.-F. Pak, and V. Dinavahi. Sept 2010. "Real-Time Simulation of Grid-Connected Wind Farms Using Physical Aggregation". *Industrial Electronics, IEEE Transactions on*, vol. 57, n° 9, p. 3010-3021.
- Johnson, D. H. 2003a. "Origins of the equivalent circuit concept: the current-source equivalent". *Proceedings of the IEEE*, vol. 91, n° 5, p. 817–821.
- Johnson, D. H. 2003b. "Origins of the equivalent circuit concept: the voltage-source equivalent". *Proceedings of the IEEE*, vol. 91, n° 4, p. 636–640.
- Kato, T., K. Inoue, Y. Kotani, and T. Ogawa. June 2013. "Generalization of parallel analysis for a power electronic system by circuit partitioning". In *Control and Modeling for Power Electronics (COMPEL), 2013 IEEE 14th Workshop on*. p. 1-7.
- Kato, T., K. Inoue, and T. Ogawa. June 2014. "Automatic circuit partitioning for parallel simulation of a power electronic system". In *Control and Modeling for Power Electronics (COMPEL), 2014 IEEE 15th Workshop on*. p. 1-7.
- Le-Huy, P., P. Giroux, and J. Soumagne. 2011. "Real-time simulation of modular multilevel converters for network integration studies". In *International Conference on Power Systems Transients, Delft, Netherlands June*. p. 14–17.
- Lesnicar, A. and R. Marquardt. 2003a. "An innovative modular multilevel converter topology suitable for a wide power range". In *Proc. IEEE Bologna Power Tech*.
- Lesnicar, A. and R. Marquardt. 2003b. "An innovative modular multilevel converter topology suitable for a wide power range". In *Power Tech Conference Proceedings, 2003 IEEE Bologna*. p. 6–pp. IEEE.
- Li, K. and C. Zhao. 2010. "New technologies of modular multilevel converter for VSC-HVDC application". In *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*. p. 1–4. IEEE.
- Li, W., L.-A. Gregoire, and J. Bélanger. 2011. "Control and Performance of a Modular Multilevel Converter System". *CIGRÉ Canada, Conference on Power Systems, Halifax*.
- Li, W., L.-A. Gregoire, S. Souvanlasy, and J. Belanger. Sept 2014a. "An FPGA-based real-time simulator for HIL testing of modular multilevel converter controller". In *Energy Conversion Congress and Exposition (ECCE), 2014 IEEE*. p. 2088-2094.
- Li, W., L.-A. Gregoire, P.-Y. Robert, S. Souvanlasy, and J. Belanger. 2014b. "Modular Multilevel Converter model implemented in FPGA for HIL test of industrial controllers". In *PES General Meeting| Conference & Exposition, 2014 IEEE*. p. 1–5. IEEE.
- Li, Z., P. Wang, H. Zhu, Z. Chu, and Y. Li. 2012. "An improved pulse width modulation method for chopper-cell-based modular multilevel converters". *Power Electronics, IEEE Transactions on*, vol. 27, n° 8, p. 3472–3481.



- Liu, H., P. C. Loh, and F. Blaabjerg. 2013. "Review of fault diagnosis and fault-tolerant control for modular multilevel converter of HVDC". In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. p. 1242–1247. IEEE.
- Mahseredjian, J., S. Dennerrière, L. Dubé, B. Khodabakhchian, and L. Gérin-Lajoie. 2007. "On a new approach for the simulation of transients in power systems". *Electric Power Systems Research*, vol. 77, n° 11, p. 1514–1520.
- Maksimovic, D., A. Stankovici, V. Thottuvelil, and G. C. Verghese. 2001. "Modeling and simulation of power electronic converters". *Proceedings of the IEEE*, vol. 89, n° 6, p. 898-912.
- Marquardt, R. June 2010. "Modular Multilevel Converter: An universal concept for HVDC-Networks and extended DC-Bus-applications". In *Power Electronics Conference (IPEC), 2010 International*. p. 502-507.
- Marti, J. and L. Linares. 1994. "Real-time EMTP-based transients simulation". *Power Systems, IEEE Transactions on*, vol. 9, n° 3, p. 1309–1317.
- Matar, M., M. Abdel-Rahman, and A. Soliman. Sept 2004. "Developing an FPGA coprocessor for real-time simulation of power systems". In *Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on*. p. 791-794.
- McLaren, P., R. Kuffel, R. Wierckx, J. Giesbrecht, and L. Arendt. 1992. "A real time digital simulator for testing relays". *Power Delivery, IEEE Transactions on*, vol. 7, n° 1, p. 207–213.
- Moreira, F. A., J. R. Martí, L. Zanetta, and L. R. Linares. 2006. "Multirate simulations with simultaneous-solution using direct integration methods in a partitioned network environment". *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, n° 12, p. 2765–2778.
- Najm, F. N., 2010. *Circuit simulation*.
- Nilsson, J. and S. Riedel, 2008. *Electric Circuits*.
- OPAL-RT. 2013a. "HYPERSIM Power System Real-Time Digital Simulator". Online <http://www.opal-rt.com/new-product/hypersim-power-system-real-time-digital-simulator>. Accessed: 2015-05-11.
- OPAL-RT. 2013b. "eFPGAsim Power Electronic Real-Time Simulator". Online <http://www.opal-rt.com/new-product/efpgasim-power-electronic-real-time-simulator>. Accessed: 2015-05-11.
- OPAL-RT. 2013c. "OPAL-RT home page". Online <http://www.opal-rt.com/>. Accessed: 2015-05-11.

- OPAL-RT Technologies Inc. 2014. *OP4500 RT-LAB-RCP/HIL system user guide*. Available from <http://www.opal-rt.com>.
- Ourari, M., L.-A. Dessaint, and V. Q. Do. June 2007. "Integration of Dynamic Equivalents in Hypersim Power System Simulator". In *Power Engineering Society General Meeting, 2007. IEEE*. p. 1-6.
- Paquin, J.-N., J. Bélanger, L. A. Snider, C. Pirolli, and W. Li. 2009. "Monte-carlo study on a large-scale power system model in real-time using emegasim". In *Energy Conversion Congress and Exposition, 2009. ECCE 2009. IEEE*. p. 3194–3202. IEEE.
- Paré, D., G. Turmel, J. Soumagne, V. Do, S. Casoria, M. Bissonnette, B. Marcoux, and D. McNabb. 2003. "Validation tests of the hypersim digital real time simulator with a large AC-DC network". In *proceedings of the 2003 International Conference on Power System Transients*. p. 1–6.
- Pekarek, S., O. Wasynczuk, E. Walters, J. Jatskevich, C. Lucas, N. Wu, and P. Lamm. Feb 2004. "An efficient multirate Simulation technique for power-electronic-based systems". *Power Systems, IEEE Transactions on*, vol. 19, n° 1, p. 399-409.
- Peralta, J., H. Saad, S. Denetière, and J. Mahseredjian. 2012a. "Dynamic performance of average-value models for multi-terminal VSC-HVDC systems". In *Power and Energy Society General Meeting, 2012 IEEE*. p. 1–8. IEEE.
- Peralta, J., H. Saad, S. Denetière, J. Mahseredjian, and S. Nguefeu. 2012b. "Detailed and averaged models for a 401-level MMC–HVDC system". *Power Delivery, IEEE Transactions on*, vol. 27, n° 3, p. 1501–1508.
- Perez, M., J. Rodriguez, E. Fuentes, and F. Kammerer. 2012. "Predictive Control of AC-AC Modular Multilevel Converters". *Industrial Electronics, IEEE Transactions on*, vol. 59, n° 7, p. 2832-2839.
- Perez, M., D. Arancibia, S. Kouro, and J. Rodriguez. Nov 2013. "Modular multilevel converter with integrated storage for solar photovoltaic applications". In *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*. p. 6993-6998.
- Rajasekar, S. and R. Gupta. 2012. "Solar photovoltaic power conversion using modular multi-level converter". In *Engineering and Systems (SCES), 2012 Students Conference on*. p. 1–6. IEEE.
- Rice, J. R. 1960. "Split Runge-Kutta methods for simultaneous equations". *J. Res. Natl. Bur. Standards B*, vol. 64, p. 151–170.
- Rohner, S., S. Bernet, M. Hiller, and R. Sommer. Aug. 2010. "Modulation, Losses, and Semiconductor Requirements of Modular Multilevel Converters". *Industrial Electronics, IEEE Transactions on*, vol. 57, n° 8, p. 2633–2642.
- RTDS. 2013. "RTDS home page". Online <https://www.rtds.com/>. Accessed: 2015-05-11.

- Saad, H., C. Dufour, J. Mahseredjian, S. Denetiere, and S. Nguefeu. 2013. "Real time simulation of MMCs using the state-space nodal approach".
- Saad, H., T. Ould-Bachir, J. Mahseredjian, C. Dufour, S. Denetiere, and S. Nguefeu. Jan 2015a. "Real-Time Simulation of MMCs Using CPU and FPGA". *Power Electronics, IEEE Transactions on*, vol. 30, n° 1, p. 259-267.
- Saad, H., T. Ould-Bachir, J. Mahseredjian, C. Dufour, S. Denetiere, and S. Nguefeu. 2015b. "Real-time simulation of MMCs using CPU and FPGA". *Power Electronics, IEEE Transactions on*, vol. 30, n° 1, p. 259–267.
- Saeedifard, M. and R. Iravani. 2010. "Dynamic Performance of a Modular Multilevel Back-to-Back HVDC System". *Power Delivery, IEEE Transactions on*, vol. 25, n° 4, p. 2903-2912.
- Schaller, R. R. 1997. "Moore's law: past, present and future". *Spectrum, IEEE*, vol. 34, n° 6, p. 52–59.
- Seny, B., J. Lambrechts, T. Toulorge, V. Legat, and J.-F. Remacle. 2014. "An efficient parallel implementation of explicit multirate Runge–Kutta schemes for discontinuous Galerkin computations". *Journal of Computational Physics*, vol. 256, p. 135–160.
- Shao, S., P. W. Wheeler, J. C. Clare, and A. J. Watson. 2013. "Fault detection for modular multilevel converters based on sliding mode observer". *IEEE transactions on power electronics*, vol. 28, n° 11, p. 4867–4872.
- Shi, J., W. Gou, H. Yuan, T. Zhao, and A. Huang. 2011. "Research on Voltage and Power Balance Control for Cascaded Modular Solid-State Transformer". *Power Electronics, IEEE Transactions on*, vol. 26, n° 4, p. 1154-1166.
- Shi, X., Z. Wang, B. Liu, Y. Liu, L. Tolbert, and F. Wang. 2014. "Characteristic Investigation and Control of Modular Multilevel Converter Based HVDC System under Single-Line-to-Ground Fault Conditions". *Power Electronics, IEEE Transactions on*, vol. PP, n° 99, p. 1-1.
- Shortt, D. and F. Lee. March 1983. "Improved Switching Converter Model Using Discrete and Averaging Techniques". *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-19, n° 2, p. 190-202.
- Slotine, J.-J. E., W. Li, et al., 1991. *Applied nonlinear control*, volume 60.
- Solas, E., G. Abad, J. Barrena, S. Aurtenetxea, A. Carcar, and L. Zajac. 2013a. "Modular Multilevel Converter With Different Submodule Concepts - Part I: Capacitor Voltage Balancing Method". *Industrial Electronics, IEEE Transactions on*, vol. 60, n° 10, p. 4525-4535.

- Solas, E., G. Abad, J. Barrena, S. Aurtenetxea, A. Carcar, and L. Zajac. 2013b. "Modular Multilevel Converter With Different Submodule Concepts - Part II: Experimental Validation and Comparison for HVDC Application". *Industrial Electronics, IEEE Transactions on*, vol. 60, n° 10, p. 4536-4545.
- Teodorescu, R., E. Eni, L. Mathe, and P. Rodriguez. 2013. "Modular multilevel converter control strategy with fault tolerance". In *International Conference on Renewable Energies and Power Quality*.
- Tomim, M., J. R. Martí, T. De Rybel, L. Wang, M. Yao, et al. 2010. "MATE network tearing techniques for multiprocessor solution of large power system networks". In *Power and Energy Society General Meeting, 2010 IEEE*. p. 1–6. IEEE.
- Tu, Q., Z. Xu, and L. Xu. 2011. "Reduced switching-frequency modulation and circulating current suppression for modular multilevel converters". *Power Delivery, IEEE Transactions on*, vol. 26, n° 3, p. 2009–2017.
- Twining, E. and D. G. Holmes. 2003. "Grid current regulation of a three-phase voltage source inverter with an LCL input filter". *Power Electronics, IEEE Transactions on*, vol. 18, n° 3, p. 888–895.
- Typhoon-HIL. 2013. "Typhoon-HIL home page". Online <http://www.typhoon-hil.com/>. Accessed: 2015-05-11.
- Vahedi, H., K. Al-Haddad, P.-A. Labbe, and S. Rahmani. 2014. "Cascaded multilevel inverter with multicarrier PWM technique and voltage balancing feature". In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*. p. 2155–2160. IEEE.
- Vajta, M. 2000. "Some remarks on Pade-approximations". In *Proc. 3rd TEMPUS—INTCOM Symp. Intelligent Systems in Control and Measurement*. p. 53–58.
- Wang, J., R. Burgos, and D. Boroyevich. Sept 2013. "A survey on the modular multilevel converters - Modeling, modulation and controls". In *Energy Conversion Congress and Exposition (ECCE), 2013 IEEE*. p. 3984-3991.
- Wang, Y., G. Delille, X. Guillaud, F. Colas, and B. François. 2010. "Real-time simulation: the missing link in the design process of advanced grid equipment". In *Power and Energy Society General Meeting, 2010 IEEE*. p. 1–8. IEEE.
- Wanner, G. and E. Hairer. 1991. "Solving ordinary differential equations II".
- Watson, N. and J. Arrillaga, 2003. *Power systems electromagnetic transients simulation*. Number 39.
- Yan, X., S. Difeng, and Q. Shi. Oct 2013. "Protection coordination of meshed MMC-MTDC transmission systems under DC faults". In *TENCON 2013 - 2013 IEEE Region 10 Conference (31194)*. p. 1-5.

- Yan, Z., H. Xue-hao, T. Guang-fu, and H. Zhi-yuan. 2010. "A study on MMC model and its current control strategies". In *Power Electronics for Distributed Generation Systems (PEDG), 2010 2nd IEEE International Symposium on*. p. 259–264. IEEE.
- Zeigler, B. P., H. Praehofer, and T. G. Kim, 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*.
- Zhang, M., L. Huang, W. Yao, and Z. Lu. 2013. "Circulating Harmonic Current Elimination of a CPS-PWM Based Modular Multilevel Converter with Plug-In Repetitive Controller". *Power Electronics, IEEE Transactions on*, vol. PP, n° 99, p. 1-1.
- Zhang, M., L. Huang, W. Yao, and Z. Lu. April 2014. "Circulating Harmonic Current Elimination of a CPS-PWM-Based Modular Multilevel Converter With a Plug-In Repetitive Controller". *Power Electronics, IEEE Transactions on*, vol. 29, n° 4, p. 2083-2097.
- Zhang, X. Y., Z. J. Wu, S. P. Hao, and K. Xu. 2012. "A Study on the New Grid Integration Solutions of Offshore Wind Farms". *Advanced Materials Research*, vol. 383, p. 3610–3616.