May 2020

A Hardware-in-the-Loop Platform for DC Protection

Mark Vygoder
University of Wisconsin-Milwaukee

A HARDWARE-IN-THE-LOOP PLATFORM FOR DC PROTECTION

by

Mark Vygoder

A Thesis Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Master of Science
in Engineering

at

The University of Wisconsin - Milwaukee

May 2020

ABSTRACT

A HARDWARE-IN-THE-LOOP PLATFORM FOR DC PROTECTION

by

Mark Vygoder

The University of Wisconsin - Milwaukee, 2020
Under the Supervision of Professor Robert M. Cuzner

With the proliferation of power electronics, dc-based power distribution systems can be realized; however, dc electrical protection remains a significant barrier to mass implementation dc power distribution. Controller Hardware-in-the-loop (CHiL) simulation enables moving up technology readiness levels (TRL) quickly. This work presents an end-to-end solution for dc protection CHiL for early design exploration and verification for dc protection, allowing for the rapid development of dc protection schemes for both Line-to-Line (LL) and Line-to-Ground (LG) faults. The approach combines using Latency Based Linear Multistep Compound (LB-LMC), a real-time simulation method for power electronic, and National Instruments (NI) FPGA hardware to enable dc protection design with CHiL. A case study is performed for a 1.5 MW Voltage Source Rectifier (VSR) under LL and LG faults in an ungrounded system. The deficiency in real-time simulation resolution of Commercial-off-the-Shelf (COTS) for dc fault transients is shown, and addressed by using LB-LMC RT solver inside NI FPGA hardware to achieve 50 ns resolution of dc fault transients.

Dedications

Some of the two great challenges humanity faces in the 21st century include climate change, and the millions of people (if not billions) without electricity and billions of people without clean cooking and internet access. To ensure a sustainable future, humanity must achieve net-zero global emissions while also lifting the remaining civilization out of poverty. To achieve sustainable energy, the solution seems pretty obvious: mass implementation of renewable energy source like PV and wind with a large amount of energy storage, both batteries and supercapacitors to support the interments of renewables, and nuclear fission for a baseline amount of power generation during the night and extended periods of no sun. Then using HVdc or high-temperature superconducting cables at the transmission level, MVdc at the distribution/industrial level, and LVdc at the residential/load level, with energy storage and renewable distributed at each voltage level. The dessert, in areas such as the southwest U.S and the Sahara in Africa, should be filled with PV farms, and then distributed with HVdc throughout the continent. This solution is sufficient until fusion made economically viable and can produce more power than it consumes, but this is still several decades away. Furthermore, EVs will replace all land-based transportation, this is just a matter of time, but hopefully sooner rather than later.

Most likely developed countries will be hybrid ac/dc systems due to the currently existing power distribution system, while developing nations will go straight to dc systems. The latter point is similar to when developing nations went straight to cell phones, skipping the landlines entirely, as a better technology existed.

All the above is enabled by the proliferation of WBG power electronic devices and power converters. To achieve the goals mentioned above will require the implementation of renewables, energy storage, and power electronics on an unprecedented scale. To do this, cabinet-level/system-level design tools are needed to develop optimized power electronics-based distribution systems. These tools should be able to go from a single die to an optimized multichip power module to an optimized power converter to an optimized cabinet of power electronic converters, to an entire optimized power electronics distribution system such as an entire microgrid or electrified shipboard

system.

Dc protection remains one of the barriers to the widespread implementation of dc power distribution system. My hope for this work was to create a real-time simulation sandbox environment to develop and assess dc protection and coordination schemes to help move up technology readiness levels quickly and rapid design iteration.

To this end, I would like to thank my advisor, Prof. Robert Cuzner for the opportunity to work on cutting edge research, solving challenges associated with power electronic-based distribution system, the opportunity to learn through both direct instruction and osmosis, as well as the various opportunity to travel, present our research both within the U.S and internationally, and collaborate with world-class colleagues. I would like to thank my family on the east coast for their support. I would like to thank wonderful mother for her support throughout my university education, and particularly in the last few years. I would like to thank Big Alex for this handiness around our condo, and Art Miller for his mentorship several years ago in photography. I would like to tip my hat to my fellow lab mates for being in the trenches with me, and in particular, Jake Gudex for his help with the controls in LabVIEW FPGA and his help with writing/editing. I would also like to thank my colleagues at the University of South Carolina, Matthew Milton, and Prof. Andrea Benigni (now with FZ-Juelich), for their collaboration of the past two years to help make this work possible. Without their inputs and efforts, much of this work, if not all, would have not been possible, and I have learned a lot through this collaboration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

+LG	Positive Line-to-Ground
-LG	Negative Line-to-Ground
ADC	Analog-to-Digital Converter
ADIO	Analog and Digital Input/Output
BJT	Bipolar Junction Transistor
BRAM	Block Random Access Memory
CLI	Command Line Interface
CLIP	Component Level Intellectual Property
CM	Common-Mode
CO_2	Carbon Dioxide
COTS	Commercial-off-the-shelf
cRIO	CompactRIO
DAB	Dual Active Bridge
DAC	Digital-to-Analog Converter
dcPEDS	dc Power Electronic-based Distribution Systems
DM	Differential-Mode
DMA	Direct Access Memory
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processor
EMI	Electromagnetic Interference
ESS	Energy Storage Systems
FIFO	First-In-First-Out
GUIs	Graphical User Interfaces
HDL	Hardware Description Language
HLS	High-Level Synthesis
HVdc	High Voltage dc
IGBT	Isolated Gate Bipolar Transistor

LB-LMC	Latency-Based Linear Multi-step Compound
LG	Line-to-Ground
LL	Line-to-Line
LUTs	Lookup Tables
LV	Low Voltage
LVdc	Low Voltage dc
MEA	More Electric Aircraft
MMC	Modular Multilevel Converter
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MVac	Medium Voltage ac
MVdc	Medium Voltage dc
NI	National Instruments
NPC	Neutral Point Clamped
OS	Operating System
PECs	Power Electronic Converters
ppm	parts per million
PV	Photo Voltaic
RC	Resistive Companion
SCR	Silicon Controlled Rectifier
SCTL	Single-Cycle Timed Loop
SFP+	Small Form-Factor Pluggable Transceiver
SiC	Silicon Carbide
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SSCBs	Solid State Circuit Breakers
TDMS	Technical Data Management Streaming
TRL	Technology Readiness Levels
VI	Virtual Instruments
VSC	Voltage Source Converter
VSI	Voltage Source Inverter
VSR	Voltage Source Rectifier

ACKNOWLEDGEMENTS

This work was made possible by support from GRAPES (GRid conected Advanced Power Electronics Systems) and the National Science Foundation (NSF). This material is based upon work supported by the National Science Foundation under Grant No. 1650470. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of this institution.

Chapter 1

Introduction

Human-made greenhouse gasses are the primary contributors to climate change [1–3, 9]. When accounting for variations from the sun's 11-year cycle, the average temperature of the earth has increased by 1°C in the last 40 years, as seen in Fig. 1-1. Fig. 1-2 shows the increase in Carbon Dioxide (CO_2) emission per year and shows the commutative amount of CO_2 in the atmosphere. Fig. 1-3a shows the impact of different scenarios of average parts per million (ppm) of CO_2 and annual CO_2 emissions. For example, if no action is taken to reduce emissions, then the average surface temperature would increase by 4°C [3], and the average CO_2 in the atmosphere would be about 1000 ppm by 2100. If significant action is taken, the average CO_2 ppm can be kept to 430-480 ppm; however, this requires reaching annual emissions of 0 or even negative emission using carbon capture. For reference, today's atmospheric CO_2 content is 414 ppm. Fig. 1-3b shows the relationship between different levels of average atmospheric ppm of CO_2, and relative temperature change. The circles are color-coated to correspond with the different scenarios in Fig. 1-3a. To prevent the average temperature from increasing above 2°C, which corresponds to an average atmospheric CO_2 content of 480-530 ppm, yearly greenhouse gas must reduce to zero, or may even need to go negative.

However, it is not just sufficient to analyze greenhouse gas emissions with today energy demands. Currently, the International Energy Agency estimates there are 860 million people without access to electricity and 2.6 billion people with access to clean cooking facilities [10]. As these developing countries gain access to reliable electricity and clean methods of cooking, ideally, these countries can leverage the latest in technology and develop the electrical infrastructure in a sustainable way,

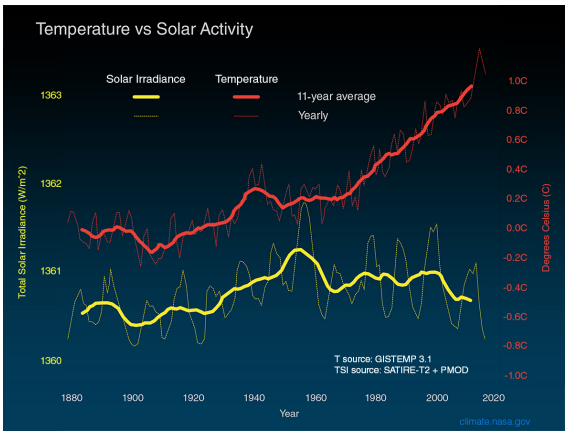


Figure 1-1: Average surface temperature and average solar irradiance between from 1880 to 2018 [1].

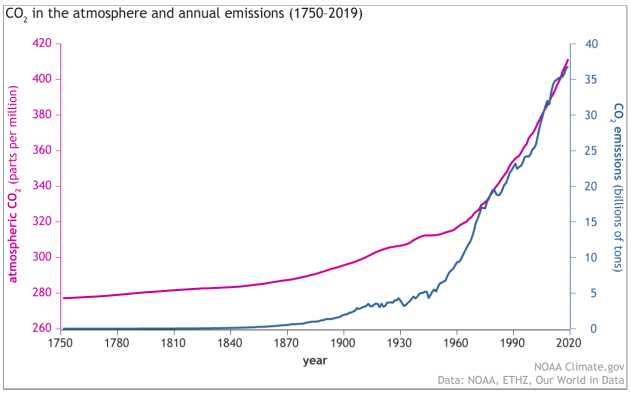


Figure 1-2: Global annual and cumulative CO_2 emissions [2].

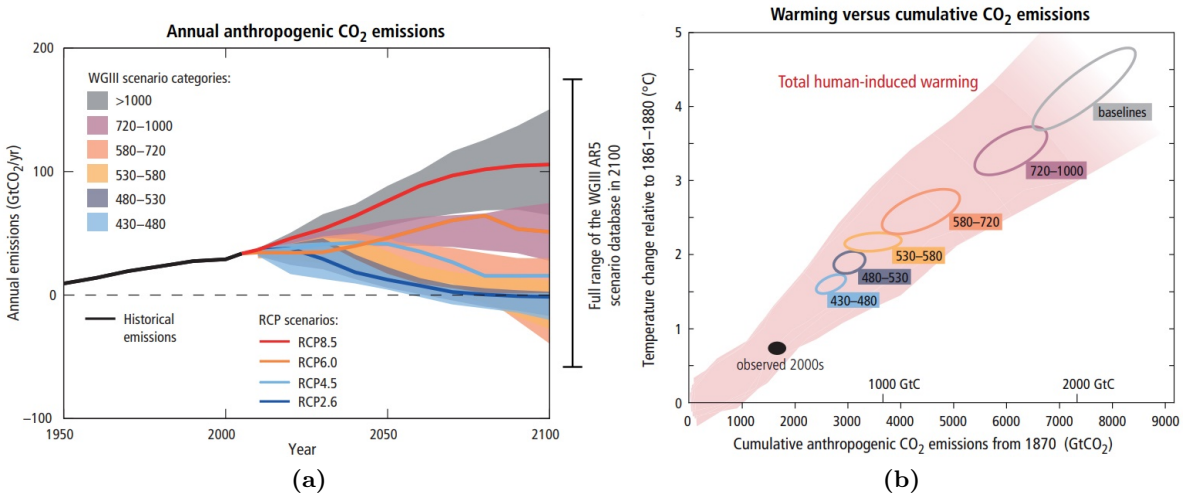


Figure 1-3: (a) Projected annual CO_2 emission based on different Representative Concentration Pathways (RCP) scenarios, and (b) relating scenarios and CO_2 ppm to temperature change [3].

such as microgrids [11–13].

Generally speaking, power electronics enable conversion of power in the four following forms: ac to ac, ac to dc, dc to dc, and dc to ac. Power electronics allows electrical sources to be different voltage level and shape than the load [14]. The debate between ac and dc distribution goes back to the 1890s between Nicola Tesla and Thomas Edison. Tesla being the proponent of ac, while Edison advocated for dc. Without power electronics, ac distribution was the clear choice from the viewpoint of being able to use reliable ac machinery for power generation and passive ac transformers for power distribution and changing voltage levels. It is fair to say that Tesla won the

battle against Edison with the installation of the world's first hydroelectric power plant in Niagara Falls with Westinghouse in 1895, which used ac generators and ac distribution.

Today, with switching power electronics, digital feedback control, low cost embedded processors and Field Programmable Gate Array (FPGA), advanced magnetic materials, and other advancements in the past 130 years, ac distribution may no longer be the better option. However, these technological advancements are relatively recent. Looking back in history, solid-state devices began to replace vacuum tubes in the 1950s with the development of the P-N junction by the research team at Bell Labs. Through the 1950s, the power diode, Bipolar Junction Transistor (BJT), and Silicon Controlled Rectifier (SCR) were developed and became commercially available. Around 1970, the power Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) became commercially available. However, it was not until the commercial availability of the Isolated Gate Bipolar Transistor (IGBT), when the field of power electronics took off. The IGBT combined the voltage-controlled gate properties of a MOSFET with the current-carrying capabilities of a BJT, which resulted in a device suitable for medium and high power applications and was much easier to control than a BJT. During this time, most of the world's power distribution was utilizing ac, as there was not a *viable*, i.e., not just technological feasible, but economical from a cost, size, and weight perspective, approach for dc distribution in most cases. However, with today's power electronics, dc distribution is not only viable but has many advantages over ac distribution. These advantages include:

1. Increased electrical efficiency. Photo Voltaic (PV) panels, Energy Storage Systems (ESS), such as lithium-ion batteries and supercapacitors, and many loads, such as LED lighting and computers, are naively dc. These can integrate into a common dc bus via dc-dc Power Electronic Converters (PECs), which are more efficient than dc-ac converters.

2. Improved power quality and more straightforward controls. A dc system is at 0 Hz and is not as susceptible to lower-order harmonics of the fundamental frequency like in ac systems.

3. Easy to parallel multiple sources. Dc systems do not have frequency and phase angle, which needs to be synchronized like in ac systems.

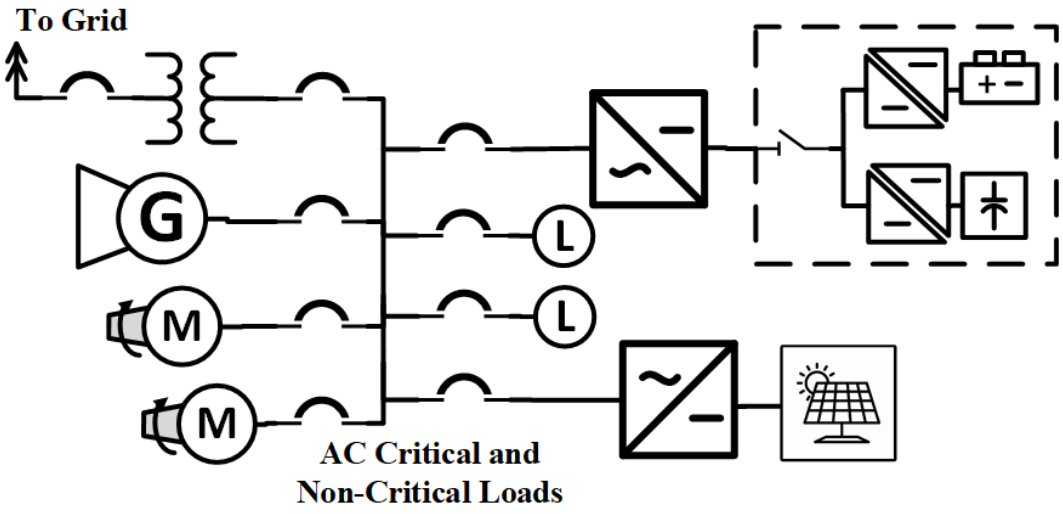
4. Interfacing ac generators or ac motor loads via power electronics decouples the ac source and load. For example, this enables both ac generators and ac motors to each operate at optimal

conditions, which reduces fuel consumption for the ac generator, and increases performance for the ac motor loads [15] [16].

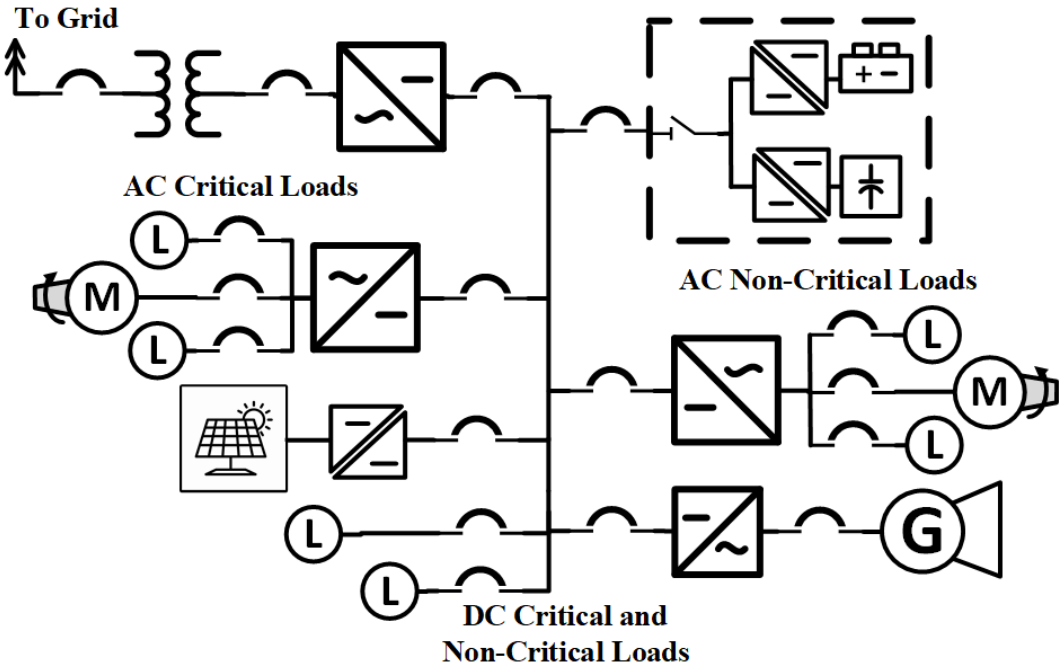
The advantages listed above do not mean dc systems are without disadvantages or challenges. For one, active and reactive power in ac systems, although perhaps more complex, allows for an additional degree of freedom in ac system controls. Another, and perhaps the main challenge, is that dc distributions systems are a *network* of PECs. In other words, with power electronic based interfaces to distributed energy resources, dc distribution systems are dc Power Electronic-based Distribution Systems (dcPEDS). The challenges of dcPEDS includes system stability [4], electrical protection [17], and creepage/clearance and insulation coordination standards for dc system. Electrical protection is the focus of this work.

Examples of dcPEDS for emerging Low Voltage dc (LVdc) and Medium Voltage dc (MVdc) applications include dc microgrids, shipboard electrification, and More Electric Aircraft (MEA). Fig. 1-4a shows a notional diagram with PV, ESS, diesel generator, ac motor loads, and resistive loads with an ac distribution, while Fig. 1-4b shows a notional diagram for a dc distribution system. Fig. 1-5a shows an electrified shipboard system with a conventional Medium Voltage ac (MVac) distribution for a cruise ship, and Fig. 1-4b shows a notional MVdc shipboard for a future US Navy destroyer. Fig. 1-6a shows an MEA with LVac distribution, and Fig. 1-4b shows an MVdc architecture. From looking at the dc implementations of the three applications listed above, one can see similarities between the architectures. They all have multiple sources and loads interfacing through power electronics, connecting to a common dc bus. The notional MVdc shipboard has two MVdc bus, one port and one starboard, for survivability purposes, but comparing one bus of the shipboard with MEA and dc building microgrid, they all look similar. Currently, the navy is pushing higher power density (MW/m^3) in shipboard applications. NASA is pushing specific power for MEA (kW/kg). For building microgrids, cost is one of the main driving factors. Regardless, all three are pushing for improvements, and since the architectures all look very similar, it is very likely improvements in one application carries over to another and to dcPEDS in general.

Protective system design is challenging for PEC-based systems in general, and specifically for dcPEDS, due to the lack of standards and experience-based design practices [18]. Microgrids, electrified ships, and MEA have short cable lengths, dc distribution, and, generally, meshed power



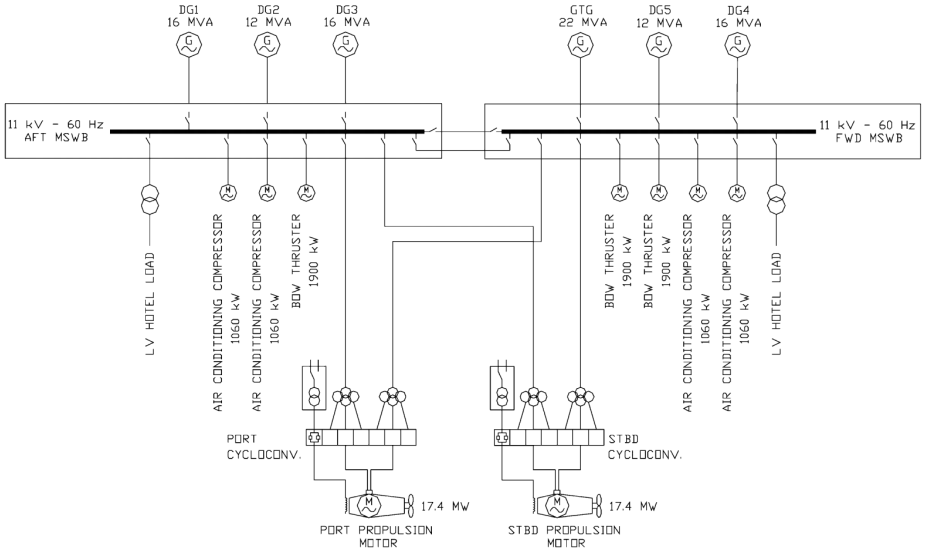
(a)



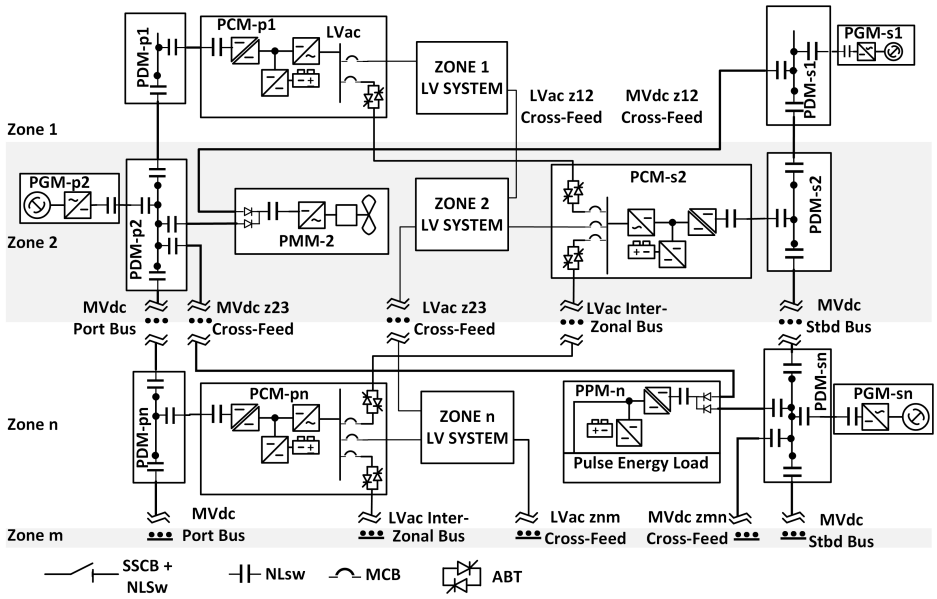
(b)

Figure 1-4: Notional building microgrid for (a) ac, and (b) dc distribution.

and energy delivery. In such systems, fault behavior at its extreme, i.e., low impedance sudden inception Line-to-Line (LL) and Line-to-Ground (LG) faults, is characterized by cable inductance and the filter capacitance of the connected PECs [19]. There is also a race condition between the time for protective circuits to respond and isolate a fault, and the actuation of internal unit-protection of PECs. These factors result in the demand for extremely fast time responses of



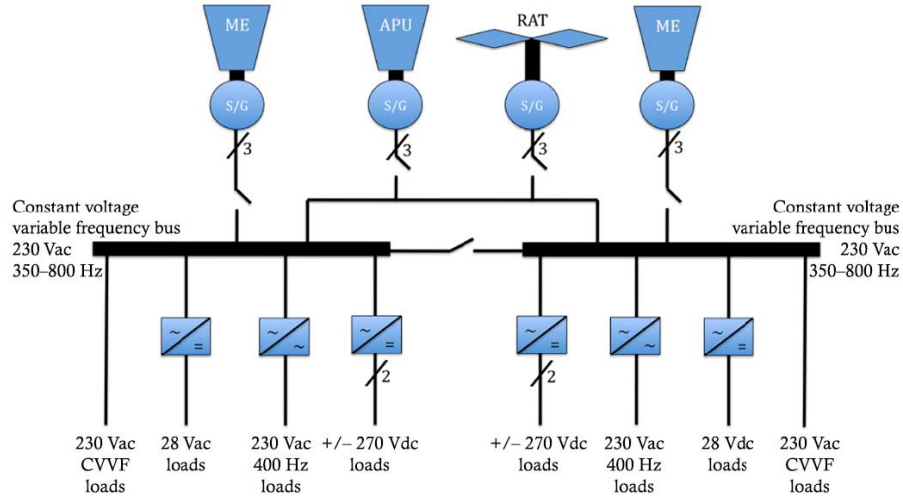
(a)



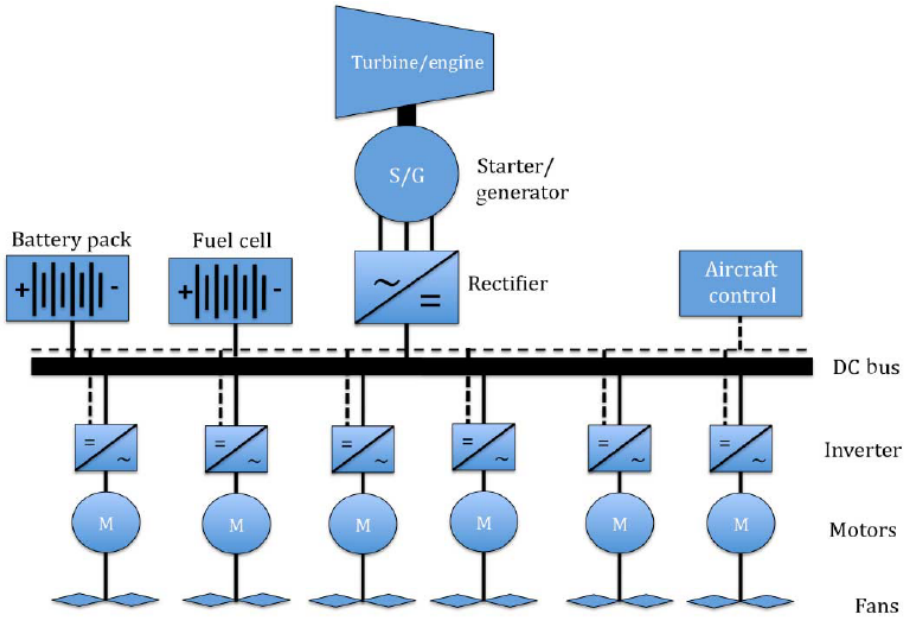
(b)

Figure 1-5: Notional shipboard with (a) MVac distribution for cruise ship [4], and (b) MVdc distribution for navy destroyer.

protective equipment and associated protection schemes to achieve fault discrimination. Solid State Circuit Breakers (SSCBs) and fast-acting no-load isolating switches installed at interconnection points within the distribution architecture enable this capability. Radially distributed architectures require extremely fast coordination to achieve overcurrent relay capability [20]. Meshed distribution architectures will, as a minimum, require high-speed communications between protective devices.



(a)



(b)

Figure 1-6: Notional MEA with (a) LVac distribution for current aircraft, and (b) MVdc distribution for future aircraft [5]

Real-time simulation of power systems provides an opportunity to advance to higher Technology Readiness Levels [21] when paired with CHiL-based real-time controls [22], [23]. Table 1.1 describes TRL as applied to power electronics development. Offline simulation can be considered as TRL 3. Examples of TRL 3 can be seen in [19, 24, 25]. Implementing the design into CHiL allows engineers

Table 1.1: TRL as applied to Power Electronics

TRL	Description	Simulation Environment
9	System proven through successful operations.	-
8	System completed and qualified through testing and demonstration activities (ready to procure).	-
7	Demonstration of a system prototype in an operational environment.	-
6	Demonstration of system or subsystem model or prototype in the relevant environment.	CHiL / PHiL
5	Component or breadboard validation in the relevant environment	CHiL / PHiL
4	Component or breadboard validation in a laboratory environment.	CHiL
3	Analytical and experimental critical function and/or characteristic proof of concept.	Offline
2	Technology concepts and/or applications formulated.	-
1	Basic principles observed and reported.	-

to move from TRL 3 to TRL 4 (or even up to TRL 6) by enabling testing and validation of control schemes in an real-time environment on real hardware.

Due to the speed and resolution requirements of Low Voltage dc (LVdc) and Medium Voltage dc (MVdc) protection systems, performing real-time CHiL simulations to evaluate and validate fault detection, coordination, and isolation schemes in Commercial-off-the-shelf (COTS) systems are limited in scope due to the simulation time step of COTS systems. This results in having to develop laboratory-scale hardware to truly understand the performance of the system. Laboratory-scale, Low Voltage (LV) hardware may also be considered TRL 4, as seen in [26–29]. A downside is this approach results in limited test coverage of the protection scheme and inevitable sub-optimal design. Furthermore, once laboratory hardware is built, it is virtually impossible to explore alternative solutions due to the costs involved in such systems.

Additionally, scaling up laboratory hardware to full-scale hardware, requires almost a full, if not a complete, redesign of the protection hardware and controls. However, running a CHiL simulation takes less development time, is safer to run (especially when performing fault protection), and allows for more design exploration. Validating the protection scheme with CHiL allows for more carryover to the full-scale system, which reduces development times and risks when progressing to

higher TRLs compared to laboratory hardware. While, from a fault standpoint, there will always be field conditions that cannot be adequately simulated with CHiL, it is reasonable to expect the fault behaviors that can be simulated, and the system response to them, should be explored rigorously in a sandbox environment before committing to hardware.

Examples of CHiL-based dc protection can be seen in [30,31], but are for High Voltage dc (HVdc) systems with kilometers of cabling or with non-zero fault impedance, which both significantly limit the di/dt of the fault transient. LVdc and MVdc systems have much shorter cable lengths, which significantly increase the di/dt of the fault transient. An extensive review of microgrid CHiL applications is given in [32], but examples of real-time CHiL of dc protection are missing.

Next, one could use Power Hardware-in-the-Loop (PHiL) simulation [33] to process to TRL 5 or 6. Examples of dc protection at TRL 5 or 6 can be found in [34–37].

Additionally, most COTS real-time simulation systems are meant for testing controls of power electronics to validate Differential-Mode (DM) behavior, but not necessarily Common-Mode (CM) nor fault transient behavior. These behaviors are particularly crucial to the design of PEC-based systems. Fault-induced resonant CM voltages dictate LG voltage stresses on the system. Furthermore, multiple LG fault-induced CM circulating currents between non-isolated paralleled converters, and through unintended fault current paths internal to PECs can cause voltage stresses in components within the system at magnitudes greater than twice the rated system voltage (depending upon meshed inter-connection schemes) [38]. Thus, CM and correct grounding must be included in the real-time simulation.

To achieve PEC-based distribution system simulations in real-time suitable for testing of protective approaches, the real-time simulation must be able to simulate:

1. Sufficient simulated fault transient resolution to test out the capability of the protective control hardware;

2. Accurate simulation of earthed neutral (TT), chassis neutral (TN) and floating (IT) grounding schemes to enable correct LG fault characterization;

3. Mixed-mode (MM) DM/CM behavior resulting from PEC network asymmetry and the impacts of controls delays and Pulse-Width Modulation (PWM);

4. Ability to perform high-speed FPGA-to-FPGA communication sufficient for high-speed protective relaying;

5. Ability to simulate large networks through plug and play parallel-ability of the real-time simulator components;

6. Ability to add limitations such as compute and sensor bandwidth, and communication delays, to derive the performance required for protection schemes, controls, and sensors.

This thesis provides an approach to addressing the above real-time simulation needs by proposing a CHiL platform via combining a Latency-Based Linear Multi-step Compound (LB-LMC) real-time simulation method with National Instruments (NI) FPGA based hardware to enable dc protection design in a real-time environment for emerging dc applications at the TRL 4 level. To this end, this work simulates LG and LL fault transients and associated DM and CM behaviors in real-time with sufficient time resolution, using a case study of a 1.5 MW, 12 kV, ac-to-dc Voltage Source Rectifier (VSR).

Chapter 2 gives a background on the evolution of real-time simulation, how to compares for dc protection, and different types of CHiL. Chapter 3 discusses LG capacitances in DC systems and shows the gap between COTS real-time simulators and time steps required for realistic cable lengths in emerging dc systems. Chapter 4 introduces the LB-LMC simulation method and a new component developed for this study. Chapter 5 describes NI platforms, LB-LMC simulation engine implementation, and LabVIEW FPGA code used for controls and data logging. Chapter 6 simulates CM voltage and current produced by the VSR in a floating ground system, and LG and LL faults transients at 50 ns time steps. Conclusions and future works are discussed in Chapter 7.

Chapter 2

Real Time Simulation of DC Protection

Fig. 2-1 shows a pseudo-quantitative-qualitative curve of CPU and FPGA based RT Simulation platforms, both COTS and LB-LMC versus their circuit size, and their application towards protection. CPU based RT solvers can achieve time steps in the tens of microseconds (μs) for small systems or milliseconds for large systems. An exponential decay is drawn to represent the trade-off between system size and achievable time step for a given amount of computing. For RT simulation of power systems, power systems faults, and line commutated devices, tens of μs of simulation resolution is adequate. These solvers can exploit natural time delays in considerable cable lengths to enable parallel computation of larger power systems.

With the introduction of power electronics into electrical distribution systems, higher frequency dynamics such as switching events, harmonics, and passive component dynamics now need to be simulated, and achieving lower time steps smaller than tens of μs became necessary. However, this became difficult for CPU based RT simulators due to the jitter of the Operating System. The FPGA's inherent ability to parallel computations, reconfigure its hardware for specific circuits, reduction of execution latency, and the lack of required OS, allowed for the hardware improvements needed for RT simulators to achieve lower time steps.

FPGA based RT simulators also have constant compute curves, with time steps ranging from a few hundred ns for minimal systems to 20 μs on the higher end. From the authors' experience, most systems of reasonable size run between 1-4 μs. This range of time steps is sufficient for most

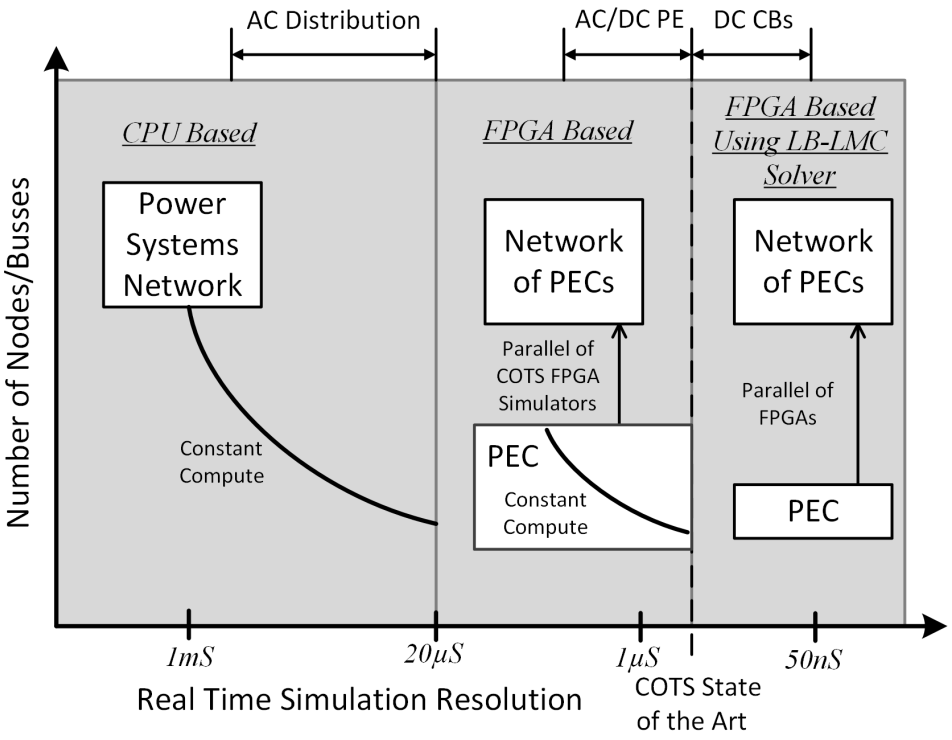


Figure 2-1: Real Time Simulation Resolution vs Model Size

power electronic systems. To achieve larger power electronics networks, FPGA based solvers need to be paralleled.

The first step in performing protection design for a distribution system, fault characterization must be performed to determine the differences between fault current at different nodes in the system. The differences in currents between these nodes determine the settings at different breaker locations. A case study was performed for an LVdc residential microgrid in [39]. It was shown, due to the short cabling inductance of the system, the time-trip characteristics between the upstream and downstream breakers were on the order of μs. In [34], MVdc SSCBs were tested for both operation and coordination for MVdc shipboard applications. In the validation of the controls, the author noted an attempt to test operation and coordination in a COTS RT environment. However, in testing fault scenarios with minimum cabling required by the protection system design specification, the time of operations of the SSCB was on the order of a μs, and the RT COTS platform did not have sufficient interface bandwidth capabilities to validate coordination and operation of SSCBs.

In both these examples, and in general, if coordination must occur on the μs time scale, then the

simulation must have a time step orders of magnitude smaller than μs to have sufficient resolution to simulate the fault transients. Additionally, the RT simulation platform's interface latency between the simulator itself and the controls or hardware under test must be low enough not to interfere with the testing. For these reasons, COTS platforms did not have sufficient resolution for the tripping and coordination being assessed. LB-LMC RT simulation method can achieve 50 ns time steps on Xilinx 7-series FPGA [7], [40]. For these cases described above, this would have provided enough resolution.

The main limitation of the circuit size that can be implemented using LB-LMC is the Digital Signal Processor (DSP) resources on the FPGA. Since the solver is performing mostly multiplications and additions, this is the resource which generally runs out first. Larger circuits can be achieved using the UltraScale+ FPGA due to the increase in DSP resources compared to prior fabrication generations. Additionally, the newer fabrication process results in the DSP multipliers on the FPGAs being more efficient. Moreover, the LB-LMC method has been extended to multi-FPGA systems [41], allow the system to run at 100 ns on 2 FPGAs in parallel.

In general, One challenge that is faced, especially with dc networks, is how to partition the model between paralleled FPGA solvers as the system grows in size. PECs interconnected by dc buses represent a contiguous system that is also the main source of fault vulnerability. If parts of the system are partitioned into separate FPGA's separated by a dc cable, which is relatively short in a dc microgrid, ship system or MEA, the routing of signals between simulation platform components may introduce unacceptable time delays that affect the fault response. The 100 ns multi-FPGA approach in [41] shows promise for dc systems.

Fig. 2-2a shows a typical CHiL interface for power electronics. The RT simulator is on an FPGA, and the controls platform is on an FPGA or DSP. The two interface with analog and digital inputs/outputs (IOs). This approach is easy to implement and works if a controller has already been selected for the application. However, if the protection scheme is still being designed/verified, then selecting a controller can limit the design prematurely as the controller requirements such as bandwidth, compute, and signal resolution may be unknown or may not be sufficient. Additionally, the Digital-to-Analog Converter (DAC)/Analog-to-Digital Converter (ADC) interface of the controller and RT simulation platform, can limit possible solutions as both have bandwidth and resolution limitations. At the 10's of MHz level, a level required to see the full resolution of a 50

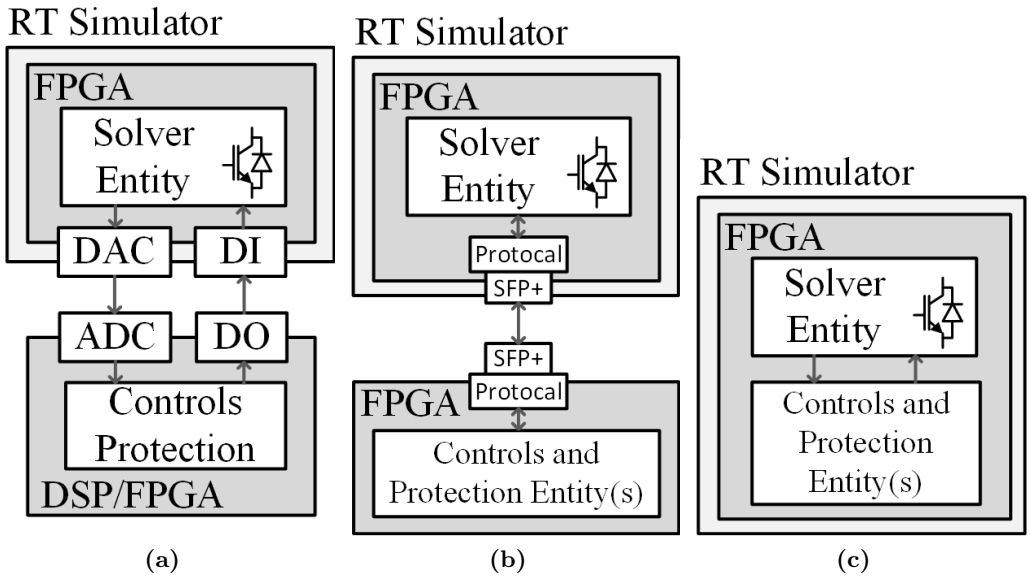


Figure 2-2: Controller Hardware-in-the-Loop interfaces between RT Simulator and Controls/Protection Scheme via (a) Analog/Digital I/Os, (b) FPGA-to-FPGA Gigabit Serial, and (c) on same FPGA.

ns RT solver, commercial DAC options that can interface to FPGAs can be limited, either by the number of output channels, price, or ease of implementation.

Fig. 2-2b shows the controls and RT simulation platform interfacing via a duplex Gigabit serial protocol Small Form-Factor Pluggable Transceiver (SFP+) based protocols. These communication protocols can transmit data over fiber optic and is capable of up to 10 Gbit/s per channel (Gigabit Ethernet could also be used). The exact latency can vary, but are generally on the order of a few hundred ns. An advantage of this approach includes no limitation on the number of signals transferring data, nor the resolution of the signals. These settings are up to the user. The downside of this approach is the user must be proficient in FPGA programming, using the Gigabit protocols, and developed one's own interface on top of the protocol to keep track of signals and their corresponding data.

Fig. 2-2c shows the solver entity and the protection logic entity on the same FPGA. This approach offers a latency of a few clock cycles between the solver and protection logic. Also, the precision and throughput of signals are limited by the area of the FPGA. This approach removes the downsides of the two methods mentioned above. Another advantage is, once the protection scheme is validated, delays and resolution limitations can be added to determine the amount of compute, allowed latency, and minimum bandwidth. A downside of this approach is that the controls must

share FPGA resources with the solver entity. If the power electronic circuit in the solver entity is close to maxing out of one of the resources of the FPGA, then there is not much room to implement the protection scheme. Another drawback is that the FPGA code needs to be recompiled every time.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Line to Ground Capacitances in DC systems

To explore the benefits of this RT simulation approach, an example was chosen of a VSR with a floating (ungrounded) system. Floating grounds or high resistances grounds are used in mission-critical systems like maritime applications [42] to maintain operability via the LL DM voltage remains constant during a single LG fault. The downside is that ground faults are more difficult to detect due to the much lower, or nonexistent, steady-state fault current compared to hard grounded or low resistance grounded system [43, 44].

Bulk capacitive elements, inductive cabling, and filter drive the fault characteristics in dc power electronics-based systems. For LG faults, the inductive elements of the system are cabling and filters, while the capacitive elements are LG capacitance, which can come from 3 sources:

1. parasitic capacitance of the baseplate found in multi-chip power electronic module;
2. parasitic capacitance of cabling;
3. capacitance of Electromagnetic Interference (EMI) filters.

A ground fault's characteristics are governed by the equivalent path L inductance and bulk LG capacitance C, which gives a resonant frequency of

$$f_{res} = \frac{1}{2\pi\sqrt{LC}} \tag{3.1}$$

Table 3.1: Baseplate-to-Ground Capacitance for Half-Bridge Modules

Parameter		CAS120M12BM2	Gen. III Power Module
Device Rating		1.2 kV / 120 A	10 kV / 240 A
Baseplate Dimensions		62 mm x 106 mm	195 mm x 125 mm
Baseplate Area		6,572 mm^2	24,375 mm^2
Area Ratio		1.000	3.709
C_U		191.0 pF	708.4 pF
C_M		255.7 pF	948.4 pF
C_L		102.6 pF	380.5 pF
C_P	$(3C_U)$	573.0 pF	2,125.2 pF
C_N	$(3C_L)$	307.8 pF	1,141.6 pF
C_O	$(3C_U + 3C_L)$	880.8 pF	3,266.8 pF
C_{AC}	(C_M)	255.7 pF	948.4 pF
C_{TM}	$(C_U + C_M)$	446.7 pF	1,656.8 pF
C_{BM}	$(C_L + C_M)$	358.3 pF	1,328.9 pF

3.1 Parasitic Capacitance of 12 kV Neutral Point Clamped Converter

A 12 kV dc bus voltage is a consideration for the MVdc power distribution in shipboard applications [42]. Using a 10 kV / 240 A Silicon Carbide (SiC) MOSFET multi-chip power module in half-bridge

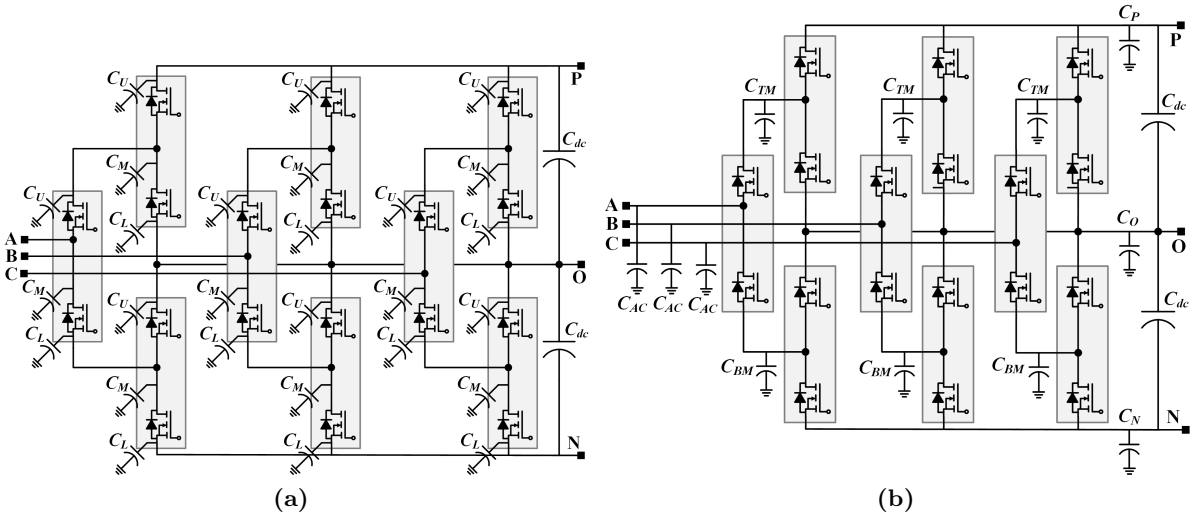


Figure 3-1: NPC Converter with baseplate-to-heat sink parasitic capacitances (a) for each module, and (b) simplified.

configuration [45] and applying 60% voltage and current derating [46] gives 6 kV / 144 A building block. These modules can be configured into a Neutral Point Clamped (NPC) VSR converter to achieve a 12 kV bus. The geometry between the half-bridge module and heat can be represented as three parasitic capacitors at the upper, middle, and lower part of the module, labeled as C_U, C_M, C_L, respectively [47]. Due to lack of availability of the 10 kV power module, the capacitances from a Cree 1.2 kV / 120 A half-bridge module taken from [48], and scaled by the increased area of the module's baseplate, and are tabulated in Table 3.4. Fig. 3-1a shows a 3 phase NPC converter made of multi-chip power module half-bridges and the parasitic capacitances. The capacitances can be added in parallel, as shown in Fig. 3-1b. Table 3.4 tabulates these values.

3.2 Parasitic Capacitance of Cabling

To determine the parasitic capacitance of cabling, some assumptions need to be made about the current and insulation rating. The current rating of the cable is assuming to be the output current rating of the power converter. Given a 3 phase, 6 kVac LL RMS input voltage, and 144 A RMS input current, the input power is about 1.5 MW. A 12 kV dc output at 1.5 MW gives an output dc current of 125 A. Assuming a current density of 3.76 A/mm^2 for copper cable [49], gives an area of 33.24 mm^2. Using a 35 mm^2 single-core, copper cable with polyethylene (XLPE) insulation, conformed to IEC 60502 from [50], the capacitance, inductance and resistance per meter can be found. Table 3.2 tabulates these values for different cable lengths and insulation ratings. (3.1) calculates the resonant frequency of the cable. Dividing the resonant simulation time step by the resonant period gives the amount of simulation time steps per resonant period. The COTS platform assumes a time step of 1 μs or a frequency of 1 MHz, and the LB-LMC solver assumes a time step of 50 ns, or a frequency of 20 MHz. Columns 6-8 of Table 3.2 tabulate these values.

Table 3.2: IEC 60502 single core $35mm^2$ copper cable parameter for various lengths, CL resonant frequencies, and time step per resonant period in COTS and LB-LMC solvers.

Cable Length [m]	V_{LN}/V_{LL} **[kV]**	**Resistance** $[\mu\Omega]$	**Capacitance [pF]**	**Inductance, Flat Spaced [nH]**	**Resonant Frequency [kHz]**	**Time Steps per Resonant Period**	
						COTS (1 MHz)	**LB-LMC (20 MHz)**
1	3.6 / 6	47	627	480	9,174.1	0.11	2.18
	6 / 10	47	561	490	9,599.3	0.10	2.08
	8.7 / 15	47	439	500	10,742.4	0.09	1.86
	12 / 20	47	370	510	11,586.0	0.09	1.73
10	3.6 / 6	470	6,270	4,800	917.4	1.09	21.80
	6 / 10	470	5,610	4,900	959.9	1.04	20.83
	8.7 / 15	470	4,390	5,000	1,074.2	0.93	18.62
	12 / 20	470	3,700	5,100	1,158.6	0.86	17.26
25	3.6 / 6	1,175	15,675	12,000	367.0	2.73	54.50
	6 / 10	1,175	14,025	12,250	384.0	2.60	52.09
	8.7 / 15	1,175	10,975	12,500	429.7	2.33	46.54
	12 / 20	1,175	9,250	12,750	463.4	2.16	43.16
50	3.6 / 6	2,350	31,350	24,000	183.5	5.45	109.00
	6 / 10	2,350	28,050	24,500	192.0	5.21	104.17
	8.7 / 15	2,350	21,950	25,000	214.8	4.65	93.09
	12 / 20	2,350	18,500	25,500	231.7	4.32	86.31
100	3.6 / 6	4,700	62,700	48,000	91.7	10.90	218.00
	6/ 1 0	4,700	56,100	49,000	96.0	10.42	208.35
	8.7 / 15	4,700	43,900	50,000	107.4	9.31	186.18
	12 / 20	4,700	37,000	51,000	115.9	8.63	172.62
250	3.6 / 6	11,750	156,750	120,000	36.7	27.25	545.01
	6 / 10	11,750	140,250	122,500	38.4	26.04	520.87
	8.7 / 15	11,750	109,750	125,000	43.0	23.27	465.44
	12 / 20	11,750	92,500	127,500	46.3	21.58	431.55

The question then becomes what constitutes a *sufficient resolution*? The resolution of the RT platform should be sufficiently greater than the dynamics of the circuit such that the resolution of the RT simulation does not interfere with the protection solution design, nor drive the design requirements. A conservative amount of samples could be said to be 100. Using the LB-LMC method, the shortest cable length a ground fault would be simulated in RT with sufficient resolution would be 50 m. Using a COTS platform, at 50 m of cabling, there would only be 4-5 samples period, which would be insufficient for any design. For this particular cable, 1 km of cabling is required to achieve 100 time steps per period during an LG fault. With a 20x reduction in time steps, 20x shorter cable lengths can be simulated in RT.

The lengths of potential dc applications are shortening. In essence, the length of a Boeing-777 or Airbus A340 is around 70-75 meters, a US Navy Zumwalt-class destroyer is 183 meters, and a city block varies from 80 to a few hundred meters. The length of the transportation system or microgrid is not an exact unit of measure, but helps to provide an intuition for cable length. Thus, for these emerging applications, cable length will be in term of tens to few hundreds of meters, as opposed to kilometers cabling. The shorter cable length means the LB-LMC solver enables RT simulation of dc protection with *realistic cable lengths* for shipboard, MEA, and microgrid applications. At the same time, the COTS solution is suitable for cable length greater than 1 km for this particle cable.

3.3 EMI Filter

For this active rectifier, an *LC* CM filter was used. During an LG fault, the LG capacitors can discharge. However, with an LG fault occurring on the dc side, the current discharging from the CM capacitors are limited by the CM inductor. If an LG fault occurs on the ac side, the fault current contribution of the CM capacitors may be more significant since the CM inductor would not shunt the current. Thus, the placement of EMI filters within the system should be considered in terms of fault current.

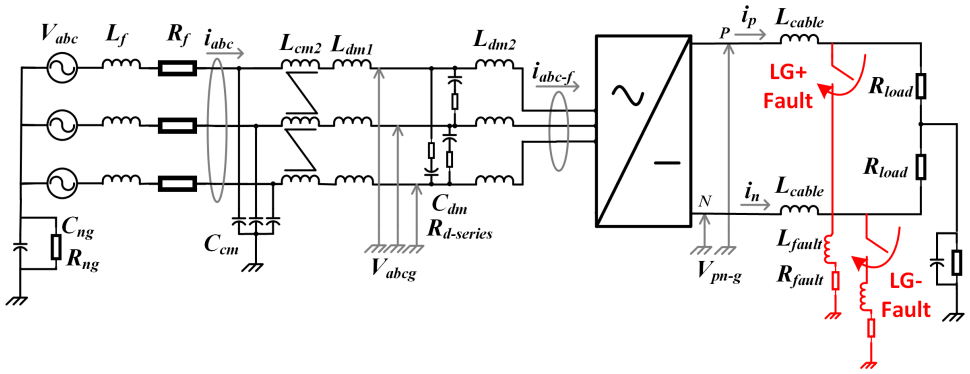


Figure 3-2: VSR circuit

3.4 Discussion on Offline Simulation of Ground Faults

Fig. 3-2 shows the VSR circuit simulated in Simulink 2017a 64-bit with PLECS v4.3 blockset with the goal of applying a Positive Line-to-Ground (+LG) fault and Negative Line-to-Ground (-LG) fault. The ac-dc converter is in Fig. 3-1a with the parameters in Table for the baseplate-to-heatsink capacitances, Table 3.3 for the di/dt limited MOSFET parameters, and Table 6.2 for the filter parameters.

A discussion can be had on how to model the cable. Conventionally, π or T models are used to include the cable's LG capacitance for power systems [51]. If an LG fault is applied on a π model, then the fault is being directly applied to the π model's capacitances, so a T model cable may be more favorable. The challenge with adding capacitances is the formation of resonances with inductances in the system. These resonances can be excited by switching events and produce high-frequency content in the simulation results that may not be present in real life due to effects not simulated like the skin-effect of cables. For this reason, these resonances have to be manually damped out. This results in a trade off of adding resistance damping to CL paths to damped noise likely not present in a real system while trying not to dampen out the effects of the sought after frequency content occurring during a ground fault. Additionally, there is a practical challenge to such simulations of stiff systems where, in this example, time constants of L's and C's are many orders of magnitudes smaller than the time constants of the controls. For a boost rectifier, like the one in this paper, it takes time to bring the bus voltage up to steady-state. Even with the dc-link capacitors initialized to the nominal bus voltage, it still takes time, once the converter starts switching, for the controls to bring the bus voltage to steady-state. Additionally, the lower time

Table 3.3: Gen. III Power Module Simulation Parameters

Parameter	Description	Value	Unit
V_{ds}	Blocking voltage	10	kV
I_d	Continuous drain current	240	A
Rds_{on}	Drain-Source On-resistance	23	$m\Omega$
Rds_{off}	Drain-Source Off-resistance	1	$M\Omega$
t_r	Rise time	60	ns
t_f	Fall time	100	ns
L_σ	Stray Inductance	15	nH
V_f	Diode Forward Voltage	3.5	V
Rd_{on}	Diode On-resistance	12	$m\Omega$

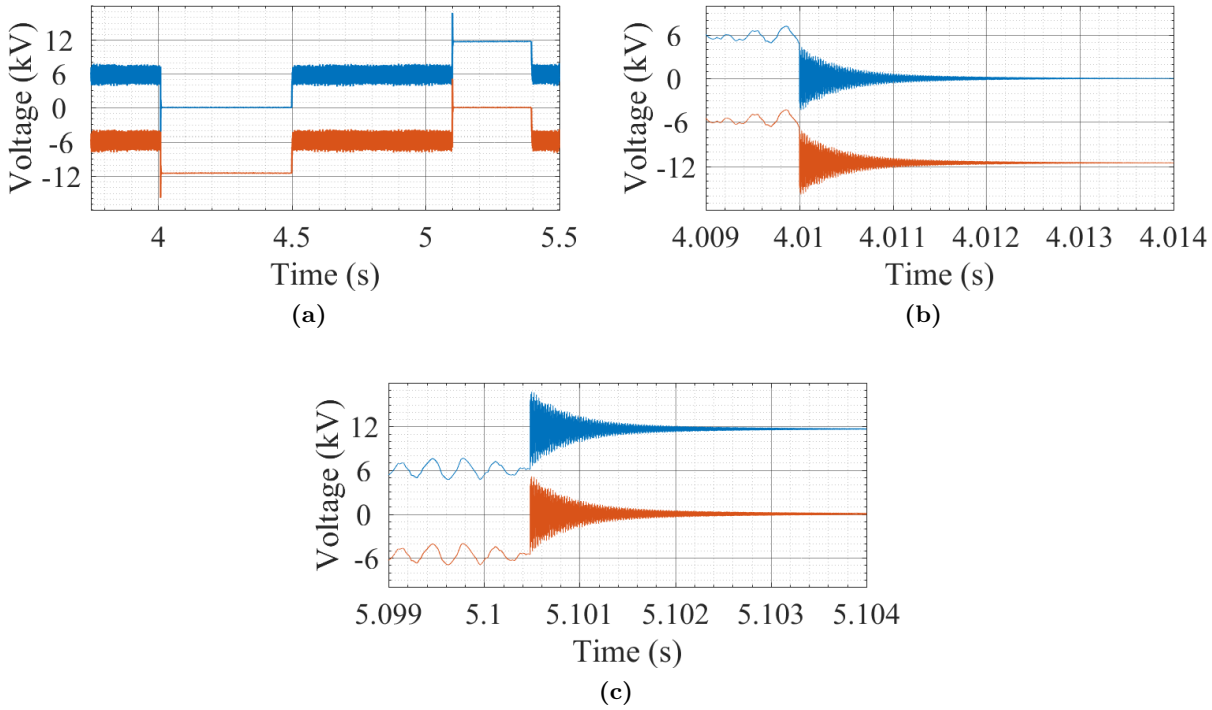


Figure 3-3: Offline PLECS Simulation of NPC based VSR: (a) +LG and -LG fault, (b) +LG zoomed, and (c) -LG zoomed

constant of the L's and C's drives the lower bound of the simulation time steps. If a variable-step solver is used, it will hit the lower bounds of the time steps and may error out due to reaching the maximum number of consecutive minimum time steps. For this reason, either a fixed time step solver should be used with the drawback of increase computational time, or these time constants need to be damped out for the variable time step solver.

Fig. 3-3a shows a +LG and -LG fault using the NPC with the equivalent baseplate capacitances

showing in Fig. 3-1b. Zoomed views are showing for +LG and -LG in Fig. 3-3b and 3-3c, respectively. Damping resistors were added between the baseplate capacitance and ground, then tuned to set the lower bounds of the simulation dynamics. The model with capacitances shown in Fig. 3-1b makes tuning the damping resistors easier compared to the model in 3-1a. These results do not claim to be the exact results during an LG fault, but gives an approach on how to simulate the correct behavior that can be adjusted based on the system parameters. As will be seen in Section 6.1, for real-time simulation, this capacitance can be lumped at the midpoint of the dc-link to ground to allow for similar fault behavior. Discussion on LG faults in IT systems is carried out in Section 6.1 as well.

Chapter 4

LB-LMC Solver method

4.1 Summary of LB-LMC Method, and FPGA implementation

This section provides a summary of the LB-LMC method as from the perspective of the author, which is directed more towards the power electronic engineer, who maybe a layman in terms of simulation methods. For the more technical description and analysis of the LB-LMC method by the creator of the method, please see [7] [52], which contains details of the method not presented here, such as stability analysis, computational considerations, and further comparison to traditional simulation methods.

The LB-LMC method is a numerical integration approach for solving transient, linear and non-linear systems such as for power electronics and multi-physics system. This method has been implemented on Xilinx 7 series FPGAs to achieve time steps in the mid-tens of ns range, such as 50 ns as in [7, 8, 40, 53]. It is possible to achieve lower time steps using the latest UltraScale or UltraScale+ FPGAs. As the circuit size increases, the compute time stays almost constant. The main operations of the LB-LMC solver are additions and multiplications, which tend to heavily utilize DSP responses on the FPGA. For this reason, the DSP resources on the FPGA is the resource type which runs out first, and is the main limitation of the circuit size one can simulate on a single FPGA. To address this, [54] implements the LB-LMC solver across multiple FPGA at 100 ns with a custom FPGA to FPGA bus interface, but is outside the scope of this work.

The LB-LMC method is a modified Resistive Companion (RC) method. The RC method is a commonly used circuit solving approach, and is used in such solvers as EMTP and SPICE based

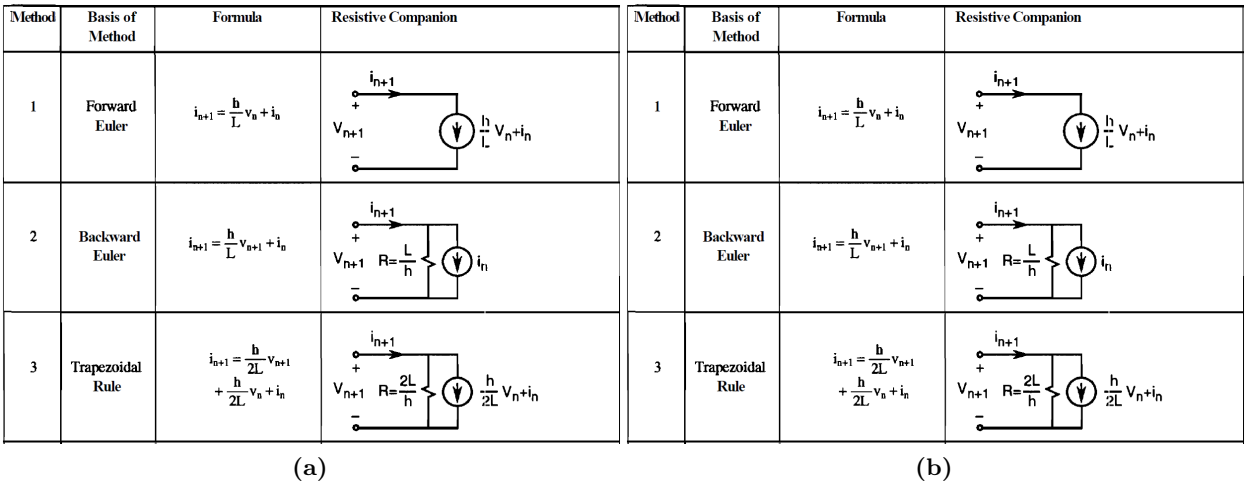


Figure 4-1: RC models of (a) Inductor and (b) capacitor for different 1^{st} order integration methods [6].

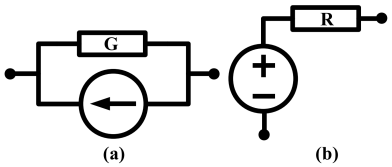


Figure 4-2: LB-LMC Component Models: (a) component with current state, and (b) component with voltage state [7].

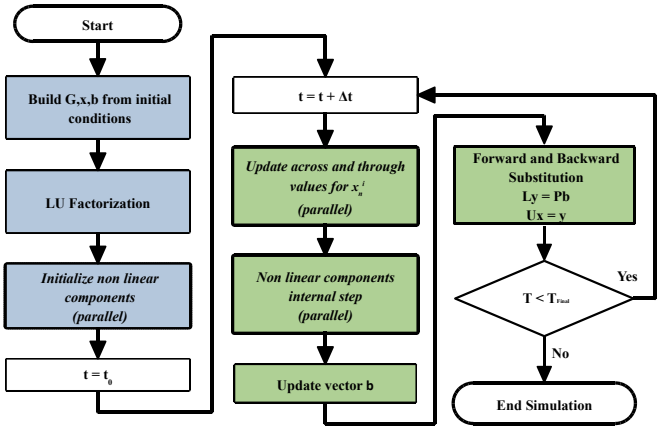


Figure 4-3: LB-LMC Solution Flow [7].

simulation software like PSCAD or LTSpice.

Fig. 4-1 shows different RC equivalent circuits for an inductor and capacitor for Forward Euler, Backward Euler, and Trapezoidal.The RC equivalent circuits translate the system in voltage sources, current sources, and resistors, and a linear set of equations can be formed. For the implicit integration method, such as Backward Euler and Trapezoidal, the current sources are accompanied

by a companion resistor, hence the name, resistive companion. These linear set of equations can then be formed into a matrix form of $Gx = b$, where G is the conductance matrix, x is a vector of the system solution (i.e., next time step), and b is a vector of the source contributions (i.e., present time step). The system solution can be solved by inverting the conductance matrix G on both sides to get $x = G^{-1}b$. G^{-1} is computed offline prior to the simulation start. This works for linear systems; however, power electronics are non-linear, piece-wise components. Implementing power electronics, or systems with non-linearities, in traditional RC method would involve recomputation of G^{-1} when the switch state changes, or storing all the G^{-1} in memory and then jumping between different G^{-1} matrices depending on the switch states. In order to keep G constant, and avoiding having to perform a matrix inverse during real-time simulation, and/or having a non-iterative solver, the non-linear components, such as PECs, are integrated with explicit integration methods like Forward Euler or Runge-Kutta, while the linear components such as inductors and capacitors are integrated with implicit integration methods like Trapezoidal. The core component model building blocks in LB-LMC can be seen in Fig. 4-2. After the integration is performed, the source contributions to each node of the circuit/each element of the b vector. Then the system solutions are found via $x = G^{-1}b$. Fig. 4-3 shows the solution flow of the LB-LMC solver. To highlight an example of the different integration methods in a circuit, Fig. 4-4 shows the active rectifier circuit, which will be simulated in Chapter 6. The VSR has an LCL DM filter with passive damping and LC CM filter powering a resistive load, and measurement points referred to throughout the paper. Fig. 4-5 shows the different integration methods used for the different components in the circuit (One of the DM inductors, L_{dm1}, is added to leakage term of the CM inductor to reduce components).

For FPGA implementation, the fixed-point data type is used to avoid iterative computation. The numerical accuracy of the LB-LMC approach running in C++ double-precision floating-point was shown to be less than 1% of traditional RC running at two orders of magnitude smaller times time steps (LB-LMC running at 50 ns and RC running at 500 ps), and indistinguishable from zero when comparing LB-LMC running in C++ double-precision floating-point vs. fixed-point [7]. In the author's own experience, this solver matches PLECs exactly, or at least (indistinguishable form the naked eye). The fixed-point word size of the system variables and internal solver signals can be larger than 64 bits. The signal size used in this model was 69 bits of 28 integer bits and 41 fractional bits. Moreover, each component's integration method and contribution to the source

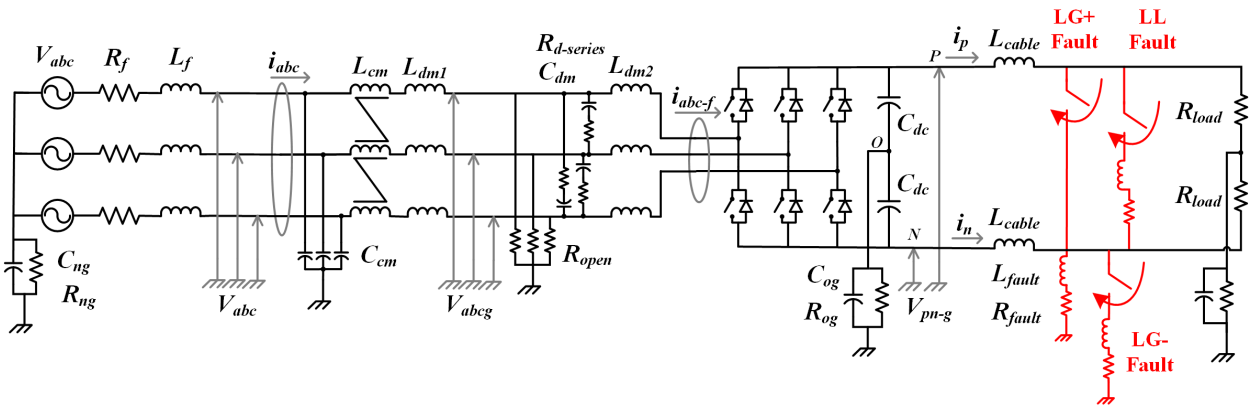


Figure 4-4: VSR Circuit with LCL DM filter, LC CM filter, floating ground, measurement points, +LG and LL faults.

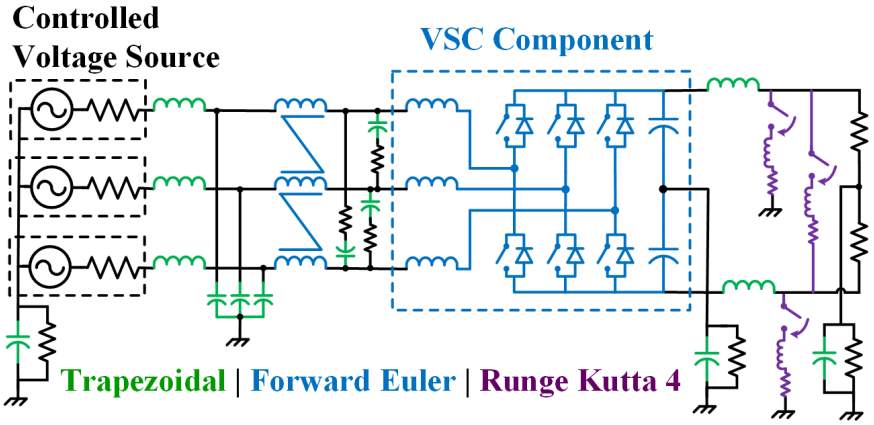


Figure 4-5: VSR Circuit showing the different integration methods used per component.

contribution vector, b, can be, and are, performed in parallel to leverage the natural strength of FPGAs. A diagram of the LB-LMC solver engine in an FPGA is shown in Fig. 4-6. To summarize how the LB-LMC solver executed on FPGAs is able to such low time steps compared to COTS simulation approaches:

1. LB-LMC maintains a fixed conductance matrix by explicit integration for non-linear components like PECS, while preserving their non-linearity.

2. The data type of signals used is set to fixed-point to avoid iterative computation, allowing the solver to execute solutions in a single clock cycle.

3. Much of the computation, including the component integration, is performed in parallel on an FPGA.

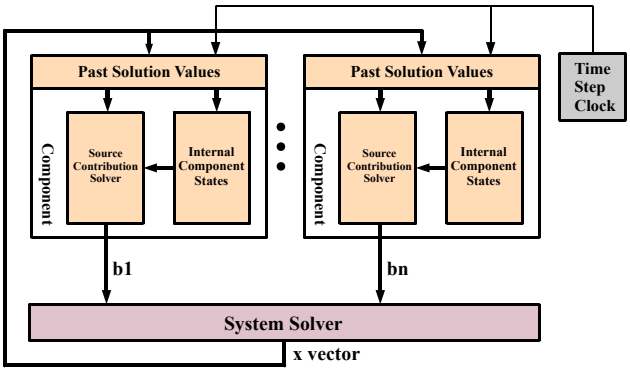


Figure 4-6: LB-LMC Solver Engine [7].

4.2 Half-Bridge State Space Model

The half-bridge module is the workhorse of the power electronics industry. An example half-bridge with anti-parallel diodes that will be used in the LB-LMC solver is shown in Fig. 4-7. This module is building block for many topologies such as buck and boost converters, single-phase, and 3-phase 2-level Voltage Source Converter (VSC), and other topologies such as, but not limited to, Dual Active Bridge (DAB), NPCs, and sub-modules of Modular Multilevel Converter (MMC). This component allows for independent switching of the upper and lower of a conventional half-bridge module. This enables testing of dead-time, and the conduction of both anti-parallel diodes simultaneously. This conduction of both anti-parallel diodes is important in evaluating power electronic converters under fault conditions. In general, the anti-parallel diodes are there to prevent output current from going to zero during dead time. In IGBT modules, the diode is added. In MOSFET modules, the body diode is inherent to the device structure. If the diodes were not present, then during dead-time,

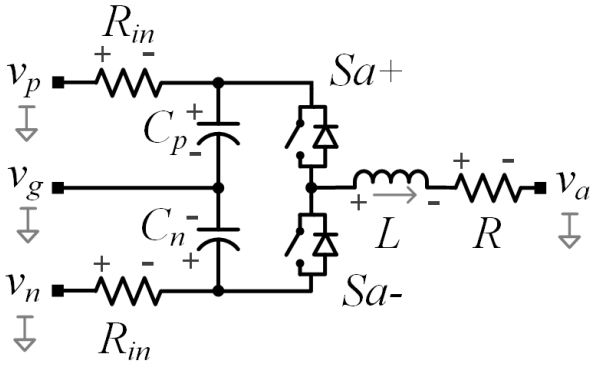


Figure 4-7: Schematic of half-bridge VSC with anti-parallel diodes

the inductor current would go to zero instantaneously, which would cause a large di/dt, and a large voltage spike. This voltage spike could damage or destroy the devices.

When operating the half-bridge module as an ac-dc boost rectifier, on startup with zero bus voltage, the anti-parallel diodes cause the boost rectifier to operate as a conventional uncontrolled diode bridge rectifier while precharging the bus voltage. Once the bus has reached a steady-state, then the boost rectifier can begin operation in a controllable manner. Additionally, if there is an LL fault on the dc side of the converter, the anti-parallel diode should conduct in a free-wheeling manner. Lastly, the component model should function as a conventional buck converter. This half-bridge module, typically called a leg, can be configured with three legs for 3 phase applications. To summarize the operational modes of the VSC for 3 phase application:

1. dc-ac, buck, Voltage Source Inverter (VSI).
2. ac-dc, boost, VSR.
3. ac-dc, diode bridge rectifier.
4. free wheeling diodes during fault conditions.

It is also possible for the three-leg VSC as an interleaved buck or boost converter, and which the component model should be able to do, but is not covered in this work.

A VSI without dead time can be implemented with just switching between the upper and lower leg. This is a typical approach used in converter legs of this type, but it does not encompass all the operational modes listed above. To allow for shoot-through conditions caused by errors in gating logic or diode free-wheeling condition caused by LL faults in the system, the circuit state with both switches should be considered. Additionally, the circuit condition with both switches are off should be included when switches are not conducting. In the context of real-time simulation, the integration is performed with fixed, non-iterative time steps (as opposed to variable or iterative), as each computation per time step must be complete before the next time step. From a modeling and simulation perspective, power electronic devices can be thought of as an open or closed switch. Depending on the switch position, a different part of the circuit is connected. The circuits equations can be written out as a set of differential equations represented in state-space form for each switching position. Then, to integrated with respect to time to see the time domain response of the system.

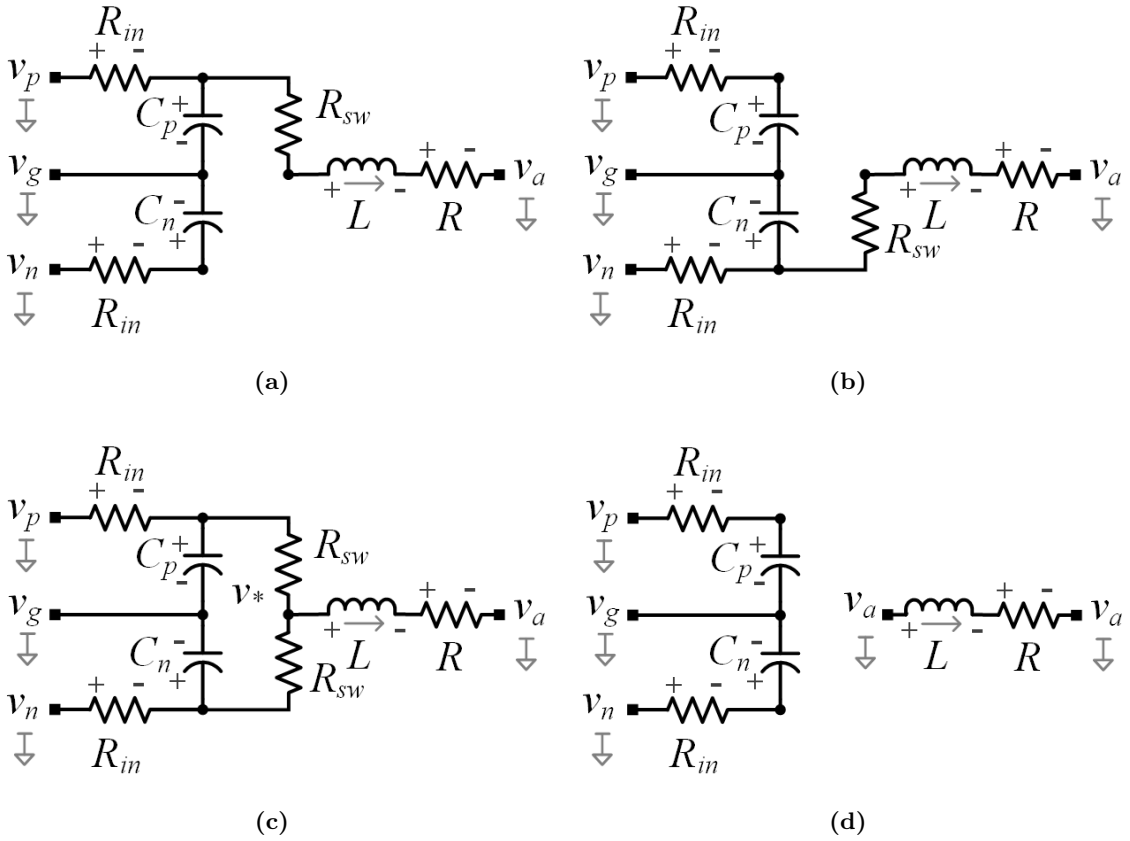


Figure 4-8: Half-bridge converter with anti-parallel diodes (a) upper switch/diode on, (b) lower switch/-diode on, (c) both switches or both diodes on, and (d) both switches and both diodes off

For the half-bridge component, first, the state-space equation should be determined for the different half-bridge states: upper switch on, lower switch on, both switches on, and both switches off. These different equivalent circuits are shown in Fig 4-8. Assumptions for modeling include ideal switches for both the switching devices and diodes, with the same fixed on-state resistance of R_{sw}. The R_{in} and R resistances are small values used for the discretization of capacitor and inductor, respectively. The state-space equations were derived with the midpoint of the dc-link capacitors as an access point to allow the implementation of various ground schemes. A complete derivation can be found in Appendix A.1. The final equations are shown in Table 4.1 for $\dot{i_{La}}$, and Table 4.2 for $\dot{v_{cp}}$ and $\dot{v_{cn}}$. The equations are organized by the different switching states. Depending on which switch state is active for a given time step, the appropriate set of differential equations will be used. There are two sets of equations for both switches on-state. One derivation was done by Matthew from USC using the voltage at the midpoint of the leg, v_*, where $v_* = \frac{1}{2}\big(v_{cp} + v_{cn}\big) + v_g - \frac{R_{sw}}{2} i_L$.

Another was done by the author without using v_*. Both give the same equations once v_* is plugged in.

Table 4.1: Switching functions for $\dot{i_{La}}$ equations

Switch State		$\frac{\Delta x[n]}{DT}$		Switching Functions											$RswR/L$	$x[n]$
SA+	SA-			v_{cp}		v_{cn}		v_g		v_*		v_x				
ON	OFF	$\dot{i_{La}}$	=	$\frac{1}{L_a}\Big[$	v_{cp}	+	0.0	+	v_g	+	0.0	-	v_a	$\Big]$	-	$(R_{sw}+R)/L$	i_{La}
OFF	ON	$\dot{i_{La}}$	=	$\frac{1}{L_a}\Big[$	0.0	+	v_{cn}	+	v_g	+	0.0	-	v_a	$\Big]$	-	$(R_{sw}+R)/L$	i_{La}
OFF	OFF	$\dot{i_{La}}$	=	$\frac{1}{L_a}\Big[$	0.0	+	0.0	+	v_g	+	0.0	-	v_a	$\Big]$	-	R/L	i_{La}
ON	ON	$\dot{i_{La}}$	=	$\frac{1}{L_a}\Big[$	0.0	+	0.0	+	0.0	+	v_*	-	v_a	$\Big]$	-	R/L	i_{La}
ON	ON	$\dot{i_{La}}$	=	$\frac{1}{L_a}\Big[$	$\frac{1}{2}v_{cp}$	+	$\frac{1}{2}v_{cn}$	+	v_g	+	0.0	-	v_a	$\Big]$	-	$R/L - R_{sw}/(2L)$	i_{La}

Table 4.2: Switching functions for $\dot{v_{cp}}$ and $\dot{v_{cn}}$ equations

Switch State		$\frac{\Delta x[n]}{DT}$		$x[n]$			Switching function	
SA+	SA-							
ON	OFF	$\dot{v_{cp}}$	=	$\frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	$-i_{La}$ $\Big]$
OFF	ON	$\dot{v_{cp}}$	=	$\frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	0.0 $\Big]$
OFF	OFF	$\dot{v_{cp}}$	=	$\frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	0.0 $\Big]$
ON	ON	$\dot{v_{cp}}$	=	$\frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big]$	+	$\frac{1}{R_{sw}C}$	$\Big[$	$v_{cp} + v_g - v_*$ $\Big]$
ON	ON	$\dot{v_{cp}}$	=	$\frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big]$	+	$\frac{1}{2R_{sw}C}$	$\Big[$	$-v_{cp} + v_{cn} + R_{sw}i_{La}$ $\Big]$
ON	OFF	$\dot{v_{cn}}$	=	$\frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	0.0 $\Big]$
OFF	ON	$\dot{v_{cn}}$	=	$\frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	$-i_{La}$ $\Big]$
OFF	OFF	$\dot{v_{cn}}$	=	$\frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big]$	+	$\frac{1}{C}$	$\Big[$	0.0 $\Big]$
ON	ON	$\dot{v_{cn}}$	=	$\frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big]$	+	$\frac{1}{R_{sw}C}$	$\Big[$	$v_{cn} + v_g - v_*$ $\Big]$
ON	ON	$\dot{v_{cn}}$	=	$\frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big]$	+	$\frac{1}{2R_{sw}C}$	$\Big[$	$v_{cp} - v_{cn} + R_{sw}i_{La}$ $\Big]$

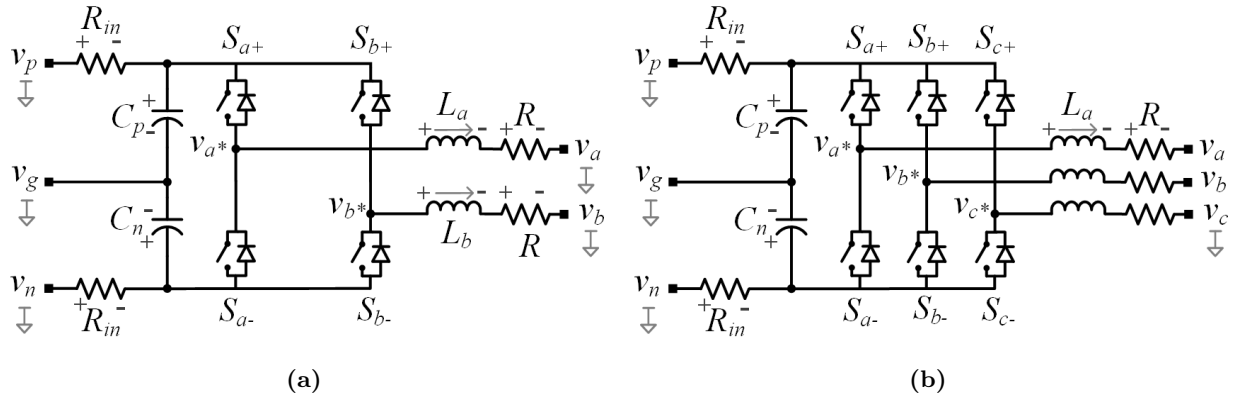


Figure 4-9: Voltage Source Converters with anti-parallel diodes (a) Full Bridge (b) 3-Leg.

Extending the state-space equations to multi-phase applications such as the full-bridge converter (Fig. 4-9a) or 3-leg, 2-level converter (Fig. 4-9b) is fairly straight forward. The switching function equations for $\dot{i_{Lb}}$ and $\dot{i_{Lc}}$ are the same as Table 4.1. For $\dot{v_{cp}}$ and $\dot{v_{cn}}$, the switching functions add the inductor current term for each of the legs. For example, if the ON switches are SA+, SB-, and SC-, then the capacitor voltage state-space equations will be:

$$\dot{v_{cp}} = \frac{1}{R_{in}C}\Big[v_p - v_{cp} - v_g\Big] + \frac{1}{C}\Big[-i_{La}\Big] \tag{4.1}$$

$$\dot{v_{cn}} = \frac{1}{R_{in}C}\Big[v_n - v_{cn} - v_g\Big] + \frac{1}{C}\Big[-i_{Lb} - i_{Lc}\Big] \tag{4.2}$$

If one of the legs has a short circuit, then the short circuit term would be added for that leg.

4.3 Logic for VSC with Anti-Parallel Diode behavior

The next step is to determine which switch should be on under which conditions. If the component was modelled as a single-pole, double-throw switch, where the switch state was drive by the gate signals, then determining the switch state would just be equal to the gate signal. As mentioned before, this approach works for VSI under steady-state operation, but does not work for the VSR due to its diode rectification upon startup, so the VSR must include both the actively gating operation and diode operation, and be able to go between the two modes seamlessly. This section describes the logic used to simulate diode behavior.

A standard diode, as shown in Fig. 4-10a, becomes forward biased when the voltage across the diode, v_f, is greater than the diode threshold voltage, v_{TH}. Once the diode is conducting, it turns off on zero current. The distinction of zero current compared to zero or negative voltage is if an ac to dc diode rectifier has an input voltage with lagging current, then it is possible for the diode to experience negative voltage across the device with positive current. To capture the voltage turn-on, and current turn-off, a diode state machine was used, as shown in Fig. 4-10b. Fig.4-11 shows the logic implemented combining the diode state machines, and gate signals, two per leg. Additional logic is added to ensure if the inductor current is nonzero, that the correct diode will freewheel, such as in a buck converter. This logic is then repeated per each leg to build the state-space equations for Forward Euler integration. The updated states are turned into source contributions, the capacitor voltages are multiplied by the conductance of R_{in}, and the inductor

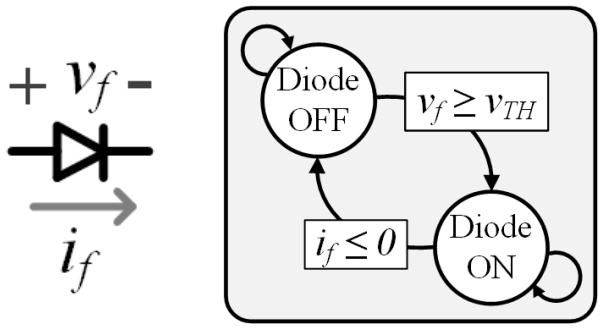


Figure 4-10: (a) Diode and (b) Diode State Machine.

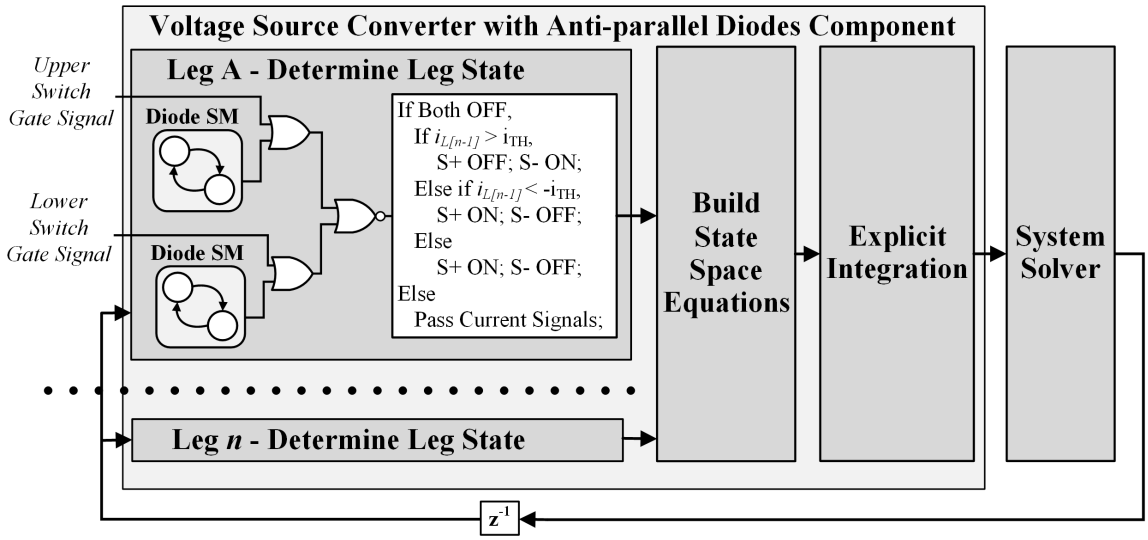


Figure 4-11: VSC with anti-parallel diode component logic flow in LB-LMC.

currents are the source contributions themselves. The source contributions for the whole circuit are then aggregated into the b vector, and the system solution found for x just as mentioned in Section 4.1.

4.4 VSR filter Implementation

An issue which was run into during the design process was that the high-order filter of the VSR consumed a large portion of available DSP resources on the FPGA. Since the filter is integrated with the trapezoidal method, an LCL DM filter, 3 phase filter, will form a 3x3 matrix block. This can be seen in the middle 3x3 matrix block in Fig. 4-12a. This figure shows the sparsity of the G^{-1} matrix for a VSI circuit with an LCL DM filter feeding an RL load. This VSI circuit is essentially the same circuit as the VSR circuit shown in Fig. 4-5, but with dc sources instead of dc loads and ac RL loads instead of ac sources. An LCL DM filter with passive damping ends up forming a 6x6 matrix block. This can be seen in the middle 6x6 matrix block of Fig. 4-12b. The matrix blocks are separated by the explicit integration components used in the circuit, as shown in Fig.

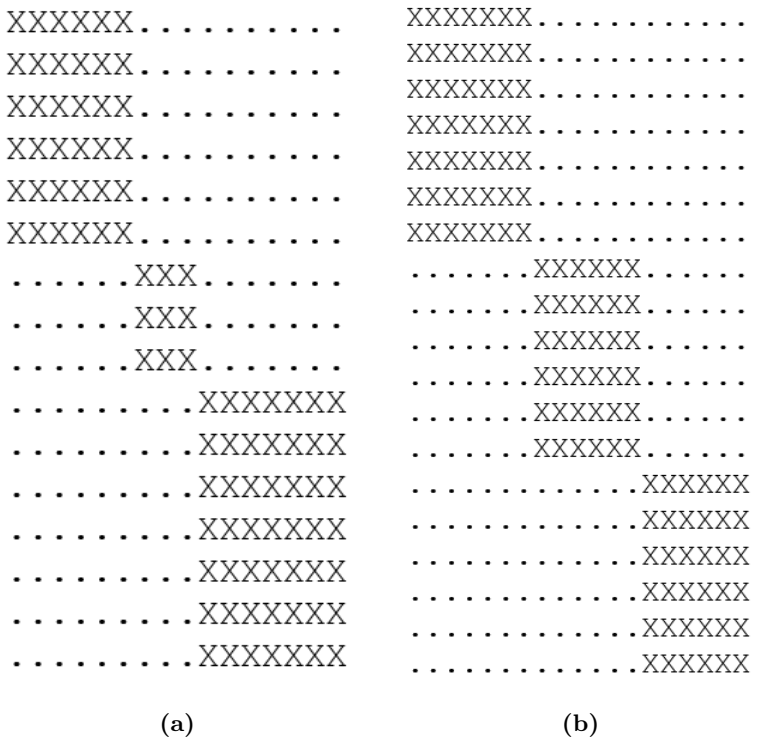
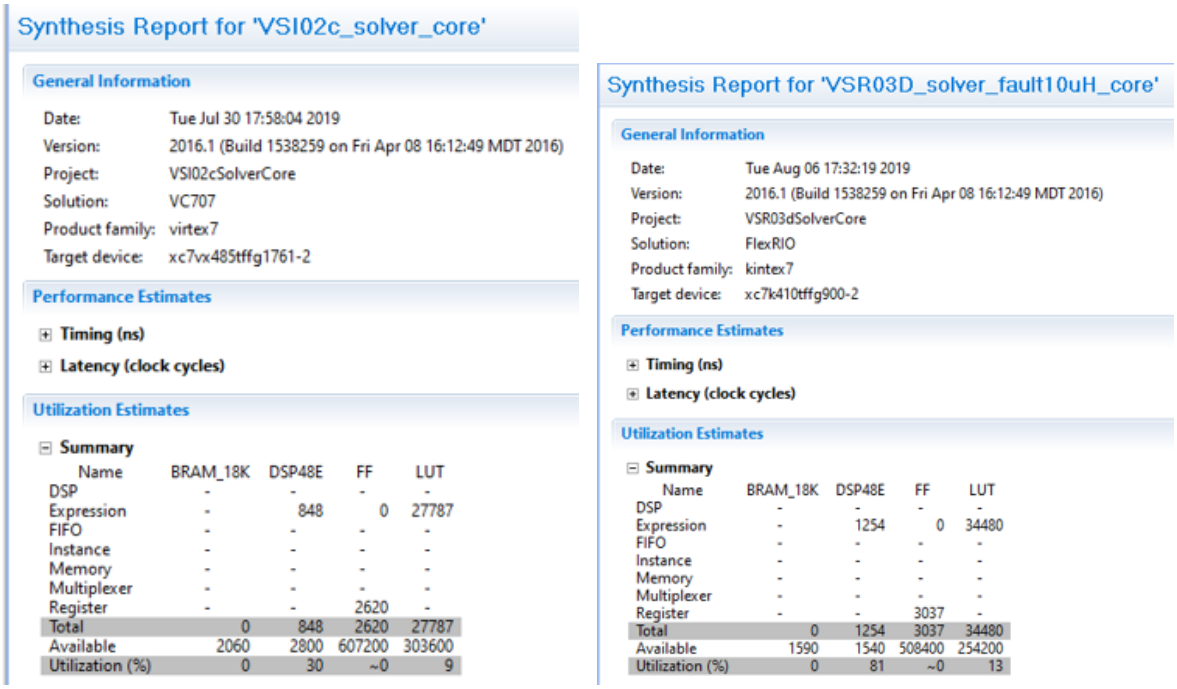


Figure 4-12: G^{-1} sparsity matrix for (a) VSI with LCL DM filter, and (b) VSR with LCL DM filter with passive damping.

4-5. Increased sparsity reduces DSP resource usage on the FPGA at the expense of using explicit integration which may lead solver instability. However, the lower time steps on the order of tens of ns allows for explicit integration to be used without accuracy or stability issues. Still, these trade-offs are something the user must be aware off.

Building a high order filters, as may be needed to meet EMI standards, should be carefully considered as they lead to a large, non-sparse matrix blocks, and consumes a fair amount of DSP resources. Originally, the VSR filter was implemented with an $LCCL$ DM filter with a passive damping and an LCL CM filter. However, this DM filter forms a 9x9 martix block, and due to the over all filter's resource usage, it was reduced to an LCL DM filer with passive damping and an LC CM filter, as shown in 4-4. An attempt was made to reduce the filter to an LCL DM filter without passive damping, but the solver went unstable, and the parameters would need to be adjusted. Sythesis reports for the VSI and VSR circuits can be seen in Fig. 4-13a, and Fig. 4-13b, respectively. Removing the passive damping from the LCL filter reduces the DSP resource usage from by about 400. This equates to about a 25% reduction of the DSP resources on the Kintex-7 410T FPGA for this circuit.



(a) **(b)**

Figure 4-13: Synthesis Reports for (a) VSI circuit, and (b) VSR circuit.

Chapter 5

National Instrument FPGA Platform and LB-LMC Solver Implementation

5.1 NI FPGA Platform for Real-Time Simulation

This chapter provides a discussion on how real-time simulation can be performed on NI FPGA-based platforms using the LB-LMC solver engine and how the controls are implemented for the VSR in LabVIEW FPGA. Further material on this subject can be found in [8].

NI is a manufacturer of widely used computing and instrumentation platforms often applied in system automation, testing, and control, with applications ranging from industrial automation to automotive to wireless to aerospace, just to name a few. Within NI's line of products are many FPGA+processor based computing platforms with modular and numerous Analog and Digital Input/Output (ADIO). Such platforms include the CompactRIO (cRIO), FlexRIO, and PXIe series of products, as depicted in Fig. 5-1. These platforms often incorporate ARM or x86 (Intel Core i and Xeon) based processors with Xilinx FPGAs such as Kintex-7, Kintex Ultrascale, and Zynq devices. Generally speaking, the cRIO provides a small FPGA with ARM-based or Intel-based embedded CPUs with a lot of I/O options that can be removed and swapped. cRIO provides up to 8 slots for different I/O modules, from digital I/O, multichannel lows sampling rate analog I/O, up to 2 channel 1MHz analog input. NI's flagship product is the PXIe chassis. This chassis allows for multiple FPGA processing and FPGA I/O cards, or other types of modules such as vector network analyzers, 5G or industrial communications, various types of analog and digital I/O together in

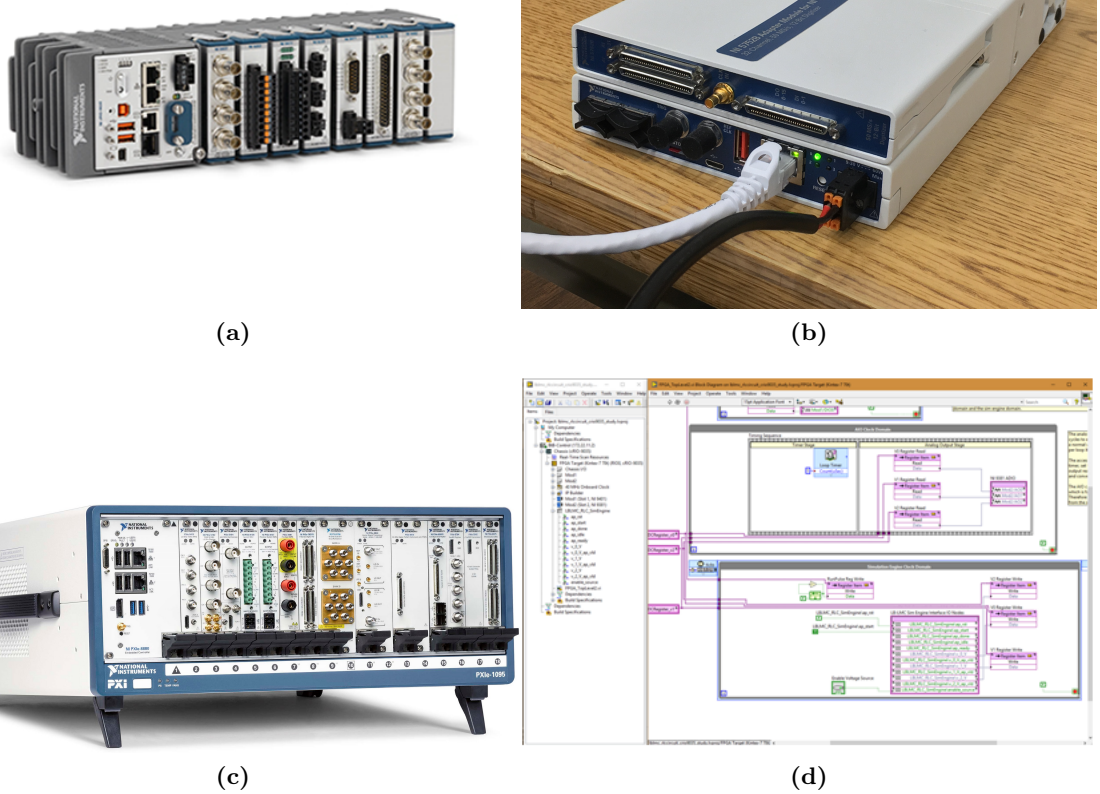


Figure 5-1: NI CPU+FPGA Platforms: (a) CompactRIO, (b) Controller for FlexRIO, and (c) PXIe Chassis; (d) LabVIEW FPGA Software [8].

one platform. The CPU can be higher-end Intel processes such as i7's or Xeon's. Most recently, NI released FlexRIO coprocessing modules based on the Kintex UltraScale series, but most of the FlexRIO I/O-type modules are either Kintex-7 series or Virtex-5 series. These I/O modules are more on the oscilloscope range with bandwidth ranging from 10's MHz to a few GHz. At the time of this work, NI does not offer Kintex UlstraScale+ FPGA options. In between these two platforms, NI offers the Controller for FlexRIO, showing in Fig. 5-1b, which is a stand-alone platform for running one FlexRIO FPGA, and one I/O module. At the time of writing this, the options FPGAs are the Kintex-7 series 325T, or 410T. The 410T version is the hardware platform used in Section 6.1.

Because these platforms are capable of CPU+FPGA architectures and I/O options, real-time simulation, HiL testing, and CHiL can be conveniently performed entirely on these platforms.

To perform real-time simulation, the LB-LMC solver engines can be implemented on the FPGA portions of the NI platforms. Furthermore, due to the ability to rack-and-stack multiple FPGAs into a single PXIe chassis, many FPGA based real-time solvers can be paralleled in one common platform.

NI offers a wide range of software tools. The most common software tool for NI platforms is LabVIEW. LabVIEW provides a graphical software environment, as shown in 5-1d, to allow users to rapidly program real-time targets, such as embedded and FPGA based systems. Moreover, LabVIEW can be thought of a software suite with different version and flavors depending on the target platform and application. For example, when programming real-time targets with an embedded processor, LabVIEW Real-Time is used. For programming FPGA based targets, LabVIEW FPGA is used. For running LabVIEW on desktops, just "regular" LabVIEW is used. Similar to Simulink, logic in LabVIEW is connected in a graphical signal flow like manner. When building logic in LabVIEW, one window is used for the logic flow, and another is used for the user interface. This user interface is versatile, and can display such items as buttons, graphs, dials, knobs, etc, or any other items a human-machine interface would have. Very fitting to the application, these files are called Virtual Instruments (VI). One .VI is written per target. Taking the FlexRIO used in this work as an example, one .VI program is used for the embedded cpu and another .VI is used for the FPGA.

NI LabVIEW allows for easy integration of I/Os, as I/Os are just another logic block, which can be written to or read from, and easy incorporated into the overall code. Using I/Os is simple for both embedded CPU and FPGA targets. Having simple I/O integration is a huge positive for the NI platform. Other code generation methods, such as Xilinx System Generator for DSP (FPGA code generation with Simulink), only generates the code for an FPGA entity for said Simulink Logic. This FPGA entity then needs to be integrated into a top-level design, which varies per FPGA chip and pcb assembly. Futuremore, if ADCs/DACs functionality is desired in the application, then the FPGA must interface to external ADCs/DACs over some type of communication protocol, such as Serial Peripheral Interface (SPI), as FPGAs do not natively have this functionality built into the silicon. This results in the user having to manually interface with the ADCs/DAC through additional Hardware Description Language (HDL) code, such as VHDL or Verilog, in the top-level design. Since NI makes both the hardware and software, they are able to integrate between the

hardware and software, which allows the FPGA code generation to integrate with NI's ADCs/DACs. This is a feature is generally not present in other FPGA code generation platforms. More over, LabVIEW FPGA is able take the .vi, and after compilation, download the compiled code straight the FPGA without the user needing to integrate the compiled code into a top-level design, etc.

There is always a trade-off between platform cost, engineering time, and performance. NI platforms generally cost more than FPGA development boards, such as Xilinx VC707 evaluation boards used in [40], [7], but NI platforms generally require much less intimate knowledge FPGAs to be productive. For reference, the author has taken one formal FPGA class at UW-Milwaukee. This class, along with the NI tutorials, gave mostly a sufficient background on FPGAs and using LabVIEW FPGA to be productive with the FlexRIO platform. LabVIEW FPGA provides a nice balance between control of the FPGA code and a level of abstractions, as generally abstraction and control are conflicting objects. In the authors experience, building most logic is straight forward with LabVIEW FPGA, but performing more complex tasks such as using the SFP+ based protocols, like Xilinx Aurora for FPGA-to-FPGA communication, proved more complicated on the FlexRIO than on a regular Xilinx VC707 evaluation board. In terms of cost, a cRIO chassis + I/Os ranges in the lower $1,000s. A FlexRIO chassis ranges in the upper $1,000's, and with an I/O module, can cost in the range of lower $10,000s. A PXIe chassis, depending on the I/O configuration can range from lower $10,000s to up to a couple $100,000s. For comparison, an equivalent Xilinx FPGA evaluation board comparable with a FlexRIO, costs in the lower $1,000s. But although the FlexRIO cost more, it also has an ARM processor, easy to interface I/O and lower development times for most applications. For a university laboratory research environment, the reduced learning curve is highly desirable as students programs are for a limited number of year compared to industry. Additionally, many students, including the author, do not have an extensive enough background in FPGAs to perform complex research objects on FPGA evaluation boards. A down side of using NI FPGA platforms is sometime the NI FPGA hardware does not use the latest FPGA, such as the Xilinx UltraScale+ FPGAs, where as these large FPGAs are available in evaluation boards.

LabVIEW FPGA also allows for external/custom HDL code integration into a .VI design through what NI calls Component Level Intellectual Property. Since the LB-LMC solver is an FPGA entity, the solver can imported via a wizard into a CLIP, and the integrated into the LabVIEW FPGA code, as shown in Fig. 5-2a. A CLIP core it just a wrapper around the HDL entity to

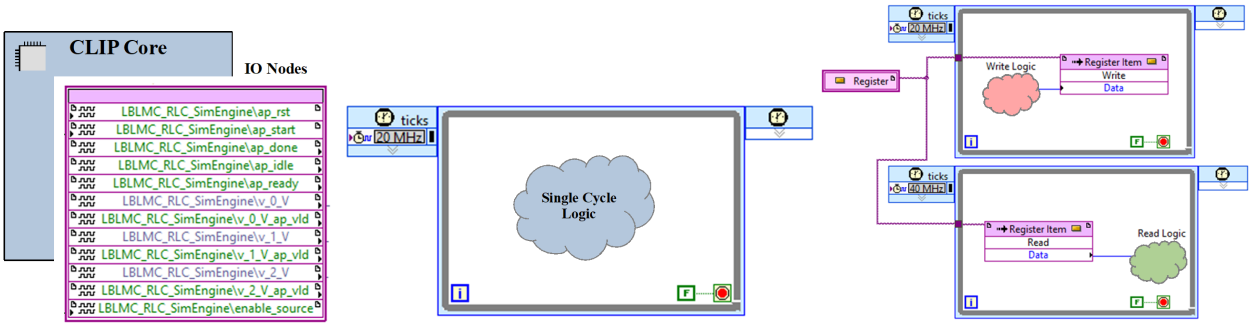


Figure 5-2: LabVIEW Simulation Setup: (a) CLIP core with I/O Nodes, (b) Single-Cycle Timed Loop (SCTL), and (c) Clock Domain Synchronization between Resources with Registers [8].

allow integration with the .VI. The CLIP core's I/O block elements correspond to ports signals in the HDL entity. This allows the user to send and receive signal to and from the custom HDL code. In the case, having the LB-LMC solver in a CLIP core allows for sending and receiving signals to an from the real-time solver.

The LB-LMC solver executes in real-time within a single clock cycle on FPGAs. Thus, the CLIP core is placed in a Single-Cycle Timed Loop (SCTL), as shown in Fig. 5-2b. In general, SCTLs forces timing requirements during compilation on the logic inside the loop structure. If timing fails, LabVIEW will let the user know which part of the logic failed timing. The time step which the LB-LMC solver was designed to execute at should match the clock period of the SCTL. For example, in this work, the LB-LMC solver is set to execute at 50 ns, so the frequency of the SCTL is set to 20 MHz. The base clock on the Kintex-7 410T is 40 MHz, so a 20 MHz clock is derived, then used to drive the SCTL and the CLIP core. The data going to and coming from the CLIP core inside the SCTL is synchronized to the 20 MHz clock.

For crossing clock domains, and moving data around on the FPGA between loops, LabVIEW FPGA provides a few options such as synchronization registers, First-In-First-Out (FIFO), or global variables to name a few. An easy choice one can implement is synchronization registers, as shown in Fig. 5-2c. This allows the user to move data between different clock domains by simply writing to and reading from a register. The clock domain crossing through registers has a few clock cycles of overhead. The registers do not guarantee lossless data transfer, and can only be read from one location at a time. Examples of using synchronization registers is shown throughout Section 5.3.

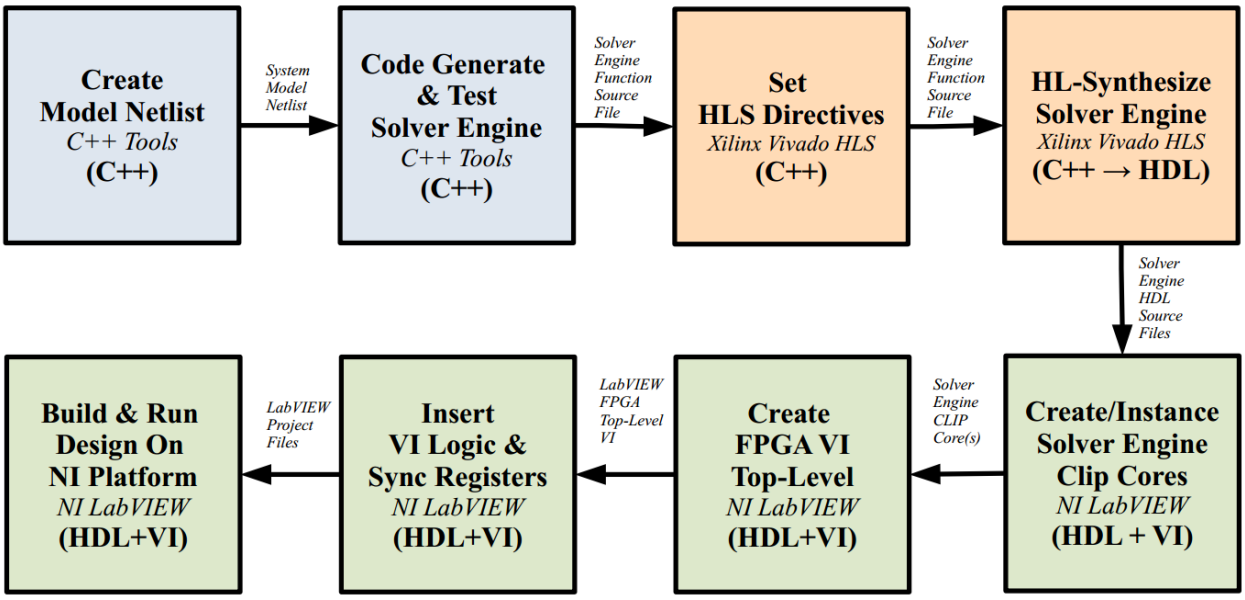


Figure 5-3: LB-LMC NI Platform Solver Engine Development Flow [8].

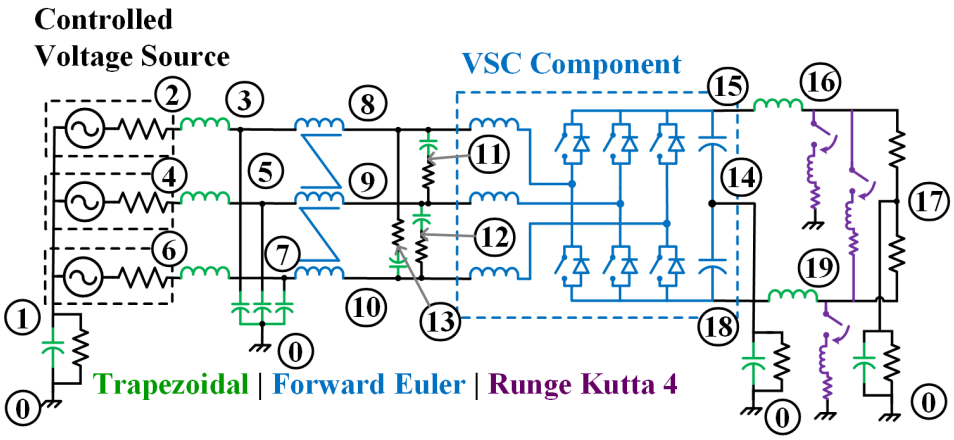


Figure 5-4: Netlist of VSR circuit.

If lossless data transfer is required FIFOs should be used. FIFOs are discussed in detail in Section 5.4.1.

5.2 NI FPGA Platform Solver Development Flow

This section provides a summary on how the real-time LB-LMC solver engine for a power electronic system is developed and then incorporated into a NI platform application. Further details on this process can be found in [8].

Fig. 5-3 shows the tool-chain and development flow used for creating an LB-LMC solver engine

for real-time FPGA execution on NI platforms. The first step involves defining a netlist of the circuit, similar to old SPICE modeling before Graphical User Interfaces (GUIs) were developed. An example of the circuit netlist in this work is shown in Fig. 5-4. This netlist is then implemented into a Command Line Interface (CLI) tool developed by Matthew Milton [55]. The parameters and netlist are listed out in a text file according to the tools syntax. At the time of this work, various components are available in the tool library such as functional and ideal voltage and current sources, standard passive components such resistors, capacitors, inductors, and 3-phase mutual inductors, a few types of PECs such as VSC, half-bridge MMC and DAB, and an RL switch for applying faults. These various components are called out in the netlist. An example of the text file for the CLI tool used in this work is provided in Appendix A.6. Then, an executable file is run which takes netlist file location as an input parameter. The executable file then generates an LB-LMC simulation engine of the netlist circuit in C++ code. This C++ code is a function, which can then be integrated into a test bench file to validate correct behaviour of the solver.

Once behaviour is consider correct, this solver is then imported into an High-Level Synthesis (HLS) tool, such as Xilinx Vivado HLS. This tool converters the C++ code automatically into either VHDL or Verilog. The user test can the LB-LMC simulation engine in fixed point to validation sufficient accuracy. HLS directives are used to set timing requirements, latency, and resource usage requirements. An example of using the HLS directives for this project is included in Appendix A.7 in lines 36, 37, 40, 43, and 105. These directives enable the FPGA to execute in a single clock cycle, tell the signal port to not have memory interfaces, which would add delays. They also tell the signals port to output array elements in parallel, and to place any nested functions into one large function, which eliminates additional signal interfacing and delays. LabVIEW FPGA's largest datatype is 64-bits, while signal width in the LB-LMC solver engine for this design was chosen as 69-bits of 28-integer and 41-fractional bits. To make the LB-LMC solver engine compatible with LabVIEW FPGA, the ports have to be no more than 64-bits. Additional logic is added which converts the data types of the input signals from 64-bits to 69-bits by padding zeros to the end, and converts output signals from 69-bits to 64-bits by truncating the last 5 fractional bits. This can be seen in lines 67-69, and 109-123, respectively, in Appendix A.7. The design is then complied, and an VHDL entity is now made. This VHDL entity is then imported into LabVIEW FPGA via an import wizard to form a CLIP core. With this, the LB-LMC simulation engine is now imported

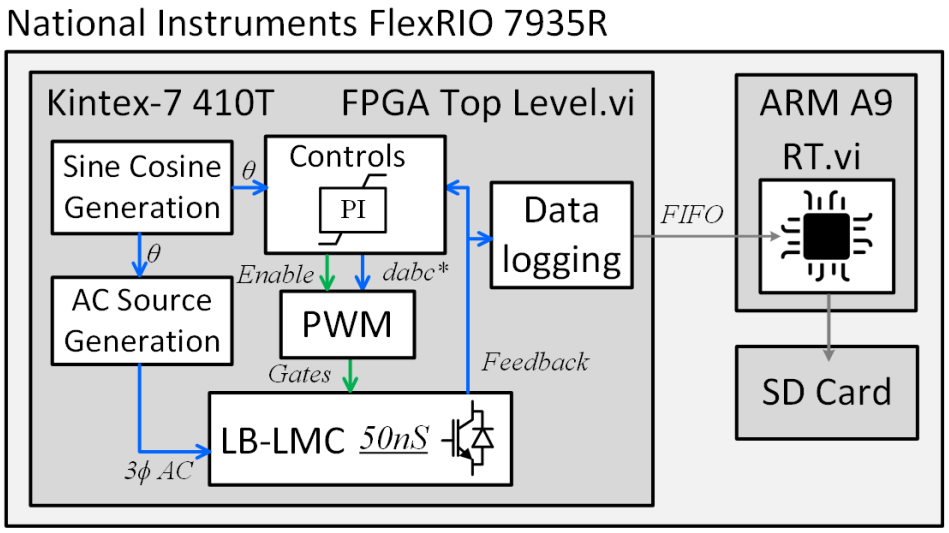


Figure 5-5: Block diagram of top level VI in LabVIEW FPGA

into LabVIEW FPGA and ready to use for HiL testing.

The author would like to would to acknowledge and thank Matthew Milton, Dr. Andrea Benigni, and the team at University of South Carolina (USC) for the C++ code, CLI tool, code generation library, and showing the author how to implement the various software tools presented here.

5.3 Controls and PWM Implementation in LabVIEW FPGA

Fig. 5-5 shows a block diagram of the Top Level VI implemented in LabVIEW FPGA. The hardware used was NI FlexRIO, model number NI-7935R (Fig. 5-1b). This particular FlexRIO model has a Xilinx Kintex-7 410T FPGA, where the LB-LMC solver, controls, and PWM are implemented. Sine and Cosine lookup tables are used to generate the $\sin\theta$'s and $\cos\theta$'s need for power-variant dq and inverse dq transforms using in the controls, as well as to synthesize the 3 phase ac input signals to the LB-LMC solver. Registers are used to cross clock domains. For data logging, LabVIEW FPGA utilized FIFOs to transfer data between the FPGA and ARM processor. Then, the ARM processor saves the data to an SD card. Due to limitations in data logging ability, longer time data are captured at a lower resolution, and the shorter times are captured at a higher resolution. All vi's and subvi's shown in this chapter were made by the author.

For the active rectifier, a 2-level converter was used, but in practical application, a high number of levels would be used to reach a 12 kV bus such as the NPC described in Chapter 3. The source

is modeled as an ideal 3-phase voltage source of 6 kV_{LL} RMS with a feeder cable. The feeder cable represents resistance and inductance from cabling and a synchronous generator's stator windings. The controls utilize decoupled voltage oriented controls in the rotating dq frame [56], but with 3^{rd} harmonic injection and per-unitized input signals, as shown in Fig. 5-11. The power invariant dq and inverse dq transforms are given by (5.1) and (5.2), respectively, where θ is the electrical angle.

$$\begin{bmatrix} d \\ q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{5.1}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} d \\ q \end{bmatrix} \tag{5.2}$$

As mentioned above, to reduce the computational burden on the FPGA of doing trig multiplication, 6 lookup tables were implemented, one for each of the angles required for the dq transform; however, these lookup tables are not able to be implemented inside SCTLs, and are instead implemented in while loops. The lookup tables are set to 60 Hz, 16 bits, with different offsets of $2\pi/3$ for the different terms. The lookup tables can also be 8 or 32 bits, and can vary in size. A screenshot is provided in Fig. 5-6. The output of the lookup table is signed integer. The controls were implemented in fixed-point, so a reinterpret block was added to move the decimal to the designated fixed-point value. These values are then saved to registers to be used in other loops, either SCTLs or while loops on the FPGA. If a register is read from multiple places at the same time in LabVIEW FPGA, data loss can occur. For this reason, the output of each lookup table saves the same data to 3 or 4 different registers to be used later at different parts of the vi, such as 1 for dq transform of voltage, 1 for currents, 1 for dq inverse transform of modulation indexes, and 1 for the 3 phase ac input. This is shown in Fig. 5-7.

Fig. 5-8 shows a PI (Proportional Integral) block diagram with clamping integral logic. This logic prevents the integral term from blowing up if the output of the PI is saturated. Also, the integral term does not engage until the enable signal is high, and clears when the enable signal is low. In a standard Simulink model, the integrator accounts for the time step in its integration, i.e.,

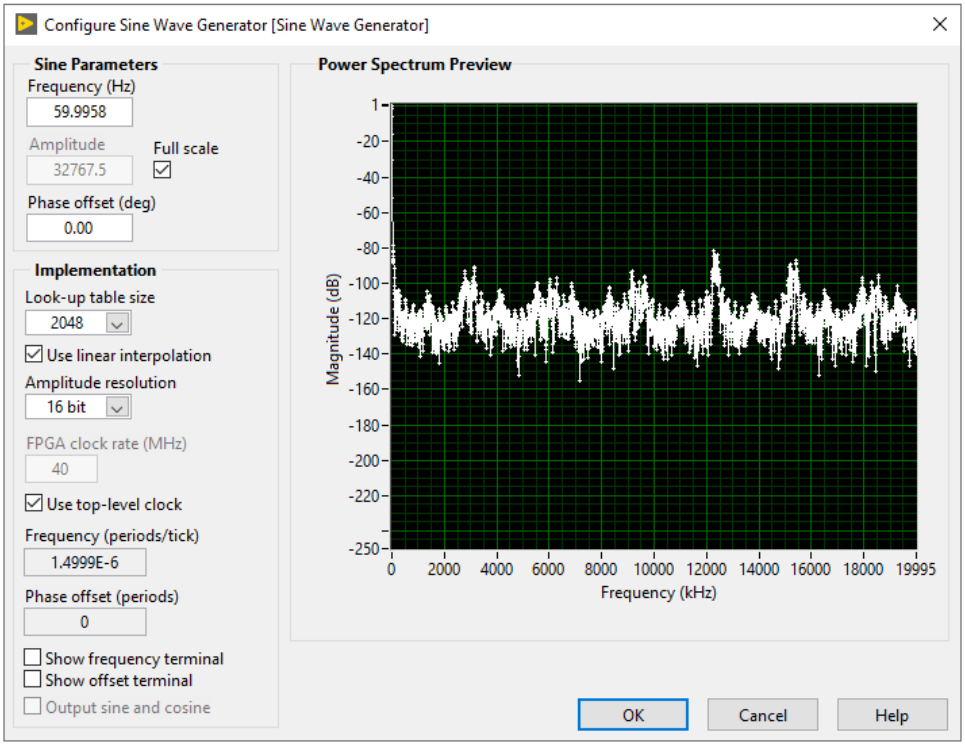


Figure 5-6: LabVIEW FPGA - sin lookup table configurator

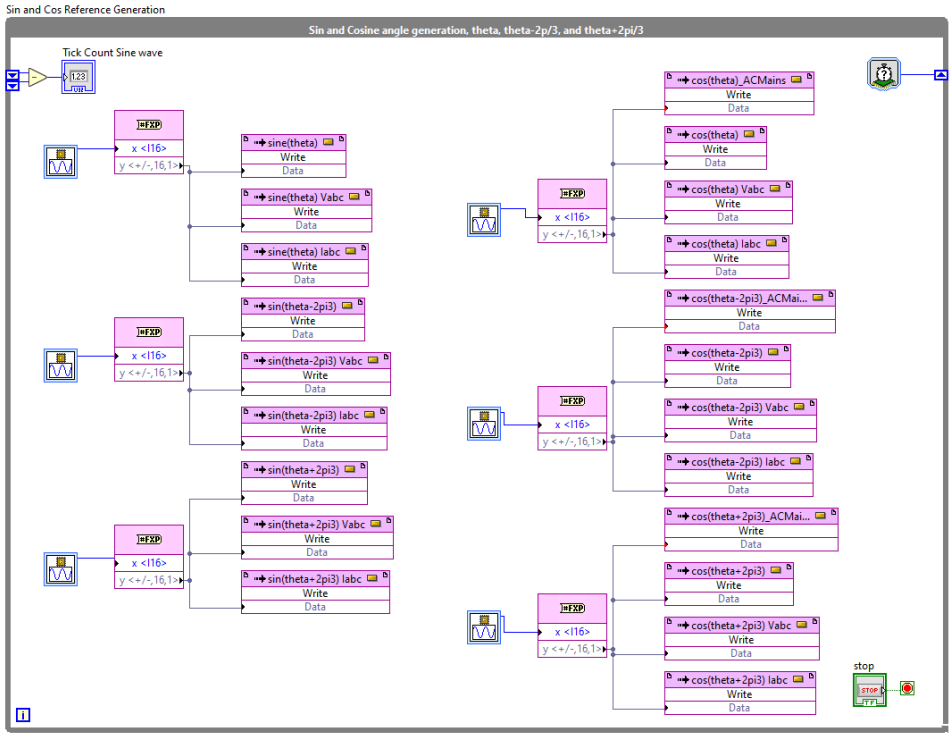


Figure 5-7: LabVIEW FPGA - saving *sin* and *cos* angles for *dq* transforms

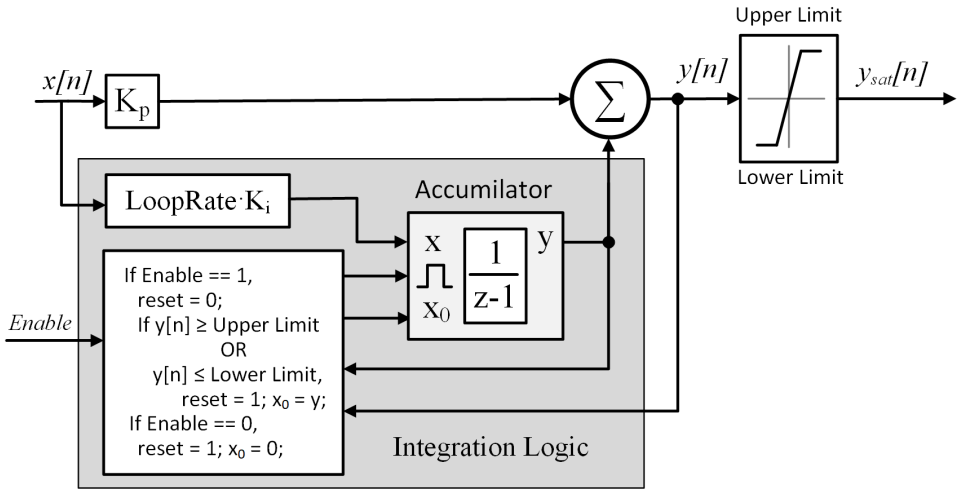


Figure 5-8: PI block with clamping integrator and saturation

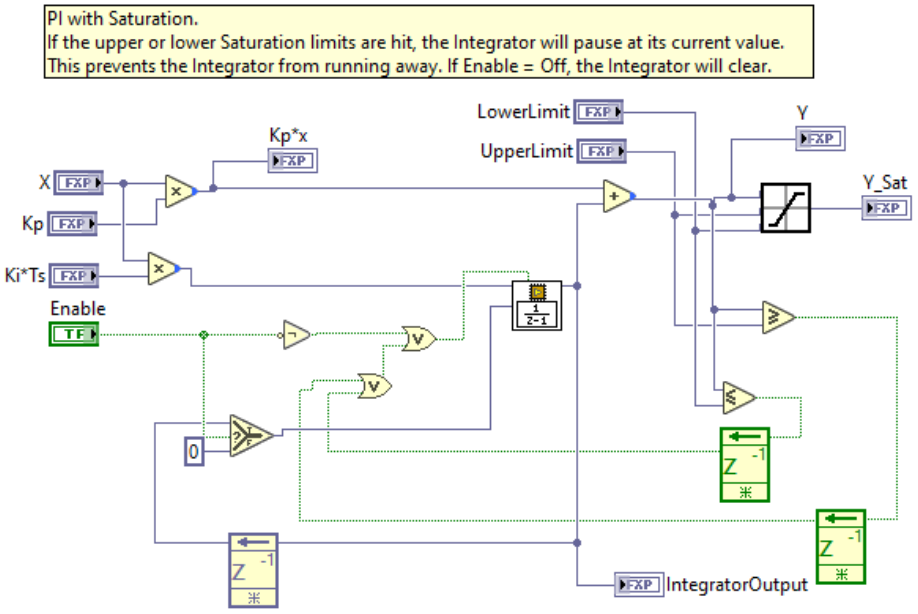


Figure 5-9: LabVIEW FPGA implementation of PI controller

the integral gain is multiplied by the time step automatically. However, in FPGA implementation, this does not automatically occur. What is conventionally thought of as an integrator is really an accumulator. To account for this, the loop rate for the SCTL must be manually multiplied by the integral gain. In the authors' experience with LabVIEW FPGA, the additional logic of the clamping integrator must be implemented in an SCTL, as opposed to while loops or other timing structures. The LabVIEW implementation of the PI block is shown in Fig. 5-9. A saturation block was made as well, shown in Fig. 5-10.

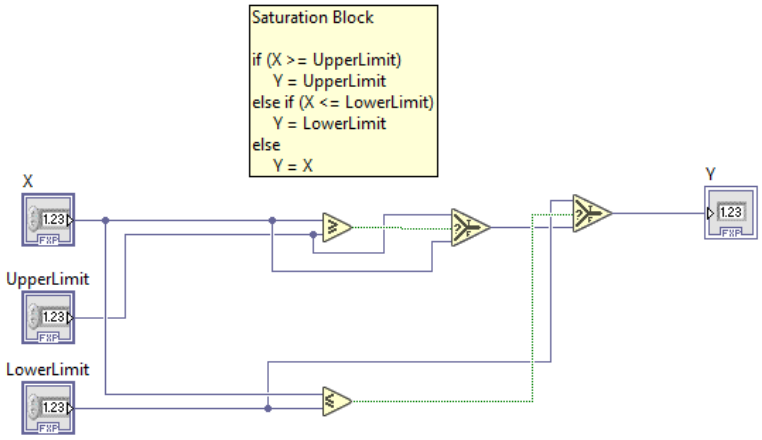


Figure 5-10: LabVIEW FPGA implementation of saturation block

With all of these pieces in place, we now have all the building blocks needed for the main VSR controls. A block diagram of the controls is shown in Fig. 5-11. The LabVIEW FPGA implementation is shown in Fig. 5-13. The controls were implemented in a 40-bit fixed-point, with an 8-bit integer part, and a 32-bit fractional part. This is not claiming to be optimal, but enough bits where the resolution would not be an issue. The resolution can be reduced if the area becomes limited on the FPGA. In general, when working with fixed-point in LabVIEW FPGA, the software will automatically increase the fixed-point signal size to account for the largest possible and smallest possible signal size. If multiplication is performed on many signals in a row, the signal size can quickly expand from a 4- or 8-bit word to a 64 bit word. The output size of mathematical operations can be manually set in LabVIEW FPGA, as well as for any signal. Incite from the designer on the appropriate signal size can be used to set the signal size sufficiently large enough to handle expected signal transients, but no larger to not use more resources than necessary. To NI's credit, this is very easy to set in the properties of the constant or math operation, as LabVIEW FPGA displays the range of the signal and precision of the fractional portion. Fig. 5-12 shows an example of the lower limit constant used, which feeds into the saturation blocks.

In Fig. 5-13, one may notice the long trailing decimals. This is due to the fixed-point representation, which, unlike floating-point, does not have the ability to create rational fractional-parts, so many fractional terms cannot be perfectly represented. For example, the upper and lower saturation limits for the *idq* PI controllers are to 1.2 and -1.2, respectively. The 0.2 term of 1.2 cannot perfectly represented with the 40-bit word chosen, and is instead represented as

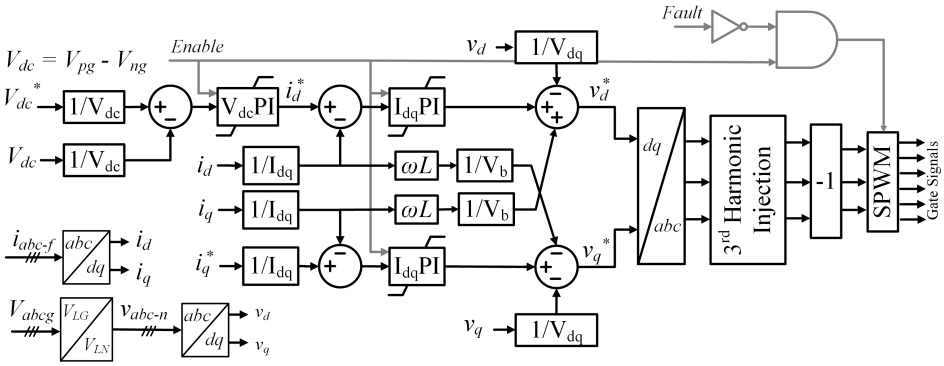


Figure 5-11: Decoupled Voltage Orient Controls for Voltage Source Rectifier

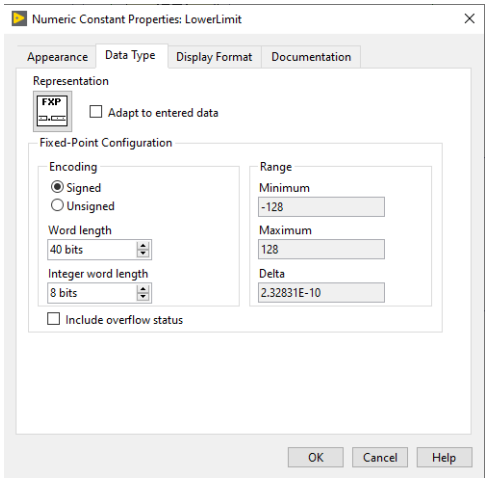


Figure 5-12: LabVIEW FPGA fixed-point properties for a contant

"1.19999999995343387126922607421875." This obviously is close enough to 1.2 due to the 32 fractional bits, but just something the designer needs to keep in mind when using fixed-point in LabVIEW FPGA.

One key point on implementing logic inside an SCTL is division operations are not allowed because divisions are an iterative operation and thus cannot be performed in a single clock cycle. The controller gains were designed in Simulink and are based on the per-unitized signals. Since divisions are not allowed, multiplications by the inverse of the constants was used. Constants such as these were manually entered.

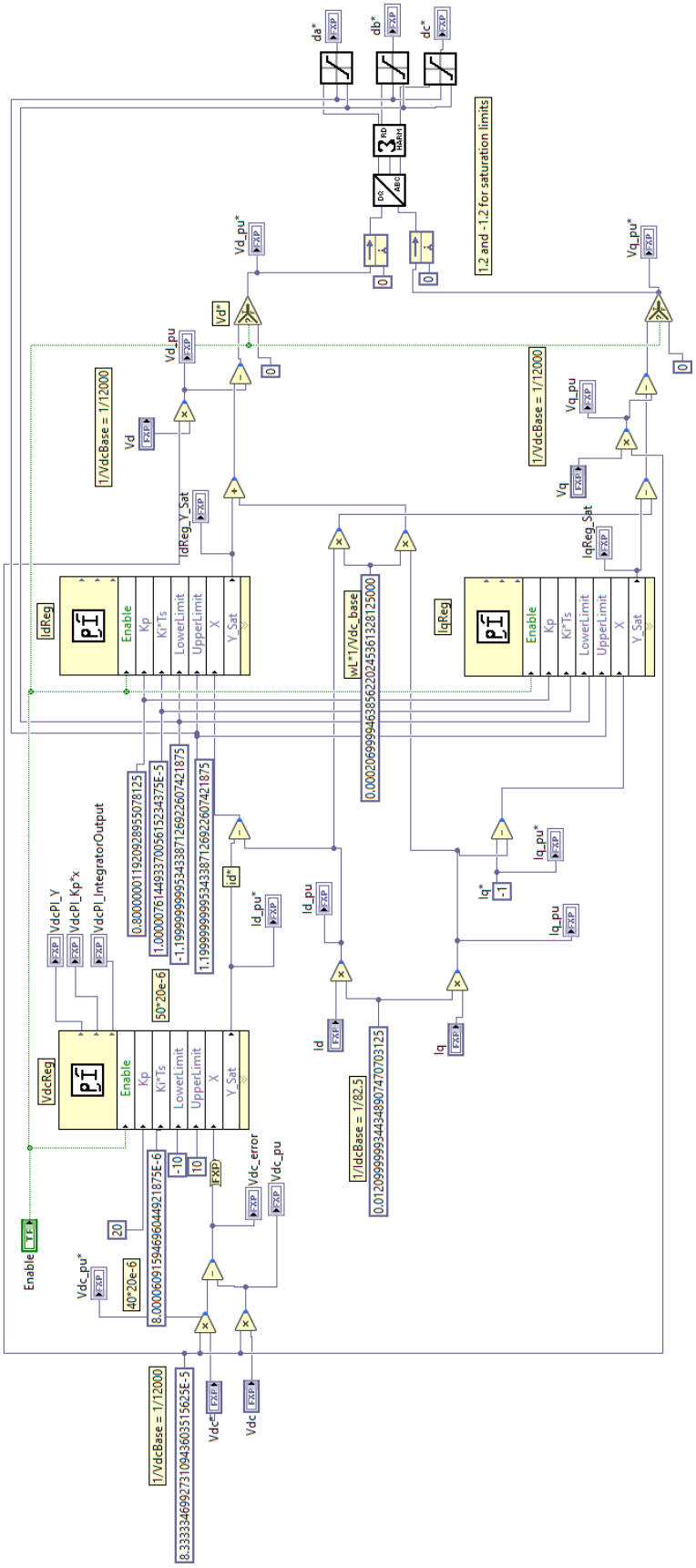


Figure 5-13: LabVIEW FPGA implementation of decoupled VOC controller

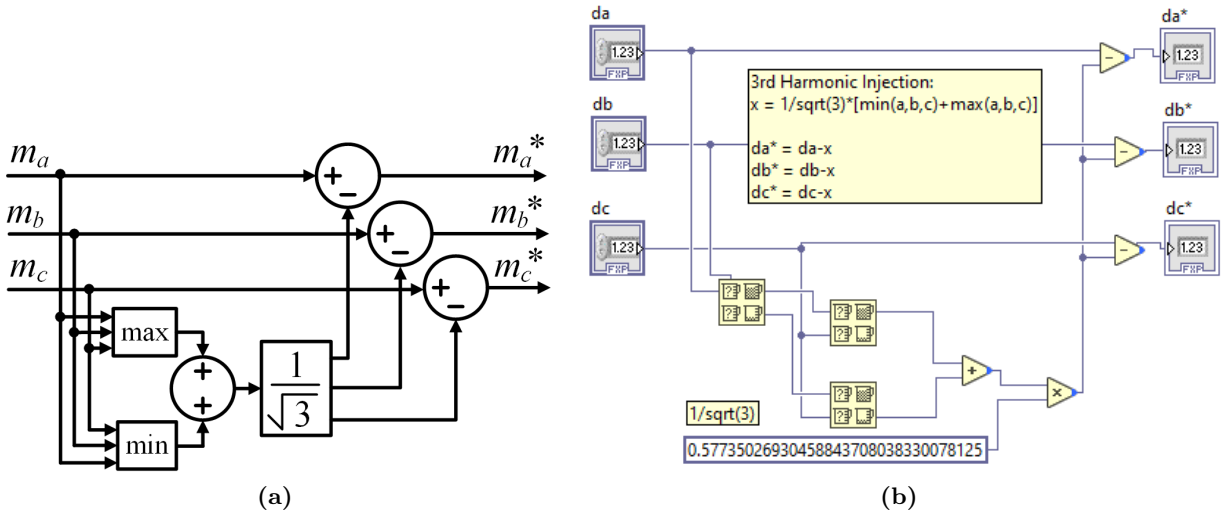
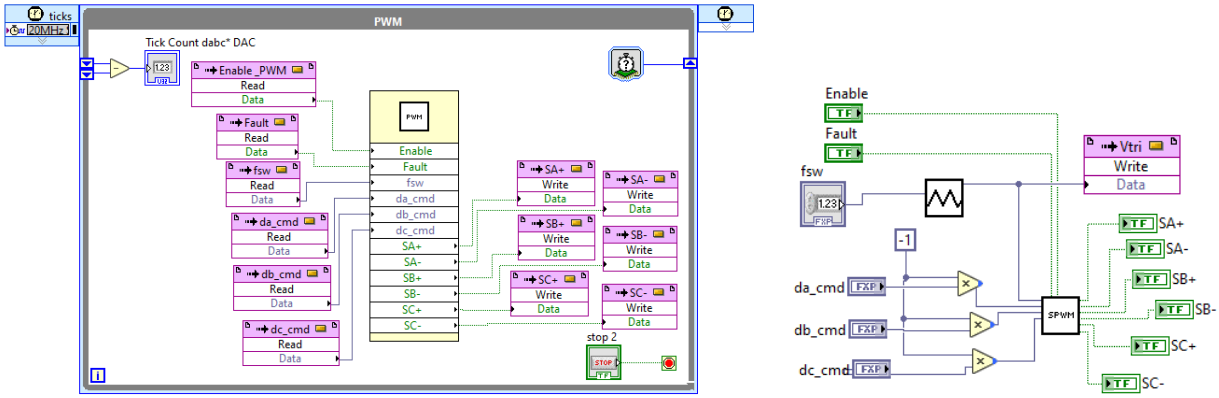


Figure 5-14: 3^{rd} harmonic injection (a) diagram, and (b) LabVIEW FPGA implementation

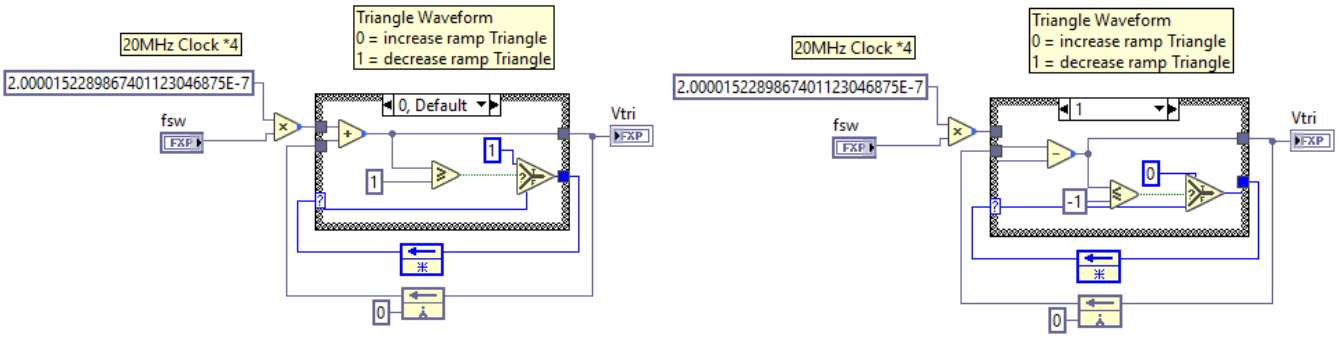
After the inverse dq transform, the 3^{rd} harmonic injection is added to the modulation signal. This increases bus voltage utilization by about 15% [57]. This means for the same bus voltage, the RMS output of an inverter can be 15% greater. Fig. 5-14a shows a block diagram of 3^{rd} harmonic injection and Fig. 5-14b shows its implementation in LabVIEW FPGA.

Fig. 5-15a shows the PWM.vi in the top level. It is implemented in its own SCTL with registers reading the duty cycles from the output of the controls, and write the outputs to registers which go to the gate signals of the simulation engine. This could have also been implemented in the same SCTL as the LB-LMC simulation engine. Fig. 5-15b shows the internals of the PWM.vi. It contains a triangle wave generator block and a PWM comparator block for a 3 phase converter. The triangle wave block generators a triangle waveform between -1 and 1, with a period of the switching period of the converter. The state machine for the triangle wave contains two states: ramping up, and ramping down, shown in Fig. 5-16. It starts at zero then counts up to 1. Once the count is equal to 1, then it counts down to -1, then back up to 1, and so on. A triangle wave ramping up and down between -1 and 1 and back to -1 has an effective slope of 4. Or thought of another way, this triangle wave is essentially an integrator with a gain of 4. If a switching frequency (f_{sw}) of 1 was desired, then the clock frequency would be multiplied by 4 and either added or subtracted onto itself. To increase the frequency of the triangle, one can simply increase the term, which is added or subtracted with each iteration. To set the frequency of the triangle wave generator at the f_{sw},



(a) (b)

Figure 5-15: LabVIEW FPGA PWM.vi (a) in Toplevel.vi, and (b) internals.



(a) (b)

Figure 5-16: LabVIEW FPGA implementation in fixed-point of triangle wave: (a) ramping up state, and (b) ramping down state.

one can multiply the clock period by 4 and by the switching frequency. This works for fixed-point representation. A similar approach can also be done with integer data types as well, but is not covered here.

To generate the AC mains input signals, which feeds into the real-time simulator as voltage source, the cos angles saved to registers earlier are multiplied by the line to neutral amplitude, then saved back into a register to be used in a different SCTL, as shown in 5-18. This multiplication could take place in another part of the .vi, not necessarily in an independent SCTL.

Fig. 5-19 shows the CLIP core of the LB-LMC simulation engine containing the VSR circuit. One can see the registers sending the signals of AC mains, data signals, and fault application and removal on the left side of the figure. The right side of the figure shows the various measurement points from inductor currents and node voltages being saved to registers, which will be either used

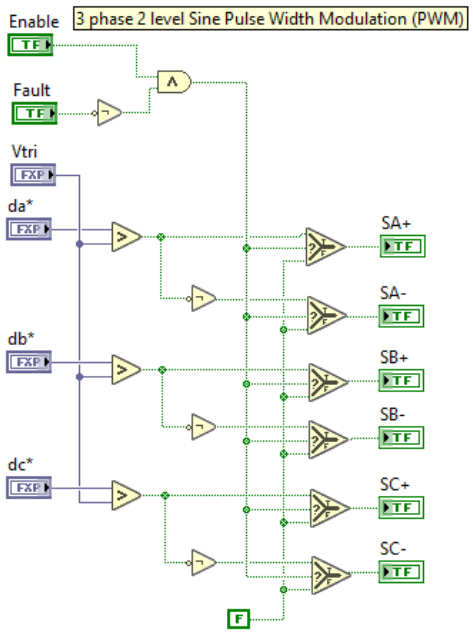


Figure 5-17: LabVIEW FPGA SPWM implementation

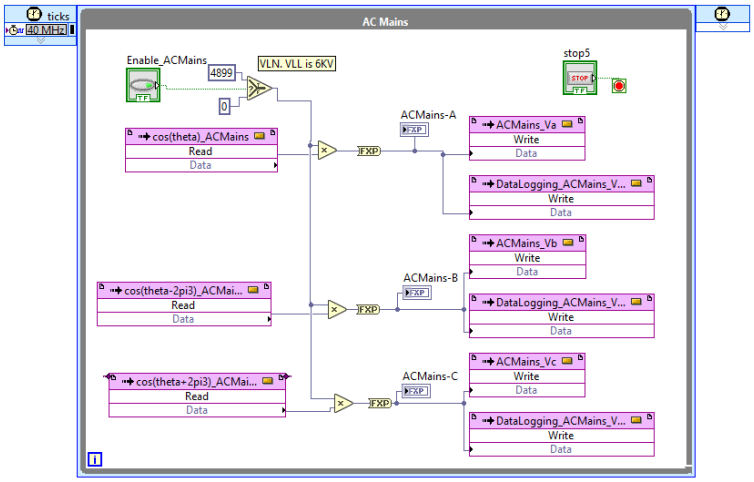


Figure 5-18: LabVIEW FPGA AC Mains implementation

for controls feedback or data logging.

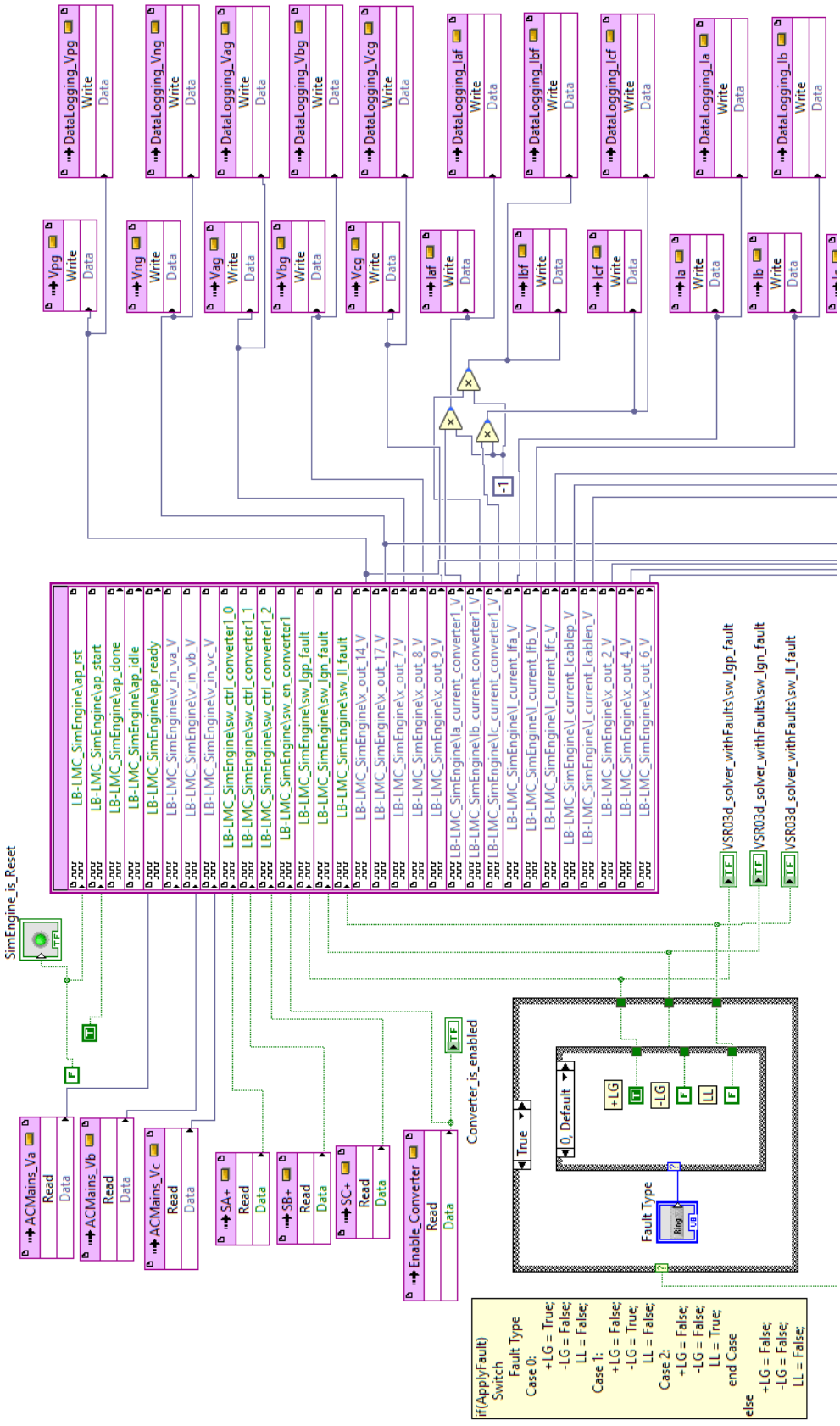


Figure 5-19: LabVIEW FPGA implementation of CLIP core contain LB-LMC simulation engine

5.4 Data logging on FlexRIO

5.4.1 Data logging in LabVIEW FPGA

The challenge of data logging on an FPGA, is the FPGA itself cannot inherently store data, nor can it directly communicate/read/write to something like a hard-drive or SD card. To save data to a file, or in this case, an SD card, the data must pass through the embedded CPU. The embedded processor is made up of a Xilinx Zynq-7020 System-on-Chip (SoC), which contains a dual-core ARM Cortex-A9 32-bit processor running NI Linux Real-Time at a clock speed of 667 MHz. The non-volatile storage of the system is only 512 MB, which is why a 32 GB SD card was used.

The main task from the user's perspective is to get the data from the FPGA to the ARM processor and saved to the SD card. Sending data between the FPGA and ARM processor is done using Direct Access Memory (DMA) FIFOs. DMA FIFOs can be configured based on the elements writing/reading to it at a time, the data type, the FIFO size, and the resource making up the FIFO. The FIFO can be made up of Flip Flops, Lookup Tables (LUTs), or Block Random Access Memory (BRAM). There is a trade-off between the writing speed and the size of memory based on the type of resource being used, but generally, BRAM is sufficiently fast and the largest memory-related resource available on the FPGA silicon. The Flip Flops are more likely used for FIFOs at higher clock rates. The Kintex-7 410T FPGA model used in the FlexRIO has 28,620 kbits of BRAM, which translates to 29,306,880 individual bits. From this, the user can divide this by the number of bits per element and the number of elements per write cycle to find how deep the FIFO can be before running out of resources. This FPGA supports up to 16 different DMA channels, but for this work, only one large DMA channel was used. All the elements written to the same channel must be the same data type. Different channels can be used when logging different data types or writing at different speeds. Fig. 5-20 shows how the lower speed data logging was performed. This data was sampled at 10 or 20 kHz. The "sampling" rate was achieved with a loop timer and logic implemented in flat sequence structure. Every n amount of μs, or in this case, every 50 or 100 μs, the sequence would kick-off, then move the next frame where the data was read from the registers and written to the FIFO. Regardless of how long the latter sequence takes, the timer in the first frame ensures the logic starts executing every n amount of μs. The loop timer can also be changed to be controlled in units of clock tick, or ms. This structure is then implemented in a true/false

case statement controlled by a button on the front panel. This allows the user to start and stop data logging at their discretion. If there is an error, such as an overflow, a fault signal flag will be raised, which will stop the loop and display on the front panel to let the user know something went wrong.

The main limitation of data logging is how fast the CPU side can read from the FIFO, while the FPGA is writing to the FIFO. If the CPU cannot read as fast as the FPGA is writing to the FIFO, the FIFO will overflow and error out. So if continuous data logging is desired, i.e., "live streaming" the data, a trade-off must be made between data type, number of elements, and data sampling rate. To capture data at 20 MHz, a "burst data capture" approach was used, where the data was written to the FIFO for a short amount of time, then read on the CPU side. A screenshot is shown in Fig. 5-21, which in the Top-Level.vi is located right under the LB-LMC simulation engine in Fig. 5-19. This was done with a state machine that would enable the 50 ns data logging for a few clock periods to capture the data at steady state. Then, apply the fault for a few hundred clock period, and remove the fault for a few hundred clock period. Lastly, the state machine will disable writing to FIFO. Enclosing the FIFO block in a True/False case statement allows for the enable/disable behavior.

An alternative to using the BRAM is using the Dynamic Random Access Memory (DRAM). DRAM is the same type of memory used on standard desktops and laptops. It is not located on the FPGA silicon, but is on the PCB assembly. NI provides an example.vi on reading/writing to the DRAM, as there is a lot of communication and data manipulation overhead to getting data from the FPGA to the DRAM and back. The example.vi can be modified and used by the user. The amount of DRAM which comes on the FlexRIO used in this work is 2 Gigabytes, which translates to 16 Gigabits. Compared to the 28 kbits of BRAM, the DRAM offers 586x the amount of memory. This translates to 586x longer waveforms being saving at the MHz level. However, in the author's experience, modifying the single-channel example of reading/writing to the DRAM working with 4 channels was not as straight forward as expected, and the author was not able to get this working. Fig. 5-22 shows how the DRAM example would have been integrated into the LabVIEW FPGA shown above.

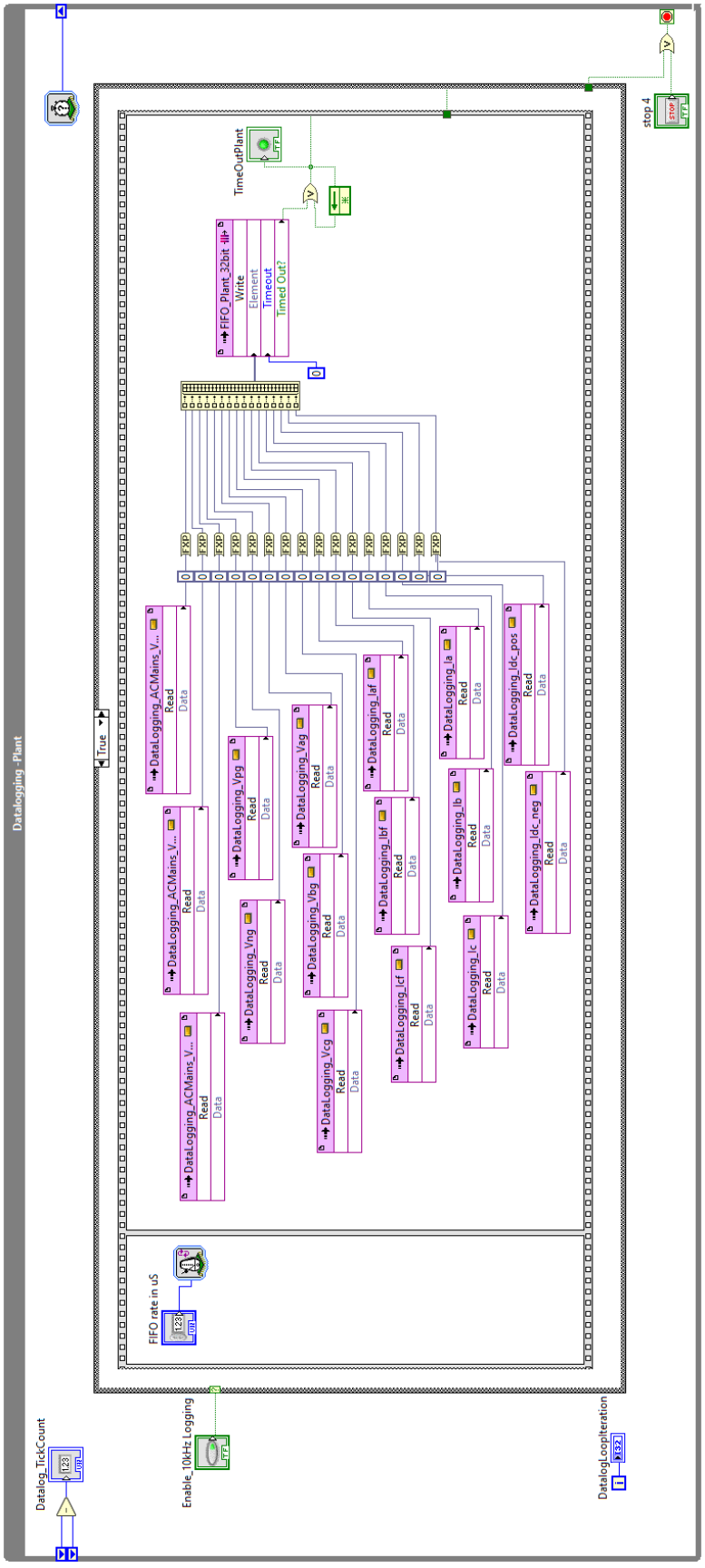


Figure 5-20: LabVIEW FPGA Data logging at 10 or 20 kHz

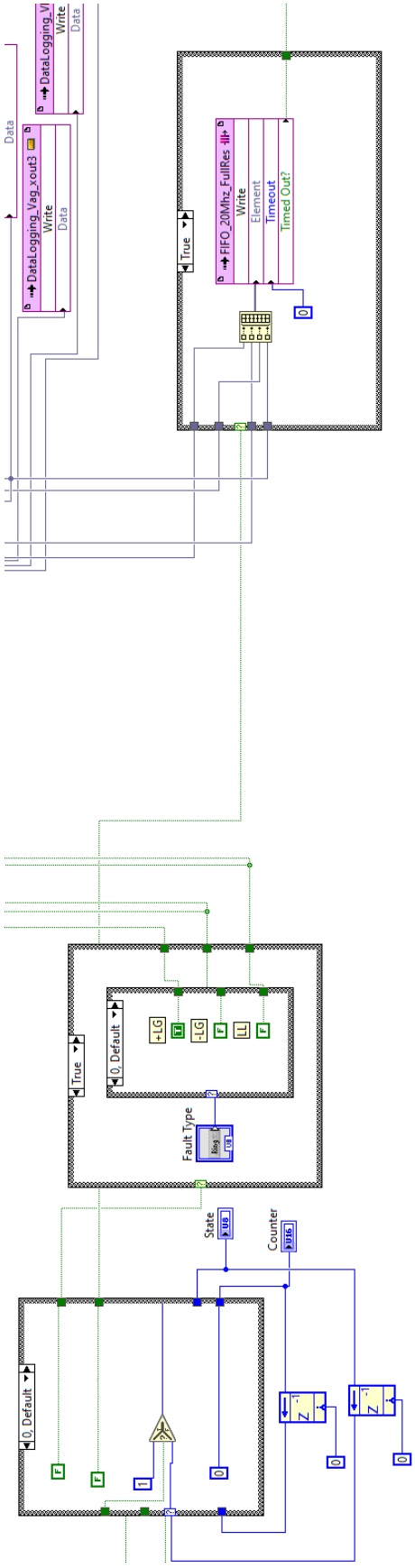


Figure 5-21: LabVIEW FPGA Data logging at 20 MHz

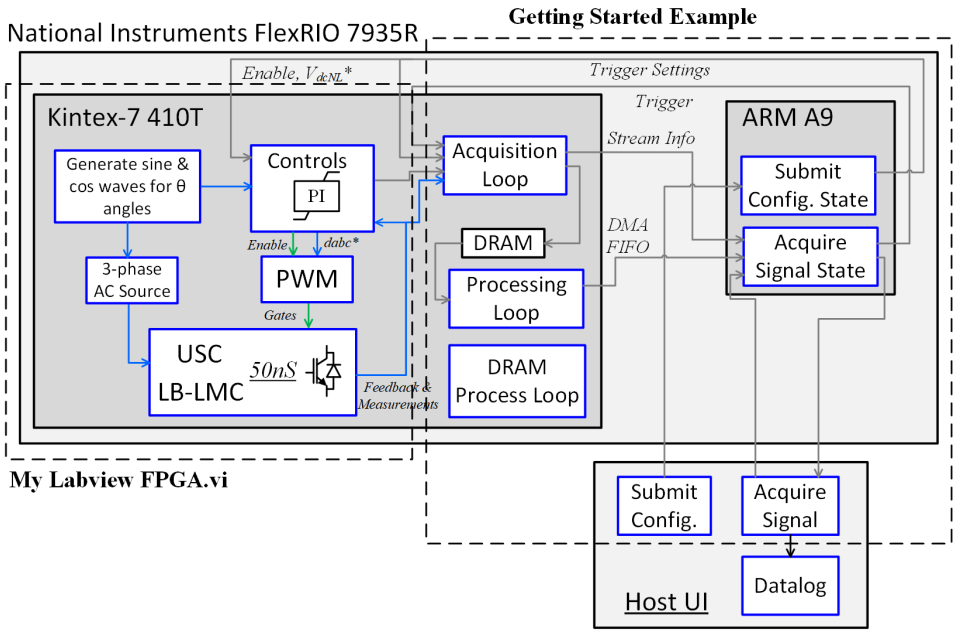


Figure 5-22: LabVIEW FPGA Data logging using DRAM

5.4.2 LabVIEW Real-Time of Data logging

This subsection describes the embedded CPU real-time side of the data logging after the FPGA has written to the DMA FIFO. However, before the reading from the DMA FIFO can begin, the real-time.vi must be configured probably. Fig. 5-23 shows an example of initial configuration, file creation, and data logging using in this work. First, the resource name block is used to select the FPGA target on the chassis. The FlexRIO only has one FPGA target, but a PXIe chassis could contain multiple FPGA targets. Then, the FPGA bitstream (the compiled FPGA code) is selected and loaded onto the FPGA. A read-write block is used to set initial condition prior to FPGA execution, such as switching frequency, FIFO sampling rating, Vdc controller command, and various flag initialization. This block has a small icon of a pair of glasses and a pencil to symbolize reading and writing. Next, the FIFO is configured and started prior to the actual execution of the FPGA target. Lastly, the FPGA target is given the "run" command to start real-time execution.

From there, the real-time.vi splits off into two parallel loops, one for data logging (Fig. 5-23) and one for real-time monitoring and control (Fig. 5-24). Data logging starts when the user selects an enable button on the front panel. Then, the .vi creates a Technical Data Management Streaming (TDMS) file with a date and timestamped file name. Setting the file directory to "/u" saves the file on the SD card. TDMS is NI's proprietary file format which was developed for handing high

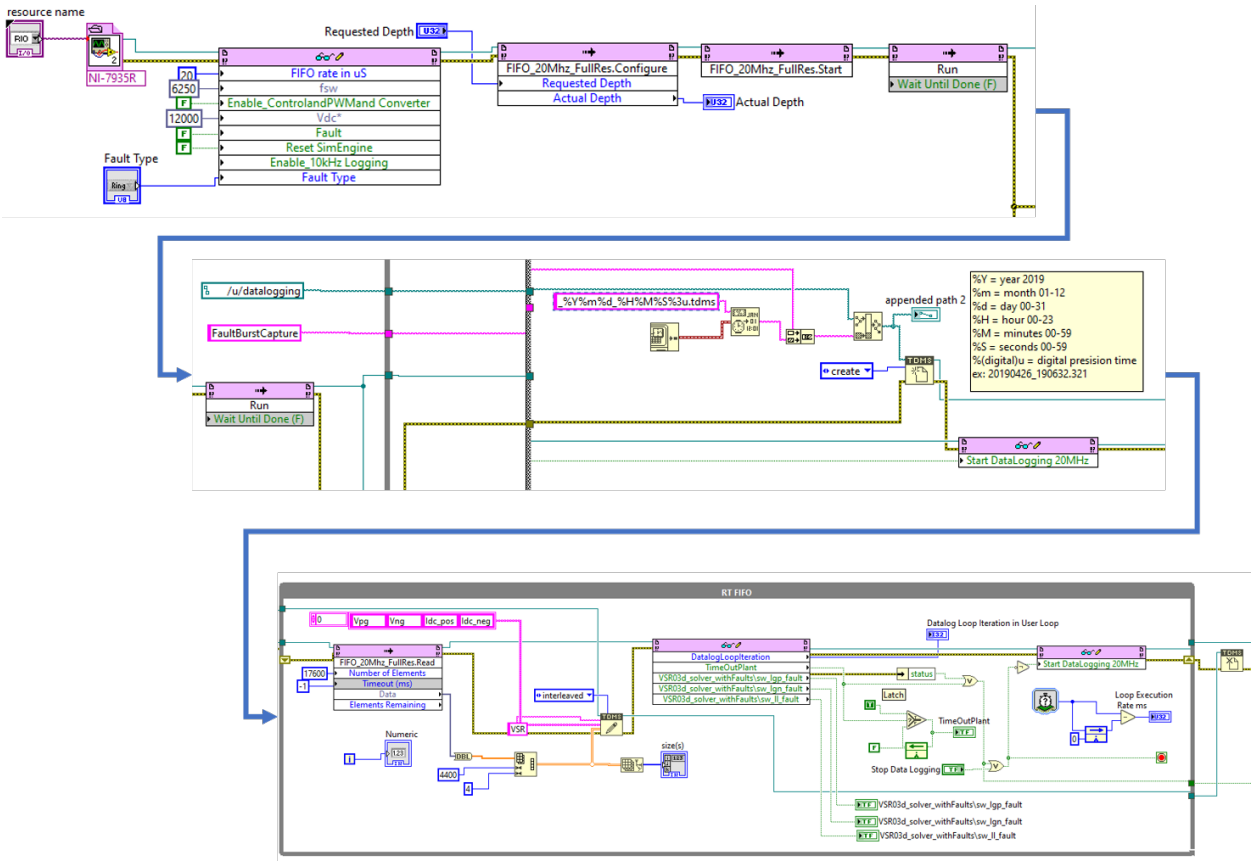


Figure 5-23: LabVIEW real-time example of data logging

data rates and Gigabits of data in a single file. Options for data logging in .csv and a few other file formats are also available. Once the TDMS file is created, a command from the real-time side is sent to the FPGA to enable one of the case statements in the FPGA.vi, such as those shown in Fig. 5-20 or Fig. 5-21, for the FPGA to start writing to the DMA FIFO. The real-time.vi will enter a while loop and begin reading from the DMA FIFO. The data from the DMA FIFO gets converted from fixed point to double-precision floating point (double), as the TDMS file requires doubles as the data type. Next, the data gets manipulated from a single column array to a 4 column array, one for each of the 4 channels being saved. Finally, the data is written to the TDMS file. The while loop keeps running until the FIFO is empty, the user selects to stop data logging, or an error has occurred, such as a FIFO overflow. The loop structure was setup so the user could log multiple TDMS files while the system is running.

Fig. 5-24 shows the monitoring and control loop, which runs in parallel to the data logging loop. This loop allows the user to enable ac mains, enable the controls, set the vdc command for

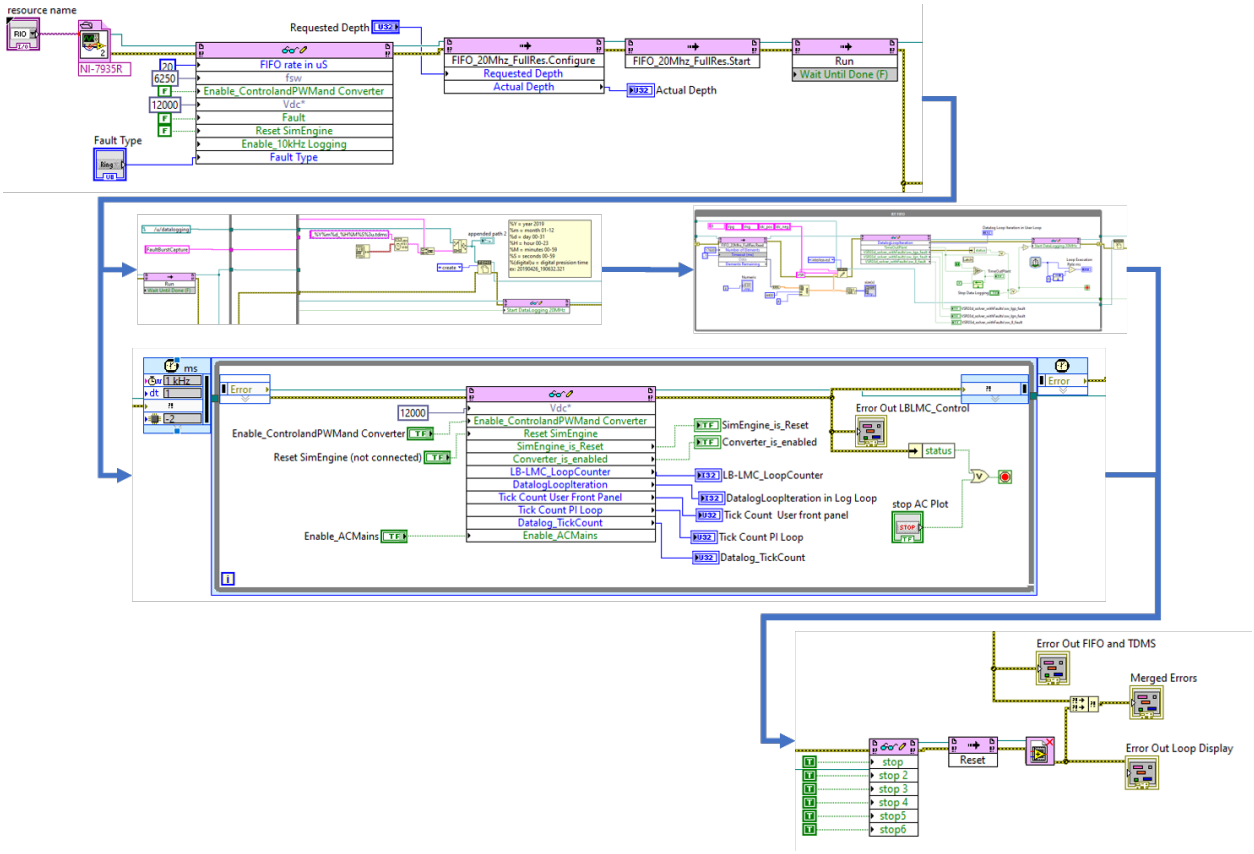


Figure 5-24: LabVIEW real-time example of monitoring

the controller, and monitor different statuses within the FPGA will the simulation is running in real-time.

If an error occurs or the user decides to finish the simulation, then both loops are exited, commands are sent to the FPGA to stop the various loop inside the FPGA, the DMA FIFOs are reset, and the FPGA reference is closed. If the DMA FIFOs are not reset, then when starting them up again in a later execution, the data from the previous run will still be present. If the reference to the FPGA is not closed, then the FPGA will still be running. Lastly, any errors occurred will display on the front panel.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Real-Time Simulation Results

6.1 Real-Time Simulation of Common Mode and +LG Fault in an Ungrounded dc System

Fig. 6-1 shows the same VSR circuit discussed in Chapter 4, which was simulated in real time for both LG and LL faults in this Chapter.

Within this circuit, LG ac and dc voltage measurements are used in the simulation, but for the controls, their respective DM parts are extracted. Table 6.1 shows the equations used to extract DM and CM from MM for 2 and 3 phases for voltages and currents. IT grounding is modeled with a capacitor (C_{og}) and resistor (R_{og}) in parallel to ground, with values of 0.1μF and 10kΩ, respectively. This represents the parasitic capacitance in an ungrounded system [58], [59].

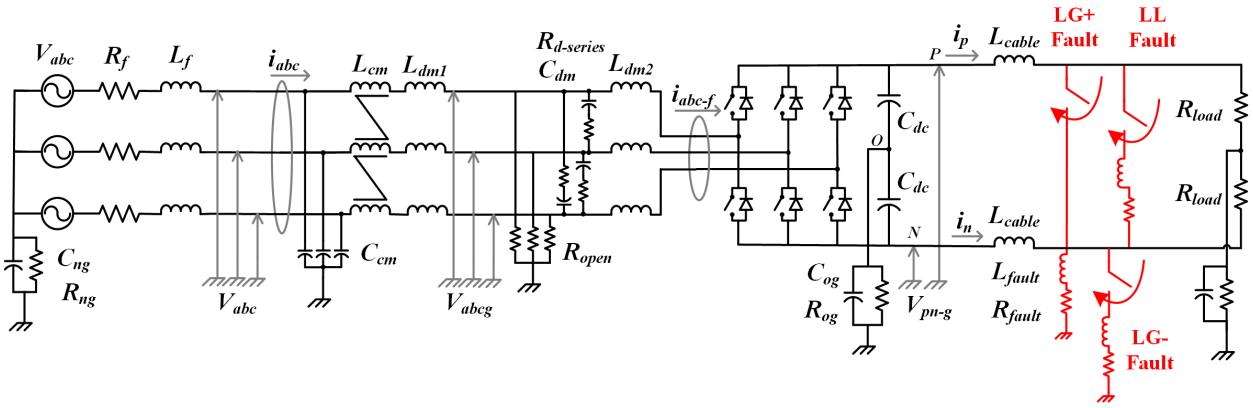


Figure 6-1: VSR Circuit with LCL DM filter, LC CM filter, floating ground, measurement points, +LG, -LG, and LL faults.

Table 6.1: Equations for extracting DM and CM from MM signals

Description	Signals	Mixed Mode	Differential Mode	Common Mode
3-phase Voltage	Vag, Vbg, Vcg	$Vabcg_{MM}$	$Vabc_{DM} = Vabcg_{MM} - Vabcg_{CM}$	$Vabcg_{CM} = (Vag + Vbg + Vcg)/3$
3-phase Current	ia, ib, ic	$iabc_{MM}$	$iabc_{DM} = iabc_{MM} - iabc_{CM}/3$	$iabc_{cm} = ia + ib + ic$
2-phase Voltage	Vpg, Vng	Vdc_{MM}	$Vdc_{DM} = Vpg - Vng$	$Vdc_{CM} = (Vpg + Vng)/2$
2-phase Current	ip, in	idc_{MM}	$idc_{DM} = (ip - in)/2$	$idc_{CM} = ip + in$

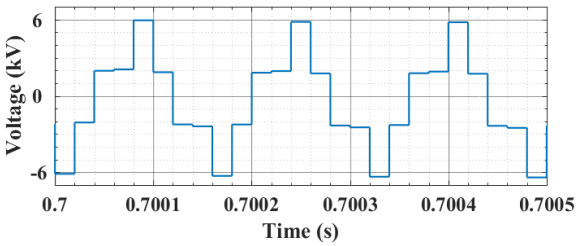


Figure 6-2: CM Voltage at $Vabcg$. Data is captured at 50 kHz.

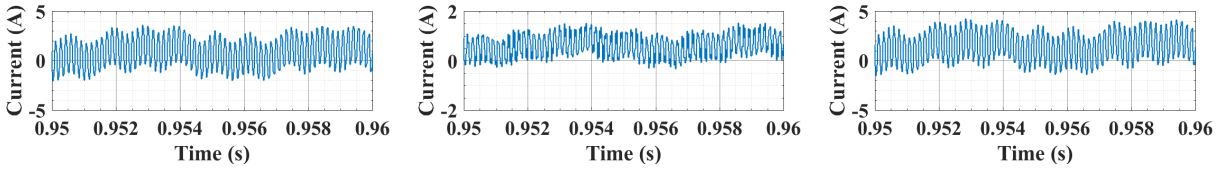


Figure 6-3: VSR CM current at (a) $iabc_f$ (b) $iabc$ (c) idc. Data is captured at 50 kHz.

The parallel resistor R_{og} is present to ensure the bus voltages returns to initial conditions. Changing this resistance affects how long the bus voltage takes to recover after a fault is removed. The voltage recovery is governed by the RC time constant of R_{og} and C_{og}. To get the exact value of the parasitics to ground, one would have to use a Network Vector Analyser (NVA) to extract S-parameters or make an FEA model, for example, with COMSOL Multiphysics [60]. However, this changes with the geometry of the system. So unless the exact system is known beforehand, the protection scheme will have to be modified once it is deployed. The section shows such high-frequency ringing during sudden inception, zero impedance LG faults can be modeled in the real-time simulation. A list of circuit parameters can be found in Table 6.2.

The 2 level, 3 phase VSR produces CM voltage of $\pm V_{dc}/2$, and $\pm V_{dc}/6$ depending on the switching state [61]. With a 12 kV bus, the CM voltage outputs are ± 6 kV and ± 2 kV, as seen in Fig. 6-2. This leads to CM currents, as seen at measurement point $iabc_f$ in Fig. 6-3a. The CM current is reduced after passing through the CM LC filter, Fig. 6-3b. This shows the LB-LMC solver is able to simulate CM currents and CM filters correctly.

Fig. 6-4a shows the path of current during an LG fault, and Fig. 6-4b shows the equivalent circuit during the fault. The equivalent circuit forms an under-damped LC circuit with a resonant frequency of (3.1). For this circuit, L is the sum of the cabling and fault inductance, 20 μH, and C are the parasitic capacitance to ground, 0.1 μF. The value of this capacitance was derived by

Table 6.2: VSR Parameters

Parameter	Description	Value	Unit
Base Power	P_b	1.5	MW
Vdc rated	V_b	12	kV
Base Current, dc	I_b	82.5 A	
ac Source, V_{LL} RMS	V_{abc}	6	kV
ac Source, Frequency	f_e	60	Hz
Base Voltage, ac Line-to-Neutral peak	V_{dq}	4899 V	
Base Current, ac peak	I_{dq}	136 A	
Switching Frequency	f_{sw}	6.25	kHz
Current Controller, Proportional Gain	I_{dq} K_p	0.8	
Current Controller, Integral Gain	I_{dq} K_I	50	
Current Controller, Upper Limit	-	1.15	
Current Controller, Lower Limit	-	-1.15	
Voltage Controller, Proportional Gain	V_{dc} K_p	20	
Voltage Controller, Integral Gain	V_{dc} K_I	40	
Voltage Controller, Upper Limit	-	10	
Voltage Controller, Lower Limit	-	-10	
3^{rd} Harmonic Inject Coefficient	-	$1/\sqrt{3}$	
Feeder Cable, Resistance	R_f	0.198	Ω
Feeder Cable, Inductance	L_f	1.8	mH
CM Filter, Inductors	L_{cm}	60	mH
CM Filter, Capacitors	C_{cm}	0.1	μF
DM Filter, Inductor 1	L_{dm1}	0.955	mH
DM Filter, Inductor 2	L_{dm2}	5.6	mH
DM Filter, Capacitor	C_{dm}	16	μF
DM Filter, Damping Resistor	$R_{d-series}$	55	Ω
dc Link Capacitance	C_{dc}	20	mF
Cable Inductance	L_{cable}	10	μF
Resistive Load	R_{load}	100	Ω
Floating Ground, Capacitance	C_{og}	0.1	μF
Floating Ground, Resistance	R_{og}	10	kΩ
Fault Inductance	L_{fault}	10	μH
Fault Resistance	R_{fault}	1	mΩ

taking a lumped sum of cabling capacitance and baseplate-to-heatsink capacitance. These values do not match exactly those provide in Table 3.2, but showcase this approach of modeling ground faults in IT systems, and the capabilities of the HiL platform. The parasitic capacitance and dc-link capacitance are in series, but since the dc-link cap is many orders of magnitude larger than the parasitic capacitance, it can be ignored. This gives a resonant frequency of 125.4 kHz or a period of 8.89μs, or slightly less than 9 data points per period. This is a poor resolution for a simulation that is trying to *characterize* fault transient to design protection schemes.

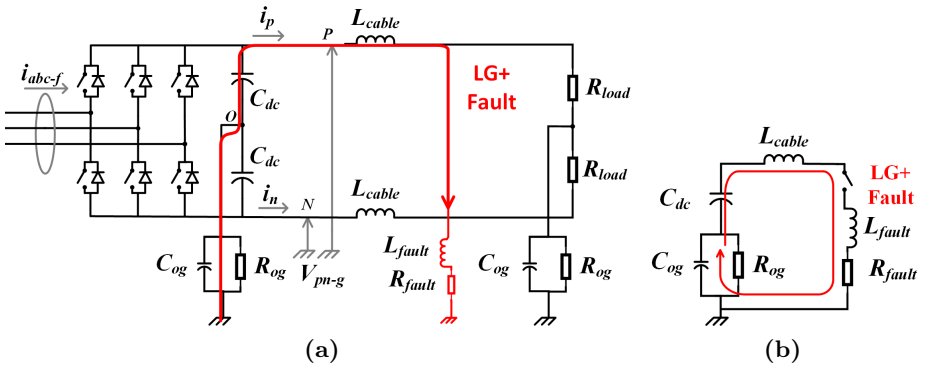


Figure 6-4: DC+ Line to Ground Fault (a) current path and (b) equivalent circuit during the fault.

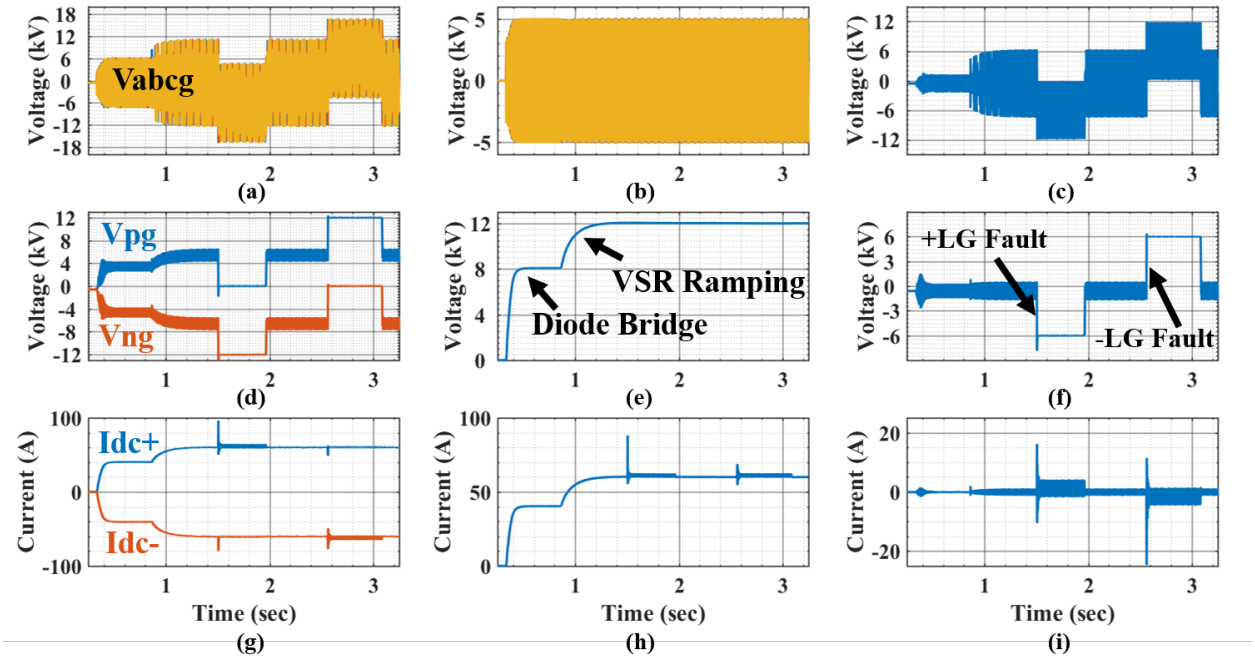


Figure 6-5: VSR ramping up as diode bridge and then actively gated. +LG and -LG are applied. Data is captured at 25 kHz. Left, Middle and Right Columns are MM, CM, and DM, respectively. Vabcg (a) MM (b) DM (c) CM; Vpng (d) MM (e) DM (f) CM; ipn (g) MM (h) DM (j) CM.

Fig. 6-5 shows various measurement points while the VSR ramping up, and the circuit's behavior under +LG and -LG fault conditions. Fig. 6-6 zooms in on the +LG fault application. Both of these figures' rows are ac-side LG voltages ($Vabcg$), dc-side LG voltages ($Vpng$), and dc-side currents (ipn). These figures' columns are broken down by MM, DM, and CM according to Table 6.1.

During an LG fault in an ungrounded system, the DM voltage remains constant while the CM shifts [62]. In normal operation, positive Vdc rail-to-ground (Vpg) is +6 kV, and negative Vdc

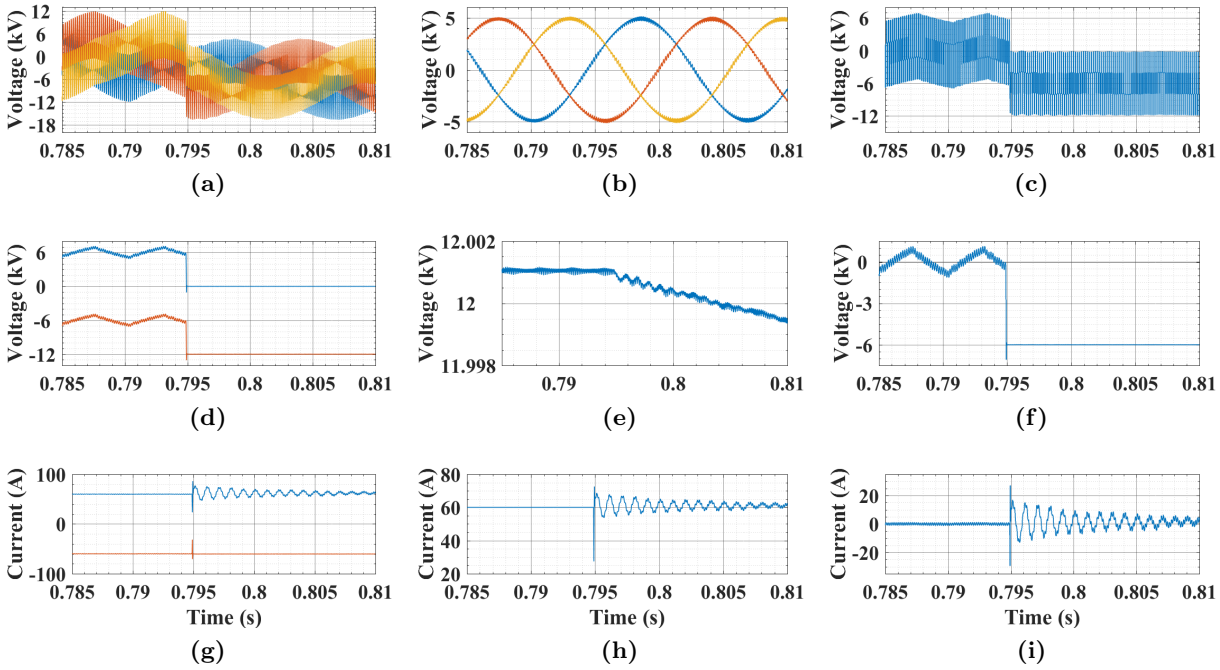


Figure 6-6: +LG fault at 0.795 seconds. Data is captured at 50 kHz. Left, Middle and Right Columns are MM, CM, and DM, respectively. Vabcg (a) MM (b) DM (c) CM; Vpng (d) MM (e) DM (f) CM; ipn (g) MM (h) DM (j) CM.

rail-to-ground (Vng) is -6 kV, giving a DM voltage (Vdc_{DM}) of 12 kV. Applying a +LG fault on the positive dc rail forces the positive rail to zero volts. The ungrounded system will keep the 12 kV DM voltage, and shift the negative rail to -12 kV. When the fault is removed, the voltage will return to nominal. All the non-electrically isolated parts of the distribution system will shift voltages, including the ac-side voltages, as shown in Fig. 6-5a and Fig. 6-6a. Fig. 6-5e and Fig. 6-6e shows the dc DM voltage remaining constant at about 12 kV during the fault. Fig. 6-5b and Fig. 6-6b shows this same behavior in the ac-side DM voltages. The voltage shifts during both +LG and -LG faults can be seen in their CM counterparts for dc and ac for in Fig. 6-5c and 6-5f, respectively. Fig.6-6c and Fig.6-6f zoom in on the +LG fault CM behavior during fault application. This behavior is characteristic of an IT system during LG faults, and shows correct behavior in real-time simulation.

The under-damped response of the LG fault causes a transient voltage on the dc bus, shown in Fig. 6-7a via Vpg and Vng. Vpg's overshoot peaks at about -5kV, and Vng's overshoot peaks at about -17 kV. This large swing in voltage could exceed the isolation rating of the power electronics module and/or stress cable insulation; however, since this transient only lasts tens of μs, the energy

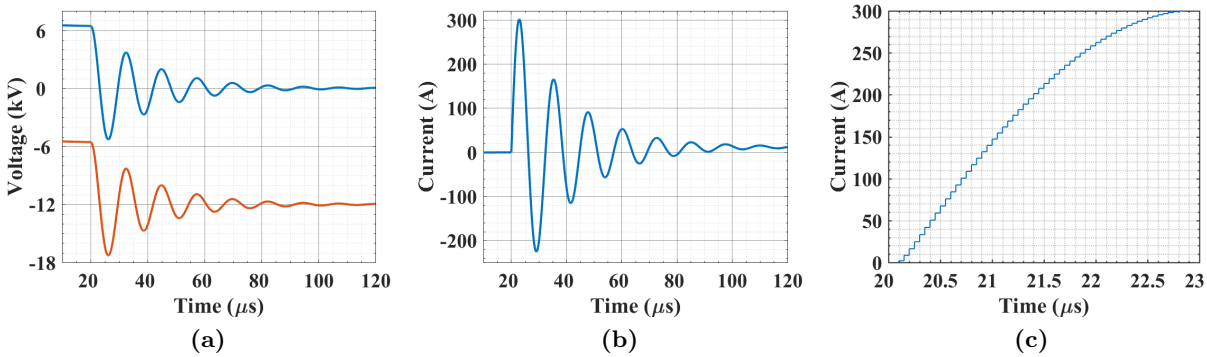


Figure 6-7: +LG fault applied at 20 μs. Data is captured at 20 MHz. (a) Vpng, (b) idc_{CM}, and (c) idc_{CM} zoomed.

content may be very low. This is an area for future research. Similar voltage transients can be expected if an LG fault was applied to the negative dc rail, just in the opposite voltage direction. The transient peaks in Fig. 6-6a and Fig. 6-6i are missed due to the lower sampling resolution of the larger time capture.

The CM dc current (idc_{CM}) transient can be seen in Fig. 6-7b. This transient is also a function of C and L. If the cable length increases, the transient will increase in period and decrease in amplitude. The significance of the idc_{CM} waveform is in a real implementation of power converters in floating system. The LG voltage measurement may not be viable, leaving only the idc_{CM} transient for ground fault detection and location. For example, one could use high bandwidth Rogowski coil spread through the network, paired with Wavelet analysis do help detection and locate ground faults [63] [64], or other methods that require high-resolution simulation [65].

With LB-LMC's 50 ns time step, this example can simulate the transient waveform with 177 data points. The resolution of the solver can be seen in Fig. 6-7c. This significantly increases the resolution of the transient, allowing for accurate fault characterization of high-frequency transient, and testing of fault detection and location algorithms for IT systems with short cable lengths.

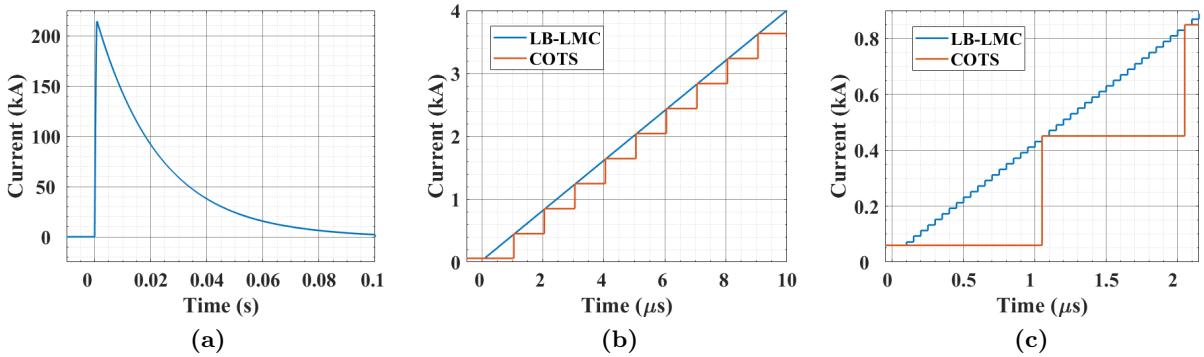


Figure 6-8: idc_{DM} during LL fault on dc bus showing (a) whole current transient, (b) current transient at fault inception comparing 50 ns resolution enabled by LB-LMC solver, and 1μs resolution if run on COTS, and (c) zoomed.

6.2 Real Time Simulation of dc LL Fault with VSR Interfacing Converter

Using the circuit shown in Fig. 6-1, an LL fault is applied on the dc bus. Analytical derivation of VSR under LL faults can be found in [24, 66, 67]; Fig. 6-8a shows the idc_{DM} transient during an LL fault. The peak is very high and is a function of vdc_{DM} at fault inception and impedance in the fault path. In [26], a peak fault current of 280 A was reached with only a 17 Vdc bus. In [24], LL faults were characterized for a 270 LVdc system for an aircraft applications. A peak fault current of several kA was simulated. In [26], a peak fault current of 280 A was reached with only a 17 Vdc bus, and in [27], a peak fault current of about 370 A was reached with a 20 Vdc bus. In [36],a phase-controlled rectifier (PCR) provided power conversion for a 1 kV high power shipboard distribution system. An LL fault was applied to test the fault detection, isolation, and recovery scheme. The fault current reached a peak of 22 kA. For MVdc systems with a bus voltage of tens of kV and fault impedance of mΩs, peak fault current of tens or hundreds of kA is possible. Once the capacitor has discharged, the fault current will go through the anti-parallel diodes of the VSR. Given that the magnitude of this current is 2-3 orders of magnitude greater than rated, most likely, catastrophic failure will occur. For this reason, the VSR will most likely need to paired with protection equipment containing current limiting or arresting capabilities to prevent such a failure.

The rise in fault current, di/dt, is a critical feature in LL faults, as coordination upstream and downstream protection equipment must occur before the current exceeds the rating of the protection

equipment. Therefore, it is vital to have sufficient resolution during this part of the fault transient to enable fault characterization and coordination between different parts of the distribution system. The di/dt is governed by vdc_{DM}/L, where L is the fault path inductance. MVdc systems can have tens of kV and μH of inductance, which leads to di/dt of hundreds to thousands of A/μs [34], [35]. For this simulation, vdc_{DM} is around 12 kV at fault inception, and L is 30 μH. Fig. 6-8b shows idc_{DM} reaching 4 kA in 10 μs, or 400 A/μs.

For MVdc systems with such high di/dt, the protection equipment must detect, coordinate, and begin to arrest the fault current on the μs time scale. Dynamics this is close to the simulation time steps of COTS platforms can be insufficient resolution during the early design stage and can lead to sub-optimal protection system design.

Fig. 6-8b shows idc_{dm} at fault inception and compares the 50 ns time steps of LB-LMC with 1 μs sampling if executed on COTS platforms. Fig. 6-8c shows the critical information missed between the two platforms.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Conclusion and Future Work

In this thesis, the gap between the requirements for real-time HiL simulation of dc protection design for LL and LG fault and COTS real-time simulation is demonstrated. A solution is proposed that implements the HiL in a NI FPGA-based platform using a LB-LMC solver. This same platform is easily expanded to a CHiL as control and protective relaying functions are added. While this CHiL concept may be overkill for a final implementation, it is extremely useful as a means of determining the protective system timing, latency requirements, and will be applied to this purpose in future research. A detailed analysis is performed of a ungrounded (or floating ground) system where the ground reference is established through LG capacitances. This system includes a 1.5 MW, 12 kV NPC VSR and dc side cabling. It is shown the parasitic capacitance of the cabling contributes more to the system's aggregate LG capacitance than the LG capacitance of the NPC VSR. It is also shown that the time step of the real-time simulator is sufficient for simulation of the impacts of limited cable lengths during LG fault scenarios. The LB-LMC solver enables fault characterization of much short cable lengths when compared with COTS real-time solvers. This capability will enable real-time simulation with realistic cable length of emerging MVdc and LVdc applications, such as dc microgrids, hybrid ac/dc microgrids, electrified shipboard distribution systems and MEA.

Additionally, CM voltage and current behaviors are simulated in real-time for the 12 kV VSR in the ungrounded system. It should be noted that any grounding system can be applied. The ungrounded system presents a challenging corner case scenario for real-time simulation. As LG fault are applied, the approach emulates overvoltage stresses and CM current transients upon sudden zero ohm fault applications with a sufficiently high resolution of 50ns–a level currently not present

in the COTS real-time simulators. Due to the scalibility of LB-LMC and the capability of achieving FPGA-to-FPGA communications using SFP-based protocols, such as Aurora from Xilinx, in plug and play systems, such as the NI PXIe, much larger PEC-based networks can be simulated without sacrificing the resolution necessary for accurate fault behavior.

Future work will demonstrate the application of this approach to shipboard MVdc systems where there are contiguous MVdc and LVdc buses interconnected through medium frequency transformer isolated dc-dc converters. The proposed approach is applicable to expandable FPGA-based systems with low timing latency FPGA-to-FPGA communications where a host of present-day communication protocols can be tested. Since the LB-LMC simulation engine is an entity on the FPGA, any logic can be added on the same FPGA, allowing testing logic without simulation platform induced sensor bandwidth limitations or communication delays. This approach allows for the deliberate introduction of delays representing a realistic analog sensing chain and even processor induced delays associated with COTS equipment to demonstrate the impacts of system resiliency. The resultant CHiL systems will enable the development of new protective schemes for dc systems where high resolution data-capture upon fault incidence and subsequent high speed data processing and communications are necessary. Such systems will be required for ground fault location in dc systems and discrimination between pulsed loading and fault conditions. Moreover, superior resolution during LL fault is demonstrated which will enable the design and coordination of LL fault protection schemes in dc systems.

With this platform now established, the work on dc protection design for emerging applications can be accelerated from concept, to design, to validation, and ultimately, transition to industry.

Bibliography

[1] NASA Global Climate Change. Vital signs of the planet. *http://climate. nasa. gov/scientific-consensus*, 2018.

[2] Rebecca Lindsey. Climate change: Atmospheric carbon dioxide. *https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide*, 2020.

[3] Rajendra K Pachauri, Myles R Allen, Vicente R Barros, John Broome, Wolfgang Cramer, Renate Christ, John A Church, Leon Clarke, Qin Dahe, Purnamita Dasgupta, et al. *Climate change 2014: synthesis report. Contribution of Working Groups I, II and III to the fifth assessment report of the Intergovernmental Panel on Climate Change*. Ipcc, 2014.

[4] Marco Cupelli, Ferdinanda Ponci, Giorgio Sulligoi, Andrea Vicenzutti, Chris S Edrington, Touria El-Mezyani, and Antonello Monti. Power flow control and network stability in an all-electric ship. *Proceedings of the IEEE*, 103(12):2355–2380, 2015.

[5] Bulent Sarlioglu and Casey T Morris. More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft. *IEEE transactions on Transportation Electrification*, 1(1):54–64, 2015.

[6] Charles A Thompson. A study of numerical integration techniques for use in the companion circult method of transient circuit analysis. *ECE Technical Reports*, page 297, 1992.

[7] Matthew Milton, Andrea Benigni, and Jason Bakos. System-level, FPGA-based, real-time simulation of ship power systems. *IEEE Transactions on Energy Conversion*, 32(2):737–747, 2017.

[8] M. Milton, A. Benigni, M. Vygoder, J. Gudex, and R. Cuzner. Power electronic system real-time simulation on national instruments FPGA platforms. In *2019 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 32–38, Aug 2019.

[9] Hannah Ritchie and Max Roser. Co_2 and greenhouse gas emissions. *Our World in Data*, 2020. https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.

[10] Paris IEA. Sdg7: Data and projections. *https://www.iea.org/reports/sdg7-data-and-projections*, 2019.

[11] R. Cuzner. The socially responsible microgrid [about this issue]. *IEEE Electrification Magazine*, 6(4):2–5, 2018.

[12] Karthik Palaniappan. A viable residential dc microgrid for low income communities–architecture, protection and education. 2019.

[13] Swachala Veerapaneni. Analysis of impacts of electrical architectures, social-economic considerations and regions, on requirements for residential combined solar and battery implementations. 2018.

[14] Thomas G Wilson. The evolution of power electronics. *IEEE Transactions on Power electronics*, 15(3):439–446, 2000.

[15] Zheming Jin, Giorgio Sulligoi, Rob Cuzner, Lexuan Meng, Juan C Vasquez, and Josep M Guerrero. Next-generation shipboard dc power system: Introduction smart grid and dc microgrid technologies into maritime electrical netowrks. *IEEE Electrification Magazine*, 4(2):45–57, 2016.

[16] John G Ciezki and Robert W Ashton. Selection and stability issues associated with a navy shipboard dc zonal electric distribution system. *IEEE Transactions on power delivery*, 15(2):665–669, 2000.

[17] Robert M Cuzner and Giri Venkataramanan. The status of dc micro-grid protection. In *2008 IEEE Industry Applications Society Annual Meeting*, pages 1–8. IEEE, 2008.

[18] Tomislav Dragičević, Xiaonan Lu, Juan C Vasquez, and Josep M Guerrero. Dc microgrids—part ii: A review of power architectures, applications, and standardization issues. *IEEE transactions on power electronics*, 31(5):3528–3549, 2015.

[19] Steven DA Fletcher, Patrick J Norman, Stuart J Galloway, Paul Crolla, and Graeme M Burt. Optimizing the roles of unit and non-unit protection methods within dc microgrids. *IEEE transactions on Smart Grid*, 3(4):2079–2087, 2012.

[20] Karthik Palaniappan, Willy Sedano, Nicholas Hoeft, Robert Cuzner, and Z John Shen. Fault discrimination using SiC JFET based self-powered solid-state circuit breakers in a residential dc community microgrid. In *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 3747–3753. IEEE, 2017.

[21] John C Mankins. Technology readiness levels. *White Paper, April*, 6:1995, 1995.

[22] MD Omar Faruque, Thomas Strasser, Georg Lauss, Vahid Jalili-Marandi, Paul Forsyth, Christian Dufour, Venkata Dinavahi, Antonello Monti, Panos Kotsampopoulos, Juan A Martinez, et al. Real-time simulation technologies for power systems design, testing, and analysis. *IEEE Power and Energy Technology Systems Journal*, 2(2):63–73, 2015.

[23] Xavier Guillaud, M Omar Faruque, Alexandre Teninge, Ali Hasan Hariri, Luigi Vanfretti, Mario Paolone, Venkata Dinavahi, Pinaki Mitra, Georg Lauss, Christian Dufour, et al. Applications of real-time simulation technologies in power and energy systems. *IEEE Power and Energy Technology Systems Journal*, 2(3):103–115, 2015.

[24] Steven Fletcher, Patrick Norman, Stuart Galloway, and Graeme Burt. Solid state circuit breakers enabling optimised protection of dc aircraft power systems. In *Proceedings of the 2011 14th European Conference on Power Electronics and Applications*, pages 1–10. IEEE, 2011.

[25] Khaled A Saleh, Ali Hooshyar, and Ehab F El-Saadany. Hybrid passive-overcurrent relay for detection of faults in low-voltage dc grids. *IEEE Transactions on smart grid*, 8(3):1129–1138, 2015.

[26] Steven DA Fletcher, Patrick J Norman, Kenny Fong, Stuart J Galloway, and Graeme M Burt. High-speed differential protection for smart dc distribution systems. *IEEE Transactions on Smart Grid*, 5(5):2610–2617, 2014.

[27] Abdullah AS Emhemed, Kenny Fong, Steven Fletcher, and Graeme M Burt. Validation of fast and selective protection scheme for an lvdc distribution network. *IEEE Transactions on Power Delivery*, 32(3):1432–1440, 2016.

[28] Nagesh Geddada, Yew Ming Yeap, and Abhisek Ukil. Experimental validation of fault identification in vsc-based dc grid system. *IEEE Transactions on Industrial Electronics*, 65(6):4799–4809, 2017.

[29] Pietro Cairoli, Roger A Dougal, and Kathleen Lentijo. Coordination between supply power converters and contactors for fault protection in multi-terminal mvdc distribution systems. In *2013 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 493–499. IEEE, 2013.

[30] Pinaki Mitra, Christian Wikström, Niclas Johannesson, and Tomas Larsson. First real-time implementation of dc grid protection strategy. 2015.

[31] Rui Bertho, Vinícius A Lacerda, Renato M Monaro, José CM Vieira, and Denis V Coury. Selective nonunit protection technique for multiterminal vsc-hvdc grids. *IEEE Transactions on Power Delivery*, 33(5):2106–2114, 2017.

[32] AS Vijay, Suryanarayana Doolla, and Mukul C Chandorkar. Real-time testing approaches for microgrids. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 5(3):1356–1376, 2017.

[33] Chris S Edrington, Michael Steurer, James Langston, Touria El-Mezyani, and Karl Schoder. Role of power hardware in the loop in modeling and simulation for experimentation in power and energy systems. *Proceedings of the IEEE*, 103(12):2401–2409, 2015.

[34] J. Langston, K. Schoder, M. Sloderbeck, M. Steurer, and A. Rockhill. Testing operation and coordination of dc solid state circuit breakers. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3445–3452, Oct 2018.

[35] Rich Schmerda, Rob Cuzner, Rodney Clark, Dan Nowak, and Steve Bunzel. Shipboard solid-state protection: Overview and applications. *IEEE Electrification Magazine*, 1(1):32–39, 2013.

[36] Mark W Rose and Robert M Cuzner. Fault isolation and reconfiguration in a three-zone system. In *2015 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 409–414. IEEE, 2015.

[37] Pietro Cairoli, Lisa Qi, Colin Tschida, VRR Ramanan, Luca Raciti, and Antonello Antoniazzi. High current solid state circuit breaker for dc shipboard power systems. In *2019 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 468–476. IEEE, 2019.

[38] Robert M Cuzner, Ashish R Bendre, Jarrod D Widmann, Kelly A Stonger, Stephen M Peshman, Jesse S Carlton, and Jeffery A Fischer. Considerations when diode auctioneering multiple dc buses in a non-isolated dc distribution system. In *2011 IEEE Electric Ship Technologies Symposium*, pages 277–282. IEEE, 2011.

[39] R. M. Cuzner, K. Palaniappan, W. Sedano, N. Hoeft, and M. Qi. Fault characterization and protective system design for a residential dc microgrid. In *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 642–647, Nov 2017.

[40] Michele Difronzo, Matthew Milton, Matthew Davidson, and Andrea Benigni. Hardware-in-the-loop testing of high switching frequency power electronics converters. In *2017 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 299–304. IEEE, 2017.

[41] Matthew Milton, Andrea Benigni, and Antonello Monti. Real-time multi-FPGA simulation of energy conversion systems. *IEEE Transactions on Energy Conversion*, 34(4):2198–2208, 2019.

[42] Ieee recommended practice for 1 kv to 35 kv medium-voltage dc power systems on ships. *IEEE Std 1709-2018 (Revision of IEEE Std 1709-2010)*, pages 1–54, Dec 2018.

[43] J. Mohammadi, F. Badrkhani Ajaei, and G. Stevens. Grounding the dc microgrid. *IEEE Transactions on Industry Applications*, 55(5):4490–4499, Sep. 2019.

[44] N. El-Sherif and S. P. Kennedy. A design guide to neutral grounding of industrial power systems: The pros and cons of various methods. *IEEE Industry Applications Magazine*, 25(1):25–36, Jan 2019.

[45] Brandon Passmore, Zach Cole, Brad McGee, Matthew Wells, Jennifer Stabach, Josh Bradshaw, Robert Shaw, Dan Martin, Ty McNutt, Edward VanBrunt, et al. The next generation of high voltage (10 kv) silicon carbide power modules. In *2016 IEEE 4th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, pages 1–4. IEEE, 2016.

[46] Ieee recommended practice for the design and application of power electronics in electrical power systems. *IEEE Std 1662-2016 (Revision of IEEE Std 1662-2008)*, pages 1–68, March 2017.

[47] Andrew N Lemmon, Robert Cuzner, James Gafford, Rasoul Hosseini, Aaron D Brovont, and Michael S Mazzola. Methodology for characterization of common-mode conducted electromagnetic emissions in wide-bandgap converters for ungrounded shipboard applications. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 6(1):300–314, 2017.

[48] A. D. Brovont, A. Lemmon, C. New, B. W. Nelson, and B. T. DeBoi. Analysis and cancellation of leakage current through power module baseplate capacitance. *IEEE Transactions on Power Electronics*, pages 1–1, 2019.

[49] M Mccandless, C Cooke, J Chalfant, C Chryssostomidis, A Colavitto, A Vicenzutti, A Contin, and G Sulligoi. Thermal analysis of mvdc power corridor. In *2019 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 106–112. IEEE, 2019.

[50] Caledonain. *Medium Voltage Cable*.

[51] John J.. Grainger and William D Stevenson. *Power system analysis*, volume 67. McGraw-Hill New York, 1994.

[52] Andrea Benigni and Antonello Monti. A parallel approach to real-time simulation of power electronics systems. *IEEE Transactions on Power Electronics*, 30(9):5192–5206, 2014.

[53] Mark Vygoder, Jacob Gudex, Robert Cuzner, Matthew Milton, and Andrea Benigni. Real time simulation of transient overvoltage and common-mode during line-to-ground fault in dc ungrounded systems. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 6451–6456. IEEE, 2019.

[54] M. Milton, A. Benigni, and A. Monti. Real-time multi-FPGA simulation of energy conversion systems. *IEEE Transactions on Energy Conversion*, 34(4):2198–2208, 2019.

[55] M. Milton and A. Benigni. Software and synthesis development libraries for power electronic system real-time simulation. In *2019 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 368–376, Aug 2019.

[56] Bin Wu, Yongqiang Lang, Navid Zargari, and Samir Kouro. *Power conversion and control of wind energy systems*, volume 76. John Wiley & Sons, 2011.

[57] D Grahame Holmes and Thomas A Lipo. *Pulse width modulation for power converters: principles and practice*, volume 18. John Wiley & Sons, 2003.

[58] AD Brovont and SD Pekarek. Equivalent circuits for common-mode analysis of naval power systems. In *2015 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 245–250. IEEE, 2015.

[59] Alexander L Julian, Giovanna Oriti, and Thomas A Lipo. Elimination of common-mode voltage in three-phase sinusoidal power converters. *IEEE Transactions on Power Electronics*, 14(5):982–989, 1999.

[60] Lukas Graber, M Steurer, J Kvitkovic, M Kofler, S Pekarek, RA Howard, A Taher, MS Mazzola, and AE Card. Time and frequency domain methods to evaluate grounding strategies for medium voltage dc shipboard power systems. In *2013 IEEE Electric Ship Technologies Symposium (ESTS)*, pages 43–48. IEEE, 2013.

[61] Di Han, Yujiang Wu, Silong Li, and Bulent Sarlioglu. Zero state common mode voltage control in motor drives through inverter topology. In *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 556–560. IEEE, 2017.

[62] Louis V Dusang. A ground fault protection method for ungrounded systems. In *2008 IEEE Canada Electric Power Conference*, pages 1–6. IEEE, 2008.

[63] Siavash Beheshtaein, Junyang Yu, and Robert M Cuzner. A novel wavelet-based feature extraction from common mode currents for fault location in a residential dc microgrid. In *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 706–711. IEEE, 2017.

[64] Ruijing Yang and Robert M Cuzner. Single ground fault location algorithm in dc microgrid based on wavelet transform. In *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 907–912. IEEE, 2016.

[65] Yan Pan, Michael Steurer, and Thomas Baldwin. Feasibility study of noise pattern analysis based ground fault locating method for ungrounded dc shipboard power distribution systems. In *2009 IEEE Electric Ship Technologies Symposium*, pages 18–22. IEEE, 2009.

[66] S. Beheshtaein, R. M. Cuzner, M. Forouzesh, M. Savaghebi, and J. M. Guerrero. Dc microgrid protection: A comprehensive review. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pages 1–1, 2019.

[67] Jin Yang, John E Fletcher, and John O'Reilly. Short-circuit and ground fault analyses and location in vsc-based dc network cables. *IEEE transactions on Industrial Electronics*, 59(10):3827–3837, 2011.

[68] ORTiS Solver Code Generation Tools. https://github.com/OpenRealTimeSimulation, 2020.

Appendix

A.1 Derivation of State-Space Equations for Half-Bridge with Anti-parallel Diodes

This Appendix derives the state space equations for the Half Bridge Converter with anti-parallel diodes. Fig. A-1 shows the schematic of the half-bridge module with anti-parallel diodes. This assumes ideals switches with a fixed on-state resistance (R_{SW}). The diodes are also assumed to be ideal with a fixed on-state of the same R_{SW} value.

A.2 Upper Switch/Diode Conducting

When the upper switch is on, and the lower switch is off, the equivalent circuit is shown in Fig. A-2. Starting off with KCL:

$$i_{Rin} - i_{Cp} - i_L = 0$$

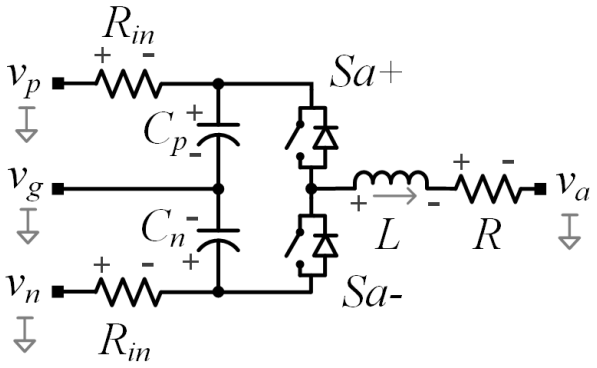


Figure A-1: Schematic of half bridge VSC with anti-parallel diodes

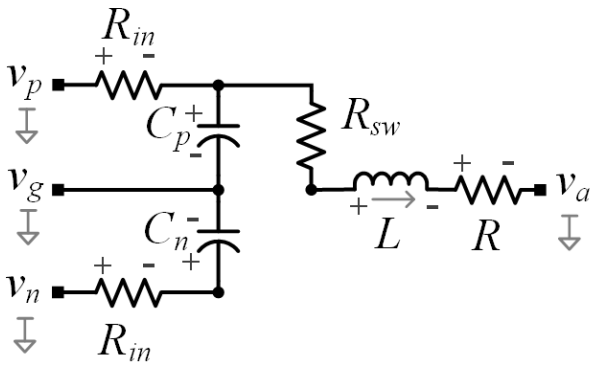


Figure A-2: VSC half-bridge equivalent circuit with upper switch on

Plugging in voltages for i_{Rin} and i_{Cp}:

$$\frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - i_L = 0$$

Solving for $\dot{v_{cp}}$:

$$\dot{v_{cp}} = \frac{1}{R_{in}C_p}\Big(v_p - v_{cp} - v_g\Big) - \frac{1}{C_p}i_L \tag{A.1}$$

For the negative side, $i_{cn} = i_{Rin}$, so doing KVL:

$$0 = -v_n + v_{Rin} + v_{cn} + v_g$$

$$0 = -v_n + i_{cn}R_{in} + v_{cn} + v_g$$

$$0 = -v_n + C_n\dot{v_{cn}}R_{in} + v_{cn} + v_g$$

Solving for $\dot{v_{cn}}$:

$$\dot{v_{cn}} = \frac{1}{C_nR_{in}}\Big(v_n - v_{cn} - v_g\Big) \tag{A.2}$$

(A.2) is similar to (A.1), but without the inductor contribution term, since the inductor is connected via the upper switch being off. Now doing KVL:

$$v_g + v_{cp} - v_{Rsw} - v_L - v_R - v_a = 0$$

Plugging in currents:

$$v_g + v_{cp} - R_{sw}\cdot i_L - L\cdot\dot{i_L} - R\cdot i_L - v_a = 0$$

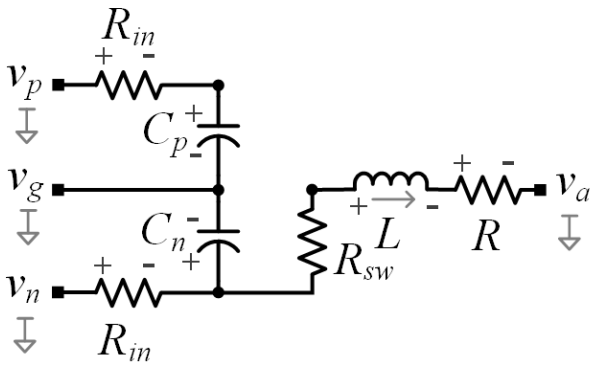


Figure A-3: VSC half-bridge equivalent circuit with lower switch on

And solving for $\dot{i_L}$:

$$\dot{i_L} = \frac{1}{L}\left(v_g + v_{cp} - v_a\right) - \left(\frac{R_{sw}+R}{L}\right) i_L \tag{A.3}$$

(A.1), (A.2), and (A.3) are the differential equations for this switch configuration. Putting this into state space form:

$$\begin{bmatrix} \dot{v_{cp}} \\ \dot{v_{cn}} \\ \dot{i_L} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_{in}C_p} & 0 & -\frac{1}{C_p} \\ 0 & -\frac{1}{R_{in}C_p} & 0 \\ \frac{1}{L} & 0 & -\frac{R_{sw}+R}{L} \end{bmatrix} \begin{bmatrix} v_{cp} \\ v_{cn} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_{in}C_p} & 0 & -\frac{1}{R_{in}C_p} & 0 \\ 0 & \frac{1}{R_{in}C_p} & -\frac{1}{R_{in}C_p} & 0 \\ 0 & 0 & \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_p \\ v_n \\ v_g \\ v_a \end{bmatrix} \tag{A.4}$$

For Forward Euler integration $x[n] = x[n-1] + DT \cdot \dot{x[n-1]}$, where DT is the solver time step, $x[n]$ is the state space variables of the current time step, $x[n-1]$ is the state space variables of the previous time step. The state space equations discretized with Forward Euler are then:

$$v_{cp}[n] = v_{cp}[n-1] + DT\left[\frac{1}{R_{in}C_p}\left(v_p[n-1] - v_{cp}[n-1] - v_g[n-1]\right) - \frac{1}{C_p} i_L[n-1]\right] \tag{A.5}$$

$$v_{cn}[n] = v_{cn}[n-1] + DT\left[\frac{1}{C_n R_{in}}\left(v_n[n-1] - v_{cn}[n-1] - v_g[n-1]\right)\right] \tag{A.6}$$

$$i_L[n] = i_L[n-1] + DT\left[\frac{1}{L}\left(v_g[n-1] + v_{cp}[n-1] - v_a[n-1]\right) - \left(\frac{R_{sw}+R}{L}\right) i_L[n-1]\right] \tag{A.7}$$

A.3 Lower Switch/Diode Conducting

When the upper switch is off, and the lower switch is on, the equivalent circuit is shown in Fig. A-3. For this switching states, the state space equations are the essentially the same, just with using lower side n components and voltages, instead of the upper side p. Thus, the state space equations are:

$$\dot{v_{cp}} = \frac{1}{R_{in}C_p}\left(v_p - v_{cp} - v_g\right) \tag{A.8}$$

$$\dot{v_{cn}} = \frac{1}{R_{in}C_n}\left(v_n - v_{cn} - v_g\right) - \frac{1}{C_n}i_L \tag{A.9}$$

$$\dot{i_L} = \frac{1}{L}\left(v_g + v_{cn} - v_a\right) - \left(\frac{R_{sw}+R}{L}\right)i_L \tag{A.10}$$

Putting (A.8), (A.9), and (A.10) into state space form:

$$\begin{bmatrix} \dot{v_{cp}} \\ \dot{v_{cn}} \\ \dot{i_L} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_{in}C_p} & 0 & 0 \\ 0 & -\frac{1}{R_{in}C_p} & -\frac{1}{C_p} \\ 0 & \frac{1}{L} & -\frac{R_{sw}+R}{L} \end{bmatrix} \begin{bmatrix} v_{cp} \\ v_{cn} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_{in}C_p} & 0 & -\frac{1}{R_{in}C_p} & 0 \\ 0 & \frac{1}{R_{in}C_p} & -\frac{1}{R_{in}C_p} & 0 \\ 0 & 0 & \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_p \\ v_n \\ v_g \\ v_a \end{bmatrix} \tag{A.11}$$

The discritized with Forward Euler:

$$v_{cp}[n] = v_{cp}[n-1] + DT\left[\frac{1}{R_{in}C_p}\left(v_p[n-1] - v_{cp}[n-1] - v_g[n-1]\right)\right] \tag{A.12}$$

$$v_{cn}[n] = v_{cn}[n-1] + DT\left[\frac{1}{R_{in}C_n}\left(v_n[n-1] - v_{cn}[n-1] - v_g[n-1]\right) - \frac{1}{C_n}i_L[n-1]\right] \tag{A.13}$$

$$i_L[n] = i_L[n-1] + DT\left[\frac{1}{L}\left(v_g[n-1] + v_{cn}[n-1] - v_a[n-1]\right) - \left(\frac{R_{sw}+R}{L}\right)i_L[n-1]\right] \tag{A.14}$$

A.4 Both Switches/Diodes Conducting

A.4.1 Using v_* at the midpoint

When both switches are on due to a "shoot-through" conditions, or when both diodes are force to conduct, such as during a Line-to-Line (LL) fault on the between v_p and v_n, the equivalent circuit is shown in Fig. A-4.To simply the calculations we can use the midpoint, v_*, where $v_* = \frac{1}{2}\big(v_{cp} + v_{cn}\big) + v_g - \frac{R_{sw}}{2} i_L$. Finding the differential equation for $\dot{v_{cp}}$:

$$0 = i_{in} - i_{cp} - i_{Rsw} \tag{A.15}$$

Capacitor current is:

$$i_{cp} = C_p \dot{v_{cp}} \tag{A.16}$$

Input current is:

$$i_{in} = \frac{v_p - v_{cp} - v_g}{R_{in}} \tag{A.17}$$

Current though the on-state switch is:

$$i_{Rsw} = \frac{v_g + v_{cp} - v_*}{R_{sw}} \tag{A.18}$$

Plugging in (A.16), (A.17), and (A.18) into (A.15).

$$0 = \frac{v_p - v_{cp} - v_g}{R_{in}} - C_p \dot{v_{cp}} - \frac{v_g + v_{cp} - v_*}{R_{sw}} \tag{A.19}$$

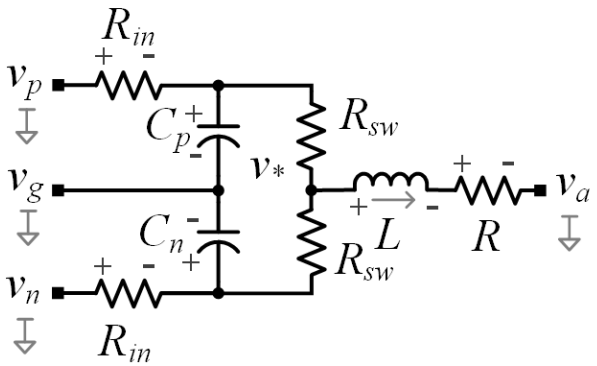


Figure A-4: VSC half-bridge equivalent circuit with both switches on

solving for $\dot{v_{cp}}$:

$$\dot{v_{cp}} = \frac{1}{R_{in}C_p}\Big(v_p - v_{cp} - v_g\Big) - \frac{1}{R_{sw}C_p}\Big(v_g + v_{cp} - v_*\Big) \tag{A.20}$$

Similarly for $\dot{v_{cn}}$:

$$\dot{v_{cn}} = \frac{1}{R_{in}C_n}\Big(v_n - v_{cn} - v_g\Big) - \frac{1}{R_{sw}C_n}\Big(v_g + v_{cn} - v_*\Big) \tag{A.21}$$

Assuming we know v_*, we can do KVL with the inductor part of the circuit:

$$0 = v_* - v_L - v_R - v_a$$

Plugging in terms for inductor voltage, $v_L = L\dot{i_L}$, and resistor voltage, $v_r = Ri_L$:

$$0 = v_* - L\dot{i_L} - Ri_L - v_a \tag{A.22}$$

Solving for $\dot{i_L}$:

$$\dot{i_L} = \frac{1}{L}\Big(v_* - Ri_L - v_a\Big) \tag{A.23}$$

A.4.2 Without using v_* at the midpoint

Right now it is not obvious to me how, v_* is derived, so let come up with the state space equations without it. For this, the subscript p denotes the "positive side" currents, voltages, and parameters, and n for the "negative side."

$$0 = i_{inp} - i_{cp} - i_{Rswp} \tag{A.24}$$

$$0 = i_{inn} - i_{cn} - i_{Rswn} \tag{A.25}$$

$$i_{cp} = C_p\dot{v_{cp}} \tag{A.26}$$

$$i_{cn} = C_n\dot{v_{cn}} \tag{A.27}$$

$$i_{inp} = \frac{1}{R_{inp}}\Big(v_p - v_{cp} - v_g\Big) \tag{A.28}$$

$$i_{inn} = \frac{1}{R_{inn}}\Big(v_n - v_{cn} - v_g\Big) \tag{A.29}$$

$$i_{Rswp} = \frac{1}{R_{swp}}\Big[v_{cp} + v_g - (v_L + v_R + v_a)\Big] \tag{A.30}$$

$$i_{Rswn} = \frac{1}{R_{swn}}\Big[v_{cn} + v_g - (v_L + v_R + v_a)\Big] \tag{A.31}$$

$$i_{Rswp} = \frac{1}{R_{swp}}\Big(v_{cp} + v_g\Big) - \frac{L}{R_{swp}}\dot{i_L} - \frac{R}{R_{swp}}i_L - \frac{1}{R_{swp}}v_a \tag{A.32}$$

$$i_{Rswn} = \frac{1}{R_{swn}}\Big(v_{cn} + v_g\Big) - \frac{L}{R_{swn}}\dot{i_L} - \frac{R}{R_{swn}}i_L - \frac{1}{R_{swn}}v_a \tag{A.33}$$

$$i_{Rswp} + i_{Rswn} = i_L \tag{A.34}$$

We can add (A.32) and (A.33) to get inductor current, i_L, (A.34). It is assumed that $R_{swp} = R_{swp}$, and $R_{inp} = R_{inn}$.

$$i_L = \frac{1}{R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{2}{R_{sw}}v_g - \frac{2L}{R_{sw}}\dot{i_L} - \frac{2R}{R_{sw}}i_L - \frac{2}{R_{sw}}v_a \tag{A.35}$$

$$0 = \frac{1}{R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{2}{R_{sw}}v_g - \frac{2L}{R_{sw}}\dot{i_L} - \frac{2R}{R_{sw}}i_L - i_L - \frac{2}{R_{sw}}v_a \tag{A.36}$$

$$0 = \frac{1}{R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{2}{R_{sw}}v_g - \frac{2L}{R_{sw}}\dot{i_L} - \Big(\frac{2R}{R_{sw}} - 1\Big)i_L - \frac{2}{R_{sw}}v_a \tag{A.37}$$

$$\frac{2L}{R_{sw}}\dot{i_L} = \frac{1}{R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{2}{R_{sw}}v_g - \Big(\frac{2R}{R_{sw}} - 1\Big)i_L - \frac{2}{R_{sw}}v_a \tag{A.38}$$

$$\frac{L}{R_{sw}}\dot{i_L} = \frac{1}{2R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{1}{R_{sw}}v_g - \Big(\frac{R}{R_{sw}} - \frac{1}{2}\Big)i_L - \frac{1}{R_{sw}}v_a \tag{A.39}$$

$$\dot{i_L} = \frac{1}{2L}\Big(v_{cp} + v_{cn}\Big) + \frac{1}{L}v_g - \Big(\frac{R}{L} - \frac{R_{sw}}{2L}\Big)i_L - \frac{1}{L}v_a \tag{A.40}$$

Plugging in v_* into (A.23) gives

$$\dot{i_L} = \frac{1}{L}\Big(v_* - Ri_L - v_a\Big)$$

$$v_* = \frac{1}{2}\big(v_{cp} + v_{cn}\big) + v_g - \frac{R_{sw}}{2}i_L$$

$$\dot{i_L} = \frac{1}{L}\Big(\frac{1}{2}\big(v_{cp} + v_{cn}\big) + v_g - \frac{R_{sw}}{2}i_L - Ri_L - v_a\Big)$$

Which is the same as (A.40). For the capacitor voltage derivative equations, we can plug in (A.26), (A.28), and (A.32) into (A.24), and (A.27), (A.29), and (A.33) into (A.25):

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - \left[\frac{1}{R_{sw}}\Big(v_{cp} + v_g\Big) - \frac{L}{R_{sw}}\dot{i_L} - \frac{R}{R_{sw}}i_L - \frac{1}{R_{sw}}v_a\right] \tag{A.41}$$

$$0 = \frac{1}{R_{in}}\Big(v_n - v_{cn} - v_g\Big) - C_n\dot{v_{cn}} - \left[\frac{1}{R_{sw}}\Big(v_{cn} + v_g\Big) - \frac{L}{R_{sw}}\dot{i_L} + \frac{R}{R_{sw}}i_L - \frac{1}{R_{sw}}v_a\right] \tag{A.42}$$

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - \frac{1}{R_{sw}}\Big(v_{cp} + v_g\Big) + \frac{L}{R_{sw}}\dot{i_L} + \frac{R}{R_{sw}}i_L + \frac{1}{R_{sw}}v_a \tag{A.43}$$

$$0 = \frac{1}{R_{in}}\Big(v_n - v_{cn} - v_g\Big) - C_n\dot{v_{cn}} - \frac{1}{R_{sw}}\Big(v_{cn} + v_g\Big) + \frac{L}{R_{sw}}\dot{i_L} + \frac{R}{R_{sw}}i_L + \frac{1}{R_{sw}}v_a \tag{A.44}$$

Now, we should be able to solve for $\dot{v_{cp}}$, and $\dot{v_{cn}}$, expect we have another derivative term, $L/R_{sw}\cdot\dot{i_L}$, in the equation. To get rid of this term, we can plug in (A.39) into both (A.43), and (A.44) and simplifing.

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - \frac{1}{R_{sw}}\Big(v_{cp} + v_g\Big) + \frac{1}{2R_{sw}}\Big(v_{cp} + v_{cn}\Big) + \frac{1}{R_{sw}}v_g \tag{A.45}$$

$$- \left(\frac{R}{R_{sw}} - \frac{1}{2}\right)i_L - \frac{1}{R_{sw}}v_a + \frac{R}{R_{sw}}i_L + \frac{1}{R_{sw}}v_a \tag{A.46}$$

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - \frac{1}{R_{sw}}v_{cp} - \frac{1}{R_{sw}}v_g + \frac{1}{2R_{sw}}v_{cp} + \frac{1}{2R_{sw}}v_{cn} + \frac{1}{R_{sw}}v_g \tag{A.47}$$

$$- \frac{R}{R_{sw}}i_L + \frac{1}{2}i_L - \frac{1}{R_{sw}}v_a + \frac{R}{R_{sw}}i_L + \frac{1}{R_{sw}}v_a \tag{A.48}$$

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} - \frac{1}{2R_{sw}}v_{cp} + \frac{1}{2R_{sw}}v_{cn} + \frac{1}{2}i_L \tag{A.49}$$

$$0 = \frac{1}{R_{in}}\Big(v_p - v_{cp} - v_g\Big) - C_p\dot{v_{cp}} + \frac{1}{R_{sw}}\left[\frac{1}{2}\Big(-v_{cp} + v_{cn}\Big) + \frac{R_{sw}}{2}i_L\right] \tag{A.50}$$

Solving for $\dot{v_{cp}}$:

$$\dot{v_{cp}} = \frac{1}{R_{in}C_p}\Big(v_p - v_{cp} - v_g\Big) + \frac{1}{R_{sw}C_p}\left[\frac{1}{2}\Big(-v_{cp} + v_{cn}\Big) + \frac{R_{sw}}{2}i_L\right] \tag{A.51}$$

$$\dot{v_{cp}} = \frac{1}{R_{in}C_p}\Big(v_p - v_{cp} - v_g\Big) + \frac{1}{2R_{sw}C_p}\Big(-v_{cp} + v_{cn}\Big) + \frac{1}{2C_p}i_L \tag{A.52}$$

$$\dot{v_{cn}} = \frac{1}{R_{in}C_n}\Big(v_n - v_{cn} - v_g\Big) + \frac{1}{2R_{sw}C_n}\Big(v_{cp} - v_{cn}\Big) + \frac{1}{2C_n}i_L \tag{A.53}$$

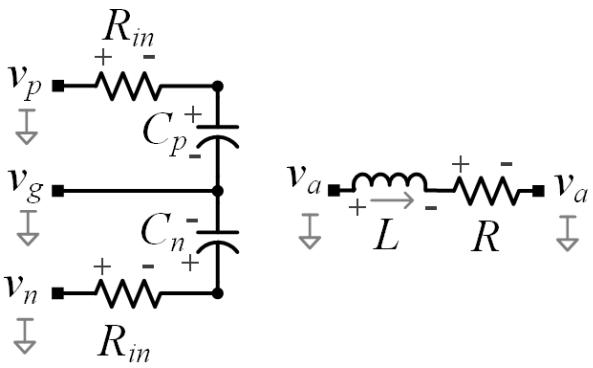


Figure A-5: VSC half-bridge equivalent circuit with both switches off

A.5 Both Switches/Diodes Off

Fig. A-5 shows the equivalent circuit for when both switches are off. An interesting questions is what is the voltage between a half bridge when both switches are off? With both switches off, there should be not current through the inductor, so the voltage midpoint of the leg would be equal to v_a. Also, with no current flowing through inductor, the di/dt is zero as well. The equations for capacitor voltage are the same for the open switches during upper or lower switch commutation, as in (A.2), and (A.8).

A.6 VSR Example using Command Line Interface Tool

Below the code used for the command line interface tool. The tool takes the inputted parameters and netlist and generators the circuit in the LB-LMC simulation engine using C++ code. The tool and code generation library are available at [68].

Listing A.1: VSR netlist file used for the CLI tool

```
%12 kV Voltage Source rectifier VSR03d with LL and LG fault
%LCL DM and LC CM filter.

#name VSR03d_anti_parallel_diodes_dt50ns_10uH

%Time Step
#const DT 50.0e-9

%Feeder cable
#const RF 0.198
#const LF 1.8e-3

%Common Mode Filter
```

```
#const LCM 60.0e-3
#const CCM 0.1e-6
#const LCM_and_LDM1 61.01494e-3

%Differential Mode Filter
#const LDM1 0.95494e-3
#const LDM2 5.6e-3
#const CDM 16.652e-6
#const RDSERIES 55.1822

%Voltage Source Rectifier
#const LDM2_ESR 0.001
#const CDC2 20e-3
#const DiodeThresholdVoltage 1.5

%DC Bus
#const LCABLE 10.0e-6
#const RLOAD 100.0

%Floating Ground
#const CNG 0.1e-6
#const RNG 10.0e3
#const COG 0.1e-6
#const ROG 10.0e3

%Fault Switches
#const LFAULT_1uH 1.0e-6
#const LFAULT_10uH 10.0e-6
#const RFAULT_1mOhms 1.0e-3

%High resistance path to ground
#const ROPEN 100.0e3

%%%%%%%%%%%%%%%%%%%%%
%%% Build Netlist %%%
%%%%%%%%%%%%%%%%%%%%%

%AC Voltage Inputs with feeder Resistor
FunctionalVoltageSource va (RF) {2,1}
FunctionalVoltageSource vb (RF) {4,1}
FunctionalVoltageSource vc (RF) {6,1}

%Feeder cable inductance
Inductor lfa (DT, LF) {2,3}
Inductor lfb (DT, LF) {4,5}
Inductor lfc (DT, LF) {6,7}

%Common Mode Cap filter
Capacitor ccma (DT, CCM) {3,0}
Capacitor ccmb (DT, CCM) {5,0}
Capacitor ccmc (DT, CCM) {7,0}

%Common Mode Inductor Filter
MutualInductance3 lcm_ldm1 (DT, LCM_and_LDM1, LCM_and_LDM1,
LCM_and_LDM1, LCM, LCM, LCM) {3, 8, 5, 9, 7, 10}

%Resistor to Ground to prevent solver instability
Resistor rcmopena (ROPEN) {8, 0}
Resistor rcmopenb (ROPEN) {9, 0}
```

```
Resistor rcmopenc (ROPEN) {10, 0}

%Differential Model Filters
Capacitor cdmab (DT, CDM) {8, 11}
Capacitor cdmbc (DT, CDM) {9, 12}
Capacitor cdmca (DT, CDM) {10, 13}
Resistor rdsab (RDSERIES) {11, 9}
Resistor rdsbc (RDSERIES) {12, 10}
Resistor rdsca (RDSERIES) {13, 8}

%3 phase 2 level converter
BridgeConverter_3LegIdealSwitchesAntiParallelDiodes VSR (DT, CDC2, LDM2,
LDM2_ESR, DiodeThresholdVoltage) {15, 14, 18, 8, 9, 10}

%Cabling
Inductor lcablep (DT, LCABLE) {15, 16}
Inductor lcablen (DT, LCABLE) {18, 19}

%Resistive Load
Resistor rloadp (RLOAD) {16, 17}
Resistor rloadn (RLOAD) {19, 17}

%Floating Ground Connection Points
Capacitor cng (DT, CNG) {1,0}
Resistor rng (RNG) {1, 0}

Capacitor cog (DT, COG) {14,0}
Resistor rog (ROG) {14, 0}

Capacitor cog2 (DT, COG) {17,0}
Resistor rog2 (ROG) {17, 0}

%Fault Switches
SeriesRLIdealSwitch ll_fault (DT, LFAULT_10uH, RFAULT_1mOhms) {16, 19}
SeriesRLIdealSwitch lgp_fault (DT, LFAULT_10uH, RFAULT_1mOhms) {16, 0}
SeriesRLIdealSwitch lgn_fault (DT, LFAULT_10uH, RFAULT_1mOhms) {19, 0}
```

A.7 Xivado HLS directives

Listing A.2: Example of HLS directives and port signal adjustments for LabVIEW FPGA CLIP core

```
#include "vsr_FullSwitching_solver_core.hpp"

void vsr_FullSwitching_solver_core
(
    //outputs
    port_real  x_out[19],
    port_real* l_current_lfa,
    port_real* l_current_lfb,
    port_real* l_current_lfc,
    port_real& positive_capacitor_voltage_VSR,
    port_real& negative_capacitor_voltage_VSR,
    port_real& leg_a_inductor_current_VSR,
    port_real& leg_b_inductor_current_VSR,
    port_real& leg_c_inductor_current_VSR,
    port_real* l_current_lcablep,
```

```
16      port_real* l_current_lcablen,
17      port_real* l_current_ll_fault,
18      port_real* l_current_lgp_fault,
19      port_real* l_current_lgn_fault,
20
21      //inputs
22      port_real  v_in_va,
23      port_real  v_in_vb,
24      port_real  v_in_vc,
25      bool         switch_gates_VSR[6],
26      bool         sw_ll_fault,
27      bool         sw_lgp_fault,
28      bool         sw_lgn_fault
29  )
30  {
31
32  //Xilinx Vivado HLS Settings
33
34      //partition array ports to be like individual element ports;
35      //to avoid ap_memory interface which has undesired latency to read/write
36      #pragma HLS array_partition variable=x_out dim=0
37      #pragma HLS array_partition variable=sw_ctrl_converter1 dim=0
38
39      //inline all code recursively to remove logic boundaries for optimization
40      #pragma HLS inline recursive
41
42      //set execution latency to be 0 clock cycles (execute in single cycle)
43      #pragma HLS latency min=0 max=0
44
45  //=========================================================================
46      //internal output signals
47
48      real  x_out_inner[19];
49      real l_current_lfa_inner;
50      real l_current_lfb_inner;
51      real l_current_lfc_inner;
52      real positive_capacitor_voltage_VSR_inner;
53      real negative_capacitor_voltage_VSR_inner;
54      real leg_a_inductor_current_VSR_inner;
55      real leg_b_inductor_current_VSR_inner;
56      real leg_c_inductor_current_VSR_inner;
57      real l_current_lcablep_inner;
58      real l_current_lcablen_inner;
59      real l_current_ll_fault_inner;
60      real l_current_lgp_fault_inner;
61      real l_current_lgn_fault_inner;
62
63
64  //=========================================================================
65      //update inner inputs from external input ports
66
67      real  v_in_va_inner = convertFromPortReal(v_in_va);
68      real  v_in_vb_inner = convertFromPortReal(v_in_vb);
69      real  v_in_vc_inner = convertFromPortReal(v_in_vc);
70      //bool inputs are left alone
71
72  //=========================================================================
73      //update simulation solver
74
```

```
75      VSR_FullSwitching_50ns_LBLMC_SimEngine_solver<0,real>
76      (
77          x_out_inner,
78          &l_current_lfa_inner,
79          &l_current_lfb_inner,
80          &l_current_lfc_inner,
81          positive_capacitor_voltage_VSR_inner,
82          negative_capacitor_voltage_VSR_inner,
83          leg_a_inductor_current_VSR_inner,
84          leg_b_inductor_current_VSR_inner,
85          leg_c_inductor_current_VSR_inner,
86          &l_current_lcablep_inner,
87          &l_current_lcablen_inner,
88          &l_current_ll_fault_inner,
89          &l_current_lgp_fault_inner,
90          &l_current_lgn_fault_inner,
91          v_in_va_inner,
92          v_in_vb_inner,
93          v_in_vc_inner,
94          switch_gates_VSR,
95          sw_ll_fault,
96          sw_lgp_fault,
97          sw_lgn_fault
98      );
99
100 //=============================================================================
101     //update external output ports
102
103     for(int i = 0; i < 19; i++)
104     {
105         #pragma HLS unroll
106         x_out[i] = convertToPortReal(x_out_inner[i]);
107     }
108
109     *l_current_lfa = convertToPortReal(l_current_lfa_inner);
110     *l_current_lfb = convertToPortReal(l_current_lfb_inner);
111     *l_current_lfc = convertToPortReal(l_current_lfc_inner);
112     positive_capacitor_voltage_VSR =
113         convertToPortReal(positive_capacitor_voltage_VSR_inner);
114     negative_capacitor_voltage_VSR =
115         convertToPortReal(negative_capacitor_voltage_VSR_inner);
116     leg_a_inductor_current_VSR = convertToPortReal(leg_a_inductor_current_VSR_inner);
117     leg_b_inductor_current_VSR = convertToPortReal(leg_b_inductor_current_VSR_inner);
118     leg_c_inductor_current_VSR = convertToPortReal(leg_c_inductor_current_VSR_inner);
119     *l_current_lcablep = convertToPortReal(l_current_lcablep_inner);
120     *l_current_lcablen = convertToPortReal(l_current_lcablen_inner);
121     *l_current_ll_fault = convertToPortReal(l_current_ll_fault_inner);
122     *l_current_lgp_fault = convertToPortReal(l_current_lgp_fault_inner);
123     *l_current_lgn_fault = convertToPortReal(l_current_lgn_fault_inner);
```