SYNCHROPHASOR-BASED MONITORING, CONTROL, AND PROTECTION FOR DISTRIBUTED ENERGY RESOURCES

Prottay Mondal Adhikari

Submitted in Partial Fullfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:
Luigi Vanfretti, Chair
Joe Chow
Koushik Kar
Stacy Patterson

Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, New York

[December 2021]
Submitted November 2021

CONTENTS

LIST OF TABLES x

LIST OF FIGURES xii

ABBREVIATIONS xvii

ACKNOWLEDGEMENT xviii

ABSTRACT xxiv

1. INTRODUCTION 1

1.1 Motivational Background 1

1.1.1 The Synchrophasor Paradigm 1

1.1.2 Mathematical Model Development for Power System Simulations . . 3

1.1.3 The Concept of Microgrid 4

1.1.4 Utilizing Synchrophasor Technology to Improve the Monitoring, Control and Protection Functionalities for Microgrids 6

1.2 Research Problem-statements 7

1.3 Scope 9

1.3.1 Exploration of Reconfigruable FPGA Based PMUs 9

1.3.2 Modeling and Simulation for Real-Time Hardware-in-the-Loop Testing 9

1.3.3 Implementation of the Control and Decision Framework 10

1.4 Thesis Organization 11

2. RECONFIGURABLE FPGA-BASED PHASOR MEASUREMENT UNITS 13

2.1 Introduction 13

2.1.1 Motivation 13

2.1.2 Related Works 14

2.1.3 Highlights of the Chapter 15

2.1.4 Structure of the Chapter 16

2.2 PMU Software, Hardware, and Design Flow 16

2.2.1 Software Implementation of the PMU 16

2.2.2 Review of the Existing PMU Hardware 17

2.2.3 Hardware Design Flow Methodology 19

2.2.3.1 Observations Related to the NI Cloud Compile Service . . . 20

2.3 Case-Studies and Experiments 20

2.3.1 Test Specimens/Designs 21
2.3.2 In Vivo Experiments on Physical FPGA Devices 22
2.3.3 In Silico Experiments using FPGA Emulation 22
2.3.4 Characterization 22
2.4 Experimental Results 23
2.4.1 Results on Physical FPGA Hardware 23
2.4.1.1 Hypothesis and Metric Derivation 23
2.4.1.2 Cross-verification of the Proposed Metric With HDL Designs 26
2.4.1.3 Errors in Different Stages for Physical PMU Implementation 26
2.4.2 FPGA Emulation Results 28
2.4.3 FPGA Selection 29
2.5 Conclusions 30
3. VERIFICATION OF THE FPGA BASED PHASOR MEASUREMENT UNITS . 31
3.1 Introduction 31
3.1.1 Motivation 31
3.1.2 Related Works 32
3.1.3 Highlights of the Chapter 32
3.1.4 Structure of the Chapter 33
3.2 PMU Pre-Compliance Testing Review 33
3.3 Proposed Testing Infrastructure 34
3.3.1 Hardware Description 35
3.3.2 Software Description 37
3.4 Results 39
3.4.1 Steady State Test 39
3.4.2 Step Change in Magnitude and Phase 40
3.4.3 Bandwidth Test 41
3.4.4 Frequency Ramp Test 41
3.5 Conclusions 42
4. REAL TIME SIMULATION OF POWER SYSTEM COMPONENTS 44
4.1 Introduction 44
4.1.1 Motivation 44
4.1.2 Related Works 45
4.1.3 Highlights of the Chapter 46
4.2 Review of the Simulation Platforms 46

4.2.1 Opal-RT 5030/4520 Real Time Simulator System 46
4.2.2 Typhoon HIL 603 Real Time Simulator 47
4.3 Test Cases 47
4.3.1 Components for Unit Testing 47
4.3.1.1 Unit 1: Synchronous Generator 47
4.3.1.2 Unit 2: Transformer 47
4.3.1.3 Unit 3: Induction Motor 48
4.3.1.4 Unit 4: Governor 48
4.3.1.5 Unit 5: Load 48
4.3.2 Description of the Microsystem 48
4.3.3 Description of the Microgrid 49
4.4 Simulation Results 49
4.4.1 Unit Tests: Simulation Results 50
4.4.2 Simulation Results for the Microsystem 53
4.4.3 Simulation Results for the Microgrid 54
4.5 Discussions 55
4.6 Conclusions 55
5. ADVANCED COMPONENT MODELING: PHOTOVOLTAIC CELLS AND ARRAYS 57
5.1 Introduction 57
5.1.1 Motivation 57
5.1.2 Related Works 58
5.1.3 Highlights of the Chapter 59
5.2 Theory 59
5.2.1 Review of the Real-time Hardware Platforms 59
5.2.2 Description of the Proposed Model 60
5.2.2.1 Thermal Voltage (V_t) Computation 62
5.2.2.2 Series Resistance (R_s) Computation 62
5.2.2.3 Open-circuit Voltage (V_{oc}) Computation 63
5.2.2.4 Short-circuit Current (I_{sc}) Computation 63
5.2.2.5 Dark Current (I_o) Computation 63
5.3 Results 64
5.3.1 Verification of the Proposed Model in Opal RT 64
5.3.2 Implementation in the Typhoon HIL Platform 65
5.3.3 Hardware Selection 66

5.4 Conclusions . . . 66
5.5 Discussions and Future Works . . . 67
6. EVALUATING OVERVOLTAGE MITIGATION STRATEGIES FOR PV INVERTERS AND THEIR IMPACT ON GROUND-FAULT OVERVOLTAGE THROUGH RT-CHIL EXPERIMENTS . . . 69
6.1 Introduction . . . 70
6.1.1 Motivation . . . 70
6.1.2 Related Works . . . 71
6.1.3 Contributions . . . 72
6.2 Experimental Setup . . . 74
6.2.1 Experiment's Hardware Infrastructure Overview . . . 74
6.2.2 Experiment's Software Overview . . . 74
6.2.3 Experiment Setup and Protocol . . . 75
6.3 Results and Analysis . . . 78
6.3.1 Tuning of Capacitor Banks for GFOV Sequence Component Analysis 78
6.3.2 GFOV and GLR . . . 82
6.3.3 Impact of Conventional GFOV Mitigation . . . 82
6.3.4 Impact of the Power Factor (PF) in the Load . . . 83
6.3.5 Impact of Δ Connected Loads . . . 87
6.3.6 Impact of SPOV . . . 91
6.4 Summary and Analysis of Experimental Results . . . 93
6.4.1 Impact of TrOV in IBRs . . . 93
6.4.2 Importance of SPOV Settings in IBRs . . . 93
6.5 Conclusions and Future Works . . . 94
7. ADVANCED COMPONENT MODELING: LI-ION BATTERIES . . . 97
7.1 Introduction . . . 97
7.1.1 Motivation . . . 97
7.1.2 Related Works . . . 98
7.1.3 Contributions . . . 99
7.2 Theory . . . 99
7.2.1 Review of the Real-time Hardware Platforms . . . 99
7.2.2 Description of the Proposed Model . . . 100
7.3 Results . . . 104
7.3.1 Model Verification and Characterization . . . 104

7.3.2 Simulation in Typhoon HIL Platform . . . 105
7.3.3 Test Case: Application of the Battery Model . . . 106
7.4 Discussions and Future Works . . . 107
7.5 Conclusions . . . 108
8. PRECISION TIMING AND COMMUNICATION NETWORKING EXPERIMENTS IN A REAL-TIME POWER GRID HARDWARE-IN-THE-LOOP LABORATORY 109
8.1 Introduction . . . 110
8.1.1 Motivation . . . 110
8.1.2 Related Works . . . 111
8.1.3 Contributions . . . 113
8.2 Review of the ALSET Infrastructure . . . 114
8.2.1 Review of the Precise Timing Layer . . . 114
8.2.2 Review of the Communication Network Layer . . . 116
8.3 Experiments . . . 117
8.3.1 Experiment 1: Demonstration of the Different VLAN Data Types . . . 117
8.3.2 Experiment 2: Demonstrating the Three Layers of ALSET Lab Through an Experiment . . . 118
8.3.3 Experiment 3: Tampering With the Network Layer . . . 120
8.3.4 Experiment 4: Tampering With the Timing Layer . . . 120
8.3.5 Experiment 5: Validating the QoS Rules for the Network . . . 123
8.3.5.1 Experiment 5(a): Validating the QoS for Priorities of VLANs 124
8.3.5.2 Experiment 5(b): Validating the Guaranteed Minimum Bandwidth for the VLANs . . . 125
8.3.6 Analysis and Synthesis of Experimental Complexity . . . 126
8.4 Conclusions . . . 128
9. IMPLEMENTATION OF SYNCHROPHASOR SYNCHRONIZATION GATEWAY & CONTROL ARCHITECTURE . . . 129
9.1 Introduction . . . 130
9.1.1 Motivation . . . 130
9.1.2 Related Works . . . 131
9.1.3 Highlights of This Chapter . . . 133
9.2 Proposed Hardware & Software Framework for SSG/SSGC Architecture . . . 134
9.2.1 Software Components . . . 135
9.2.2 Hardware Components . . . 135
9.3 Experimental Case Studies and Analysis . . . 136

9.3.1 Test case 1: Usage of SSG Hardware to Parse PMU Streams 137
9.3.2 Test case 2: Evaluating the SSG Under Network Delay 139
9.3.3 Test Case 3: Dynamic Measurements Through Tampered Networks . 143
9.3.4 Test Case 4: Extending the Architecture in SSG and Control of a Micrgrid Model in Real-time 146
9.3.4.1 Software and Hardware Components for Developing the SSGC and Testing it With a Microgrid Model 150
9.3.4.2 Implementation of the Control Architecture for the Microgrid Model 151
9.3.4.3 Evaluation of the Proposed Control Architecture Under Varied Network Conditions 153
9.4 Directions for Future Experiments 154
9.4.1 Adjustments to Adhere to the Constraints of the Battery 155
9.4.2 Alternate PMU Placement and Utilizing Different System Variables for Controlling DERs 155
9.4.3 Incorporation of the Battery State-of-charge (SOC) Within the Control Action 156
9.5 Conclusions 156
10. SUMMARY AND FUTURE WORKS 157
10.1 Summary 157
10.2 Future Works 159
10.2.1 Extension of the Experiments into the Paradigm of Power Hardware in the Loop (PHIL) Configurations 159
10.2.2 Efforts Towards Model Standardization 160
10.2.3 SSGC Based Control System Development for Microgrids and DERs . 160
REFERENCES 161

APPENDICES

A. CODE FOR PV CELL MODELING 174
B. THEORETICAL ESTIMATION OF GFOV FOR SYNCHRONOUS MACHINES AND IBRS 175
B.1 Theoretical GFOV Calculation in Synchronous Generators 175
B.2 Theoretical GFOV Calculation in Inverters 176
C. THEORETICAL ESTIMATION FOR IMPEDANCE IN GROUNDING TRANSFORMERS 178

D. HARDWARE AND SOFTWARE UTILIZED FOR CHIL EXPERIMENTS 180

D.1 Hardware Utilization 180

D.1.1 HIL-604 Real-Time Simulator 180

D.1.2 ASGC CHIL 180

D.1.3 Interface Between HIL-604 and ASGC 181

D.2 Software Utilization 181

D.2.1 Schematic Editor 181

D.2.2 HIL SCADA 181

D.2.3 aBoot Flasher 181

D.2.4 Firmware Manager 182

E. SUPPLEMENTAL MATERIALS 183

LIST OF TABLES

2.1 Hardware Platforms, FPGAs and Processors 18

2.2 Time Comparison of Report-Generation For Various Stages 23

2.3 Synthesis Statistics for Hardware Compilation for Different Designs 24

2.4 Cross Validation of the Derived Metric with HDL (Verilog) Designs 24

2.5 Assessment of the Proposed Metric Based on FPGA Emulation Results For Unknown Hardware 28

3.1 Specifications of the PMU Testing Hardware and PMU Under Test 36

3.2 PMU FE Performances During Frequency Ramp Test Available in the Literature 43

4.1 Internal Parameters for Power System Components 47

4.2 Simulation Settings 50

4.3 Simulation Results for Model Load 51

4.4 Results From Individual Buses of the Microgrid 54

5.1 Performance Characterization of the Proposed PV Model in Opal-RT Platform 65

5.2 Memory Utilization For Simulating the PV Model on Typhoon HIL 603 in Real-time 65

6.1 Software Used in the CHIL Experiment 73

6.2 Distribution System Specifications 77

6.3 Sequence Components of Figure 6.5(a) at t=0.65 Sec 82

6.4 Sequence Components of Figure 6.5(b) at t=0.65 Sec 88

6.5 Effect of GTFs on TrOV With Varying Share of Δ Connected Load 88

6.6 TrOV Variation With and Without GTF 91

6.7 Impact of (i) SPOV Settings and (ii) Percentage of Y-connected Loads- on the Maximum Overvoltage 94

7.1 Real Time Hardwares: A Comparison of Features 100

7.2 Performance Characterization of the Proposed Battery Model in Opal-RT Platform104

7.3 Memory Utilization to Simulate the Battery Model in Typhoon HIL 603 105

8.1 Specifications of the Existing ALSET VLANs 117

8.2 Reviewing the Complexities of the Performed Experiments 127

9.1 Delay Statistics for All the PMUs (NI Network Published Shared Variable Used for Measurements) 139

9.2 Delay Reduction After Removal of NI Network Published Shared Variable Infrastructure 145

9.3 Testing the Resilience of the SSGC Framework Under Varying Network Conditions154

LIST OF FIGURES

1.1 Proliferation of PMU Technology in USA 2

1.2 Proliferation of PV Systems Across the World (Figure from [6]) 3

1.3 Proliferation of Battery Energy Storage Systems Across the World (Figure from [7]) 4

1.4 Projected Growth for Microgrids' Capacity Across the World (Figure from [11]) 5

1.5 Current Infrastructure for Control Using Networked PMUs and PDCs 7

1.6 Proposed Control Infrastructure for Microgrids With Networked PMUs and SSG/SSGC 8

1.7 Visual Representation of the Organization for This Thesis 11

2.1 Software and Hardware Resources of the PMU Implementation 17

2.2 Block Diagram for the Hardware Architecture of the PMU Assembly 18

2.3 Physical PMU Hardware in a Compact-RIO 9068 19

2.4 The FPGA Based Hardware Implementation Flow and Report Generation . . . 20

2.5 Comparison of Pre-Synthesis, Synthesis and Post-Map/PnR Results 27

2.6 Xilinx Spartan 6 FPGA Family IC's Variation of Cost and Hardware Resources 29

3.1 PMU Testing Infrastructure Connected With the PMU Under Test 34

3.2 PMU Testing Hardware Connected to the PMU Prototype 35

3.3 Connection Diagram for GPS Signal Acquisition 37

3.4 Organization of the Testing Software in LabVIEW Including the Real-time and FPGA Dependencies 38

3.5 Sample GUI to Control/Modulate a Compliance Test 38

3.6 PMU Compliance Test Results for the PMU Under Steady-state Test Using the Proposed PMU-testing Hardware (Left) Total Vector Error (TVE), (Right) Frequency Error (FE) 39

3.7 PMU Compliance Test Results for Magnitude (0.1 pu) and Phase (10^o) Step-Change: (Left) Total Vector Error (TVE), (Right) Frequency Error (FE) . . . 40

3.8 PMU Compliance Test Results for Voltage and Frequency Modulation Test: (Left) Total Vector Error (TVE), (Right) Frequency Error (FE) 41

3.9 Frequency Error (FE) of the PMU Under Test During Frequency Ramp Test . 42

4.1 Simplified Governor Model Implemented Using Typhoon HIL Schematic Editor 48

4.2 Schematic of the Micro-system Developed in the Typhoon HIL Schematic Environment 49

4.3 Schematic of the Microgrid Developed in Typhoon HIL Schematic Environment (the Red Box Outlines the Induction Motor Model Described in Section 4.3.1) 50

4.4 Step Response of the Modeled Governor 51

4.5 Response of the Transformer Model While Subjected to Rated Input Voltage . 52

4.6 Transient Response of the Motor 52

4.7 PET Parameter Estimation Results for J=Motor Intertia, L_{ls}=Stator Inductance, R_s=Stator Resistance, R_{snub}=Snubber Resistance 53

4.8 Transient Responses of the Generator in the Micro-System 54

5.1 Single Diode Model to Represent a PV Cell 60

5.2 Block-Diagram Containing Different Components of the Model of the PV Cell . 61

5.3 Comparison of V-I Characteristics for the Existing PV Models 64

5.4 Comparison of the PV Model Responses Between Typhoon HIL and Opal-RT . 67

6.1 Pictorial Representation of the Grid Connected PV Distribution Feeder Used in This Work 73

6.2 (a) Pictorial Representation of the Experimental Setup, (b) The Physical Experimental Setup, (c) The Interface Between ASGC Controller and Typhoon HIL 604 74

6.3 CHIL Simulation under GLR 1.0: (a) Operation of the Breakers, (b) Phase Voltages, (c) Phase Currents (PV-Inverter) (d) Active Power Provided by the Utility and the PV Source 76

6.4 Effect of Capacitance-Tuning: (a) Phase Voltages Under SLG Without Tuned Capacitor Bank, (b) Phase Voltages Under SLG With Tuned Capacitor Bank . 78

6.5 Sequence Diagram Under SLG: (a) Without the GTF, (b) With the GTF . . . 79

6.6 Currents When There is No GTF in the Circuit: (a) 3-Phase Currents From PV Inverter, (b) 3-Phase Load Current, (c) Sequence Components of Currents From PV Inverter, (d) Sequence Components of Load Current 81

6.7 Variation of Overvoltage With Increasing GLR 84

6.8 Overvoltage Simulation With GLR=1.0 and GTF: (a) Switching Sequence, (b) Phase Voltages . . . 84

6.9 Currents With GTF in the Circuit: (a) 3-Phase Currents From PV Inverter, (b) 3-Phase Load Current, (c) Sequence Components of PV Inverter Currents (d) Sequence Components of Load Current (e) Sequence Components of the GTF Current . . . 85

6.10 SPOV Interaction With the TrOV Simulations for GLR=1.30: (a) When SPOV is Disabled at 1.3 (b) When SPOV is Enabled at 1.3 . . . 86

6.11 Comparison of Sequence Components of the Voltages With and Without the Grounding Transformer . . . 86

6.12 Phasor Diagrams: (a) No GTF Inserted, GLR=1, (b) GTF Inserted, GLR=1, (c) No GTF Inserted, GLR=1.2, (d) GTF Inserted, GLR=1.2. (Red=Positive Sequence, Blue=Negative Sequence , Green=Zero Sequence, Black= Phase Voltages) . . . 87

6.13 TrOV Experimentation With Non-UPF (0.9 Lag) Loads: (a) While GLR=1.0, (b) While GLR=1.2 . . . 89

6.14 Overvoltage Simulation With GLR=1.0 Under UPF, With No GTF Present: (a) While Δ-Share =25%, (b)While Δ-Share =50%, (c) While Δ-Share =75% . 90

6.15 Overvoltage Simulation With GLR=1.0 Under UPF, With GTF Present: (a) While Δ-Share =25%, (b)While Δ-Share =50%, (c) While Δ-Share =75% . . . 92

6.16 Cumulative Overvoltage Duration Calculation (a) GLR=1.3 (b) GLR=1.4 . . . 95

7.1 The Existing Comprehensive Battery-Model . . . 100

7.2 Proposed Simplified Equivalent Circuit Model for a Battery . . . 100

7.3 Charging Mode: Performance Comparison Between the Proposed Model and the Benchmark Model . . . 101

7.4 Discharging Mode: Performance Comparison Between the Proposed Model and the Benchmark Model . . . 101

7.5 Charging Mode: Terminal Voltage and SOC Variation For the Proposed and Benchmark Models . . . 102

7.6 Discharging Mode: Terminal Voltage and SOC Variation For the Proposed and Benchmark Models . . . 102

7.7 Flowchart of the Proposed Model Implementation . . . 103

7.8 Microgrid Implementation to Validate the Battery Model . . . 105

7.9 Step Increase in the Reference Active Power for the Battery Unit 106

7.10 Output Current From the Battery Unit 106

7.11 Terminal Voltage of the Battery 107

7.12 Response of the Battery-SOC 107

8.1 The Source of Timing Information: As Seen on the Substation Clock GUI . . . 115

8.2 Architecture of Timing Layer for the ALSET Lab 116

8.3 Architecture of the Network Layer of the ALSET Lab Illustrating all the Virtual LANs 117

8.4 Wireshark Screen Captures of the Data Packets Through Different VLANs Implemented in ALSET Lab 118

8.5 Illustration of a Simple Real-time Experiment that Utilizes all Three Layers of the ALSET Lab Hierarchy 119

8.6 Observing the Corrupted Timestamp of SEL-421 121

8.7 Experimental Setup to Tamper with the Precise Timing Layer of the ALSET Lab121

8.8 Experimental Synchrophasor Observation After Tampering the Timing Layer . 122

8.9 Effect of the QoS Priority Rule, Under Varying Network Bandwidth 123

8.10 Experimental Setup to Demonstrate the QoS Rules 124

8.11 Effect of the QoS Guaranteed Minimum Bandwidth Rule Under Varying GOOSE-Data Stream 125

9.1 An Overview of the Hardware and Software Components for the SSG Implementation 133

9.2 Block Diagram of the Experimental Setup for Test Case 1 134

9.3 Photograph of the Experimental Setup 134

9.4 Frequency Measured in PMU (white) and the Measured Frequency Received by the SSG (green) 135

9.5 Timing Diagram of the Experimental Setup 136

9.6 Networking Infrastructure to Tamper a PMU-SSG Network Utilizing the CT-910 Network Traffic Generator 139

9.7 Example of 100 ms Delay Injection in the Communication Link 140

9.8 Tampering the Network Between PMU_1 and the SSG 141

9.9 PMU_1 to SSG Delay Under Varying Injected Delay: (a) Measured Absolute Delay Between PMU_1 & SSG After Variable Additional Network Traffic Injection (b) Estimated Pass-through (EPT) Delay for PMU_1 to SSG Delay After Variable Additional Traffic Injection . . . 141

9.10 Histograms for Measured Delay in Different Network Traffic Injections . . . 142

9.11 Fitted Distributions of the Delay-Histogram for Different Values of Injected Delays 142

9.12 Real-time Delay Measurement With No Injected Delay, and With Very High Injected Delay: (a) PMU_1 to SSG Delay Without Using NI Network Published Shared Variable (b) Absolute PMU_1 to SSG Delay With Very High Delay-injection (750 ms) . . . 143

9.13 Dynamic PMU Selection Based on the SSG . . . 144

9.14 Experimental Setup to Use the SSG For Reading Data From PMU Streams With Different Network Delays . . . 145

9.15 Tracking the Frequency of the Synchrophasor Data Received Through Two Networks With Different Network Delays . . . 146

9.16 The Interaction Between the Microgrid Model on Typhoon HIL 604 and the SSGC Running on cRIO . . . 146

9.17 Hardware Arrangement for Testing Microgrid With the SSGC Hardware: (a) Connection Between the Microgrid & the PMUs, (b) PMUs Receiving Timing Information, (c) Conversion of RT Low-voltage Signals into Current Signals, (d) SSGC Connected Remotely to the Microgrid . . . 147

9.18 Microgrid Controller Utilized in the Experimental Setup . . . 148

9.19 Simplified Microgrid Model Implemented in Real-time . . . 148

9.20 300 kW Load Injection by Switching the Interruptible Load . . . 149

9.21 Response at the Output of the PI-controller Inside SSGC . . . 149

9.22 Active Power Output from the Battery Energy Storage System (BESS) . . . 149

9.23 Control Signal Received in the Typhoon HIL Microgrid Model From the SSGC Under Varying Network Delay and Data-drop Rates . . . 152

B.1 Ground Fault in an Ungrounded Synchronous Generator Feeding a 3 Phase 4 Wire System . . . 175

B.2 Ground Fault in a Three Phase Inverter Feeding a 3 Phase 4 Wire System . . . 176

C.1 Reference to the Sequence Diagram of IBRs Under SLG: (a) Without the GTF, (b) With the GTF, Reproduced From Figure 6.5 . . . 178

ABBREVIATIONS

AMS	Adaptive Multichannel Source
BESS	Battery Energy Storage System
CHIL	Controller Hardware in the Loop
cRIO	Compact Reconfigurable Input Output
DER	Distributed Energy Resources
EPT	Estimated Pass Through
ESS	Energy Storage System
FPGA	Field Programmable Gate Array
GFOV	Ground Fault Over Voltage
GLR	Generation to Load Ratio
GOOSE	Generic Object Oriented Substation Event
GPS	Global Positioning System
GTF	Grounding Transformer
HIL	Hardware in the Loop
IBR	Inverter Based Resources
IED	Intelligent Electronic Devices
IRIG-B	Inter-Range Instrumentation Group - Time Code Format B
LROV	Load Rejection Overvoltage
NI	National Instruments
PDC	Phasor Data Concentrator
PHIL	Power Hardware in the Loop
PMU	Phasor Measurement Unit
PV	Photo Voltaic
RT	Real Time
RTS	Real Time Simulator
SCADA	Supervisory Control and Data Acquisition
SDM	Single Diode Model
SLG	Single Line to Ground
SPOV	Self Protection Overvoltage
SSG	Synchrophasor Synchronization Gateway
SSGC	Synchrophasor Synchronization Gatewat and Controller
TCP/IP	Transmission Control Protocol/Internet Protocol
TrOV	Transient Overvoltage
VLAN	Virtual Local Area Network
WAMPAC	Wide Area Monitoring Protection and Control
WAMS	Wide Area Measurement System

ACKNOWLEDGEMENT

So, it is time to say those two words. THANK YOU.

Thank you Dr. Vanfretti for all your support, help, assistance and teachings. Thank you for buying and arranging for all those fancy equipment and hardware I get to use in the lab every single day. Thank you for always being there whenever I needed any assistance, academic or otherwise. Thank you for introducing me to amazing people starting from professors across the world, to researchers from the industry, to congresspeople. Thank you for giving me the opportunity to mentor bright undergraduate students with their researches, which almost always turned out to be extraordinary learning opportunities for me. Thank you for your unlimited excitement when experiments worked, and boundless patience when they didn't. I earnestly cherish all our interactions in JEC-6220 and JEC-6119, which started with undecipherable scribbles on the whiteboard, and later transformed into various diagrams in this thesis. I hope to live by the high standards you instilled in me, both professionally and personally.

I would also express my gratitude to the committee members for their support and encouragements. A big thanks to Dr. Koushik Kar for his valuable suggestions regarding setting up the networking experiments. My sincere-most gratitude to Dr. Joe Chow, from whom I've learnt a great deal during his courses. I am truly honored to have him in my dissertation committee. Dr. Stacy Patterson, has been extremely helpful and encouraging through all the interactions I've had with her.

I truly have a dissertation committee full of amazing people.

Looking back, it is important for me to acknowledge my privilege to have been taught by some of the finest teachers one can ever hope for. Starting from my high school physics tutor Dr. Debangshu Dey who is one of the reasons I decided to major in Electrical Engineering in the first place, to Dr. Abhinandan De whose lucid technique in explaining various concepts in circuit theory and electrical machines at the undergraduate level still inspires me. The pedagogy demonstrated by professors like Dr. Anindita Sengupta, Dr. Aparajita Sengupta, Debjani Ganguly, and Dr. Paramita Chattopadhyay from my undergraduate days in Indian Institute of Engineering Science and Technology, Shibpur played instrumental roles in inspiring me to pursue graduate studies.

Speaking of graduate studies, oftentimes graduate students' default state of mind is - *'nothing works, nothing at all'.* A big part of graduate school experience is learning to live with it. The person who taught me that by instilling in me the mindset - *'Of course things don't work. That's why I am being paid to be in here'* - was Dr. Madhav Desai. I would like to express my sinceremost gratitude towards him and Dr. Mukul Chandorkar for providing me with an interesting research problem, taking time to explain the details to me (sometimes repeatedly), and being patient with my research during my tenure at IIT Bombay. From those days, I would also like to thank Dr. Joseph John, who introduced me to LabVIEW- which became an invaluable tool later during my doctoral journey.

I am lucky to have befriended some amazing people throughout my career so far. It is time to extend my recognition towards them. A huge thanks to Soham for helping me learn a wide array of concepts in electrical engineering (and suggesting great films that go well with those). Thanks to Digbalay for sticking with me through thick and thin, and supporting me through the graduate school application process (plus, we survived the agonizing death of Game of Thrones, *together*). Thanks to Kaustav for all our amazing calls, quirks, chats, and gossips. Lastly, thanks to Mahasweta for her endless support, bearing through our weird calls at ungodly hours, and being a friend with whom I can vent about anyone and everyone ... from James Charles to Yann LeCun. Thanks to the friends I made online- Habib, Harold, Mads, Dan, Nishant, Siddharth, Daniel, Sarvesh, and Mac. The occasional conversations we have been having for the past few years have been golden. Thanks for letting me be my truest, strangest and weirdest self with you.

Lastly, I must extend my gratitude to all my friends at Troy- Airin, Anirban, Ashraful, Indrani-di, Jayantika, Kanu-da, Karabi, Rikhiya-di, Sounak, Saswata, Sarthak, Sudeshna, Saptarshi-da. Their physical presence in the neighborhood often made the mundane days enjoyable, and the downright horrible ones somewhat tolerable. They truly were there for me when, *'it hadn't been my day, my week, my month or even my year'.*

I must thank Dr. Hossein Hooshyar for his infinite support, endless efforts in explaining and re-explaining concepts to me, and his out-of-this-world documentation skills. His reference documentations on the ALSET architecture still help us get going, when things go wrong. A big thanks to Dr. Hamed Nademi for helping with any and every issue I had with MATLAB. Thanks to Dr. Rohit Suryadevara for giving me company during our all-nighters in JEC-6119. A gigantic thanks to my friends and colleagues inside the ALSET Lab- Sergio, Marcelo, Meaghan and Giuseppe. While Giuseppe and Marcelo always kept a steady head on their shoulders, Marcelo and Meaghan were always joining voices with me whenever I threw a melodramatic tantrum of - *'Nothing works! Electricity is a lie!'*. Probably this beautiful dichotomy made ALSET Lab such an enjoyable place to work for me. The next token of gratitude is towards the Mikes - Dr. Michael Ropp (Sandia National Lab) and Michael Ruppert (JEM Engineering), for keeping faith in us to work on the ground fault overvoltage problem, and assisting me with their valuable insights.

I am thoroughly impressed by the help I received from the undergraduate students I worked with. Cici, Emmett, Jordan, Brennan, James - you people were amazing, and while the arrangement was for me to teach you things, I am pretty sure it worked the other way around. Thanks for being as amazing as you all were. Randall, Mick and the late Jerry, you were not just helpful in making the ALSET lab. You *made* the ALSET lab. Most of the experiments in this thesis would never have been possible without your contributions.

My four year long doctoral journey was made infinitely easier by the technical assistance I received from Ivan & Murillo (Typhoon HIL), Roland & Banja (Austrian Institute of Technology), Isaac (CandelaTech), Shijia (Opal-RT). Needless to say, I could not have finished the work I did, without the crucial supports from numerous support engineers at Schweitzer Engineering Laboratories and National Instruments - whose names I don't remember.

On a more formal note, let me use this opportunity to thank the funding agencies. The research presented in this dissertation was funded in part by the New York State Energy Research and Development Authority (NYSERDA) under agreement numbers 137948, and 149165; in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under Award EEC-1041877, in part by the CURENT Industry Partnership Program, in part by the Center of Excellence for NEOM Research at King Abdullah University of Science and Technology, and in part by Dominion Energy.

Now, it is time to talk about some of the most important people in my life who've been supporting me through the last thirty years. Thanks to Bua (my maternal aunt) who've always played an instrumental role in making me the person I am today since my formative years. Thanks to my first and most significant teacher- Maa (my mom), without whose love, persistence, composure, and perseverance I could never have been in this place today. Thanks for nurturing the love of science and math in me, since primary school. Lastly, thanks to Baba (my dad) for being my first role-model through his endless patience, love, guidance and support. Thanks to both my parents for being available, both emotionally and intellectually, whenever I needed them (and I still do). Most importantly, thanks for bearing with my tantrums (these are some of my bigger ones) whenever the internet is even marginally spotty, during our daily video-calls. With all that being said, and hoping that I haven't missed any major names, I dedicate this dissertation to the two of you.

In memory of *Jerry Dziuba*, without whom ALSET Lab would not have existed the way it does today. Whether it was setting up the electrical connections, arranging experimental setups, wiring high voltage equipment, or quite literally - ‘heavy lifting’ - *Jerry* had been instrumental in every way perceivable. May he rest in peace and power.

**বেড়ার নিয়ম ভাঙলেই পথের নিয়ম আপনিই বেরিয়ে পড়ে,
নইলে এগোব কী করে**

রবীন্দ্রনাথ ঠাকুর (তাসের দেশ)

Translation:

Barriers have their own set of rules, so do trailways.
All it takes to impose those rules of trailways
is to break a barrier down...
And that’s how we go on.

Rabindranath Tagore[1] *(Kingdom of Cards)*

[1] Bengali polymath from late 19th century and the first Nobel laureate from Asia

ABSTRACT

The proliferation of distributed energy resources (DERs) are transforming the landscape of today's electrical grid. Non-conventional DERs such as photovoltaic systems, Battery Energy Storage Systems (BESS), and wind turbines exhibit strikingly different dynamics when compared to traditional energy sources driven by synchronous generators, due to their power electronic interfaces with the grid. Thus, a modern electrical grid that utilizes these new energy resources, would require a reconfigurable, fast, accurate, and time critical monitoring, protection and control framework in order to operate harmoniously and resiliently. Synchrophasor technology and Phasor Measurement Units (PMUs) provide a unique, time-critical, accurate, reliable and standardized framework for power system measurements under these new energy resources.

This thesis proposes a synchrophasor-based monitoring, management, protection, and control architecture for power systems with DERs. The usage of PMUs and synchrophasor technology is appropriate for such applications, because they provide accurately time-stamped measurements for power network management functions such as monitoring and control. This approach proposes substantial benefits as compared to conventional supervisory control and data acquisition (SCADA) systems in terms of accuracy, reliability and speed.

Because, it is logistically impossible to operate and test the proposed architecture in a real-life power system, real-time hardware-in-the-loop (HIL) simulators are used to mimic the behavior of a power system, containing multiple DERs such as photovoltaic systems, BESSs, and diesel generators. During this process, different types of real-time hardware-in-the-loop (HIL) simulator hardware were explored and their performances were compared. The proposed synchrophasor-based network management and control infrastructure uses deterministic real-time embedded systems with Field Programmable Gate Arrays (FPGA) for data acquisition and timing-management, and real-time processors are utilized for networking and inter-communication purposes. To test the resiliency of the proposed architecture, additional network traffic generator hardware were connected to the communication network that houses the controller and the PMUs. Through the course of experimentation, multi-platform homogeneous real-time simulation models were developed for photovoltaic cells and Li-ion batteries. In the domains of power system protection, controller-hardware-in-the-loop (CHIL) experiments were performed to investigate whether traditional protection methods are suitable for the integration of inverter based DERs. These experiments revealed that unlike traditional plants with synchronous generators, grounding transformers are not required in order to protect inverter based DERs from ground-fault overvoltages (GFOV) under single-line-to-ground (SLG) faults. However, most utilities are still reluctant to abandon the inclusion of grounding transformers. Thus, this observation has the potential to reduce the cost for planning and implementing DER based substations significantly, if adopted by the operators.

To perform all these experiments seamlessly within a real-time digital power system simulation ecosystem, a laboratory architecture featuring networking, timing and electrical functionalities was standardized and experiments were performed on it. One such key experiment featured the design and implementation of a synchrophasor synchronization gateway and controller (SSGC) hardware. This hardware has the functionalities to parse multiple synchrophasor streams in real-time, compute and monitor their respective network delays, and provide supplementary control functionalities based on the information retrieved from those synchrophasor streams. This hardware was utilized to monitor and control a microgrid (containing BESS and PV systems) running on a real-time simulator. To test the resilience of this proposed hardware, its communication network was tampered with external hardware, and its performance was analyzed under such conditions.

CHAPTER 1
INTRODUCTION

The global energy landscape has been changing rapidly through the last decade. The proliferation of different distributed energy resources, coupled with the meteoric growth in the fields of communications, sensor fabrication, and computing hardware development is enabling a transformation in power generation, transmission & distribution, monitoring, and control. Emerging technologies like phasor measurement units (PMUs) and synchrophasor are reshaping the way the power grid is monitored across wide spatio-temporal scales by utilizing the latest developments in the fields of satellite communications and edge technologies. On the other end, distributed energy resources (DERs) such as solar photo-voltaic (PV) systems and battery energy storage systems (BESS) are gaining popularity both in developed and developing countries' energy grids.

This chapter provides the motivational background behind the current thesis, and introduces the research problem statements this thesis aims to address. Following those, the scopes and limitations of this thesis is presented. Finally, this chapter concludes with a visual presentation of the organizational structure of the entire thesis.

1.1 Motivational Background

This section reviews the phasor measurement units (PMUs) and their growing relevance in today's power systems, discusses the motivation for accurate and homogeneous modeling for various power system components and real-time environments, introduces the concept of microgrids, and explores how synchrophasor technologies can be employed to improve the monitoring, protection, and control of distributed energy resources and microgrids.

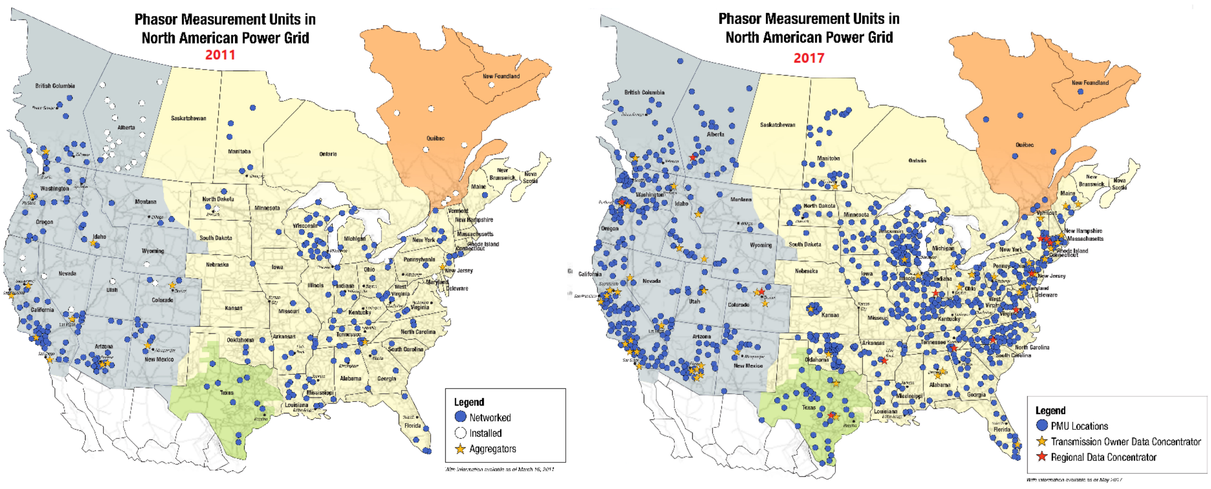
1.1.1 The Synchrophasor Paradigm

The mathematical foundations of representing periodically-varying quantities, like AC voltage and current, by a rotating line(phasor) dates back to 1893 [1]. Almost hundred years later in 1988, Dr. Arun G. Phadke and Dr. James S. Thorp at Virginia Tech proposed the application of such techniques to measure voltage and currents in power system. Their work led to the invention of world's first Phasor Measurement Unit (PMU) in 1992 [2].

These PMUs were capable of voltage and current phasor measurement in real-time and all the measurements were synchronized to a universal time reference obtained from the Global Positioning System (GPS). Because the phasor measurement technology is synchronized to an absolute clock, it is sometimes referred to as the synchrophasor technology.

Since its inception in early 90s, PMU technology has become more common in the power industry. Figure 1.1 compares the growth [3] of PMU technology in the 2010s in the US. With over 2500 PMUs currently installed across power-stations and sub-stations in the USA, they provide a foundation for real-time monitoring. In India, the number of operational PMUs back in 2013 was only 62 [4]. This number increased to 1500 by the end on 2020, as reported by the survey in [5]. Clearly, synchrophasor technology is changing the landscape of power system monitoring across the world.

With the ability to compute and stream upto 50/60/100/120 measurement data per second, PMU technology is well suited for tracking power system dynamics in real-time. Thus, they are the foundation for modern power system network management architectures like Wide Area Monitoring System (WAMS) or Wide Area Monitoring Protection and Control (WAMPAC).


Phasor Measurement Units in North American Power Grid
2011
Legend
● Networked
○ Installed
☆ Aggregators
Phasor Measurement Units in North American Power Grid
2017
Legend
● PMU Locations
☆ Transmission Owner Data Concentrator
★ Regional Data Concentrator

Source: North American Synchrophasor Initiative (NASPI)

Figure 1.1: Proliferation of PMU Technology in USA

Wide Area Measurement System (WAMS) is a technology that enhances the observability and situational awareness within the power system. It is dependent on real-time synchrophasor data to monitor the status of the power grid. WAMPAC builds upon the architectural foundation of WAMS and providess real-time protection and control functionalities for the power grid. WAMS/WAMPAC are very sophisticated technological infrastructures and require different components such as substation Phasor Data Concentrators (PDC), aggregation PDCs, and additional communication network installations apart from those required by the PMUs. This increases the complexity of the overall system, and leads to an increase in the latency of the communications network if not properly addressed. This thesis, explores the efficient utilization of synchrophasor technologies, communication methods, and reconfigurable hardwares to reduce these complexities.

1.1.2 Mathematical Model Development for Power System Simulations

It is obvious that, any proposed new monitoring, control or protection- hardware cannot be tested in the field by experimenting with an actual power system. Hence, Real-Time (RT) Simulators become neccessary. Real-time simulators are high-end computer systems which can execute the mathematical model of a power system at the same rate as the ”wall-clock”. Most industrially available real-time simulators have the ability to not only solve the model of a physical system in real-time, but also generate proportional analog and digital signals corresponding to selected measurements from the real-time model. These real-time signals are physically interfaced or networked to test the new control, monitoring or protection equipments from the safety of a laboratory.

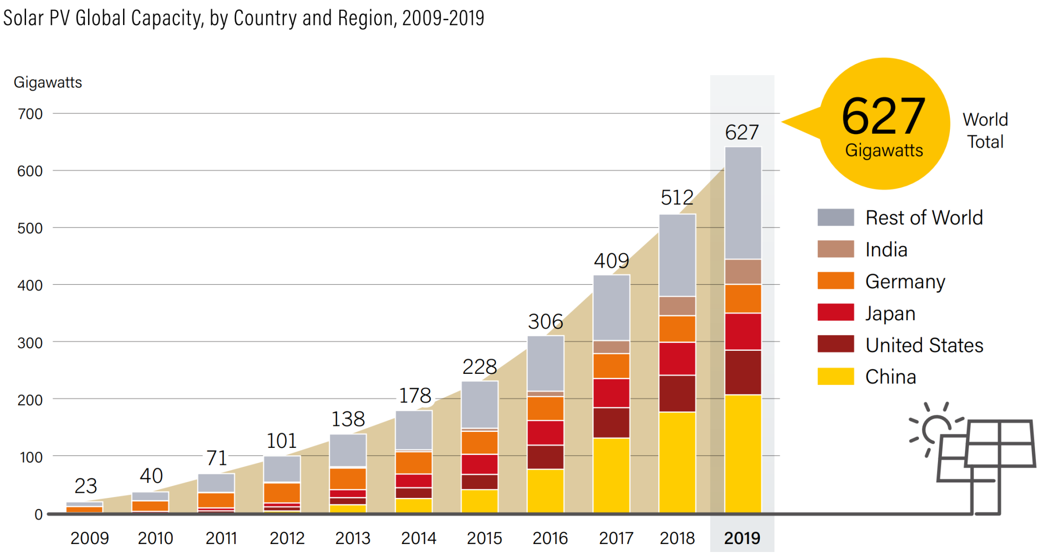


Figure 1.2: Proliferation of PV Systems Across the World (Figure from [6])

However, under this experimental methodology, it becomes crucial to have mathematical models that can replicate the behaviour of a power system accurately. While the modeling

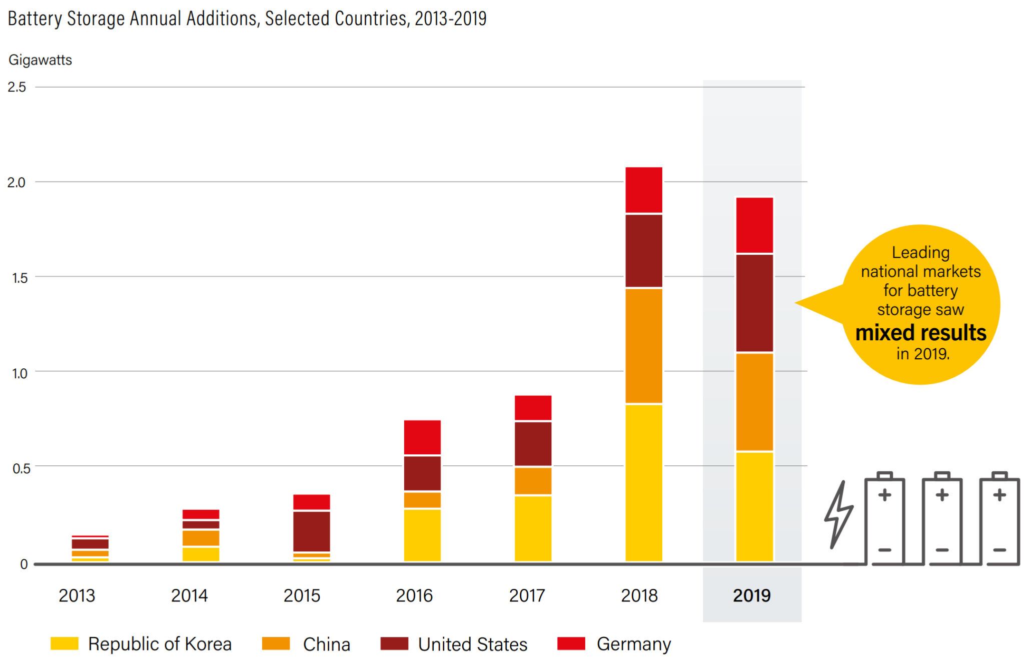


Figure 1.3: Proliferation of Battery Energy Storage Systems Across the World (Figure from [7])

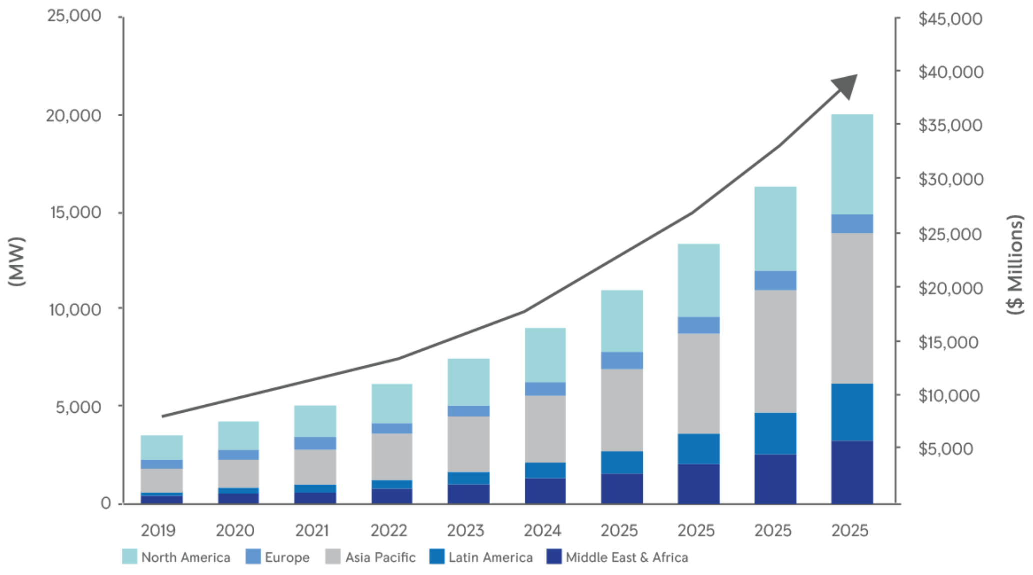
of conventional power system components are well studied and standardized, the emerging distributed energy resources (DERs) remains less explored in terms of model-standardization. When simulation models across different hardware architectures are considered, this problem becomes even more prominent. DERs such as PV-systems and Battery energy storage systems have gained massive popularity in recent times, as demonstrated in Figures 1.2 and 1.3 respectively. In spite of their recent proliferation, there have been little efforts towards standardization of these models. This reduces the reproducibility of results in power systems research, to a great extent. To address this issue, this thesis proposes homogeneous cross-platform models for PV-cells and Li-ion batteries which can be utilized to construct accurate models for PV or BESS based DERs across a wide variety of real-time (RT) hardware.

1.1.3 The Concept of Microgrid

Recently, the power grid has gone through significant structural and operational changes due to the advent of renewable energy sources. With more renewable energy resources and energy storage systems integrated into the electric power network, the power-system-monitoring dynamics are changing, because of the potential for these new sources to operate on a much lower (distribution level) voltage level. All these distributed energy resources have the ability to supply some loads, while being disconnected from the bulk grid. At the same time, these sources may have the capacity to connect to the grid and operate in a synchronous grid-connected mode. To enable these functionalities, the concept of microgrid is becoming more and more relevant as it provides a unique way of operating those small-scale and distributed energy resources in the existing framework of the power grid. The term *MicroGrid* was first formally defined by R.H.Lasseter in [8] as:

> *The MicroGrid concept assumes a cluster of loads and microsources operating as a single controllable system that provides both power and heat to its local area. To the utility the MicroGrid can be thought of as a controlled cell of the power system.*

Microgrids are gaining a lot of relevance both in the energy systems of developed and developing nations. According to a recent survey [9], the total capacity of microgrids in the USA has increased three-folds in the last 5 years. The projection reported in Figure 1.4 shows similar trend across the world.



Source: Navigant Research

World Markets: 2019 - 2028

Figure 1.4: Projected Growth for Microgrids' Capacity Across the World (Figure from [11])

A conventional fixed SCADA (Supervisory Control and Data Acquisition) system functions well for a standalone microgrid (e.g. marine applications). However, in real power systems multiple microgrids must operate as a *"cluster"* based on various operational parameters and network status as described in [10]. When the microgrid infrastructure is scaled up to microgrid *'cluster's* that is expected to operate depending on various factors like power generation, stored energy, climate, load demand, to name a few. A conventional SCADA system does not provide the flexibility to operate and control all those different sources of power generation. Hence, the application of Synchrophasor based WAMS/WAMPAC systems (which were traditionally used in transmission level) to form, operate, and control microgrids or *'cluster'*s of microgrids has become a relevant area of research.

Because those *'cluster'*s of microgrids will typically have a lot of small-scale sources (e.g. microturbines, fuel cells, PV panels), they are controlled individually via their power electronic interfaces. On top of these individual microsources, there should be a functional energy management system that will optimize the operations of all the microsources, minimize their losses, maximize their operational efficiencies, ensure that the microgrid satisfies the operational contracts with the existing transmission system, and provide necessary ancillary control and protection functionalities. This requires that the overall control-architecture becomes hierarchical and reconfigurable.

1.1.4 Utilizing Synchrophasor Technology to Improve the Monitoring, Control and Protection Functionalities for Microgrids

For microgrid applications featuring various DERs, an extensive infrastructure like WAMS/WAMPAC would require substantial instrumentation-hardware (substation PDCs and aggregrator PDCs), network extensions and controller installations. Thus, the overall implementation would be cost-prohibitive. To address this issue, this work proposes a dedicated centralized synchronization hardware to replace aggregation PDCs, and some decision making, into a single piece of hardware. This particular hardware is termed as **Synchrophasor Synchronization Gateway (SSG)**. The differences between the existing architecture and the proposed architecture, are illustrated in Figures 1.5 and 1.6.

Because, the SSG is implemented on a reconfigurable platform, it is possible to extend its functionalities. Since, microgrids employ a hierarchical control architecture as suggested by [8], [12], and [13], it is possible to partition the various control actions, and implement some part of those control functionalities inside the SSG hardware. This upgraded hardware is referred as the **Synchronized Synchrophasor Gateway and Controller (SSGC)** hardware. When compared with each other, Figures 1.5 and 1.6 reveal that the proposed SSGC based architecture has the potential to reduce the complexity and latency of the network, when put within a real-life WAMS/WAMPAC configuration.

Thus, this work comprises of the development of an SSG, implementation of PMUs that will monitor the microgrid, modeling of a microgrid and its components in real-time simulation platforms, and improvement of the SSG by incorporating intelligent control actions for distributed energy resources (DERs) and power-dispatch facilities.

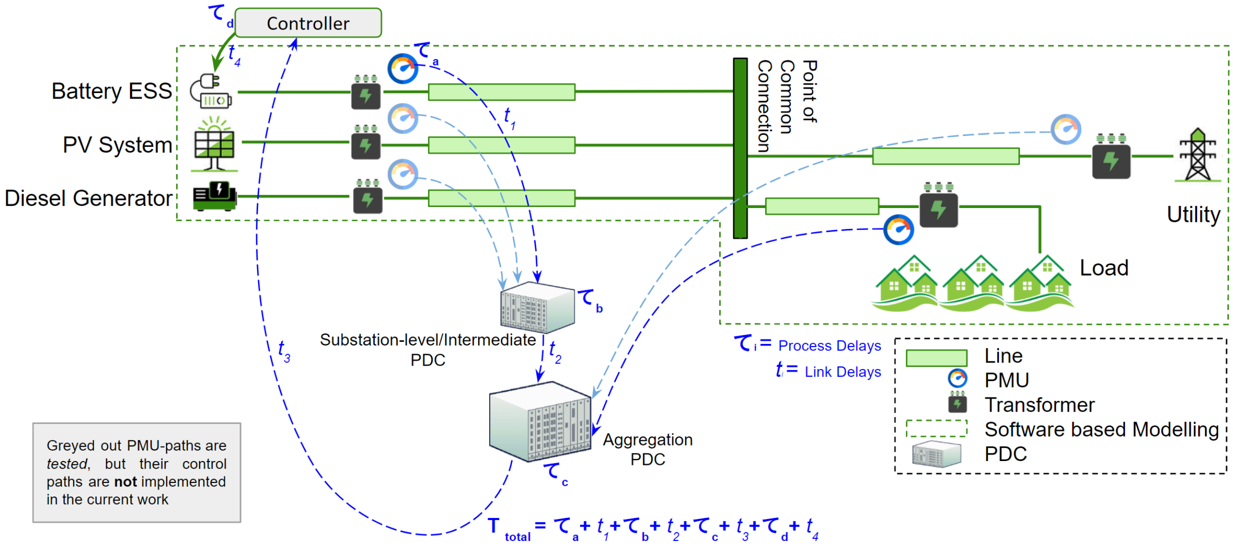

Controller
τ_d
t_4
τ_a
t_1
Battery ESS
PV System
Diesel Generator
Point of Common Connection
Utility
Load
τ_b
Substation-level/Intermediate PDC
t_2
t_3
Aggregation PDC
τ_c
τ_i = Process Delays
t_i = Link Delays
Line
PMU
Transformer
Software based Modelling
PDC
Greyed out PMU-paths are *tested*, but their control paths are **not** implemented in the current work
$T_{total} = \tau_a + t_1 + \tau_b + t_2 + \tau_c + t_3 + \tau_d + t_4$

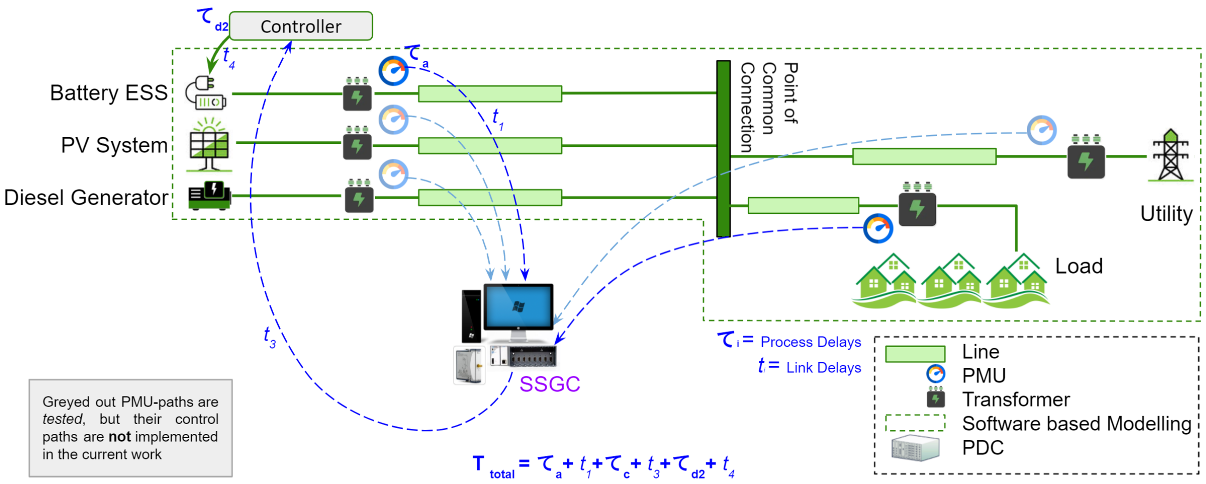
Figure 1.5: Current Infrastructure for Control Using Networked PMUs and PDCs

1.2 Research Problem-statements

This thesis addresses the following questions:

- While Field Programmable Gate Arrays (FPGAs) provide a lucrative platform for designing PMUs, *are there definite hardware-specifications which must be met to implement PMUs on a certain FPGA?*

- Industry-grade PMUs must satisfy a minimum performance-criterion when put under certain compliance-tests. *How to design a user-friendly, and configurable hardware to perform those tests on a custom PMU?*



Greyed out PMU-paths are *tested*, but their control paths are **not** implemented in the current work

Figure 1.6: Proposed Control Infrastructure for Microgrids With Networked PMUs and SSG/SSGC

- Within the realms of real-time modeling and simulation of power system components, different real-time simulators have different hardware architectures and different software libraries. *Is it possible to implement homogeneous power system components which can give identical simulation results across different real-time (RT) simulation platforms?*

- Distributed Energy Resources (DERs) often contain components whose behaviour is determined by complex physical phenomena, e.g. Photo-voltaic cells and Li-ion batteries. *How to ensure uniform performance, while such components are implemented across two different RT-simulation hardwares?*

- Traditional synchronous generators are prone to massive overvoltages under single-line-to-ground (SLG) faults. To mitigate such overvoltages, standardized 'effective grounding' has been proposed in the literature. *Are inverter based DERs susceptible to similar overvoltages when subjected to SLG faults? Are traditiona 'effective grounding' techniques suitable for application on inverter based DERs?*

- It is challenging to conduct experiments on a real power system. Thus, the development and proper network configuration for digital power grid simulation laboratory is necessary to perform real-time experiments. The main issues related to the laboratory architecture, which would be addressed in this thesis are- *How to ensure the* *reliable transmission of power system data through the network, following different protocols within the laboratory architecture? How does the timing network interact with the intelligent electronic devices (IEDs) within the laboratory?*

- *Is it possible to utilize synchrophasor streams to reduce the complexity of the traditional WAMS/WAMPAC architecture targeted for DERs and microgrids?* To answer this question, a configurable real-time hardware was designed which can operate as a gateway to synchronize multiple incoming synchrophasor streams. This hardware is further extended to incorporate supplementary control functionalities for microgrid applications. Once the proposed hardware was operational, it was stress-tested under varying network conditions.

1.3 Scope

In this section, a summary of the scopes and limitations of the current research is presented.

1.3.1 Exploration of Reconfigruable FPGA Based PMUs

As mentioned before, the WAMS/WAMPAC infrastructure is based on synchrophasor measurements obtained from PMUs. Thus, multiple PMUs were implemented on National Instruments' proprietary prototyping platform, i.e. the compact Reconfigurable Input Output (cRIO) family of devices. These devices present an excellent prototyping opportunity incorporating both a programmable hardware and a customizable graphical user interface.

Limitation: Even though these hardware family is user-friendly and flexible, their high cost makes them quite unsuitable for mass-scale production or industrial deployment. There exists other low-cost development platforms like Raspberry-Pi which had been utilized to implement PMU hardware [14], [15]. However, those implementations are not explored in the current work.

1.3.2 Modeling and Simulation for Real-Time Hardware-in-the-Loop Testing

Real-time simulators were used to model the behavior of a microgrid and all its internal sub-systems. In the current research, usage of two different real-time simulators Opal-RT and Typhoon HIL were explored. By implementing similar microgrid components in two entirely different hardware platforms, it was possible to investigate the issues that arise due to the differences between different families of real-time simulators, while migrating a simulation model from one platform to another.

Limitation: While real-time simulation of microgrid is within the scope of this thesis, no studies are to be performed in any actual power network. The synchrophasor based techniques proposed and implemented in this research will not be tested out in an actual power grid. Thus, the scope of this work is limited upto testing the proposed hardware in connection with a microgrid simulation running on real-time simulators in the lab environment.

1.3.3 Implementation of the Control and Decision Framework

Control of various energy sources in a microgrid is particularly challenging as different energy sources have different operational characteristics and capabilities. Controlling the power flow from an energy resource has been an important topic of research for a long time now [16], [17]. However, with the advent of microgrids [8] in the power and energy infrastructure, the control of power flow has become more challenging as a research problem. Efficient and autonomous management of power network is important for reliable operation of a power system. This work explores the possibility of implementing a PMU-based control and decision framework to manage a such a power network for microgrids. For the communications between the different equipments, standard TCP protocol was used. In the existing infrastructure, due to the presence of multiple substation PDCs (Phasor Data Concentrators), aggregator PDCs, PMUs and controllers in the network, latency of the decision making path increases. This work targets to reduce that latency by replacing some of that hardware with a standalone hardware termed as **Synchrophasor Synchronization Gateway and Controller (SSGC)**. In this thesis, the SSGC is utilized to implement part of the control system that controls the power flow for the battery energy storage system as demonstrated in Figure 1.6.

Limitation: Recently, substantial effort has been made to incorporate sophisticated control and decision techniques in the paradigm of power network management. A significant portion of these approaches involve the application of various machine learning methodologies. This work does not explore any such advanced algorithms. The focus of the current work is incorporating a very basic control infrastructure in a hardware absed real-time experimental setup. The scope of the proposed control system is also limited to the control

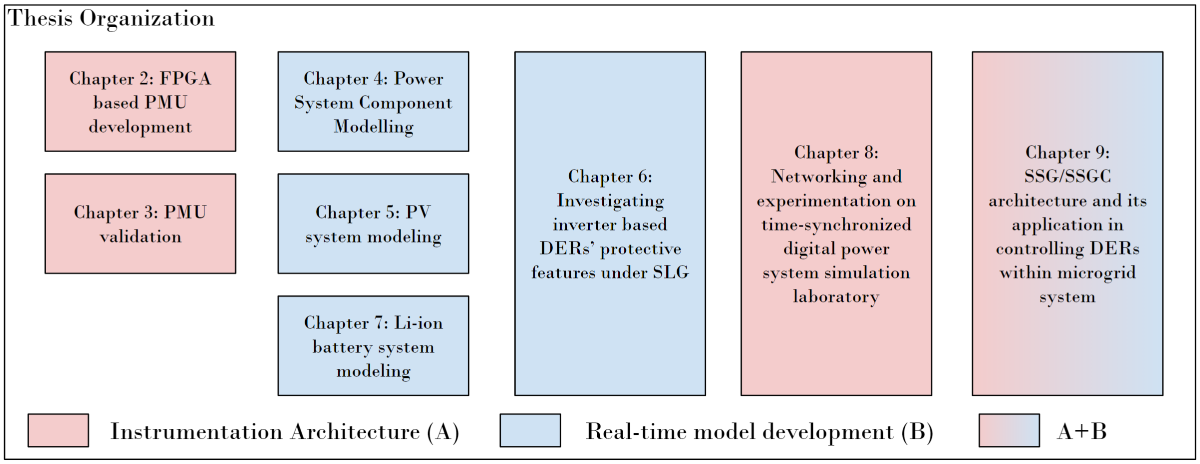

Thesis Organization
Chapter 2: FPGA based PMU development
Chapter 3: PMU validation
Chapter 4: Power System Component Modelling
Chapter 5: PV system modeling
Chapter 7: Li-ion battery system modeling
Chapter 6: Investigating inverter based DERs' protective features under SLG
Chapter 8: Networking and experimentation on time-synchronized digital power system simulation laboratory
Chapter 9: SSG/SSGC architecture and its application in controlling DERs within microgrid system
Instrumentation Architecture (A)
Real-time model development (B)
A+B

Figure 1.7: Visual Representation of the Organization for This Thesis

of battery energy storage systems. This can be extended to the control of diesel generators, PV-systems or other DERs in future.

1.4 Thesis Organization

The remainder of this document contains nine chapters. Each chapter begins with an introduction, which highlights the research objectives of that chapter, and presents a short literature review relevant to that research. Every chapter ends with a conclusion of the research contributions made in the respective chapter.

Chapter two introduces the usage of FGPAs to implement real-time PMUs. It focuses mostly on hardware requirement aspects of these PMUs. This work is based on the National Instruments' proprietary *Advanced PMU Development* infrastructure. Chapter three focuses on developing a low-voltage user-friendly PMU testing hardware, which is capable of performing the pre-compliance tests for a given PMU. Chapters four, five, six and seven explore the development of portable, cross-platform compatible power system components for real-time simulation. The cross-platform compatibility has been validated by using **Opal RT** and **Typhoon HIL** real-time simulator hardwares. While chapter four investigates the performances of basic power system components across these two hardware platforms, chapter five explores the possibilities of implementing portable models for more complicated power-systems functionalities that are relevant for microgrids, e.g. PV systems. In chapter six, the protective features of an inverter based DER under SLG faults are evaluated by performing controller hardware in the loop (CHIL) experiments. This test utilizes the PV cell modules from chapter five. In chapter seven, cross-platform modeling of Li-ion batteries are presented. Chapter eight explores the networking and experimental features of the current simulation laboratory set up at RPI. In chapter nine, the fundamental implementation and preliminary tests on the proposed *Synchrophasor Synchronization Gateway & Controller* are reported. A test where the BESS responds to the step change in the load utilizing this SSGC hardware, is also demonstrated. The SSGC hardware is then stress-tested under different network conditions to examine for its resilience and robustness. The final chapter concludes this thesis.

For the convenience of the reader, all the upcoming chapters are classified within two broad categories. These two categories are - (i) Implementation of the instrumentation architecture, (ii) Real-time model development. Chapters which introduce power system simulation models or demonstrate the experiments performed within the real-time-simulators by simulating those models, are placed in the *real-time model development* category. The chapters which focus on PMU implementation, networking and SSG/SSGC development, are classified within the *implementation of instrumentation architecture* category. This organization is visually presented in Figure 1.7. Notably, the ninth chapter connects these two worlds as it illustrates experiments with the SSGC hardware utilizing PMUs, which are connected to a simulator running real-time power system models.

CHAPTER 2
RECONFIGURABLE FPGA-BASED PHASOR MEASUREMENT UNITS

It has been mentioned in the previous chapter that, Phasor Measurement Units (PMUs) are becoming intrinsic components of modern power systems. The synchrophasor estimation algorithms in PMUs pose stringent computational demands, which makes the application of Field Programmable Gate Arrays (FPGA) highly attractive. This chapter explores the implementation of different PMU designs on multiple FPGA targets using Xilinx and NI software and hardware infrastructures and toolsets. In this process, a metric has been formulated to predict FPGA-target hardware requirements. The metric allows to predict if an FPGA-target meets the needs to deploy a given PMU design resulting in significant engineering design time savings. Because, compilation/synthesis on FPGAs is a time consuming job, this metric can reduce the implementation time for FPGA based PMUs drastically and can help in determining if additional functionalities can be added on an existing hardware platform.

2.1 Introduction

This section contains the motivation behind this part of the work, followed by a brief chapter-specific literature review. It also highlights the key research-objectives of the research illustrated in this chapter.

2.1.1 Motivation

Field Programmable Gate Arrays (FPGA) have become a very appealing hardware platform for algorithm implementation and embedded controls in various electrical engineering applications. Authors in [18] have explored that FPGAs provide a better throughput when compared to high-performance DSPs. It is also important to note that, FPGAs can

- Portions of this chapter previously appeared as: P. M. Adhikari, H. Hooshyar, and L. Vanfretti, "Experimental quantification of hardware requirements for FPGA-based reconfigurable PMUs," *IEEE Access*, vol. 7, pp. 57527-57538, Apr. 2019, doi: 10.1109/ACCESS.2019.2911916.

typically provide a sampling frequency upto 100 Mhz. (i.e. for Xilinx Spartan 6 family of FPGAs the primary clock frequency is 100 MHz. For more advanced FPGAs (e.g. Xilinx Ultrascale Virtex 7 family) the maximum frequency can be as high as 500 MHz [19]. In the current architecture though, the voltage and current sensing is executed via NI C-series modules, which have a maximum voltage and current sampling rate of 50000 samples/second [20]. Because, the computing hardware is capable of functioning at much higher frequency, it is possible to generate data seamlessly in the current setup, without the need of any further optimization at the hardware level.) In power system engineering, Phasor Measurement Units (PMU) are used for real-time synchornized measurement of various power system quantities. Given the computational demands of PMU algorithms and their functionalities, the use of FPGAs for PMU implementation has become of recent interest.

Currently, there are only a few available implementations for FPGA-based PMUs in the literature. The notable implementations are the Reason MU320 from General Electric[21], the National Instruments' Advanced PMU Development System which is also used in [18], and another NI based design reported in [22] which is commercialized by Zaphiro Technologies. This paper is limited to the National Instruments (NI) based implementations only. The default National Instruments implementation uses the real-time compact-RIO 9068 with a Xilinx Artix-7 FPGA for their PMU. However, the possibility of implementing the same PMU design across other FPGAs has not been reported, and no guideline for multiple-target hardware implementations is available. This paper investigates the possibility of using the same PMU design on other real-time targets with FPGAs from different families. The same is performed for the PMU in [22]. In this work, a new metric has been formulated to determine the requirements to deploy the two different PMU designs in terms of hardware consumption. These metrics can help to quantify hardware specifications for these PMU-designs, and reduce the design time and effort drastically.

2.1.2 Related Works

The major advantages of using FPGAs for complex system implementation are listed and described in [23]. With the configurable hardware architecture, FPGAs are extremely efficient for implementing high-speed, data-intensive applications. With the inherent reprogramming features of the hardware, they can be programmed for any particular application at a much lower level. Hence, as a platform FPGAs provide much better real-time performance. Available research where FPGAs were used to design PMU functions are discussed in [22], [24], [25], [26], [27]. The implementation in [22] was based on the hardware platform from *National Instruments technologies*, similar to the infrastructures described in [24], [25], [27], [28], [29] and [18] where various applications and observations for single-platform FPGA-based PMUs are described. Because [22] uses a similar infrastructure as described in this paper, the same PMU design is used herein while characterizing different FPGA platforms. However, none of these previous works, discusses the challenges related to single-design-multiple-platform implementations.

[30], [31] and [32] present an extension of the design in [22], in terms of analysis, estimation and decision making in a power system. In [33], the challenges of protocol implementation for FPGA based PMUs using open source software are reported. [34] and [35], on the other hand, presented the application of FPGAs to implement real time control hardware. A recent report on using the similar infrastructure for PMU implementation has been published by the processor industry-leaders in [18].

Some standard methodologies for bench-marking FPGA cores are described extensively in [36]. In [37], FPGA cores are characterized in relevance to DSP algorithms. In [38], Xilinx Virtex-1 to Virtex-5 families are reviewed in terms of their applicability in critical operations.

2.1.3 Highlights of the Chapter

The key research contributions encompassed in this chapter are:

- To describe methods to deploy PMU designs (implemented using National Instruments' infrastructure) on multiple FPGA targets. This single-design-multiple-platform infrastructure allows easier implementations of real-time PMUs on FPGAs.

- To derive a metric that helps to predict if a PMU design can be implemented and executed on a certain FPGA target. It needs to be noted that this metric is specific to existing PMU implementations only, and not expected to work for any generic FPGA-based digital system.

- The proposed metric is then validated using different FPGA targets for different PMU designs. Both physical hardware and virtual FPGA emulation were used for this validation process.

- Finally, comments are made regarding the cost-effectiveness FPGA based PMU implementations in light of the experimental results along with some of the recent developments in this field by the industry.

2.1.4 Structure of the Chapter

In the next section the overall infrastructure of the FPGA based PMU implementation is discussed. The detailed description of the PMU software, hardware, and overall work-flow for hardware implementation are described. Case Studies and Experiments are described in Section 2.3 which is divided into four subsections, describing test specimens, test-procedures using Physical FPGA hardware, test-procedures with virtual FPGA emulations and lastly, methods for characterization of error. While describing experimental results in Section 2.4, a metric for projecting the hardware requirements for FPGA based PMUs is proposed. That proposed metric is verified using both physical FPGA targets and virtual FPGA emulation techniques. Finally, cost vs capacity trade-off for the selection of FPGAs is explored.

2.2 PMU Software, Hardware, and Design Flow

This section describes the PMU designs (software), hardware and the design flow methodology used in this study. The generic PMU-implementation provided by National Instruments as a part of their Advanced PMU Development System and the implementation in [22] are used. The PMU designs used are only supported by proprietary Compact Reconfigurable I/O (cRIO) devices. It is important to note that all the experiments and observations presented in this paper were based on M-class PMU designs.

2.2.1 Software Implementation of the PMU

The PMU design was implemented in the LabVIEW Virtual Instrumentation (VI) environment. The organization of each design block is shown in the block diagram of Figure 2.1. Software components are divided into the host PC and the client compact RIO device. The compact RIO houses a RT-microprocessor and an FPGA. The FPGA target is used to implement the PMU design, whereas the rest of the cRIO runs the TCP/IP communication interfaces and performs other auxiliary functions. The overall execution on the cRIO depends on LabVIEW libraries (Electrical Power Library) and DLLs. The cRIO chasis contains the physical FPGA device. The FPGA device incorporates the PMU design, along with FPGA-

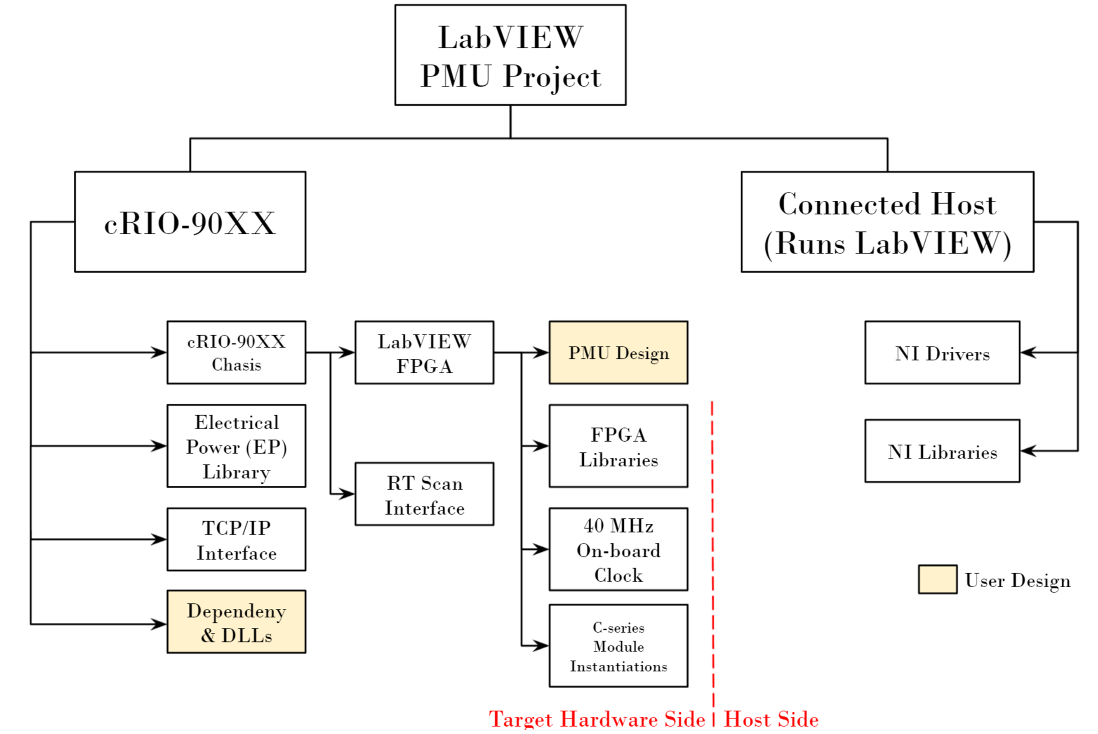


Figure 2.1: Software and Hardware Resources of the PMU Implementation

specific library files from National Instruments, and all the instantiations of the C-series modules for analog to digital signal acquisition.

The host PC communicates with the cRIO (running the PMU and TCP communication), via NI drivers, and uses NI library functions for this communication. The PMU design described so far is supported by the latest LabVIEW release of NI [19], and will be referred as NI PMU from here on. The PMU design from [22] will be referred as Beta PMU.

2.2.2 Review of the Existing PMU Hardware

NI provides a PMU implementation for the cRIO-9068 device (Figure 2.3). A cRIO-9068 houses a Xilinx Artix-7 FPGA and an ARM Cortex A9 processor as its real-time processing unit. The 3-phase voltage and current inputs were taken using NI-9225 and NI-9227 C-series modules that are capable of obtaining 50 K samples/second, with 24-bit resolution. The master time is specified by the NI-9467 GPS synchronizer module. When used together with an FPGA, this module synchronizes the internal 40 MHz FPGA time keeper clock within (+/-)100 ns to the GPS 1 PPS received from GPS satellites. All the

Table 2.1: Hardware Platforms, FPGAs and Processors

Compact-RIO Device	FPGA Family/Device	Real-Time Processor
cRIO 9068	Artix 7 /ZYNQ 7020	ARM Cortex A9
cRIO 9082	Spartan 6/LX 150T	Intel Core i7-660UE
cRIO 9081	Spartan 6/LX 75T	Intel Celeron U3405
cRIO 9074	Spartan 3/ XC3S200	PowerPC

outputs from these modules are used by the PMU algorithm running on the FPGA hardware.

The FPGA-based PMU communicates with the real-time unit running on the cRIO, which handles initialization, real-time communications, and broadcasting of the PMU output. The PMU output follows the IEEE C37.118-2005 protocol, and is broadcasted through an ethernet port using TCP/IP.

Similar hardware configurations were used in four different compact RIO platforms, with different specifications as listed in Table 2.1. For all of them, the detailed synthesis results provided by the Xilinx Vivado tools were studied and analyzed. One of the PMU hardware platforms used, is shown in Figure 2.3.

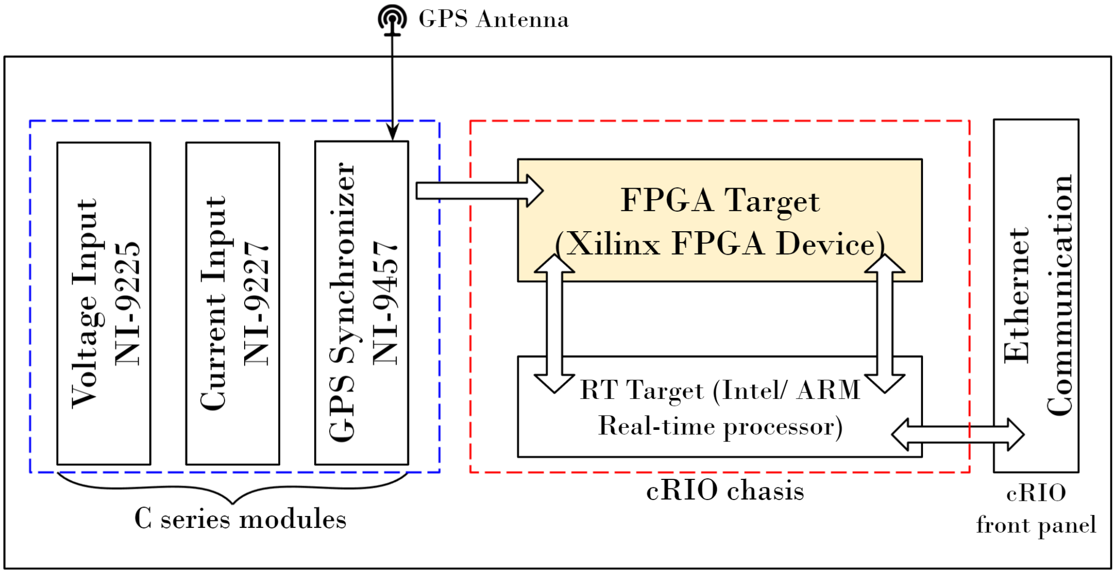


Figure 2.2: Block Diagram for the Hardware Architecture of the PMU Assembly

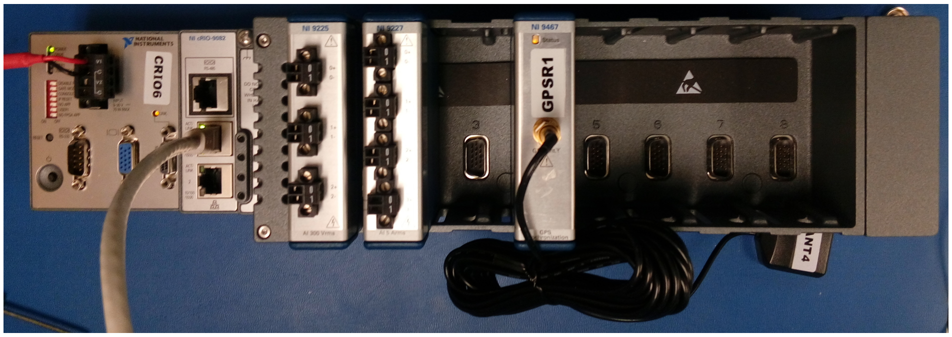


Figure 2.3: Physical PMU Hardware in a Compact-RIO 9068

2.2.3 Hardware Design Flow Methodology

In this subsection, the process for designing FPGA based hardware is described. The design flow methodology adopted in this work is described in the flow-chart shown in Figure 2.4. Note that the flow contains software infrastructure from both National Instruments (LabVIEW libraries) and Xilinx (Vivado synthesis tools). It can be observed that the PMU design is carried out in the LabVIEW Virtual Instrumentation environment from NI. The design is then converted into intermediate HDL (Hardware Description Language) files by NI libraries. Those files are converted into a bit-stream by Xilinx Vivado synthesis tools. This is a lengthy process, and it utilizes the libraries provided by Xilinx. This part of the design-flow consists of four steps: analysis, synthesis, mapping, and place & route. The Xilinx framework is designed to generate a report (pre-synthesis) after analysis, a report after synthesis (post-synthesis) and a last report for the place & route process (post place & route report).

These reports were analyzed to develop the design metrics for two different PMU designs. Note that, for some families of FPGAs, the pre-synthesis report is not generated. Because, this part of the work is time-consuming and computationally intense, National Instruments' online Cloud Compile Service (a high-performance-computing cluster) was used.

The same flow was used for all FPGAs studied and the reports were analyzed to develop metrics that can determine if a PMU design can be put on a certain FPGA. Because the pre-synthesis report takes the least amount of time, the developed metric analyzes the pre-synthesis report for one specific FPGA. Once the metric derived allowed the prediction of the hardware resource requirements, further experiments were conducted using the FPGA emulation for verification.

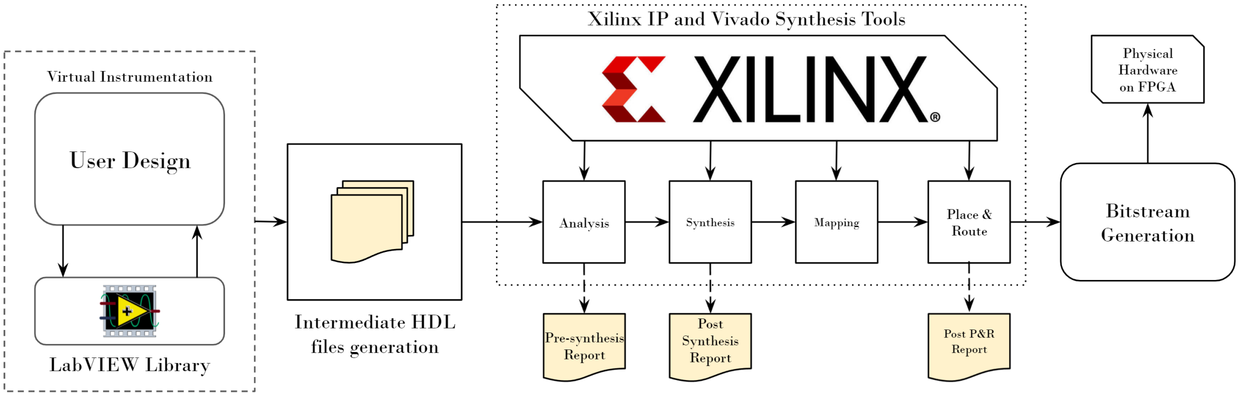


Figure 2.4: The FPGA Based Hardware Implementation Flow and Report Generation

2.2.3.1 Observations Related to the NI Cloud Compile Service

As mentioned before, the National Instruments high performance computing cluster service (Cloud Compile Service) was used for compilation and synthesis of the designs. It is to be noted that, theoretically this entire procedure can be executed offline. However, the compilation time is usually 6-7 times more in a standard work-station (Intel i7, Gen 3 processor with 16 GigaBytes of RAM) when compared to the Cloud Compile Service. Because, the procedure is computationally demanding, sometimes the resources of standalone PCs are not enough to complete the compilation. Hence, it was recommended by NI to always use their compile service for this part of the job. Usually, compilation time on such servers depends on the number of jobs it is handling at a given time. However, in this particular infrastructure, the compilation time was always similar, but the queuing time/ waiting time used to vary depending on the number of compilation jobs it was handling at the given time.

2.3 Case-Studies and Experiments

The experiments performed were classified in two categories, which are described in subsections 2.3.2 and 2.3.3 below. The test specimens are described in subsection A. The design flow in Figure 2.4, was carried out for several targets using the re-designed PMU software and reports were regenerated.

2.3.1 Test Specimens/Designs

Three test-specimens were used to characterize hardware requirements. Two of them were the PMU implementations (NI PMU and Beta PMU described in Section II.(a). Additionally, a simple 64-point FFT design was used across all the platforms for additional validation of the proposed hypothesis. A short description of these specimens is given below.

- Dummy Design: A 64-point FFT algorithm that was expected to be synthesized successfully in all available platforms. It is to be noted that, there is no direct relation between this design and the actual PMUs implemented in the cRIO devices. The results for this design is reported only to infer the fact that even though smaller designs can be synthesized on lower cost hardware with less resources, larger real-life applications such as a PMU requires stronger hardware for implementation. As it will be shown in Section 4, the two PMU designs can not be implemented on some of the targets due to insufficient hardware resources. For those targets it was necessary to use a simpler design (FFT Algorithm) to compare reports generated in all the different platforms.

- NI PMU : The latest PMU design provided by NI, which incorporates all the latest LabVIEW libraries. This design is implemented and released by NI as a part of their Advanced PMU Development System. Their implementation used a ZYNQ-7020 Artix-7 FPGA along with a NI-9068 cRIO and was modified for synthesis in other platforms.

- Beta PMU: This particular implementation [22] was developed for the NI-9076 with a Spartan 6 LX45 FPGA. Although, it has similar functionalities, the hardware requirements are different from the NI PMU design.

It is to be noted that, even though the two PMUs share the same functionalities, the phasor estimation algorithm used was different. The NI PMU utilizes a recursive DFT algorithm for frequency estimation which is well documented by the authors in [39]. The Beta PMU, however, utilizes a recently published algorithm - iterative Interpolated DFT (i-IPDFT) for estimating the frequency [22]. As the i-IPDFT algorithm is iterative, and the DFT is recursive, the FPGA hardware requirements are expected to be sufficiently different.

2.3.2 In Vivo Experiments on Physical FPGA Devices

The test specimens described in Section 2.3.1, were used to synthesize the functions for the physical hardware. The hardware platforms, which were used for this purpose are listed in the Table 2.1.

cRIO displays the detailed results for all the synthesis-jobs carried out. In column 6, the number of LUTs used are displayed, and column 5 displayed the number of available LUT-s in that specific FPGA core. In the case of the cRIO 9082, the FPGA core is a Spartan 6 LX150T which contains 92152 LUTs, each having 6 inputs. Out of those 92152 LUTs, 28826 were used as shown in column 6. Column 10 shows the number of available Block RAMs in that FPGA core. This number is 268 for the Spartan 6 LX150T FPGA. Out of these BRAMs, only 23 were used. As shown in Table 2.3, the Beta PMU design stresses the Block-RAM (BRAM) demand and the NI PMU stresses the LUT (Look Up Table) demand of the FPGAs. Experiments based on the Dummy (FFT) design were performed to establish that the proposed metric in Section 4 can be extrapolated for different algorithms other than PMUs.

2.3.3 In Silico Experiments using FPGA Emulation

National Instruments provides tools to virtually implement a design on an FPGA, without the need to access the physical FPGA hardware. This process is known as FPGA emulation. In this process, Xilinx Intellectual Property(IP) is used to emulate the actual behavior of an FPGA. FPGA emulation was used to predict whether or not the PMU designs can be executed on the other cRIO devices with different FPGA targets and results were compared against the proposed metric. As seen from Table 2.5(second row), it predicted and verified that for the Spartan 3 XC3S2000 FPGA, the available LUTs are not sufficient to host the PMU design. Based on the proposed metric, it was successfully predicted whether the PMU would fit in for other FPGA cores as well.

2.3.4 Characterization

As seen from the flow chart in Figure 2.4, Xilinx provides important information in different stages of the overall process. The first set of reports is generated after the analysis stage, and requires relatively lower time to generate because it does not involve time consuming stages, i.e. synthesis, place & route and mapping. However, this report only gives

Table 2.2: Time Comparison of Report-Generation For Various Stages

	Pre-Synthesis	Synthesis	Post P & R
Spartan 6 LX 150T	9 Min	22 Min	122 Min
Spartan 6 LX 75T	12 Min	30 Min	187 Min

a rough estimate of the actual device utilization. Regardless, the entire procedure tends to take hours. It is observed in Table 2.2 that for a Spartan 6 LX-150T the time for place & route increases by 13.5 fold, and for a Spartan 6 LX-75T it increases 15.6 fold as compared to generating pre-synthesis results. However, pre-synthesis reports are not certain and they do not guarantee that the identified requirements at this stage will remain the same when proceeding with the subsequent stages. The results in Table 2.2 show that it is extremely inefficient to run a complete synthesis job resulting in unsuccessful execution due to hardware constraints. Therefore the metric proposed in this work uses the pre-synthesis report to predict whether or not a certain design can be implemented and executed on a certain FPGA target as shown next.

2.4 Experimental Results

2.4.1 Results on Physical FPGA Hardware

2.4.1.1 Hypothesis and Metric Derivation

As mentioned before, three designs were implemented on four physical hardware platforms. The final hardware consumption by these designs across the 4 different FPGA devices are reported in Table 2.3.

It is to be noted that, the cRIO 9068 is the only device, for which the NI PMU implementation was originally developed and the cRIO 9076 in the case of the Beta PMU. It was observed from the synthesis reports, that the two determining factors that decide whether a design can be implemented on a certain FPGA target are the number of available Look-Up-Tables (LUT-s) and number of available Block-RAM units. The functional blocks such as adders, multipliers, multiplexers, are all implemented using LUTs and BRAMs through the synthesis process. Hence, the number of LUTs and number of BRAMs should be the most important metric to quantify the hardware consumption on FPGAs for PMU implementation.

However, the most interesting observation across all the different platforms was that

Table 2.3: Synthesis Statistics for Hardware Compilation for Different Designs

1	2	3	4	5	6	7	8	9	10	11	12
Design	Hardware	FPGA	FPGA	Available	LUT-s	LUT Size	Available	Consumed	Available	Used	Synthesis
		Family	Device	LUT-s	Used (L)	(Inputs) (I)	LUT × Input	LUT × Input	BRAM	BRAM	Outcome
NI PMU	cRIO 9074	Spartan 3	XC3S2000	40960	42892	4	163840	171568	40	26	**Failed**
	cRIO 9076	Spartan 6	LX45T	27288	28823	6	163728	172938	116	26	**Failed**
	cRIO 9081	Spartan 6	LX75T	46648	28824	6	279888	172944	172	23	Passed
	cRIO 9082	Spartan 6	LX150T	92152	28826	6	552912	172956	268	23	Passed
	cRIO 9068	Artix 7	XC7Z020	53200	34909	5/6[a]	319200 [b]	174545	140	21	Passed
Beta PMU	cRIO 9074	Spartan 3	XC3S2000	40960	6539	4	163840	26156	40	48	**Failed**
	cRIO 9076	Spartan 6	LX45T	27288	4324	6	163728	25944	116	48	Passed
	cRIO 9081	Spartan 6	LX75T	46648	4326	6	279888	25956	172	48	Passed
	cRIO 9082	Spartan 6	LX150T	92152	4344	6	552912	26064	268	48	Passed
	cRIO 9068	Artix 7	XC7Z020	53200	17226	5/6	319200	86130	140	33	Passed
64 Point FFT	cRIO 9074	Spartan 3	XC3S2000	40960	4561	4	163840	18244	40	33	Passed
	cRIO 9076	Spartan 6	LX45T	27288	3188	6	163728	19128	116	26	Passed
	cRIO 9081	Spartan 6	LX75T	46648	3193	6	279888	19158	172	29	Passed
	cRIO 9082	Spartan 6	LX150	92152	3215	6	552912	19290	268	29	Passed
	cRIO 9068	Artix 7	XC7Z020	53200	13245	5/6	319200	66225	140	28	Passed

[a] LUTs are of configurable sizes, while most of them were conFigured as 5 (some as 6) input LUTs

[b] Estimated Maximum Count assuming each of the LUTs were utilized to their fullest extent of 6 inputs

Table 2.4: Cross Validation of the Derived Metric with HDL (Verilog) Designs

Design Under Test	Spartan 3 XC3S2000: LUTs Consumed	Spartan 3 XC3S2000: LUT Size	LUT Consumed x LUT Size	Spartan 6 LX75T: LUTs Consumed	Spartan 6 LX75T: LUT Size	LUT Consumed x LUT Size
Square root Finder	27	4	108	19	6	114
64 bit ALU	118	4	472	79	6	474

the same PMU design consumed different amount of hardware in terms of LUTs in different FPGA targets, which contradicts the previous assumption to take number of available LUTs as the metric to quantify the hardware consumption.

To address this issue, together with cRIO, detailed specifications of various FPGA families were studied revealing that the size of a single LUT is different among those families. For example, the Spartan 3 family of FPGAs have LUTs of width 4, where as Spartan 6 FPGAs have LUTs of width 6. Another interesting observation was that the product of the LUT size and the number of consumed LUTs for synthesis, were similar across different FPGA-targets for the same design.

Consequently, the metric in equation (2.1) (LUTs × LUT-size) was observed to be the determining factor on whether a design would fit into a certain FPGA. For the NI PMU design, noting the results in column 9 of Table 2.3, the base value of 175000 was selected as the lower bound to develop the design metric. This choice was made based on the fact that the maximum observed value of ($N_{\text{LUT}} \times N_{\text{LUT-Size}}$) was 174545 (for Artix 7) across all the tested/emulated hardware (discarding Artix 7, the maximum bound would be 172956). This value was determined by computing the nearest round number above the highest reported hardware consumption (of Artix 7), while determining the exact value of N_{Max}. However, it is not recommended to use this proposed metricw Artix FPGAs .

$$(N_{\text{LUT}} \times N_{\text{LUT-size}})_{\text{new-target}} > N_{\text{max}} \tag{2.1}$$

However, as discussed, Artix 7 has configurable LUT blocks, and hence its synthesis process is more complicated. For such FPGA cores, where pre-synthesis reports are not generated, synthesis reports are considered for our proposed decision making process. Even using synthesis reports, one can save significant amount of time savings by avoiding the time consuming P & R (Place and Route) steps, and post P & R report generation during the hardware selection process. For the Beta PMU, the Block-RAM was the determining factor as it consumes very little space in terms of LUT-s.

It is to be noted that the architecture of Xilinx Artix family is significantly different from the other families. The LUTs of the Artix 7 are re-configurable in nature and can be of size 5 or 6 according to [40]. In fact, a careful evaluation of the reports from 2.4 reveals that the synthesis tools actually conFigures quite a few of those LUTs with smaller (of size 1-4) size. Hence, the proposed metric is perfectly consistent for every target, except when using Artix 7 family of FPGAs. Note that, the Artix 7 implementation is also reported and documented by National instruments in their PMU Development system in [40]. In order to maximize the savings in terms of computational effort and time-consumption, the earliest estimations of device utilization were used to construct the upper bound N_{max} for the proposed metric. For Artix family, it was provided during the synthesis for all other tested FPGAs.

To better incorporate scenarios where the post-synthesis resource allocation is more than the pre-synthesis resource allocation, a pessimistic adjustment to the margin is be proposed. [41] specifies that the accuracy for resource allocation for pre-synthesis analysis by Xilinx has a ±15% error. Keeping that in mind, the proposed metric can be scaled by a

Where N_{LUT} is the number of available LUT-s and $N_{\text{LUT-size}}$ is the number of inputs in each LUT. N_{max}=175000 factor of 1.15. As a result, equation 2.1 is modified to

$$\frac{1}{1.15} \times (N_{\text{LUT}} \times N_{\text{LUT-size}})_{\text{new-target}} > N_{\text{max}} \tag{2.2}$$

However, this will lead to a conservative estimate of the ability of the target being analyzed, and reduction in cost/performance analysis; see Section IV.C.

2.4.1.2 Cross-verification of the Proposed Metric With HDL Designs

To investigate the validity of the proposed metric without the need to use National Instruments software tools, two simple digital designs written in Synthesizable Verilog were considered:

- Square-root finder: This was a successive approximation circuit that calculates the square root of an 8 bit number.
- 64 bit Arithmetic Logic Unit: This was a 64 bit ALU, with basic arithmetic and logical operations with two 64 bit inputs and one 128 bit output.

Columns 4 and 7 of Table 2.5, show that the hardware consumption based on the proposed metric is valid for a single digital design across different FPGA cores. Thus, it can be concluded that the proposed metric is legitimate, since it holds for completely different code-generation work-flows with or without the usage of National Instruments' libraries. It needs to be noted, that Verilog codes are at a much lower level of abstraction than LabVIEW VIs, which results in overall much lower hardware consumption.

2.4.1.3 Errors in Different Stages for Physical PMU Implementation

This section analyzes the accuracy of the intermediate reports depicted in Figure 2.4. More specifically it is of interest to determine the accuracy of the pre-synthesis report as it takes substantially less time, it will be used by the design metric (2.1). The time taken by the NI-cloud compile service to complete different stages of synthesis and generate the report is given in Table 2.2. The NI PMU design was used for the observations.

Where N_{LUT} is the number of available LUT-s and $N_{\text{LUT-size}}$ is the number of inputs in each LUT. N_{max}=175000

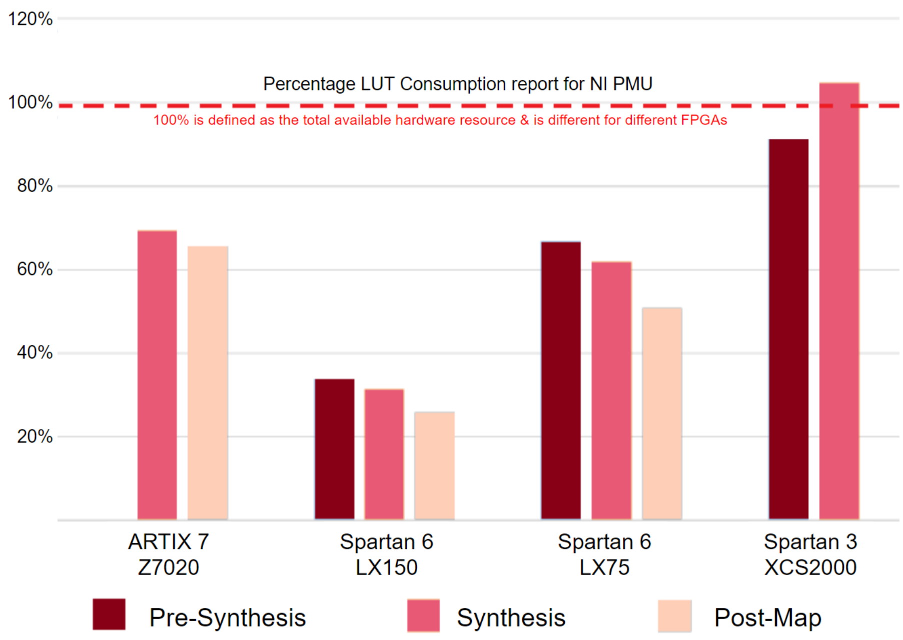


Figure 2.5: Comparison of Pre-Synthesis, Synthesis and Post-Map/PnR Results

From Table 2.2, it is seen that the use of the proposed metric can provide drastic time savings. This is relevant because one can take decisions based on pre-synthesis results, instead of running the entire compilation and synthesis job until completion.

In Figure 2.5, the results of pre-synthesis and synthesis stage are summarized with respect to the actual post place & route results. This Figure is key to understand the need for the proposed metric. The four groups displayed in this bar graph represents 4 physical FPGA cores tested in the experiments. However, it can be seen that for the Artix 7 family, there is no pre-synthesis column, as the compilation procedure for Artix family is entirely different. For Spartan-3 device, the post-map results are not reported because the synthesis job failed during synthesis due to lack of hardware resources.

The bar chart of Figure 2.5 shows the percentage LUT consumption, the available number of LUTs in each FPGA device is taken in percent. For the second bar chart, the final compilation results (post-map for successful compilations) are taken as the base value. In general, the LUT-consumption varies slightly between pre-synthesis, synthesis and post-map, because of the additional optimizations carried out by the Xilinx tools in each stage. It was observed that, during synthesis, the design is flattened out (resulting into a larger design), optimized and packed, resulting in a reduction of the hardware consumption of the

Table 2.5: Assessment of the Proposed Metric Based on FPGA Emulation Results For Unknown Hardware

Compact RIO	FPGA Family	FPGA Device	LUT Count(A)	LUT Size(B)	A × B	BRAM Count	Compilation Prediction	FPGA Emulation Results	Matched	Percentage Consumption
cRIO-9030	Kintex 7	XC7K70T	41000	6	244000	24	Success	Success	Yes	69 %
cRIO-9072	Spartan 3	XC3S1000	17280	4	69120	24	Failure	Failure	Yes	200 % +
cRIO-9113	Virtex 5	LX-50	28800	6	172800	48	Uncertain[a]	-	-	98.9 %
cRIO-9032	Kintex 7	XC7K160T	101400	6	608400	48	Success	Success	Yes	27.1 %
cRIO-9036	Kintex 7	XC7K70T	41000	6	244000	24	Success	Success	Yes	69.1 %
cRIO-9035	Kintex 7	XC7K70T	41000	6	244000	24	Success	Success	Yes	69.1 %

[a] This was a marginal case with the design consuming 99 % of the available resources

flattened design. In the case of a failed synthesis, the packing and optimization steps are skipped and the reported post-synthesis resource allocation is based on the larger flattenned out design, which is larger than the pre-synthesis resource estimate. This can be observed for the case of Spartan 3 XCS2000 device. (Column 4, in Figure 2.5)

2.4.2 FPGA Emulation Results

With the actual compilation results from four physical FPGA targets, the metric described previously was developed. It was quite successful in predicting if the FPGA targets were capable of supporting different designs. For further verification, the metric needs to be tested using a few other FPGA cores. However, it is costly to test the same design on every other physical hardware.That is why FPGA emulation was used to further test the metric. Before beginning this verification, it is necessary to determine the accuracy of the FPGA emulation tool.

Table 2.4 clearly exhibits that FPGA emulation tools were sufficiently accurate in mimicking the performance of a real FPGA target. In fact, the detailed synthesis results for FPGA emulation matched perfectly with the synthesis results of the physical FPGA targets, when the synthesis was run for the same device.

Using the proposed metric it was now possible to predict the prospective synthesis results for other FPGA targets, for which physical hardware were not available. It can be seen from Table 2.5, that in each of the test cases the proposed metric was successful in determining whether the PMU design can be implemented on the test-FPGA target.

In Table 2.5, the last two rows suggest that, the compilation job, when applied to the same FPGA hardware across different compact RIO hardwares, produces exactly same results. In this particular case results, from cRIO 9035 and cRIO 9036 (both of them contain the same Kintex 70T FPGA) are reported.

2.4.3 FPGA Selection

In this subsection, Xilinx FPGA devices are compared in terms of their cost and projected resource consumption based on the proposed metric. This set of results is an example of how this metric can be used in practice for FPGA selection. The graph in Figure 2.6 shows how the cost and projected resource consumption for PMU implementation varies across various FPGA devices of the same family. This particular test case shows results from all the devices of Spartan 6 family.

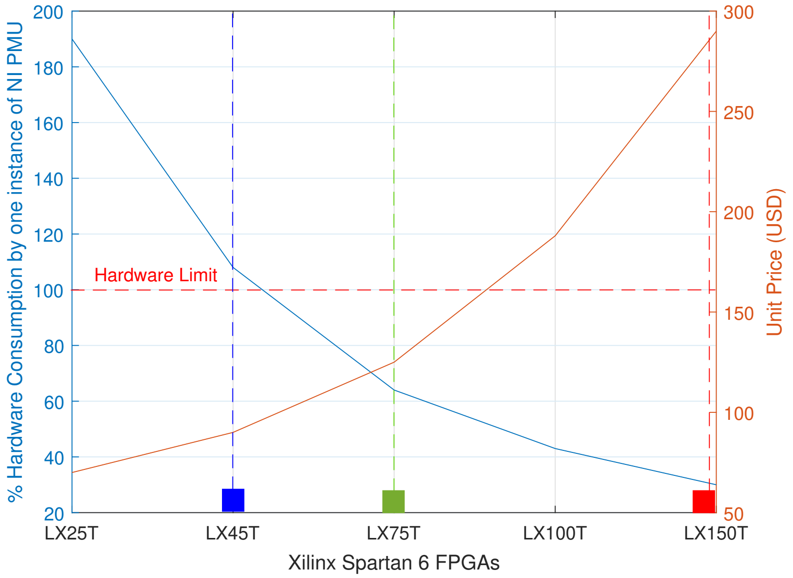


Figure 2.6: Xilinx Spartan 6 FPGA Family IC's Variation of Cost and Hardware Resources

It is to be noted, that the price of only the FPGA IC is reported in this paper. Similar FPGA ICs are also available, embedded in a development board (manufactured by both Xilinx and other third party vendors) with multiple connectivities at significantly higher prices, which are not reported in this paper.

2.5 Conclusions

In this chapter, the methods for deploying the same PMU design on different FPGA targets were discussed. That makes the single-design-multiple-target infrastructure easier to implement. In that process, a metric was developed to predict the FPGA-hardware requirements for different PMU implementations.

It is clear that, the metric proposed in the current work gives accurate prediction of the FPGA hardware requirement with satisfactorily lower effort. The experimental results show that the usage of the metric can drastically reduce the implementation time for FPGA-based PMU-s, specially in low to medium capacity FPGA targets. In the case of the cRIO 9036, it was predicted using the metric that the PMU can be implemented in the hardware and it would approximately consume 175000/244000 = 71% of the hardware. Where as, the experimental results in Table 2.5 shows that the PMU was successfully implemented in the hardware, and it utilized 69.1% of the hardware. This implies that, if a new function requiring the FPGA and consuming up to 29% of the resources was added, the target would be able to support the function. Finally, it can be concluded that the methodology proposed, provides a systematic way to chose FPGA targets for PMU implementations considering the tradeoff between cost and hardware resources.

CHAPTER 3
VERIFICATION OF THE FPGA BASED PHASOR MEASUREMENT UNITS

Chapter 2 described the details of FPGA based Phasor Measurement Unit (PMU) implementation. This chapter, aims to explore the methodologies to verify and validate such PMUs. Pre-compliance tests are intended to be used to assess PMUs' performance during their design and implementation before sending them for certification or using them for grid dynamic measurements. Methodologies for pre-compliance testing of PMUs based on the dynamic requirements of the IEEE C37.118.1-2014 standard are discussed and reported in recent literature. However, these tests are mostly performed through software simulations and there are limited test-hardwares available. To address this issue, a simple reconfigurable hardware prototype is proposed in this chapter. The implementation is carried out using National Instruments' Compact RIO family of reconfigurable hardware and it provides a fast, time-synchronous and efficient way to perform all the compliance tests on the PMU under investigation.

3.1 Introduction

This section outlines the motivations behind this part of the research, followed by a brief literature review specific to this chapter and highlights the key contributions of the research documented in this chapter.

3.1.1 Motivation

The IEEE Std C37.118-2011 and its 2014 update specifies permissible error limits for PMUs under both normal and dynamic conditions. The standard phasor estimation algorithms are designed to work with perfect time invariant sinusoidal set of waveforms. However, most of those phasor estimation algorithms are not robust enough to deal with dynamic conditions. Thus, different tolerances are specified by the standard under nominal and dynamic

- Portions of this chapter previously appeared as: P. M. Adhikari, L. Vanfretti, and H. Hooshyar, "A reconfigurable hardware prototype for pre-compliance testing of phasor measurement units," in *Int. Conf. Smart Grid Synchronized Meas. & Analytics*, College Station, TX, USA, May 2019, pp. 1-6, doi: 10.1109/SGSMA.2019.8784573.

conditions. According to the 5.5-5.7 subclauses of the standard, the PMU is expected to perform sufficiently well under certain test conditions. However, it is unrealistic to emulate these test conditions in the real-life power system. The authors in [42] incorporated these test-scenarios by feeding a PMU Software model with standard waveform files. We generated the same test scenarios in a low-voltage hardware prototype and tested two PMU systems with them. The main motivation of the current work is to provide a test-hardware prototype which can be programmed and used to test any PMU under certain conditions.

3.1.2 Related Works

Authors in [42] suggested and specified all the tests that needs to be carried out to characterize the performance of PMUs under both steady-state and various dynamic conditions in pre-compliance testing. These test conditions were provided to a PMU in [42] through standard waveform files/.csv files. In the current work we propose a hardware prototype which provides similar voltage and current waveforms in real time. Authors in [43] reported some important results of steady state compliance testing of phasor measurement units (PMUs) based on a standard industrialized relay-test set. [44] also reports similar experimental results, using a dedicated test-signal generator. However, these implementations do not include the dynamic compliance tests. In [45], all the compliance tests were performed using a Doble 6150 advanced relay test-set. Virtual Instrumentation (VI) based testing for PMUs were proposed in [46]. However, this work did not cover all the required pre-compliance tests suggested by [42]. The authors of [46] did elaborate experimental analysis for only one of the pre-compliance tests. Because, all the testing for PMUs need to be time-synchronized, time-requirements are crucial for these tests. This was explored by the authors in [47]. Overall, the literature survey revealed the lack availability for a completely autonomous and programmable test-suite for testing PMUs.

3.1.3 Highlights of the Chapter

The key research contributions described in this chapter are

- A test-infrastructure is proposed. It uses low-voltage signals to validate the functionalities of a PMU under test. The test infrastructure is based on National Instruments' hardware.

- The test-hardware can be easily controlled by a user from the GUI-based software designed in LabVIEW.
- A prototype PMU is put under the tests (using our test-suite) and its performance is analyzed and compared with the others available in literature.

3.1.4 Structure of the Chapter

In section 3.2, a brief review of the PMU compliance tests is presented. In section 3.3, the detailed description of the PMU-testing hardware is given. This section contains information about both the hardware and the software components of the proposed test-infrastructure. Special emphasis is given on the hardware components that were used to acquire the GPS signals. In the software section, the hierarchy of the source code is discussed, along with the monitoring tools used to oversee and capture the broadcasted PMU data. In section 3.4, the results of the PMU compliance tests (performed with the proposed test-hardware) by the PMU under test are reported.

3.2 PMU Pre-Compliance Testing Review

Broadly, there are four different pre-compliance tests as reported by [42], for which test conditions are briefly reviewed in this section.

- Steady-State test: Balanced three phase voltage of nominal frequency are provided to the PMU. The computed phasors must be within a specified limit set by the standard.
- Bandwidth test: In this test, sinusoidal amplitude and phase modulation is applied to a balanced set of three phase voltage and current waveforms. This can be mathematically expressed as below, where K_x is the amplitude modulation factor, K_a is the phase angle modulation factor, X_m is the amplitude of the signal, ω is the modulation frequency and ω_0 is the nominal frequency of the system.

$$X_1 = X_m[1 + K_x cos(\omega t)] \cos(\omega_o t + K_a cos(\omega t - \pi)) \tag{3.1}$$

- Frequency Ramp test: A linear increase in the system frequency is provided. This increase is applied across all the three phases. Hence the positive sequence component

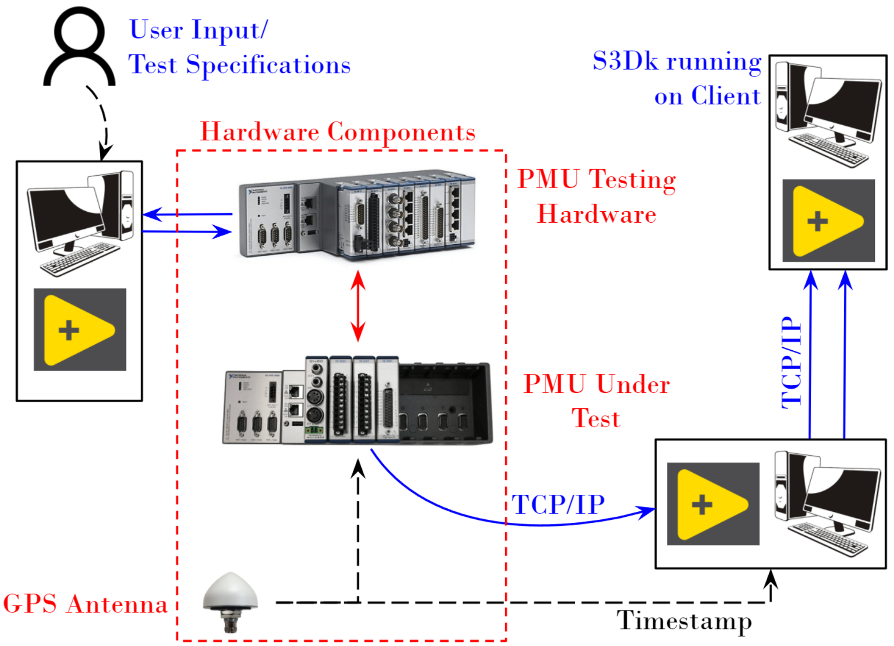


Figure 3.1: PMU Testing Infrastructure Connected With the PMU Under Test

of the system can be expressed mathematically as

$$X_1 = X_m cos(\omega_0 t + \pi R_f t^2) \tag{3.2}$$

where X_m is the amplitude of the applied signals, ω_0 is the nominal system frequency and R_f is the ramp rate in Hz/sec.

- Amplitude/Phase Step Increase test: Sudden step change in phase and amplitude are applied to the signals and the response time, overshoot and delay time is determined in the PMU measurements. Mathematically the input signal for this test can be expressed as the following.

$$X_1 = X_m[1 + K_m f_1(t)] \times \cos(\omega_o t + k_a f_1(t)) \tag{3.3}$$

where X_1 is the positive sequence component of the signal, K_m is the step size in magnitude of signals, K_a is the step size in the phase of the signals.

3.3 Proposed Testing Infrastructure

Our basic testing infrastructure consists of hardware components and software components.

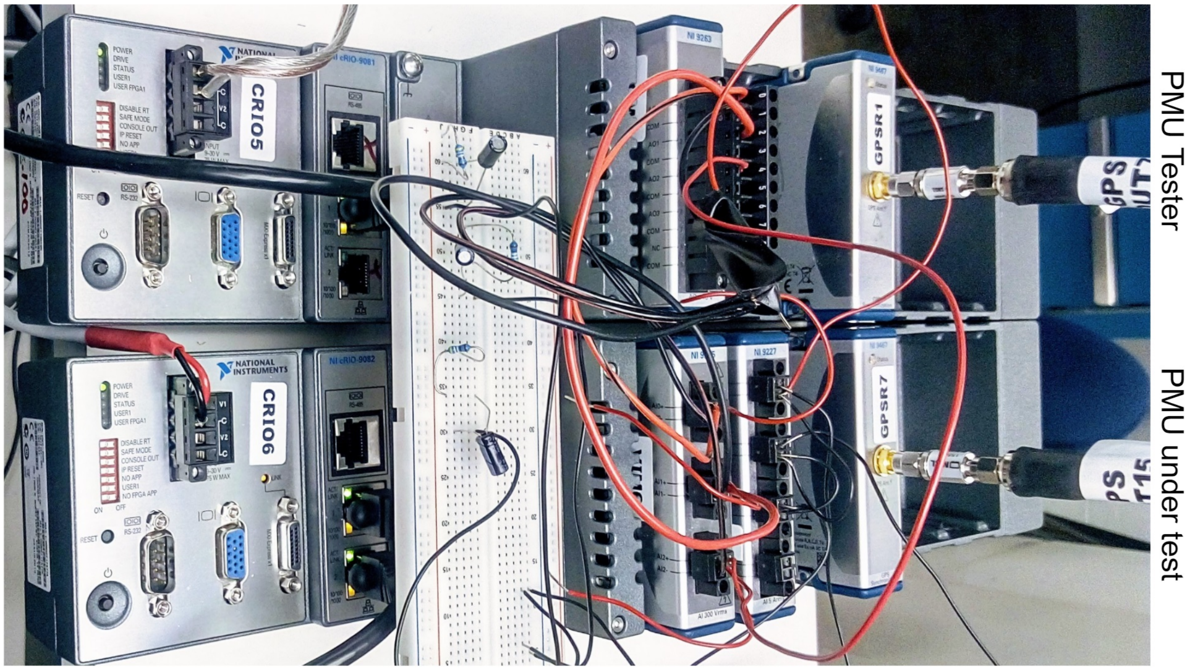


Figure 3.2: PMU Testing Hardware Connected to the PMU Prototype

3.3.1 Hardware Description

The hardware is built on a National Instruments' compact RIO 9081 device with a NI-9263 voltage output module and a NI-9467 GPS acquisition module. The data acquisition for the GPS module and data transmission for the voltage output module is controlled by the code running on the Xilinx Spartan 6 FPGA that is the in-built (inside) the compact RIO 9081 device.

Figure 3.1 shows the hardware and software parts of the test infrastructure along with a PMU prototype, which is designed and implemented in a compact RIO 9082 device. The PMU algorithm implemented in the PMU prototype is National Instruments' proprietary design for phasor estimation, which is part of their advanced PMU development system [48]. To synchronize the test-hardware with the PMU, the universal GPS time-stamp which can provide an accurate time-stamp upto 10 ns range, was used. This range was set by the specifications of the C series module NI-9467. Both cRIOs 9081 and 9082 can access the GPS signal through the NI-9467 module. (Figure 3.2)

The PMU prototype under test includes a current input module NI-9227, a voltage input module NI-9225 and a GPS acquisition module NI-9467. The voltage input module is connected to the output of the test-hardware made with the compact RIO 9081. The output

Table 3.1: Specifications of the PMU Testing Hardware and PMU Under Test

Hardware	Voltage Rating	Current Rating	Time Accuracy
PMU Tester	0-10V (Output)	0-10mA RMS (Output)	up to 10ns
PMU Prototype	0-200V (Input)	0-5A RMS (Input)	up to 10ns

of the test-hardware is connected to a simple start-connected RC load. The NI-9227 current input module is used to sense the phase current in this load, which is processed by the PMU to compute current estimates.

Figure 3.2 shows the compact RIO-based test-hardware connected to the compact RIO-based PMU prototype. Both of the compact RIO devices are receiving GPS signals from antennas via NI-9467 modules and attenuators. A three phased configurable voltage supply is configured using the NI-9263 module. On the bread-board a balanced 3 phase load is implemented using standard resistors and capacitors. For each phase a resistor of $1k\Omega$ and a capacitor $10\mu F$ was connected in series. The currents are measured from this load via the NI-9227 module by the compact RIO based PMU prototype.

The current and voltage specifications of the proposed test infrastructure and the designed PMU prototype under test are tabulated below. The voltage and current limits stated in columns 2 and 3 clearly exhibit that the PMU tester voltage and current limits are well within the that of the PMU prototype.

It can be seen from Figure 3.1 that a GPS antenna is used to provide timing information to both the PMU test hardware and the PMU under test. This connection requires a set of specific hardware as shown in Figure 3.3.

Connection Specifications for GPS Signal Acquisitions

In the current setup, a SEL 9524 GPS/GNSS antenna was used to obtain GPS signals. The antenna was connected to an RMS 116 GPS splitter since more than one GPS output are needed. To ensure that the GPS signal level from the RMS 116, is within the range of the NI-9467's input range, a DC Blocker and an attenuator were added to the connectors. All the connections were made with high quality LMR 400 cables.

A simple connection diagram for GPS signal acquisition system used in the lab is shown in Figure 3.3.

3.3.2 Software Description

Figure 3.4 shows the detailed structure in which the test-hardware is programmed. Two programs run simultaneously on the compact RIO and they interact with each other. One of them runs on the FPGA and takes care of the GPS time-stamp acquisition and voltage signal transmission to the NI-9263 voltage output module. The host-side software VIs are used to generate those voltage signals. It can be seen that, we have four different codes for four different compliance tests discussed in the previous section.

Each of the four tests are controlled from their own graphical user interfaces (GUI). One such GUI is shown below in Figure 3.5. It can be observed that, it has the necessary controls to enable or disable the applied function. Also, it can be used to control the input magnitude of the applied voltages to the PMU. As mentioned before the upper limit for this applied voltage is only up to 10V restricted by the ratings of NI-9263 voltage output module (Table 3.1). Any modification from this end, should be visible in the waveform window present in the GUI instantaneously.

The PMU readings are broadcasted via the laboratory's managed TCP/IP network, and are available to be monitored by any client connected to the network. In this particular experiments, the Smart Grid Synchrophasor Software Development (S3DK) ToolKit [49] was used to monitor the PMU measurements from remote end.

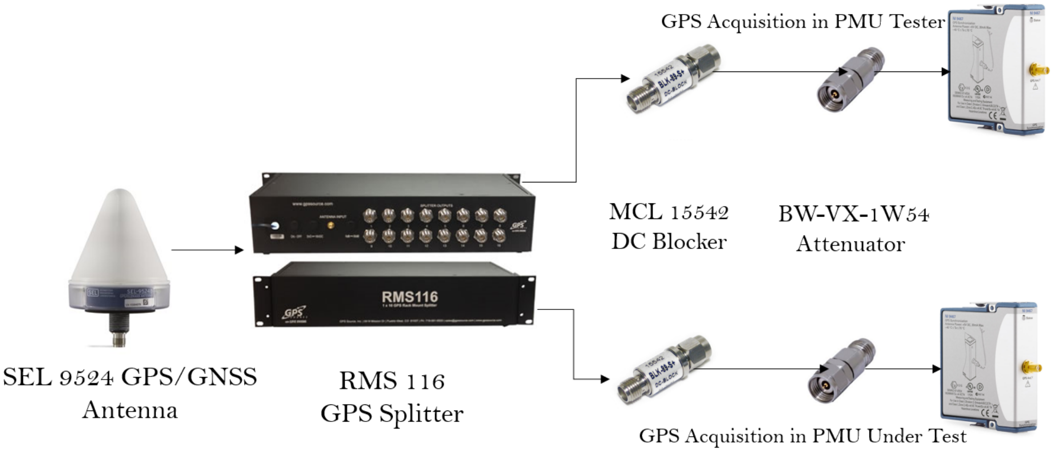
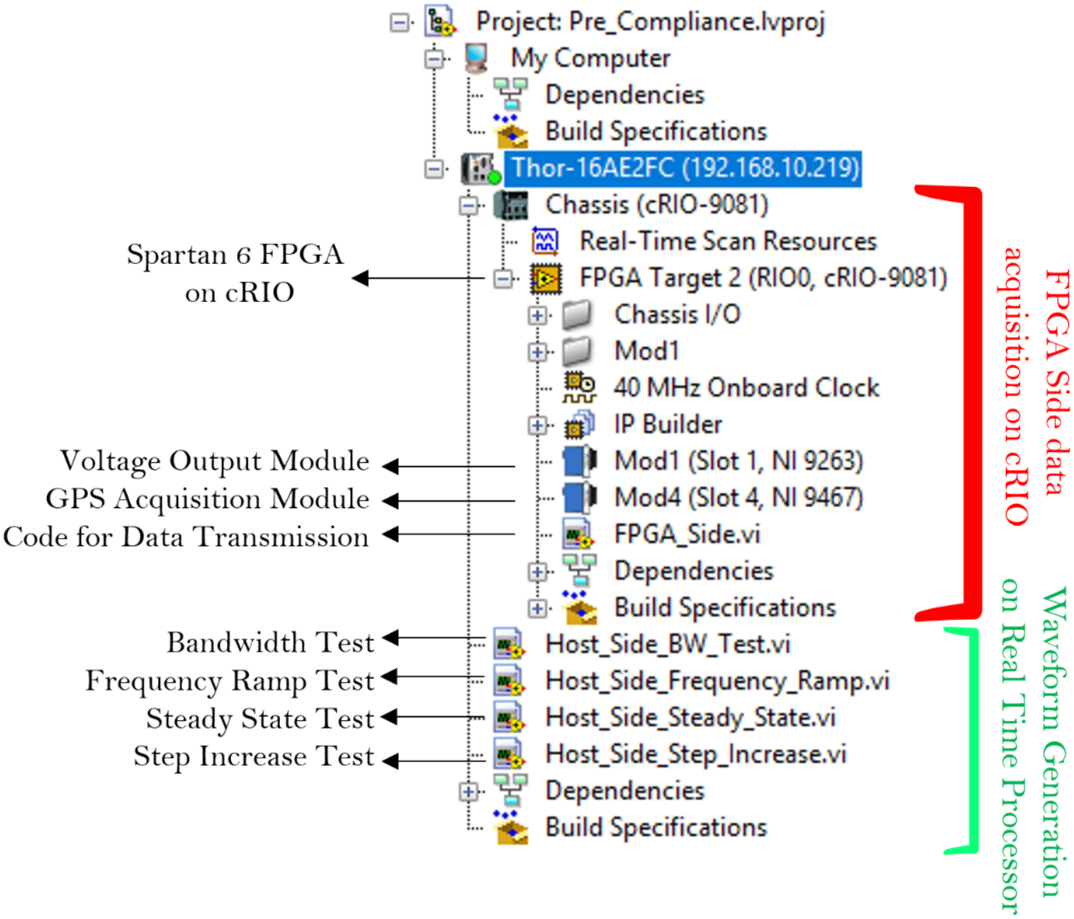
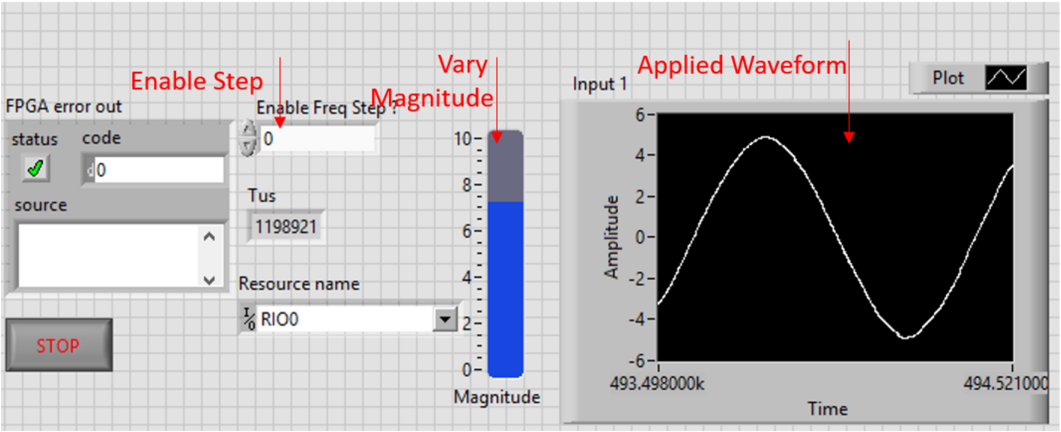


Figure 3.3: Connection Diagram for GPS Signal Acquisition



- Project: Pre_Compliance.lvproj
  - My Computer
    - Dependencies
    - Build Specifications
  - Thor-16AE2FC (192.168.10.219)
    - Chassis (cRIO-9081)
      - Real-Time Scan Resources
      - FPGA Target 2 (RIO0, cRIO-9081)
        - Chassis I/O
        - Mod1
        - 40 MHz Onboard Clock
        - IP Builder
        - Mod1 (Slot 1, NI 9263)
        - Mod4 (Slot 4, NI 9467)
        - FPGA_Side.vi
        - Dependencies
        - Build Specifications
    - Host_Side_BW_Test.vi
    - Host_Side_Frequency_Ramp.vi
    - Host_Side_Steady_State.vi
    - Host_Side_Step_Increase.vi
    - Dependencies
    - Build Specifications

Figure 3.4: Organization of the Testing Software in LabVIEW Including the Real-time and FPGA Dependencies



FPGA error out

status ✓ code 0

source

Enable Freq Step ? 0

Tus 1198921

Resource name RIO0

Figure 3.5: Sample GUI to Control/Modulate a Compliance Test

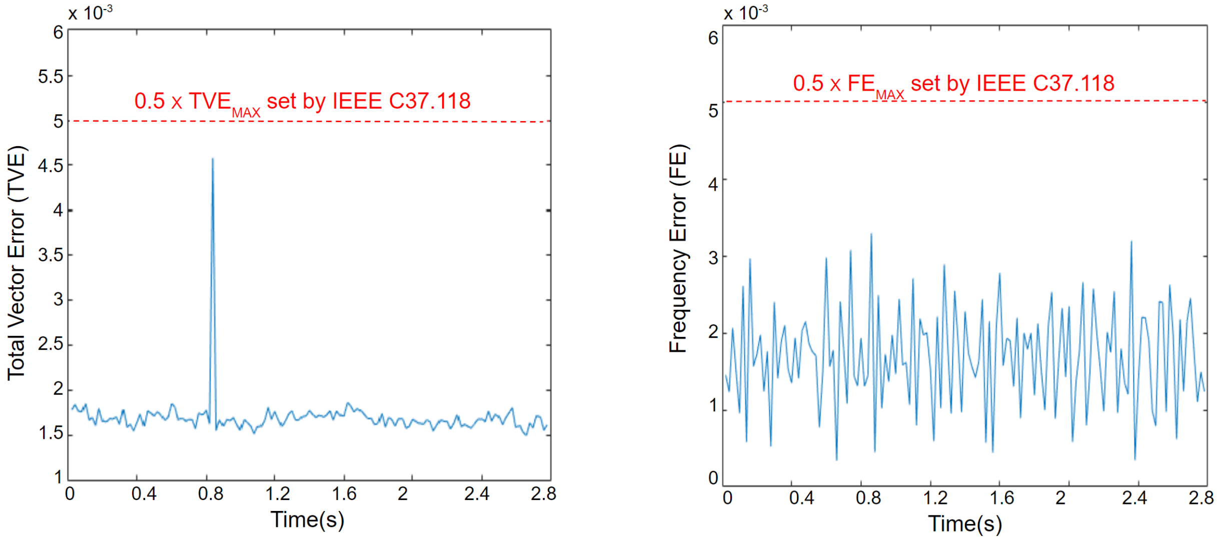


Figure 3.6: PMU Compliance Test Results for the PMU Under Steady-state Test Using the Proposed PMU-testing Hardware (Left) Total Vector Error (TVE), (Right) Frequency Error (FE)

3.4 Results

The performance of a PMU is quantified by two metrics in IEEE C37.118 standard. Those are (a) Total Vector Error (TVE) and (b) Frequency Error (FE). For a given complex variable, $X(n)$ the TVE is given by the following mathematical expression

$$TVE = \sqrt{\frac{(X_{Re_M}(n) - X_{Re}(n))^2 + (X_{Im_M}(n) - X_{Im}(n))^2}{(X_{Re}(n) + X_{Im}(n))^2}} \tag{3.4}$$

where the variables X_{Re_M} and X_{Im_M} denotes the real and imaginary part of that complex variable as measured by the PMU.

Another important metric to characterize a PMU's performance is FE, which can be computed by the following expression

$$FE = |f_{Measurement} - f_{Actual}| \tag{3.5}$$

3.4.1 Steady State Test

Ideally, the PMU under test should pass all the compliance tests. However, in reality, it has been reported that, in most cases (e.g. in [42], [45], [50]) PMUs fail to pass some of the

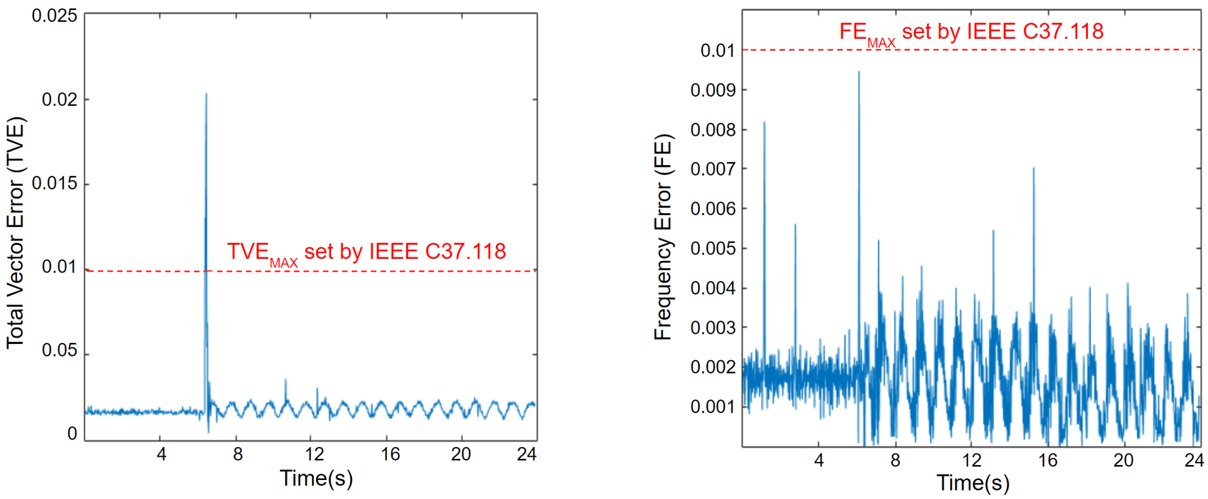


Figure 3.7: PMU Compliance Test Results for Magnitude (0.1 pu) and Phase (10^o) Step-Change: (Left) Total Vector Error (TVE), (Right) Frequency Error (FE)

dynamic compliance tests. The steady state compliance tests are successful for all PMU-s. In this section, the steady state compliance tests are first reported followed by, the results from dynamic performance testing.

Figure 3.6 shows the Total Vector Error (TVE) and Frequency Error (FE) under steady state operation of the PMU under test. Since, this PMU is based on National Instruments' *Advanced PMU Development System*, it is expected to work satisfactorily at least in steady state. Figure 3.6 also shows that the performance of the PMU is well within the compliance limits expected in steady state. In fact, the errors were observed to be within half of the maximum permissible error limits at all times.

3.4.2 Step Change in Magnitude and Phase

The experiment for step change, was performed by applying 10^o step change in phase, and a 0.1 pu change in magnitude. The changes were applied in all the phases at the same side from the GUI. The data reporting rate was set at 50 samples per second. It can be clearly investigated from Figure 3.7, that the quality of performance of the PMU deteriorates when compared to the steady state tests. However, the error of the TVE and FE are still within the specified range of IEEE C37.118 standard.

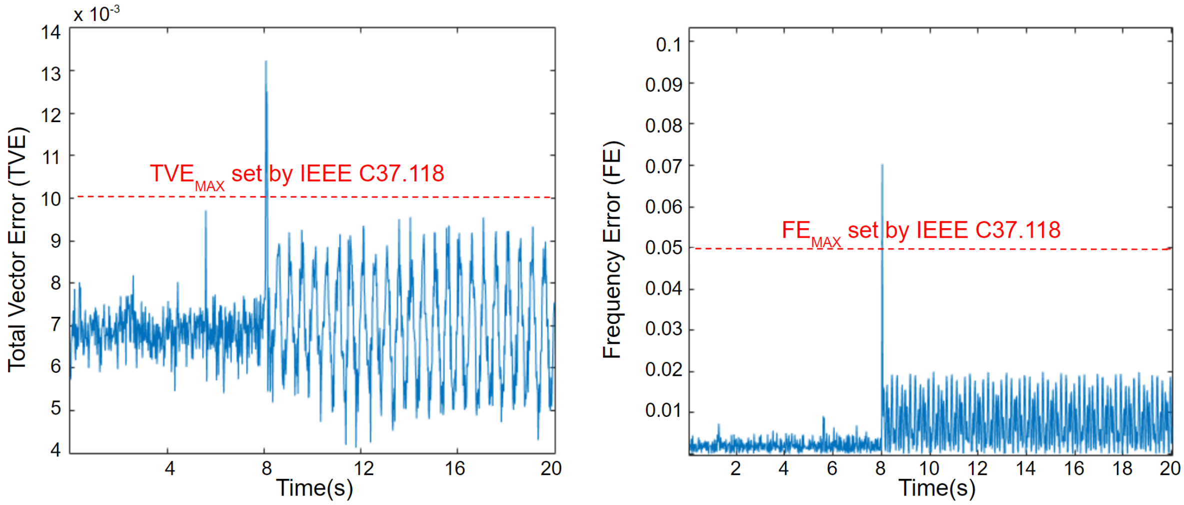


Figure 3.8: PMU Compliance Test Results for Voltage and Frequency Modulation Test: (Left) Total Vector Error (TVE), (Right) Frequency Error (FE)

3.4.3 Bandwidth Test

For the bandwidth test, the modulation factor for both the phase angle and the magnitude are chosen to be 0.1. Figure 3.8 shows that there are very few samples during the beginning of the modulation, where the PMU performance is not sufficient to pass the compliance tests. However, the overall performance during the modulations seems satisfactory. In fact, during some test runs, the PMU under test manages to perform satisfactorily. The root cause of for this behavior merits further investigation.

3.4.4 Frequency Ramp Test

As shown in Figure 3.9, the PMU under test failed in the frequency ramp test. A 1 Hz/sec frequency ramp was applied to the signal. The range was from 48-55 Hz. The observed FE was more than 0.2 Hz, which is way above the certified limit (0.05) of Frequency Error set by IEEE C37.118 standard.

However, it must be noted that frequency ramp test is the most challenging test from the point of view of the PMU. Most of the literature existing in this domain reported a failed

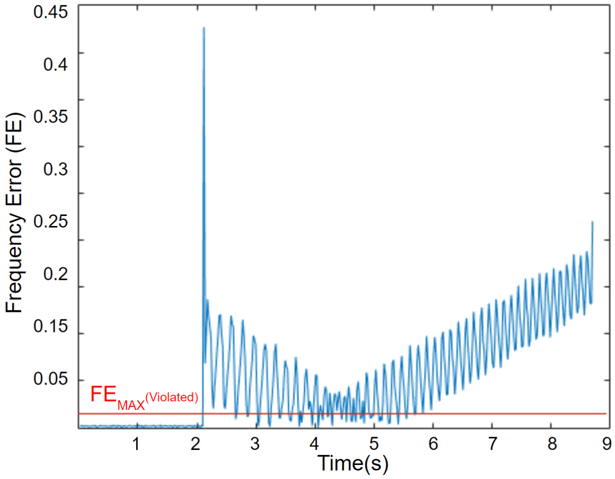


Figure 3.9: Frequency Error (FE) of the PMU Under Test During Frequency Ramp Test

frequency test.

Table 3.2 shows that this is not uncommon. In fact it can be further seen that, the PMU models in [42] and [45] also perform in a similar range as the PMU prototype tested in the experiments in this paper.

3.5 Conclusions

The experimental results in this chapter show that the PMU under test performs satisfactorily in steady state condition, and performs reasonably well under all dynamic conditions except the frequency ramp test. Even though the frequency ramp test failed for the PMU, it is clearly a limitation in the PMU technology and product of the test-infrastructure. The proposed contribution of the current work was to provide a user friendly test-infrastructure, which would be able to evaluate the performance quality of a PMU under test. The presented experimental results clearly show that the test-hardware for testing PMU functionalities can be used for PMU pre-compliance testing with confidence. One of the advantages of the testing infrastructure proposed is that it is completely configurable from the software side. This makes it a very attractive solution for testing PMUs in laboratory environment during their development process.

Table 3.2: PMU FE Performances During Frequency Ramp Test Available in the Literature

PMU Under test	Testing Hardware	Test Verdict	Max Error
PMU A as in [50]	Freja 300 Relay Set	Fail	0.015
PMU B as in [50]	Freja 300 Relay Set	Fail	0.006
PMU C as in [50]	Freja 300 Relay Set	Fail	0.022
PMU from [42]	Waveform Files	Fail	0.153
PMU Prototype on cRIO based on NI	PMU Testing Hardware proposed in this paper	Fail	0.22
PMU A as in [45]	Doble 6150 Relay Set Signals	Fail	0.35
PMU B as in [45]	Doble 6150 Relay Set Signals	Fail	0.44
PMU C as in [45]	Doble 6150 Relay Set Signals	Fail	0.08

CHAPTER 4
REAL TIME SIMULATION OF POWER SYSTEM COMPONENTS

Real-time simulators are essential tools for the research of new cyber physical power system applications, specially in the development and testing of smart grid functionalities. Like in many other fields of research, one of the major challenges when using simulation is that of reproducibility of results. This problem becomes even more complex for the case of real time simulation when different simulation targets (i.e. hardware platforms) are considered. In this chapter, comparisons between two simulators with different hardware architectures are presented. The targets compared are Opal-RT real time simulators with Intel Xeon quad-core processors and Kintex 7 FPGAs, and the Typhoon HIL 603 simulator with a combination of ARM R-class processors and Xilinx Virtex 6 FPGAs. For this comparison, standard power system components were implemented in these two environments, and simulation results from test models were compared. This chapter reports the discrepancies and similarities between the results obtained using the two platforms, and proposes a practical approach to reduce the differences found between results from both platforms. The work herein discusses on the need for the use of open standards for model exchange suitable for real time simulation.

4.1 Introduction

4.1.1 Motivation

The increase in sensor-data and the need of networking of Information and Communication Technologies (ICT) to support 'smart grid' functionalities is transforming power grids into Cyber Physical Systems (CPS). In conventional power system modeling, the grid is represented as a time varying continuous system, updating its state continuously according to *physical laws*; while ICT systems follow *rules of algorithms* that are modeled with different formalisms, such as discrete equation systems, state machines, etc. The major challenge in

• Portions of this chapter previously appeared as: B. Azimian, P. M. Adhikari, L. Vanfretti, and H. Hooshyar, "Cross-platform comparison of standard power system components used in real time simulation," *7th Workshop on Model. & Simul. Cyber-Physical Energy Syst.*, Montreal, QC, Canada, Apr. 2019, pp. 1-6, doi: 10.1109/MSCPES.2019.8738789.

the integration of models of ICT systems with those of the power grid, typically addressed through co-simulation, as illustrated in [51] and [52]. A complementary approach is the use of real time Hardware-In-the-Loop (HIL) simulation, where real-time simulators are used to model the behavior of an actual power system so to test 'real world' devices or an entire ICT system in HIL configuration. While the HIL approach is difficult to scale, it provides an essential tool for development, testing, and validation of 'smart grid' functionalities. In such case, the development of one of such functionalities is intrinsically dependent on a *specific* real time simulator, which opens the question: *will the functionality perform consistently if tested with a different real time simulator?*

In typical digital real time simulators used for simulating power grid, the simulation executes in discrete time with a fixed step, while time moves forward an equal amount in 'wall-clock'. In order to achieve this, the simulator needs to solve the model equations for that fixed time-step, acquire external inputs, and send outputs to the hardware-in-the-loop within the same amount of time as in real life. Thus, the performance of the simulator depends on many factors including the hardware architecture, numerical technique used, and solver parameters (e.g. step size), to name a few. Studying the effect of different numerical techniques and step size can be carried out using a single real time simulator hardware, examples of such experiments are carried out in [53] and [54]. This chapter, instead, apprises simulation results from two different hardware platforms and proposes possible design modifications on the model software side to obtain similar simulation results across these two different hardware architectures.

4.1.2 Related Works

Cross-platform comparisons have been carried out in other domains of electrical engineering. In the communication and networking domains, an extensive study comparing various network simulators was reported in [55], where the authors compared the performance of various network simulators including ns-2, ns-3, OMNet++, SimPy, and JiST. In the area of Magnetics, a similar survey was reported in [56], where the Rhombic Wire Simulator and the Harvard EMP Simulator were compared and analyzed. Meanwhile, only a limited number of published studies comparing real time simulators in the domain of power engineering have been carried out [57],[58], to the best of authors' knowledge. Although there are no cross platform analysis published, substantial work has been done on both the platforms (Opal-RT and Typhoon HIL) individually, by both vendors and users .

4.1.3 Highlights of the Chapter

- The current work provides a comparative analysis of how the two different real time simulators perform while simulating similar models and analysis scenarios. For an extensive comparison, basic power system component models (blocks) like synchronous machines, transformers, governors, induction machines and simple loads are considered in this work. All these blocks were implemented simultaneously in both Opal-RT 5030/4520 and Typhoon HIL 603. The blocks are subjected to identical simulation scenarios and their responses were recorded and compared.
- Having verified the individual component models, they were used to construct identical power systems in the two different platforms and the response of those systems were recorded and compared.
- Additionally, some possible measures are proposed to minimize mismatches observed between the simulation results of the two simulators.

4.2 Review of the Simulation Platforms

In this section the basic architecture of the two representative real time simulation platforms are reviewed

4.2.1 Opal-RT 5030/4520 Real Time Simulator System

Opal-RT real-time simulator consists of 32 cores intel Xeon quad-core series processor (housed in OP-5030/4520) and an array of 7 series Kintex FPGAs providing computational and I/O capabilities (housed in OP-4520), respectively. Connectivity is supported with high speed fiber optical cables with a maximum transmission capability of 5MBit/sec. The schematics are designed using standard MATLAB/Simulink libraries and additional simulink libraries are provided by Opal-RT that enable model compilation and execution on the CPU cores and the FPGAs inside the OPAL hardware. In principle it is possible to simulate systems with a minimum time step of 250 ns in this architecture.

There are studies comparing real time simulation tools with offline simulation tools like PSSE and RTDS [59], however, this is out of the scope and interest of this research.

Table 4.1: Internal Parameters for Power System Components

(a) Synchronous Generator Parameters

D Axis Reactances	$X_d = 2.2$ $X'_d = 0.3$ $X''_d = 0.2$
Q Axis Reactances	$X_q = 2$ $X'_q = 0.4$ $X''_q = 0.2$
Stator Resistance	$R_s = 0.003$
Inertia Coefficient	$H = 2$
Leakage Reactance	$X_l = 0.15$

(b) Induction Machine Parameters

Stator Side Resistance	$R_s = 0.029\Omega$
Rotor Side Inductance	$R_r = 0.029\Omega$
Stator Side Inductance	$L_s = 0.39965mH$
Rotor Side Inductance	$L_r = 0.39965mH$
Mutual Inductance	$L_m = 0.0346H$

4.2.2 Typhoon HIL 603 Real Time Simulator

Typhoon HIL 603 Real Time Simulator consists both Xilinx 6 series Virtex FPGAs and ARM R-class Processors. The communication links are standard ethernet, and this system is capable of simulating systems reliably with a minimum time step of 1us.

4.3 Test Cases

In order to compare the two different real time simulators, different basic blocks used for power system simulation were considered. In this chapter, simple system models are designed to test an individual component, including synchronous generator, transformer, governor, induction machines and loads. These 'unit-test' models were implemented in the two different platforms and their performances are compared. This part of the work is hereinafter referred to as **unit-testing**. After the successful unit testing, two power system models consisting of those tested units were simulated in the two platforms. These power system models are referred to hereinafter as **Microsystem** and **Microgrid**.

4.3.1 Components for Unit Testing

4.3.1.1 Unit 1: Synchronous Generator

A 3 phase, 50MW, 20kV synchronous generator was implemented in both Opal-RT/ Simulink and Typhoon HIL Schematic. The parameters of that generator are given in table 4.1a below.

4.3.1.2 Unit 2: Transformer

A 3 phase 100 MVA Y-Y 20kV-230kV transformer with both primary and secondary resistance of 0.002 pu and inductance of 0.08 pu was tested in both the real time simulation platforms.

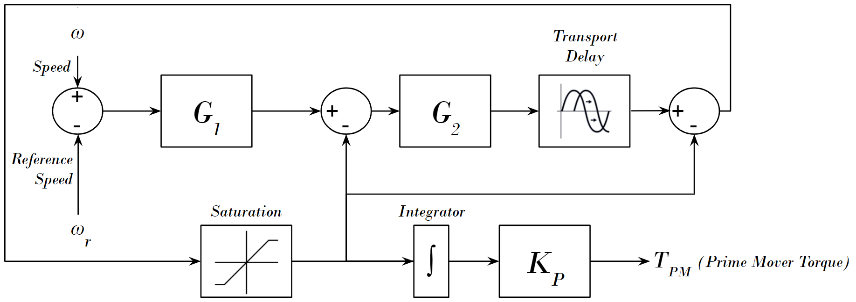


Figure 4.1: Simplified Governor Model Implemented Using Typhoon HIL Schematic Editor

4.3.1.3 Unit 3: Induction Motor

A 3 phase 480V (line to line), 200 kW Induction motor was considered. The parameters of this induction motor are given in Table 4.1b.

4.3.1.4 Unit 4: Governor

For this block, the default single mass Tandem compound turbine governor system in the Simulink library was considered. To reduce complexity, all the turbine time constants except T_1 were set to zero, and the turbine torque fractions F_3 to F_5 were set to zero. The speed-and-govern system consists of a proportional regulator, speed relay and a servo motor controlling the gate opening. The associated turbine has a maximum of four stages, while each of them can be individually modeled as a first order transfer function. The detailed modeling of this block can be found in [60]. The simplified model that was used in this work is shown in Figure 4.1.

4.3.1.5 Unit 5: Load

A simple constant impedance balanced 3 phase star connected load was implemented in both the simulators . Each phase has a resistance of 1 Ω and inductance of 1 mH.

4.3.2 Description of the Microsystem

The microsystem consists of a few of the unit-test models that were already implemented and tested in 4.3.1. It is shown in Figure 4.2, that a synchronous generator (Unit 1) is followed by a step up transformer (Unit 2). The HV side of the transformer is floating, while the generator is operating at rated conditions. At time t, a star connected balanced load (Unit 5) is switched on across the HV side of the transformer. The generator is speed-controlled by the simple governor, that was modeled as described in the previous section. The transient behavior of the system during the switching of the load is analysed in the sequel.

4.3.3 Description of the Microgrid

This models consists of an 18 bus distribution network as shown in Figure 4.3. The induction motor used in this system is parameterized as the one used in unit-testing and is connected at bus 13.

4.4 Simulation Results

It is to be noted, that for all the unit-tests and the microsystem, the authors simulated the Opal-RT models (which are also used in Opal-RT) as the reference and tried to recreate those simulation results in the Typhoon HIL platform. For the microgrid example, however, the Typhoon HIL model was taken as a reference, and an identical system was modelled in Opal-RT to recreate the simulation results obtained from Typhoon HIL. The simulation specifications are mentioned in table 4.2 below. In order to compare both simu-

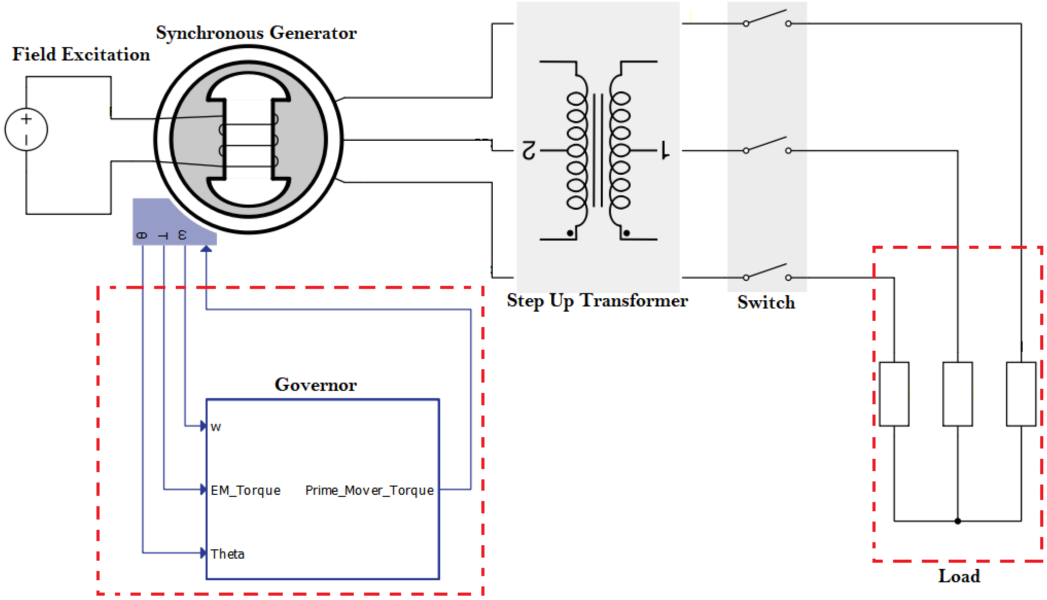


Figure 4.2: Schematic of the Micro-system Developed in the Typhoon HIL Schematic Environment

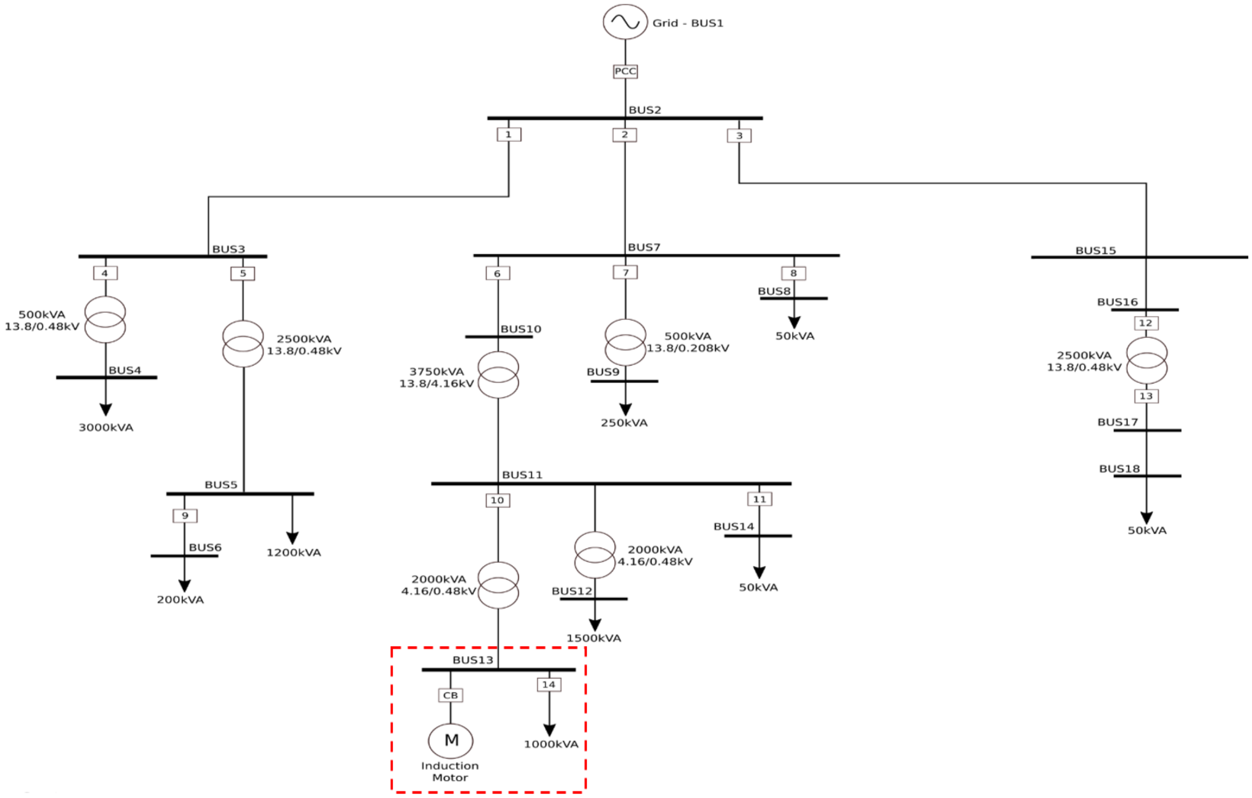


Figure 4.3: Schematic of the Microgrid Developed in Typhoon HIL Schematic Environment (the Red Box Outlines the Induction Motor Model Described in Section 4.3.1)

Table 4.2: Simulation Settings

RT Simulator	Typhoon HIL	Opal-RT
Solver Simulation Step Size	Trapezoidal 2 us	Trapezoidal 2 us

lators fairly, the same discritization methods and simulation step sizes are chosen across the two simulators.

4.4.1 Unit Tests: Simulation Results

- Load: The load as described before, was connected to a fixed balanced 60 Hz, 480V three phase supply voltage, and the active and reactive powers consumed by the load is observed in the two different platforms, for different step-sizes. Observe that the two platforms present a small deviation from the theoretical results.

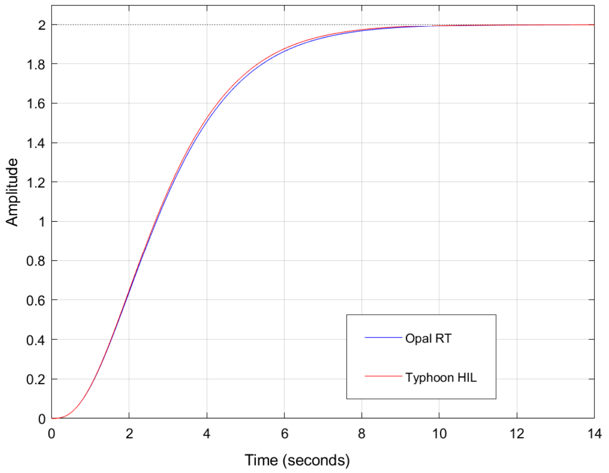


Figure 4.4: Step Response of the Modeled Governor

- Governor: The simplified mass-tandem governor is modeled in Typhoon HIL and its step response is compared to that with the step response of the model implemented in Opal-RT . Figure 4.4 shows that the responses match with negligible error and hence, the mass-tandem governor implementations are homogeneous in their responses.

- Transformer: The modeled transformer is subjected to its rated primary side voltage and the secondary side voltage is recorded for both simulation platforms. Figure 4.5 shows that the secondary voltage of the transformer model is the same in the two platforms.

- Induction Motor: The transient response for the induction motor is plotted in Figure 4.6. Both Opal-RT and Typhoon HIL results are shown in the same plot for comparison. It can be seen that the OPAL-RT does not undergo a oscillation as shown by the Typhoon model. In addition, the initial overshoot for OPAL-RT is lower. A

Table 4.3: Simulation Results for Model Load

Step	Typhoon HIL		Opal-RT		Theoretical	
	P(W)	Q(VAr)	P(W)	Q(VAr)	P(W)	Q(VAr)
4us	201771	76087	201686	76012	201729	76050
2us	201749	76068	201708	76031	201729	76050
1us	201739	76059	201718	76040	201729	76050

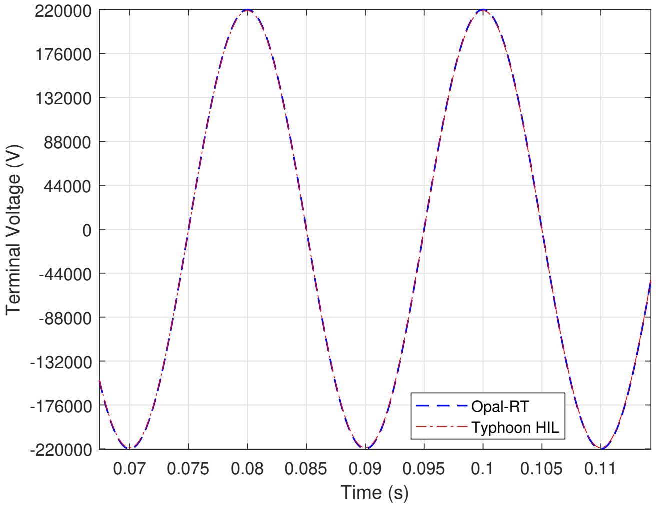


Figure 4.5: Response of the Transformer Model While Subjected to Rated Input Voltage

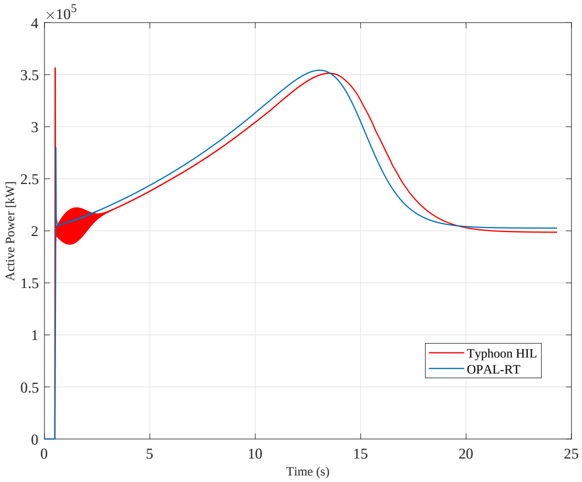


Figure 4.6: Transient Response of the Motor

subtle difference can be seen in transient response as well as the steady state value of active power consumption of the motor. These observations, however, are consistent with the results seen in the cross platform transient simulation of Microsystem in the next section. A corrective measure is proposed here in order to adjust the parameters in the OPAL-RT model in order to minimize the differences between both simulation platforms. The “parameter estimation tool” (PET) in MATLAB/Simulink is used. The motor behavior in Typhoon can be captured and imported to PET as a reference signal. PET adjusts and optimizes the motor parameters of Opal-RT model to produce a similar response to the reference signal from Typhoon. The parameter calibration results are shown in Figure 4.7. From this analysis it can be concluded that modifying the snubber resistance helps reducing mismatches between the two real-time simulators.

4.4.2 Simulation Results for the Microsystem

A load switching scenario is now used to assess the overall power system model assembly and simulation results. The ratings of the transformer, and the synchronous generator selected are those mentioned in the section 4.3. The load in section 4.3 is parameterized to represent a simpler three phase resistive load, so as to only consume active power. In Figure 4.4, at time t= 6s, a balanced resistive load of 35 MW is switched on. The resulting output of the synchronous generator is expected to decrease, as there is no feedback controller to maintain the speed in this simulation scenario. With these results, the outputs from both the platforms were investigated. Both simulators provided almost identical steady state response, but the transient responses were different. This is shown in Figure 4.8.

It is to be noted that the hardware architecture of Typhoon HIL is significantly different from that of the Opal-RT environment. The signal processing, measurement and monitoring blocks of the Typhoon HIL schematic runs with a time step of T_s, where as the FPGA based circuit solver runs on the FPGA cores with a step size of T_{exec}. In order to run the entire

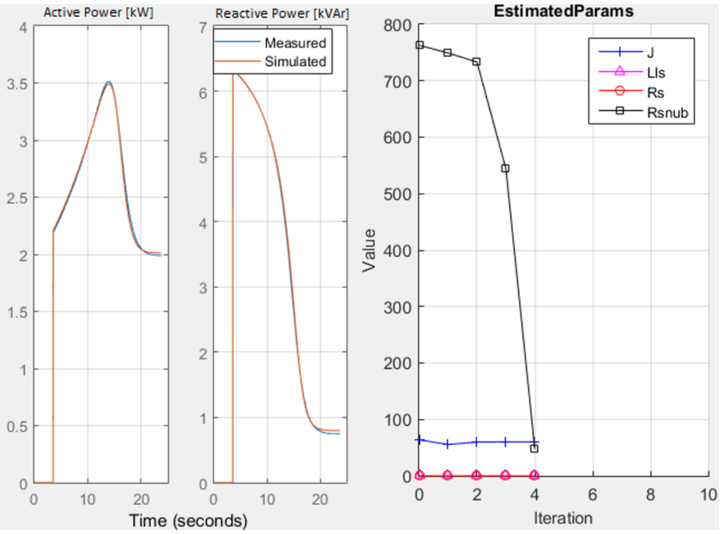


Figure 4.7: PET Parameter Estimation Results for J=Motor Intertia, L_{ls}=Stator Inductance, R_s=Stator Resistance, R_{snub}=Snubber Resistance

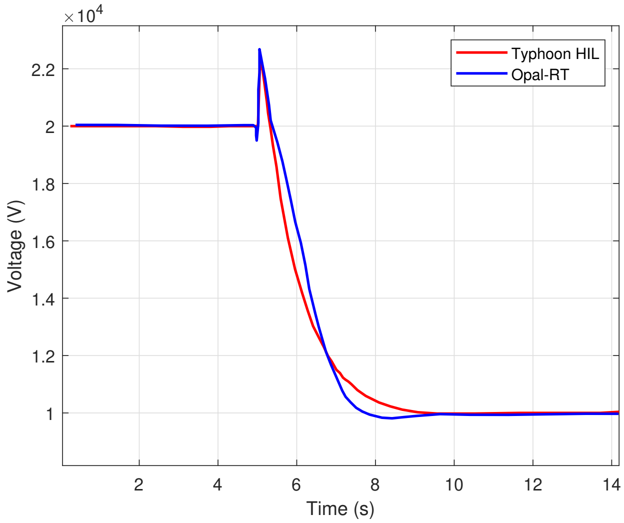


Figure 4.8: Transient Responses of the Generator in the Micro-System

system with a universal step size, multiple rate transition blocks were introduced in the schematic. Even after that, the transient responses of the two simulation results remained different.

4.4.3 Simulation Results for the Microgrid

Table 4.4: Results From Individual Buses of the Microgrid

	P(MW)	Q (MVAR)	P (MW)	Q (MVAR)
Injected Power	4.16	1.79	4.13	1.82
Bus Number	Voltage	Angle	Voltage	Angle
5	474.5	-30.42	473.87	-30.67
8	13753.13	-0.12	13753.14	-0.91
12	465.84	-61.75	466.12	-61.78
16	13793.07	-0.02	13785.16	0.02
	Opal-RT		Typhoon HIL	

Steady state analysis was carried out to investigate the differences for this system. The total active and reactive power injected from the main grid; voltages and angles at some buses are compared based on the “equilibrium” solution (i.e. when the simulation reaches a new steady state). These results are presented in table 4.4. Both platforms have to decouple the distribution network into sub circuits to be able to solve the differential equations of this system. In Typhoon HIL transformer-based decoupling (called coupling core) is used while in some cases small snubbers are used to improve numerical stability. OPAL-RT uses the state-space nodal solver (SSN) method for decoupling. The SSN method avoids numerical problems and gives more realistic results without the need of snubbers. [61] Five coupling cores and SSN blocks are used in Opal-RT, to divide the whole system into 6 sub circuits in both platforms. To match the behavior of the Typhoon model, snubbers are added to the model in Opal-RT based on similar analysis (PET) as explained for the induction machine earlier.

4.5 Discussions

Reproducibility of real time simulations across different platforms, using disparate software and hardware architectures, is a major challenge as illustrated in the experiments above. In these studies the major issues faced by the authors were:

- Modelling Coverage: It was observed that, certain Opal-RT blocks (e.g. governor) were missing in the Typhoon HIL Schematic library. Hence the user is required to create custom models in the Typhoon HIL environment which will be able to replicate the performance of those in other environments.

- Indiscriminate Simulation: Typhoon HIL provides the option to simulate using a solver called 'exact' method. On the other hand Opal-RT has multiple available methods to simulate a system, while none of them are as accurate as the 'exact' settings in Typhoon. The only option for fair comparison was to use the less accurate 'trapezoidal' methods in both platforms.

- Architecture Differences: The architecture of the Typhoon HIL uses two very different step sizes for simulation (T_s and T_{exec}). This requires to use additional rate transition blocks in Opal-RT. In practice, this makes it impossible to create two identical simulation conditions in the two different simulators.

4.6 Conclusions

This work is the first step towards 'homogeneous' modeling of power system components across different real-time platforms. The experiments reported in this work show how model portability and model exchange standards being adopted in other engineering fields, such as the Modelica language [62] and the Functional Mock-Up Interface [63], would be desirable for power systems simulation in general [64], but even more in real-time hardware-in-the-loop simulation to minimize modeling uncertainties. Currently, there are only two power system domain-specific tools that support the FMI standards, namely, Opal-RT's ePhasor-Sim [65] (only providing support for positive sequence models as a 'master' co-simulator solver) and EMTP-RV [66]. As described in [63] model exchange faces tremendous challenges in conventional power systems tools used in off-line analysis, and thus, this chapter aims to build the body of evidence documenting challenges with model portability for real time simulation where Modelica and the FMI standards can play a major role.

Because, only one type of solver was tested. with a single reasonable step size in all the experiments, it calls for a future exploration of the performance comparison with varying step size, and with different solvers. It is also noted that, in the presented experiments, none of the RT models are interacting with any physical hardware from the external world. Experiments with RT models interacting with real life external hardware are essential for further advancements in the area of cross-platform homogeneous modeling of power system components.

CHAPTER 5
ADVANCED COMPONENT MODELING: PHOTOVOLTAIC CELLS AND ARRAYS

The previous chapter illustrated the implementation of fundamental components for power system simulation, across different real-time simulators. This chapter, explores such cross-platform implementation for a more complicated power system component- PV array.

The increase of solar power generation in the last decade makes reliance on this energy source of great importance, and highlights the need of studying photo-voltaic (PV) cells and modeling their behaviors. Consequently, efficient and accurate mathematical modeling and simulation of PV cells have become very important for different studies, e.g. their use in microgrids and as distributed energy resources in the grid. This chapter reports a custom, homogeneous, cross-platform, real-time simulation model for PV cells, modules & arrays; and characterizes their performances when compared to other existing models. Once the model was successfully verified, its performance was analyzed in two different real-time hardware architectures: (i) OPAL RT OP5030/4520 and (ii) Typhoon HIL 603.

5.1 Introduction

5.1.1 Motivation

Solar power generation is gaining relevance every year as one of the fastest growing sources of renewable energy. Solar (photo-voltaic) power generation is environment friendly, cost-efficient and ideal for distributed generation technologies. Photo-voltaic (PV) power generation is dependent on PV cells, which convert the radiant energy received from sunlight into electrical energy. However, an individual PV cell is too small for practical applications, which leads to the concept of PV modules and PV arrays. Multiple PV cells are connected in series or parallel connections to create a PV module, which meets required output voltage and current specifications. Multiple PV modules are then combined in parallel or series to make a PV array. The behavior of PV arrays or PV modules are highly non-linear in

- Portions of this chapter previously appeared as: X. Jia, P. M. Adhikari, and L. Vanfretti, “Real-time simulation models for photovoltaic cells and arrays in Opal-RT and Typhoon-HIL,” in *IEEE Power & Energy Soc. General Meeting*, Montreal, QC, Canada, Aug. 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9282171.

nature, and require analysis in order to derive a viable mathematical model which can be used for simulations. These mathematical models are also useful to design, manufacture and test power-converters which will be used to convert the DC power obtained from PV arrays into AC power. From the utility stand point, modelling of PV arrays are of paramount importance because they are used to study and assess power system planning and operating impacts in any system containing PV generation.

The time-critical nature of electrical power systems, calls for simulation models which are not only accurate but also efficient for real-time simulation. Real-time simulations and Hardware-in-the-Loop simulations are critically important when testing control and protection systems related to PV systems. However, a major challenge arises when dealing with different real-time simulators that have different hardware architectures and different modeling libraries. Hence, any proposed model for PV arrays, needs to be tested on different real-time simulators to ensure its cross-platform compatibility and result homogeneity.

5.1.2 Related Works

Authors of [67] introduced and discussed the single-diode equivalent circuit model to represent a PV cell. That model had been modified and enhanced in [68]-[71]. Authors in [68], [69], [71] reported an accurate methodology to determine the parameters of the single-diode equivalent circuit of a PV cell. In [70], the single-diode model was extended to be used to represent PV panels made with amorphous silicon technology. While other, more complicated models, had been proposed by different authors, the simple representation of the single-diode model is still appealing and sufficiently accurate as reported in [72].

The simplicity of the single-diode representation makes it suitable for real-time simulation. Authors in [73] incorporated the single-diode PV model coupled with a fully functional inverter and controller into a real-time power-system model, along with detailed results for EMT-like real-time simulations using the Opal-RT hardware. In addition MathWorks has a very detailed PV model [74] that uses an improved version of single-diode representation of PV cells. This model is suitable for off-line simulations and was used as a benchmark for the work in this part of the work. Note that, this benchmark model cannot be modified, it is a “black-box” with limited documentation and limited access. The proposed model in this chapter aims to address this limitation, so to facilitate real-time simulation.

Because, reproducibility of results is a critical issue in power systems research, it was important to extend this study to other real-time simulation hardwares apart from Opal-RT. In this work, the Typhoon HIL platform provided an excellent opportunity for this study.This chapter reports that, even though the hardware architectures of these two real-time simulators are significantly different, their simulation results can be made similar if adequate care is taken during model implementation. Hence, to test that hypothesis further, the proposed PV model in this chapter was implemented in both Opal RT (OP 5030/4520) and Typhoon HIL (HIL 603) real-time simulators separately and their performances were analyzed.

5.1.3 Highlights of the Chapter

- A simple model is proposed to represent a PV system in real-time simulation environments. The accuracy and performance of this model (in Opal-RT real-time simulator) is compared with the PV model from *MathWorks*.

- The proposed model was re-implemented and migrated to Typhoon HIL 603 hardware-infrastructure. The real-time simulation results in this new platform are compared with those obtained using the Opal-RT real-time simulators.

- Comparisons between the proposed model and the existing model from the MathWorks in terms of hardware usage and efficiency are presented.

- The models are made available as open source software in the GitHub repository : `github.com/alsetlab`

5.2 Theory

5.2.1 Review of the Real-time Hardware Platforms

The *Opal-RT Real Time Simulator* was used to simulate the proposed PV system in real-time. The hardware consists of 32 cores of *Intel* Xeon quad-core series processors and a 7 series *Xilix* Kintex FPGA which provides I/O capabilities. The high speed optical fibre cables are able to support upto a speed of 5 MBit/sec. The standard Simulink (along with some additional libraries provided by Opal-RT) environment were used to design the proposed PV system model.

The *Typhoon HIL 603 Real Time Simulator* has Xilinx's Virtex 6 series FPGAs and ARM R-class Processors. The communication links used are standard Ethernet, and this hardware is capable of reliable simulations upto a minimum time-step of 1us. Typhoon has its own HIL Schematic model design environment which was used to design the PV system shown in Figure 5.2.

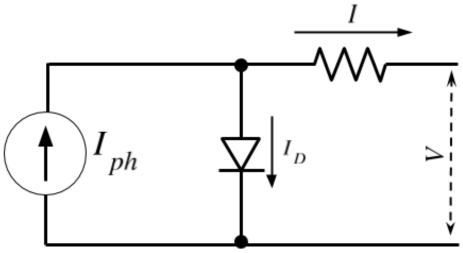


Figure 5.1: Single Diode Model to Represent a PV Cell

5.2.2 Description of the Proposed Model

The PV array model is constructed by series-parallel connection of multiple PV modules and each of these PV modules contains the same number of PV cells. A single PV cell is represented by the single-diode model shown in Figure 5.1.

In this model, I_{ph} is the photoelectric current, I_D is the diode current and I_0 is the dark current. Additionally, the series resistance R_s is used to model the losses due to poor conductivity in the solar cell, and V_t represents the thermal voltage. Using the method described in [73], if a short circuit is applied, then the resultant photoelectric current I_{ph}, is defined as the short circuit current I_{sc}.

If an input voltage V is applied to the PV cell, the output current I is given by

$$I = I_{ph} - I_D = I_{ph} - I_0 \times e^{\frac{V+IR_s}{V_t}} \tag{5.1}$$

Hence, the parameters of interest are I_{sc}, I_0, V, V_t and R_s. While these variables are known, it is possible to compute the output current of a single PV cell, and together with with the knowledge of the series parallel configuration of the PV array, they can be used to compute the output current of the system, as shown in eqn (5.1).

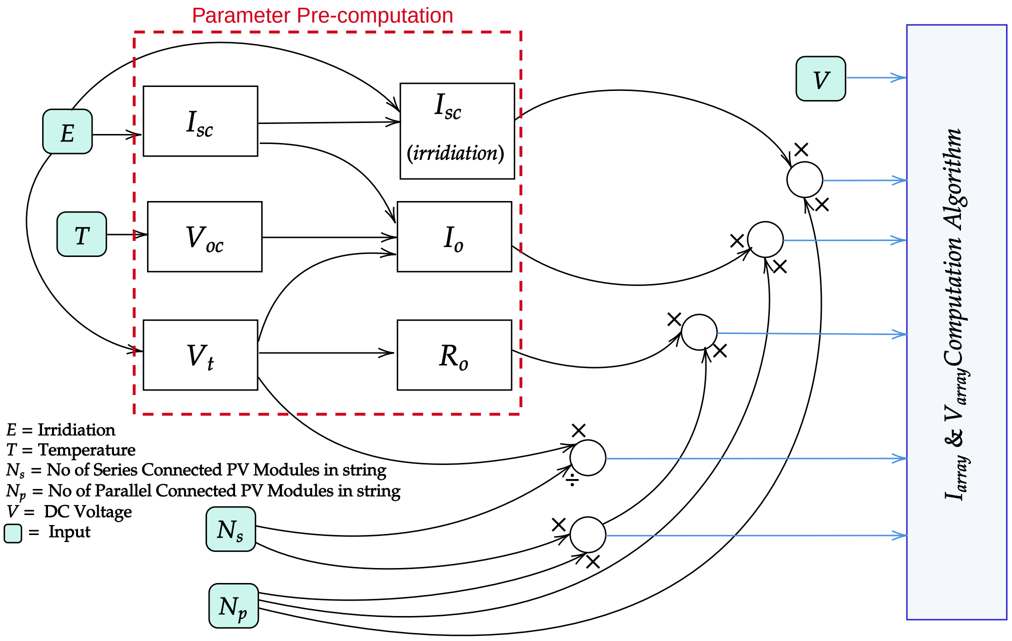

Parameter Pre-computation
E
T
I_{sc}
V_{oc}
V_t
I_{sc} (irridiation)
I_o
R_o
V
N_s
N_p
I_{array} & V_{array} Computation Algorithm
×
÷
E = Irridiation
T = Temperature
N_s = No of Series Connected PV Modules in string
N_p = No of Parallel Connected PV Modules in string
V = DC Voltage
▢ = Input

Figure 5.2: Block-Diagram Containing Different Components of the Model of the PV Cell

Algorithm 1 Calculation of I_{array} and V_{array}

```
Require: $n \geq 0 \vee x \neq 0$
Require: $I_{sc}, V_t, N_s, R_s, I_o, V_{DC}$
Require: Define:
    $f(I_n) = I_n - I_{sc} + I_o \times ((exp(V_{DC}/N_s + I_n \times R_s)/V_t) - 1)$
Ensure: $I_{start} = 0, e = 10^{-3}, L = 0$
  while $L \neq 1$ do
    if $n = 0$ then
      $I_n = I_{start}$
    else
      $n++$
      $I_n = I_{n-1} - f(I_{n-1})/f'(I_{n-1})$
      $e_n = I_n - I_{n-1}$
      if $\mid e_n \mid \leq e$ then
        $I_{array} \leftarrow I_n$
        $L = 1$
      end if
    end if
    $V_{array} \leftarrow V_{DC}/N_s$
  end while
```

Once, the output current of the PV cell is computed, and based on the series-parallel configuration the PV system, the output current of the entire PV array can be calculated as detailed in Algorithm 1. The variable e is introduced inside the algorithm to ensure that the solution has converged.

5.2.2.1 Thermal Voltage (V_t) Computation

To compute the magnitude of thermal voltage V_t of the PV array, the first step is to calculate the magnitude of the thermal voltage of a single PV cell. The mathematical expression to derive the thermal voltage of a single PV cell is given by:

$$V_{t_{cell}} = \frac{AkT}{q} \tag{5.2}$$

where, A is the diode quality factor, k is the Boltzmann's constant, q is the charge of an electron and T is the temperature in Kelvin. For simplicity, the diode quality factor is often assumed to be 1. Additionally, it is assumed that there are N_{cell} of PV cells in each PV module. The output thermal voltage V_t of PV array is given by:

$$V_t = V_{t_{module}} \times N_s = V_{t_{cell}} \times N_{cell} \times N_s \tag{5.3}$$

where N_s is number of series-connected PV module in each string.

5.2.2.2 Series Resistance (R_s) Computation

The series resistance for individual PV modules can be mathematically formulated as:

$$R_{s_{module}} = \frac{V_{t_{module}} \ln(1 - \frac{I_{mpp}}{I_{sc_{STC}}}) + V_{OC_{STC}} - V_{mpp}}{I_{mpp}} \tag{5.4}$$

where $I_{sc_{STC}}$ and $V_{oc_{STC}}$ are the short circuit current and open circuit voltage under standard time condition (STC), I_{mpp} and V_{mpp} are the current and voltage at maximum power point, respectively. The series resistance of the PV array is further computed from here, according to the following relationship where N_p is the number of parallel connected PV module strings:

$$R_s = R_{s_{module}} \times \frac{N_s}{N_p} \tag{5.5}$$

5.2.2.3 Open-circuit Voltage (V_{oc}) Computation

This subsystem computes the effect of temperature on the open circuit voltage of the PV module. The procedure is described in [74]. It estimates the open circuit voltage V_{oc} at temperature T when the open circuit voltage at STC ($V_{OC_{STC}}$) is known. The mathematical relationship that estimates the opne circuit voltage is given by

$$V_{oc_T} = V_{oc_{STC}} + V_{oc_{STC}} \times \beta_{V_{oc}}(T - 25) \tag{5.6}$$

where T is the input operating temperature for the PV array.

5.2.2.4 Short-circuit Current (I_{sc}) Computation

This subsystem computes the effect of temperature on the short circuit current of the PV module. The procedure is described in [74]. It estimates the open circuit voltage I_{sc} at temperature T when the open circuit voltage at STC ($I_{SC_{STC}}$) is known. The mathematical relationship that estimates the open circuit voltage is given by

$$I_{sc_T} = I_{sc_{STC}} + I_{oc_{STC}} \times \alpha_{I_{sc}}(T - 25) \tag{5.7}$$

I_{sc_T} is further modified to take into account the effect of irradiation to compute the actual short circuit current of the PV module $I_{sc_{module}}$, as follows:

$$I_{sc_{module}} = I_{sc_T} \times \frac{E}{1000} \tag{5.8}$$

Finally, the short circuit current of the PV array is finally estimated as

$$I_{sc} = I_{sc_{module}} \times N_p \tag{5.9}$$

5.2.2.5 Dark Current (I_o) Computation

The Dark Current computation as reported in [73] and is represented by the following mathematical relationship:

$$I_o = N_p \times \frac{I_{sc_{module}}}{e^{\frac{V_{oc_{STC}}}{V_{t_{module}}}}} \tag{5.10}$$

With all the computations above completed, I_{sc}, I_o, V_t and R_s are fed to a Newton-Raphson algorithm to solve eqn (5.1) , which gives the values of V_{array} and I_{array}. In the

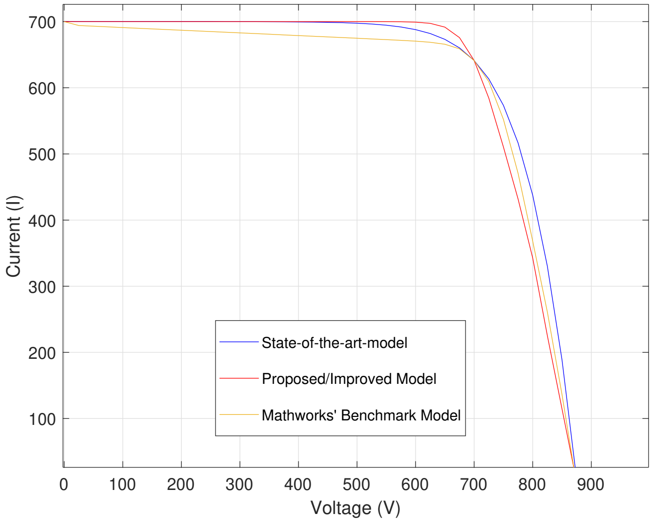


Figure 5.3: Comparison of V-I Characteristics for the Existing PV Models

Opal-RT platform it was implemented using a MATLAB script, where as in Typhoon-HIL platform the implementation was based on Typhoon's advanced C functionality support.

5.3 Results

5.3.1 Verification of the Proposed Model in Opal RT

The model reported in [74] was taken as the benchmark model, and the proposed model in the current work makes the enhancements discussed next.

The following enhancements to the model proposed in [73] are proposed in this work to make its behavior similar to that that of [74]

- The computation of the thermal voltage V_t is improved by introducing the N_{cell} variable as described in section 5.1.2. Change in thermal voltage results in a change in the dark current I_o computation as well.
- The computation of open circuit voltage V_{oc} is improved by introducing the $\beta_{V_{oc}}$ parameter.
- The computation of short circuit current I_{sc} is improved by introducing the $\alpha_{I_{sc}}$ parameter.

Figure 5.3 shows the V-I characteristics of the model in [73], the benchmark model in [74] and the modified model proposed in this chapter on the same plot. It can be seen that in terms of accuracy, the proposed model is closer to the benchmark than the model in [73] for the range of [700-900]V. It is important to note that, the proposed model is simple and contains only simple mathematical functions along with a Newton-Raphson solver, making it more hardware resource efficient than the benchmark model itself, which uses closed source library functions and blocks. This claim is justified by further observations, presented in table 5.1. It can be seen that, in terms of hardware resource consumption, the proposed PV model is more economical than the benchmark model. However, the proposed model is also, a little slower than the benchmark model from MathWorks (it takes about 15.5% more computation time), due to the use of the iterative solver. However, the increase in computation time is insignificant when considering time-steps of 5us or higher.

Table 5.1: Performance Characterization of the Proposed PV Model in Opal-RT Platform

	MathWorks' Benchmark PV Model	Proposed PV Model	Change
SSN: State Space Operation Count	110	80	-27.27% -
SSN: Memory Usage	0.003528	0.002568	-21.21%
Network Info	7	6	-14.28%
Computation Time	1.098 us	1.268 us	+15.48%

5.3.2 Implementation in the Typhoon HIL Platform

The proposed model was re-implemented for the Typhoon HIL-603 real-time simulation hardware to provide reproducibility of the results across different hardware platforms. The *Newton-Raphson* V_{array} and I_{array} calculation-block was replaced with the *Advanced-C-function* functionality (Figure 5.3) available in *Typhoon HIL schematic* environment. It can be seen in table 5.2, that the memory consumption during simulating the PV model is well

Table 5.2: Memory Utilization For Simulating the PV Model on Typhoon HIL 603 in Real-time

Memory Type	Used Memory	Available Memory (kb)	Utilization
Internal Memory	60	256	23.57%
Code Segment Size	41	256	16.33%
Data Segment Size	18	256	7.24%

within reasonable limit for microgrid applications.

Studies were performed to compare the model's response in the two different real-time hardware architectures. Figure 5.4 shows that the results in these two platforms are identical. This observation confirms that the proposed model can be successfully implemented in different real time hardware platforms and the results are reproducible if meticulous attention is paid to their implementations. A simulation time-step of 20us was used for both the simulation platforms.

5.3.3 Hardware Selection

The computational performances were also identical between HIL 603 and HIL 604 as they have very similar hardware architecture and specifications, and because the model does not put a large computational requirements. Observe that this is possible because the proposed model does not have any library dependence and utilizes simple mathematical functions only. Hence, it is fair to say that the model can run successfully in any Typhoon HIL hardware above and including HIL 603, and the computational performance will be better for Typhoon simulators with better specifications. For comparatively lower-end models, it is still expected to perform successfully, but further testing would be necessary to verify this. For Typhoon hardware the model was simulated with 20us and 10us (minimum), successfully.

In Opal-RT real-time simulator OP5030/4520, the proposed model was run successfully with a minimum step-size of 20 us. Because, the model has no library-dependency, it can be concluded that the model will (at the very least) run successfully at 20 us time-step in any Opal-RT hardware, provided that the target has similar or better processor specifications. This is particularly important as Opal-RT uses a wide array of Intel processors with different specifications, the target used in this work has 2 × Intel Xeon E5-4660V4 with 16 cores. This work only utilized 2 cores, one for monitoring and one for execution, so it is fair to assume that any target with similar processor characteristics should be capable of running the model.

5.4 Conclusions

This chapter reported a simplified PV model, improved upon previously reported single-diode models. With the simplification of the PV model, the hardware resource requirements for simulating it in real time (Opal-RT platform) were reduced. It was also investigated whether the accuracy of performance deteriorates by the proposed simplifica-

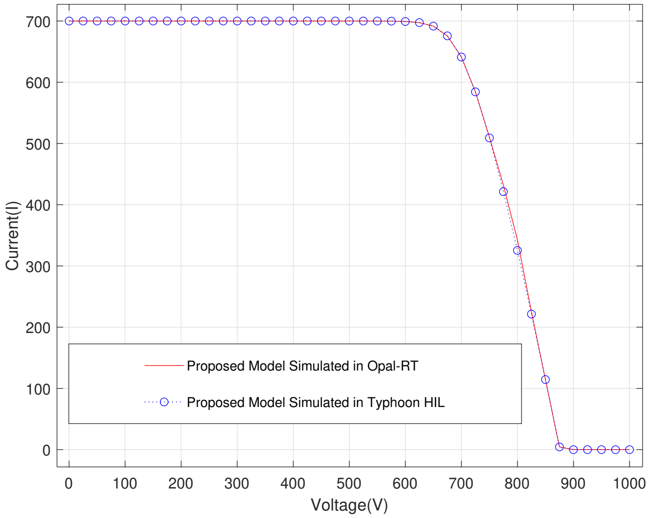


Figure 5.4: Comparison of the PV Model Responses Between Typhoon HIL and Opal-RT

tion of the model. The accuracy of the reported model is determined in comparison with the benchmark PV model from MathWorks. Once, the results were found to be satisfactory, the same model was implemented and simulated in Typhoon HIL platform. The results in Typhoon HIL were compared with those obtained in Opal RT. It was concluded that, the real-time simulation results for the proposed model was reproducible across the two different hardware architectures, thanks to the care taken to make nearly identical implementations.

5.5 Discussions and Future Works

Ensuring the consistency of model response is crucial in order to reduce the uncertainty while simulating a model across simulators of different hardware architectures. Modern model exchange interportability standards such as *Modelica* and the *FMI* (Functional Mock-up Interfaces) are already being used in some other domains of engineering and sciences. However, very few research in that regard is done in the domain of power system.

This chapter reported that the existing library models for similar components across these two different simulators can give different results. Hence, to obtain compatible simulation models for PV cells and arrays, it was necessary to build them from the scratch. Only simple operators, mathematical functions, and numerical programs (Newton-Raphson) were used. This practice is encouraged and recommended for further implementations of more complex systems across different hardware architectures, until modern model exchange standards are adopted in the power systems domain.

The current work successfully proposes a simplified model for PV cells and arrays. It is also established that, if proper care is taken, those models can be ported across two different simulator hardwares.

CHAPTER 6
EVALUATING OVERVOLTAGE MITIGATION STRATEGIES FOR PV INVERTERS AND THEIR IMPACT ON GROUND-FAULT OVERVOLTAGE THROUGH RT-CHIL EXPERIMENTS

In power systems, Single-Line-to-Ground (SLG) faults are the most common type of fault. When a three-phase four-wire system supplied by an ungrounded synchronous generator is subjected to SLG faults, the unfaulted phases are expected to exhibit significant ground-fault over-voltage (GFOV). Mitigation of this is via effective grounding, as described in IEEE Std 62.92.2. However, for inverter-based renewable energy sources (IBRs), the physical mechanism that leads to GFOV in synchronous machines is not present. This paper investigates whether GFOV is a problem in IBRs, and whether conventional mitigation requirements, such as providing a grounding transformer (GTF), are suitable for IBR installations. To answer these questions, a Controller Hardware-in-the-Loop (CHIL) performance analysis is conducted. To this end, different simulation models have been developed to analyze the IBR's control and protection response. The models are comprised of a 13.2 kV, 500 kW distribution system fed by a grid connected PV inverter which was simulated in Typhoon HIL 604 real time simulator, with a IEEE Std 1547-2018 compliant external physical controller connected in the loop. The experimental set-up and tests conducted are explained and results are analyzed, showing that effective grounding requirements are much different than those for traditional generators.

- Portions of this chapter previously appeared as: P. M. Adhikari, L. Vanfretti, M. Ruppert, M. Ropp, "Real-time controller hardware-in-the-loop (RT CHIL) analysis of ground fault overvoltages (GFOVs)," presented at the *CIGRE US Nat. Committee (USNC) Grid of the Future (GOTF) Symp.*, Providence, RI, USA, Oct. 17-20, 2021, Paper B5.
- Portions of this chapter have been submitted as: P. M. Adhikari, L. Vanfretti, A. Benjac, R. Bründlinger, M. Ruppert, M. Ropp, "Overvoltage mitigation strategies for PV inverters: An RT-CHIL based experimental investigation," submitted for publication.

6.1 Introduction

This paper investigates the schemes for protecting PV inverters from transient overvoltages (TrOV) under single-line-to-ground (SLG) faults. To carry out this investigation, Typhoon HIL based real-time controller hardware in the loop (CHIL) models for a grid connected PV-inverter were developed. The chapter is structured into five sections. The first section focuses on the motivation, presents a brief literature review, and enumerates the specific contributions of this article. The second section describes the experimental setup and methodologies in details. The third section presents the detailed simulation results, and the fourth section analyzes these results in the context of inverter protection. The final section concludes this article with additional discussions on the findings.

6.1.1 Motivation

It has been observed that up to 80% [75] of all the faults that occur in power systems, are single line to ground (SLG) faults. Theoretically, SLG faults occurring on a system supplied by ungrounded synchronous generators can lead to a Ground Fault Overvoltage (GFOV) of up to 173% of the nominal voltage on the unfaulted phases[76]. The theoretical derivation for this observation is carried out in Section 6.2. To mitigate this over-voltage problem the IEEE std C62.92.2 has proposed certain *effective grounding techniques* for synchronous generators. However, such existing methods require further validation and analysis in the context of their application in IBR based systems. To date, there is a gap in understanding on how IBR based systems would respond under SLG if existing *effective grounding techniques* are employed, and whether or not they are actually required for IBRs. Thus, the current work investigates the GFOV phenomenon and the application of existing mitigation techniques in the context of IBRs. Inverters, whether used for photovoltaic (PV) or energy storage facilities, typically include internal fast overvoltage protection mechanisms designed primarily to protect the inverter itself from damaging transients. These mechanisms, referred to as Self Protection Over-Voltage (SPOV) mechanisms, have the added benefit of causing the inverter to cease to energize when the circuit voltage exceeds certain limits. The SPOV mechanisms thus can mitigate both ground-fault overvoltage (GFOV), and load-rejection overvoltage (LROV). With the SPOV function included in the inverter, the main purpose of this work is to demonstrate that overvoltages are mitigated with or without a grounding transformer when Yg-Yg GSU transformer is used. To carry out this analysis, a controller-hardware-in-the-loop (CHIL) study is conducted by coupling a power system model designed specifically for this GFOV analysis to a *real-world* control/protection hardware device for IBRs. To this end a model for a 13.2 kW, 500 kW distribution system including a grid connected PV unit was implemented in a real-time simulator (RTS).

This choice was made because the response of IBRs is largely determined by the *real-world* software/hardware implementation of their control and protection features. The CHIL approach for testing control/protection equipment coupling an RTS with the actual control/protection hardware allows for a test-setup which reproduces highly transient power system behaviour with full feedback between the control system and the plant-hardware. In RT-CHIL experiments, RTSs are used to simulate the response of a power system in real-time. In this work, the Typhoon HIL 604 RTS was used to model the grid connected PV system. To control the PV inverter, an IEEE Std 1547-2018-compliant control hardware, the Austrian Institute of Technology Smart Grid Controller (known as the ASGC or AIT SGC), was coupled with the RTS[77]. This configuration allows to evaluate how controller-hardware connected to the PV system would react to the SLG fault and to assess a possible GFOV. To modify the SPOV settings, the firmware on the controller was manipulated. This procedure is elaborated in Section 6.3.6

6.1.2 Related Works

The severity of GFOV observed in synchronous generators has been documented and theoretically analyzed previously [78]. This work showed that ungrounded 4 wire/3-phase distribution systems supplied by synchronous generators are susceptible to an over-voltage of up to 1.73 pu under SLG conditions, and evidence of this phenomena in various actual systems has been obtained through the survey presented by the authors in [79]. SLG faults are studied in great details for Romanian power grid in [80], and similar studies focusing on North America, have been published in [81]. Hence, there is a good understanding of GFOV due to faults for systems with conventional generators.

A recent study addressed the GFOV phenomena for IBR-based systems [82] and demonstrates that the resulting over-voltages in three-phase four-wire circuits serving Y-grounded impedance loads could be expected to reach no more than about 122% of the nominal voltage, which is below the requirements for effective grounding. However, they cannot be directly mitigated by the existing techniques in the IEEE Std 62.92.2 [83]. The results demonstrated by the authors in [84] established that if standard symmetrical-component analysis is used to solve for the impedance of a hypothetical grounding bank that would ensure that the voltages of both the unfaulted phases during an SLG fault would remain below the definition of "effectively grounded" above a certain GLR, it would yield a negative value. Thus there's no real-world solution for designing a GTF that would mitigate the overvoltages IBRs face under an SLG fault when they are operating under a high GLR (over 4.5). This phenomenon was explained by the fact that the negative-sequence voltage starts to dominate above that GLR.

This is because, typically grid-following IBRs cannot be modeled by ideal voltage sources. Instead, the responses of IBRs is largely defined by their fast-acting internal control and protection strategies, implemented in the firmware. It is also important to note that the IBRs can exhibit both GFOV and Load Rejection Over-voltage (LROV), when subjected to an SLG condition; and their Self-protection Overvoltage (SPOV) might interact with the GFOV or LROV phenomena. These hypotheses are addressed in this paper.

The use of digital real-time simulators is becoming increasingly relevant for technical performance analysis of energy resources and IBRs when integrated into the electrical grid. It is crucial to note that different existing RTS are widely disparate in terms of their modeling environments, component model libraries and the hardware they use. The surveys presented in [85] and [86] summarized intricate comparisons between these various real-time simulation platforms including but not limited to those from companies such as Opal-RT, Typhoon HIL, dSPACE, RTDS, etc. In this work, the discussion is constrained to the RTS from Typhoon HIL and related SW/HW infrastructure. The decision to use this simulation hardware was based on the compatibility of the ASGC control hardware with Typhoon HIL 604 RTS. Typhoon HIL supports SunSpec compliant inverter controller hardware, which needs to be connected to the real-time simulator that runs the inverter model via a breakout board. The ASGC is one of such controllers, while [77, 87, 88] have reported detailed descriptions of this hardware for different applications, such as the control the distributed energy sources (DER) being simulated on the real-time hardware.

6.1.3 Contributions

- This paper validates the utilization of appropriate SPOV settings to mitigate the TrOV and keep it from violating the constraints reported in IEEE std 1547-2018.

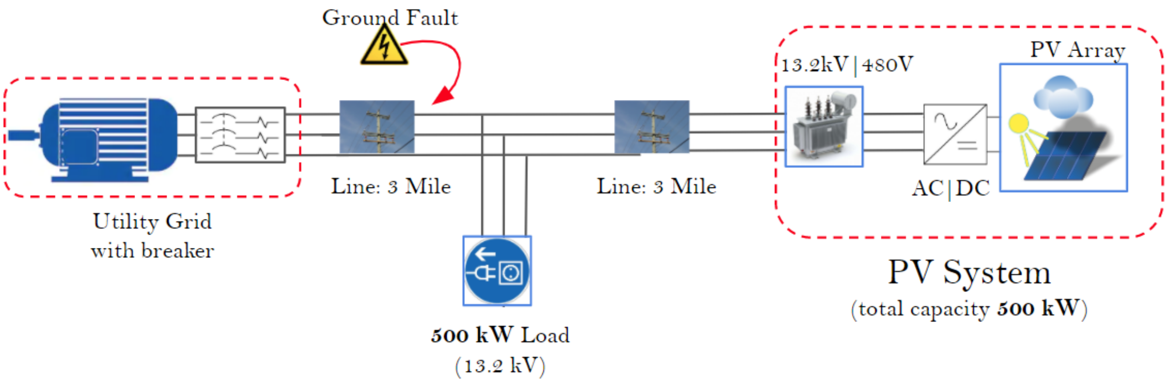


Figure 6.1: Pictorial Representation of the Grid Connected PV Distribution Feeder Used in This Work

- This paper reports experimental results to demonstrate that traditional effective grounding techniques like grounding transformers are not suitable for mitigating TrOV for inverters under SLG, operating with higher GLRs (On predominantly Y-connected loads). Extensive sequence analyses were carried out to provide a theoretical justification of this phenomenon. The experimental observations were found to be corroborating to the research published in existing literature.

To achieve these two objectives, a Typhoon HIL based RT-CHIL test-setup was modeled and interfaced with an IEEE 1547 compliant controller hardware.

Table 6.1: Software Used in the CHIL Experiment

Software Type	Version	Function	Library/Selection
Firmware Manager	2020-release	Simulator Configuration	Control of Processor cores and Machine Cores, Memory Allocation
aBoot Flasher	3.1.1	Controller Configuration	Control of current & voltage configuration of the ASGC controller
Schematic Editor	2019/20	Designs the power circuit	Typhoon's propreitery library, User defined C functions, Initialization
HIL SCADA	8.4	Interact with the circuit running on simulator	HIL SCADA Widgets, Monitoring, Data Logging and Visual Library
	3.7	Interact with the controller hardware in loop	HIL SCADA Widgets, Monitoring, Data Logging and Visual Library

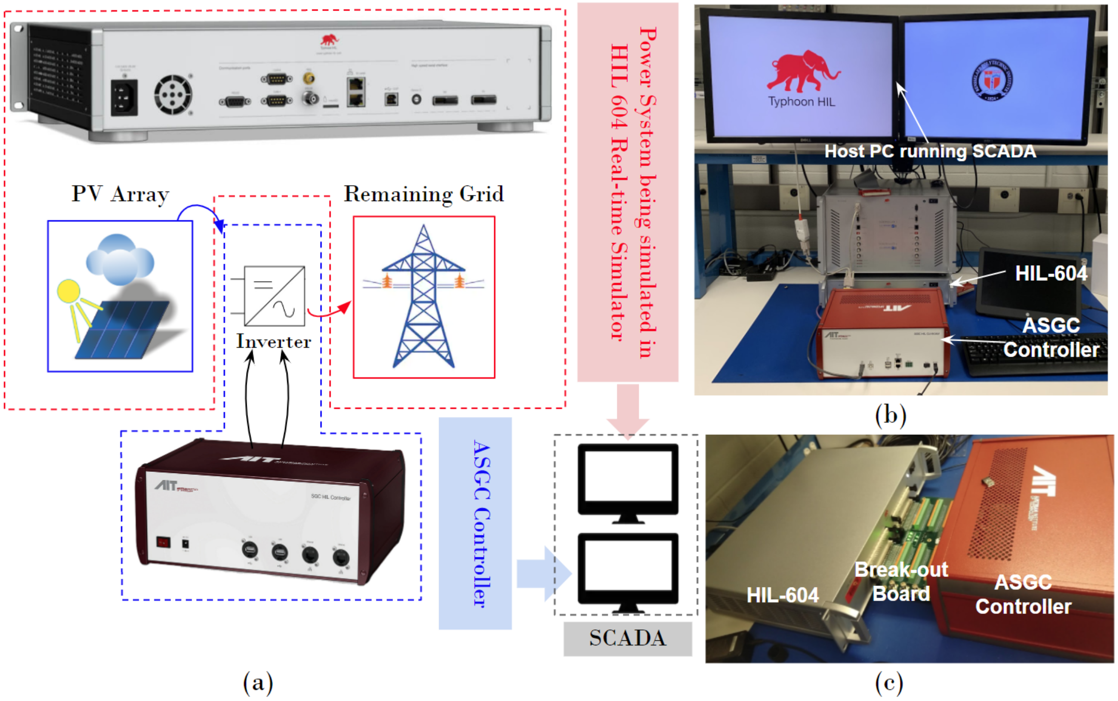


Figure 6.2: (a) Pictorial Representation of the Experimental Setup, (b) The Physical Experimental Setup, (c) The Interface Between ASGC Controller and Typhoon HIL 604

6.2 Experimental Setup

6.2.1 Experiment's Hardware Infrastructure Overview

The experiments conducted in this paper were carried out by simulating the power distribution circuit represented in Figure6.1 using the Typhoon HIL-604 RTS. The inverter is controlled by the external controller-hardware (ASGC) generated PWM signals. Figure 6.2.(b) shows a picture of the test bench with all these hardware components. Detailed description of these individual hardware components is presented in Appendix D.

6.2.2 Experiment's Software Overview

Table 6.1 summarizes the software tools utilized to carry out the GFOV experiment. It can be seen that most of these software tools are proprietary ones provided by Typhoon HIL and AIT. It needs to be noted that two different versions of the HIL SCADA software package are required. The latest version(8.4) interacts with the real-time simulator running the inverter model, and the older version (v3.7) interacts with the controller connected to the simulator. The functions of the different software are explained in detail in Appendix D.

Algorithm 2 Protocol for GFOV Experiments

```
/* SLG Fault Introduction                                         */
if $\angle V_a - \theta < \epsilon$ then
    if Faulter=1 then
        $BKR_F = CLOSE$
        State= FLT
        EXIT to next function
    else
        $BKR_F = OPEN$
else
    Continue Simulation
/* Breaker Opening                                                */
while State=FLT do
    Wait (35 ms) ;
    do in parallel
        if $\angle I_a < \epsilon$ then
            $BKR_A = OPEN$
        if $\angle I_b < \epsilon$ then
            $BKR_B = OPEN$
        if $\angle I_c < \epsilon$ then
            $BKR_C = OPEN$
```

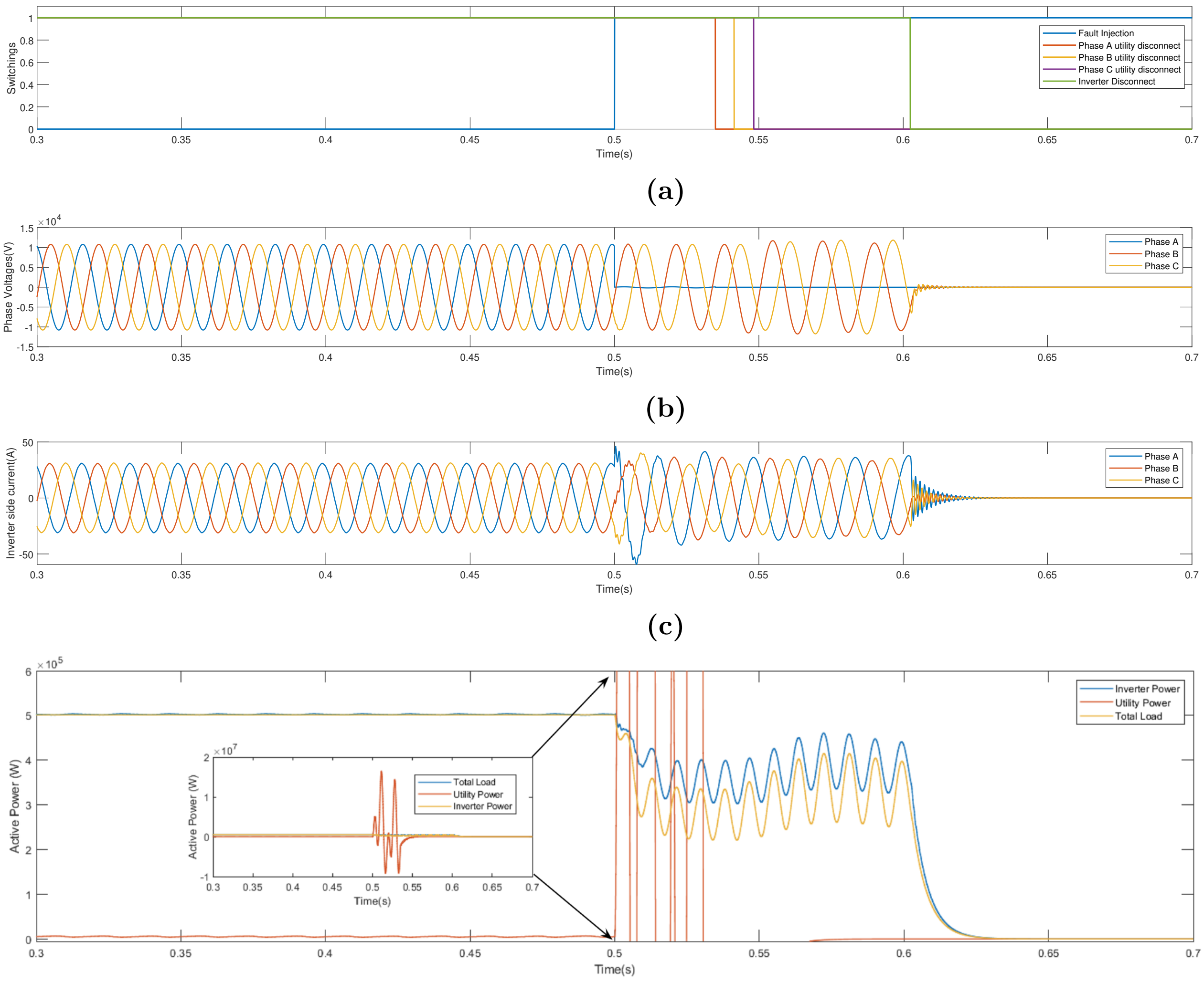
6.2.3 Experiment Setup and Protocol

This section describes how they were used to develop the experimental setup developed to analyze the GFOV phenomena.

Figure 6.1 illustrates the model that was simulated on the Typhoon HIL 604. The specifications of the model's components are presented in table 6.2. It needs to be noted that, the system shown in fuigure 6.1 also includes a capacitance-bank in parallel with the load. This capacitance-bank balances out the inductive reactive power consumed by the magnetization path of the 480V/13200V distribution transformer in order to help stabilize the frequency when the grid is disconnected.

To perform the GFOV experiments, the experimental protocol described in Algorithm 2 was followed. There are two interventions that need to be carefully applied, and to implement them one macro was written inside the Typhoon HIL SCADA (v8.4) tool.

First, the SLG fault is applied on the load bus. Next, the utility-side breaker waits 35 ms and then disconnects itself from the faulted portion of the system, leaving only the PV-inverter to feed the fault current. To ensure reproducibility in the experiments, the switch



(a)

(b)

(c)

(d)

Figure 6.3: CHIL Simulation under GLR 1.0: (a) Operation of the Breakers, (b) Phase Voltages, (c) Phase Currents (PV-Inverter) (d) Active Power Provided by the Utility and the PV Source

that applies the SLG was set to only operate when the voltage at phase A is at its highest (i.e. $\angle V_A = 90^o(\theta)$). However, it was observed that, if this equality-relation ship is enforced, then the simulator often misses the 90^o mark, because the PLL loop that computes the angle of the voltage signals had an update-rate of only 100us. Thus, an additional parameter ϵ was introduced, which is used to define a finite but small window during which the fault can strike. ϵ was set to 2.5^o. This allowed to detect when the angle of the voltage waveform was within 2.5^o of the 90^o mark in order to apply the SLG. This sequence of events can also be visualized through Figure 6.3.(a) and Algorithm 2.

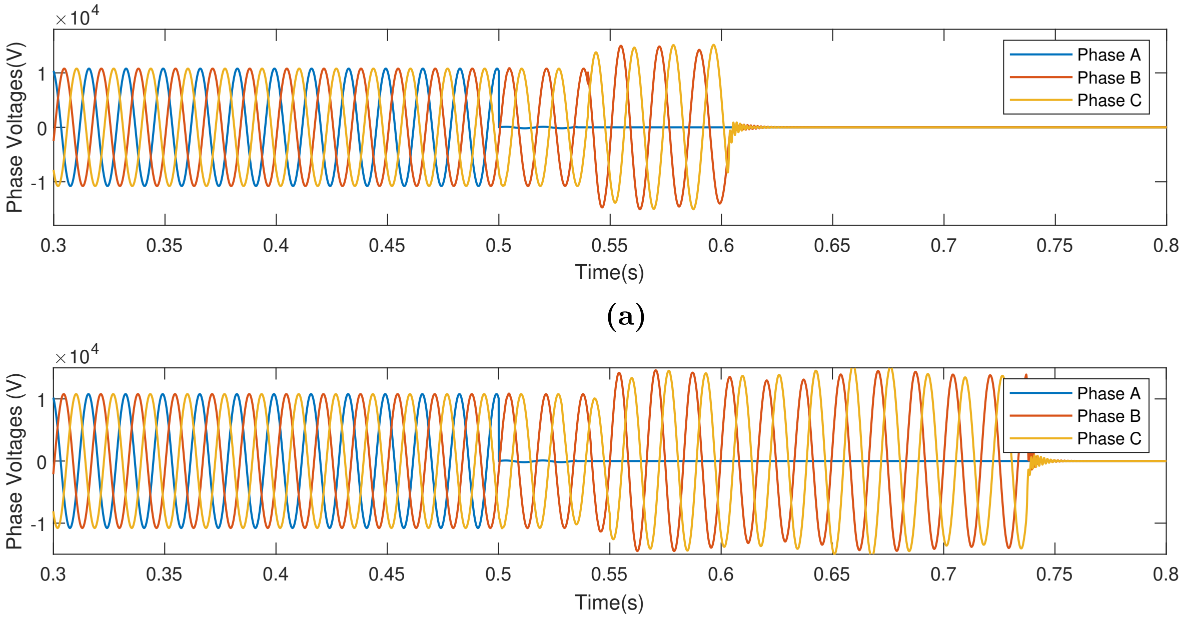
Table 6.2: Distribution System Specifications

Component	Specifications
Utility Source	Line-to-line Voltage, 13.2 kV 3 phase Short Circuit MVA, 180×10^6 X/R = 8 Line Construction = 336AA (phase), 3/o (Neutral)
Distribution Feeder	$Z_1 = 0.278 + j\ 0.682$ $Z_0 = 0.7575 + j1.9532$ Distance from PV to Load, 3 miles Distance from Load to Breaker, 3 miles
DG Transformer	3 phase, 480 V/13.2 kV kVA = 500, Z = 5%, X/R = 10 Configuration Yg-Yg
GTF (*if used*)	3 phase, 13.2 kV/13.2 kV kVA = 500, Z = 5%, X/R = 10 Configuration Yg-Δ
PV System	500 kW, 0 kVAR
Load	Constant Power, Phase to Ground kW = 500, kVAR = 0 Compensating Capacitance: 29.5 kVAR *(at untuned condition)*

Second, a similar approach was followed when disconnecting the grid from the remainder of the system. Upon waiting for 2 cycles (35 ms), the grid will be disconnected one phase at a time, only when the current on that phase is close to the zero-crossing. This was implemented by using the same parameter ϵ.

It is important to note that 35 ms is a very fast time for a standard distribution level breaker. However, in order to maximize the duration for which the inverter manages to provide the fault current (before disconnecting due to its own protection functionalities), the speed of the breaker was increased. The duration for which the inverter provides the fault current is the duration for which the GFOV may be observed.

Consequently, all the analyses were performed on the measurements taken during this period.



(a)

(b)

Figure 6.4: Effect of Capacitance-Tuning: (a) Phase Voltages Under SLG Without Tuned Capacitor Bank, (b) Phase Voltages Under SLG With Tuned Capacitor Bank

6.3 Results and Analysis

This section summarizes results from different analysis conducted.

6.3.1 Tuning of Capacitor Banks for GFOV Sequence Component Analysis

The first set of results presented in this section is a CHIL simulation of the system with a Generation to Load Ratio (GLR) of 1.0. These simulation results are summarized in Figure6.3.(a)-(d). It can be seen from Figure6.4.(a) and Figure6.4.(b), that after the utility detects the SLG fault and disconnects itself, the inverter only manages to sustain itself for 3 cycles. During these 3 cycles, there was a maximum of 8% over-voltage observed on the unfaulted phases. This can be seen upon careful observation of the waveforms in Figure6.4.(b) for $0.55s \leq t \leq 0.65s$. Clearly, this magnitude of over-voltage is much lower than what was observed (73%) in synchronous generators.

It was also observed that during these 3 cycles, the frequency varies rapidly and the inverter ultimately disconnects due to the operation of over-frequency relay. Because, the frequency was not constant, the angles were varying, and thus, it is theoretically difficult to apply symmetrical components to this situation. However, in order to evaluate the pro-

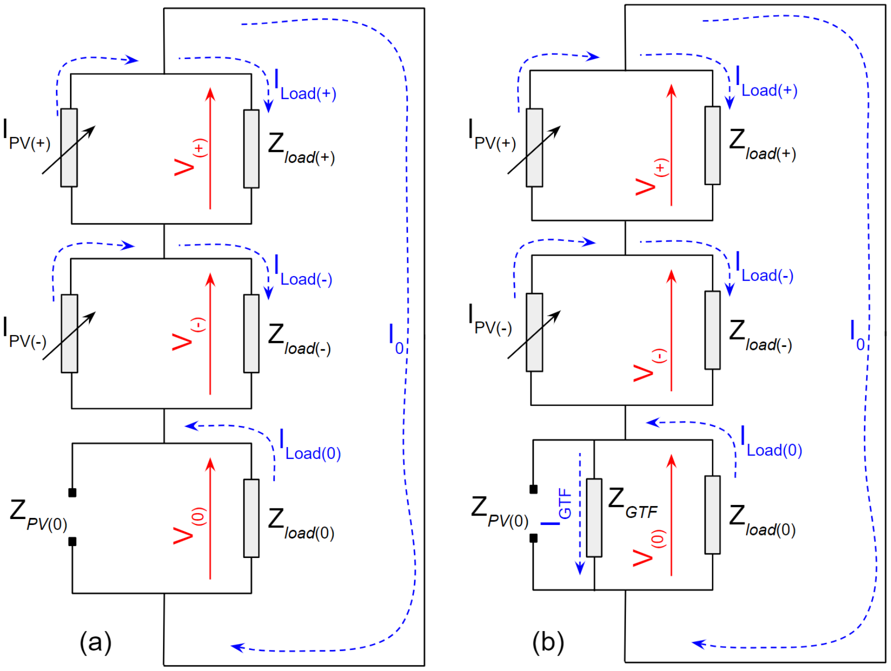


Figure 6.5: Sequence Diagram Under SLG: (a) Without the GTF, (b) With the GTF

tection schemes for such systems under SLG faults, and better understand the observed behaviours, it is useful to perform the symmetrical component analysis. To apply the symmetrical component analysis frequency needs to be relatively constant. To achieve this, the capacitance bank which was connected in parallel with the resistive load was *tuned*. Tuning of capacitance-banks is well-researched topic which has been addressed in [89]. For this particular research, tuning was done by trial and error. The tuned capacitance bank had a capacitance of 1.75 uF per phase . Upon successful tuning, the inverter manages to operate at a reasonably fixed frequency for a few cycles even after the utility has been disconnected. In order to investigate the nature and source of the overvoltage, a detailed symmetrical component analysis was performed on this modified model. The three-phase voltage-responses of the system with and without tuned capacitors are shown in Figure 6.4.(a). It can be seen that the inverter can operate for a much longer duration when the capacitor banks are carefully tuned. Under these new conditions the sequence diagram under faulted condition can be determined. When the utility has already disconnected itself from the inverter and the load, the sequence diagram is shown in Figure 6.5.(a).

Figures 6.6.(a)-(b) exhibit the phase currents from the inverter's side, and the phase currents measured on the load. Figures 6.6.(c)-(d) illustrate the symmetrical components of the currents plotted in Figure 6.6.(a)-(b) respectively. Notably, for the current from the PV-inverter side, the zero-sequence current was negligible while the negative-sequence current was significant. For the currents observed from the load, both the negative-sequence and zero-sequence currents are significant. This is expected because the phase-A current is zero while the phase-B and phase-C currents are higher than their unfaulted values.

To establish how these results relate to the circuit of Figure 6.5.(a), instantaneous values observed at $t = 0.65s$ seconds were considered. Applying KCL in the circuit of Figure 6.5.(a) gives:

$$I_0 = I_{PV(+)} - I_{load(+)} \tag{6.1a}$$

$$I_0 = I_{PV(-)} - I_{load(-)}. \tag{6.1b}$$

Next, applying KVL on Figure 6.5(a) yields:

$$V_+ = I_{load(+)} \times Z_{ph}, \tag{6.2a}$$

$$V_- = I_{load(-)} \times Z_{ph} \tag{6.2b}$$

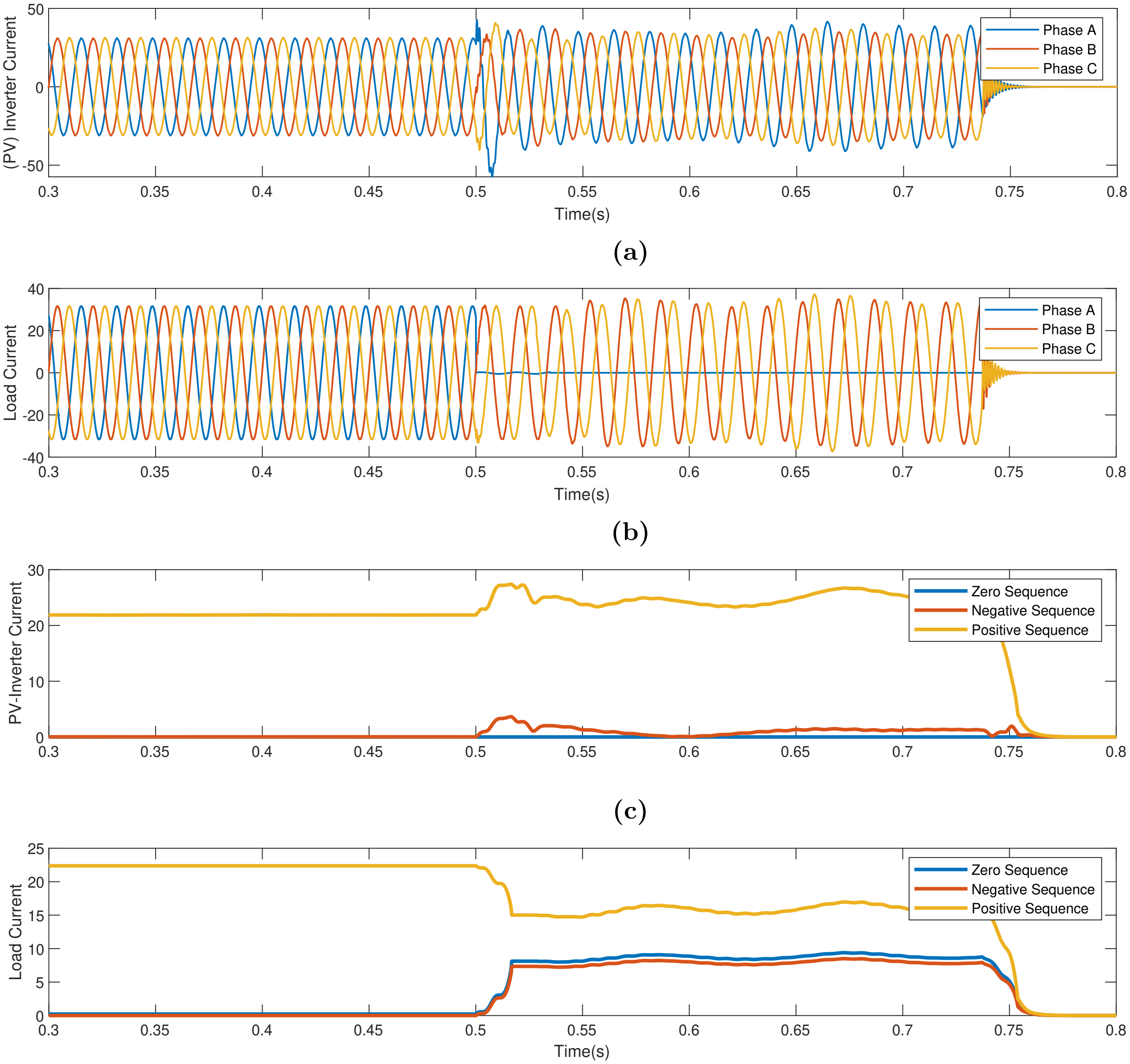
$$V_0 = I_{load(0)} \times Z_{ph}. \tag{6.2c}$$

The values of the phase currents for the PV inverter and the load at t=0.65 were logged from the observation made in Figure 6.5. These values are verified by using (6.1) and (6.2). The results are summarized in the first two rows of table 6.3. Having calculated the voltage-sequence components, those values were used to verify the phase voltages using:

$$V_b = a^2V_+ + aV_- + V_0 \tag{6.3a}$$

$$V_c = aV_+ + a^2V_- + V_0. \tag{6.3b}$$

The results for the unfaulted phases B and C are summarized in the third row of Table 6.3. It can be seen that there are some minor over-voltages in the unfaulted phases. This is consistent with the observations made in Figure 6.6.(b).



(a)

(b)

(c)

(d)

Figure 6.6: Currents When There is No GTF in the Circuit: (a) 3-Phase Currents From PV Inverter, (b) 3-Phase Load Current, (c) Sequence Components of Currents From PV Inverter, (d) Sequence Components of Load Current

Table 6.3: Sequence Components of Figure 6.5(a) at t=0.65 Sec

Eqn	Numerical Validation
KCL(6.1)	$I_{PV(-)} - I_{load(-)} = 1.93\angle{-17.6^\circ} - 7.34\angle{-178^\circ}$ $= 9.18\angle{-3.1^\circ}$ $I_{PV(+)} - I_{load(+)} = 26.07\angle{0^\circ} - 16.9\angle{1.7^\circ}$ $= 9.175\angle{-2^\circ}$ $I_{load(0)} = I_0 = -9.18\angle{178^\circ} = 9.18\angle{-2^\circ}$
KVL(6.2)	$V_- = Z_{ph} \times I_{load(-)} = 339\angle{-12.1^\circ} \times 7.34\angle{-178^\circ}$ $= 2488.26\angle{-190.1^\circ}$ $V_+ = Z_{ph} \times I_{load(+)} = 339\angle{-12.1^\circ} \times 16.9\angle{1.7^\circ}$ $= 5729.1\angle{-10.4^\circ}$ $V_0 = Z_{ph} \times I_{load(0)} = 339\angle{-12.1^\circ} \times 9.18\angle{-182^\circ}$ $= 3112.02\angle{-194.1^\circ}$
(6.3)	$V_b = a^2V_+ + aV_- + V_0 = 8361\angle{-134.4^\circ}$ $= 1.096pu$ $V_c = aV_+ + a^2V_- + V_0 = 8273\angle{112.8^\circ}$ $= 1.085pu$

6.3.2 GFOV and GLR

The next set of experiment was to vary the GLR and study the effect of GLR on GFOV. This was achieved by varying the load while keeping the generation from the PV-inverter constant. This type of study is reported in IEEE Std 62.92.6.-2017. In this work, the trend of GFOV with varying GLR was compared with those reported in the IEEE Std. This comparison is shown in Figure 6.7. It is also observed that overvoltage for IBRs is a bigger problem for larger values of GLR than it is for smaller values of GLR. It is also illustrated that at a GLR of 1.0, the GFOV is not as severe as it is for synchronous generators.

6.3.3 Impact of Conventional GFOV Mitigation

The next step in this experimentation was the application of traditional over-voltage mitigation techniques in this system. For this particular work, a grounding transformer was incorporated within the circuit (and the capacitance bank had to be re-tuned to obtain well-sustained results). Two different types of connections are used in grounding transformers

(GTF): (a) ZigZag or (b) Yg- Δ. Because the ZigZag transformer was not available in the model library, the Yg- Δ connection for the GTF was used. The specifications of the GTF are the same as the distribution transformer used in the model. blueIt is to be noted that this GTF is a large one, which was chosen to emphasize its impacts on the circuit. Under an SLG fault, the sequence diagram is also modified to include the GTF. This modified sequence diagram is illustrated in Figure 6.5.(b). The modified system model was simulated for a GLR of 1.0. The KCL applied to this sequence diagram should yield:

$$I_0 - I_{GTF} = I_{PV(+)} - I_{load(+)} \tag{6.4a}$$

$$I_0 - I_{GTF} = I_{PV(-)} - I_{load(-)} \tag{6.4b}$$

Equations (6.4) and (6.2) were used to verify the experiment results for the modified system with GTF in table 6.4. It can be seen that the zero-sequence voltage is significantly reduced by the addition of the grounding transformer, but this reduction in zero-sequence voltage was not reflected in the negative sequence circuit. In fact, there was a minor increase in the negative sequence voltage upon the introduction of the GTF.

Hence, the improvement in TrOV by incorporating GTFs is negligible. It can also be seen that one of the unfaulted phases bears a lower voltage than the other. This is consistent with the observation in Figure 6.8.(b), and those reported in [84]. It is also to be noted, that the GTF provides a significant amount of zero sequence current into the network as shown in Figure 6.9.(e). This influx of zero sequence current changes the behavior of the circuit significantly, but it does not improve the existing overvoltage (around 10%) in the system. The phasor diagrams corresponding to the observations noted in table 6.3 and table 6.4 are shown in Figure 6.12.(a)-(d).

6.3.4 Impact of the Power Factor (PF) in the Load

In this experiment the configuration of the existing load was modified to include inductive components within it (The capacitance bank was re-tuned to achieve well-sustained results). The resultant power factor after this modification was set to be 0.9 (lag). Figure 6.13.(a) shows the GFOV while the load pf is 0.9 and the active power is set to 500 kW, resulting in an apparent power of 556 kVA. The observed maximum overvoltage in this experiment was close to 9%. Similarly, Figure 6.13.(b) demonstrates the overvoltages when

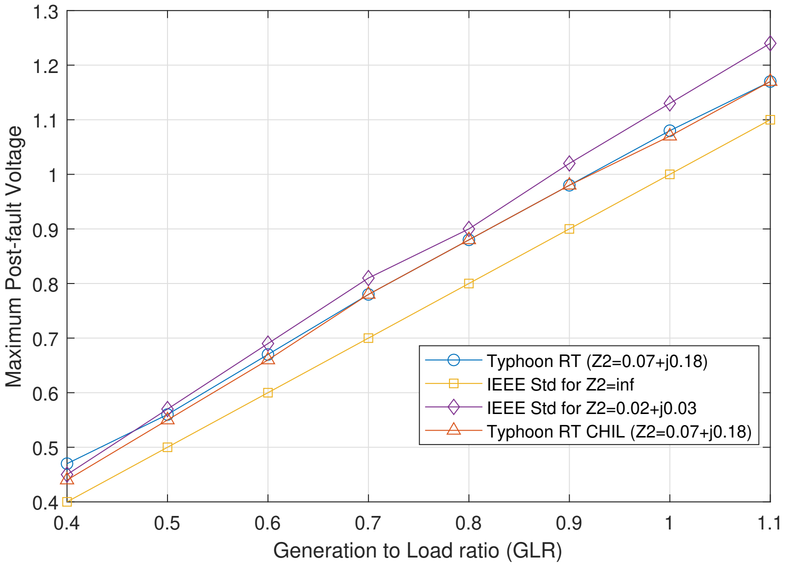
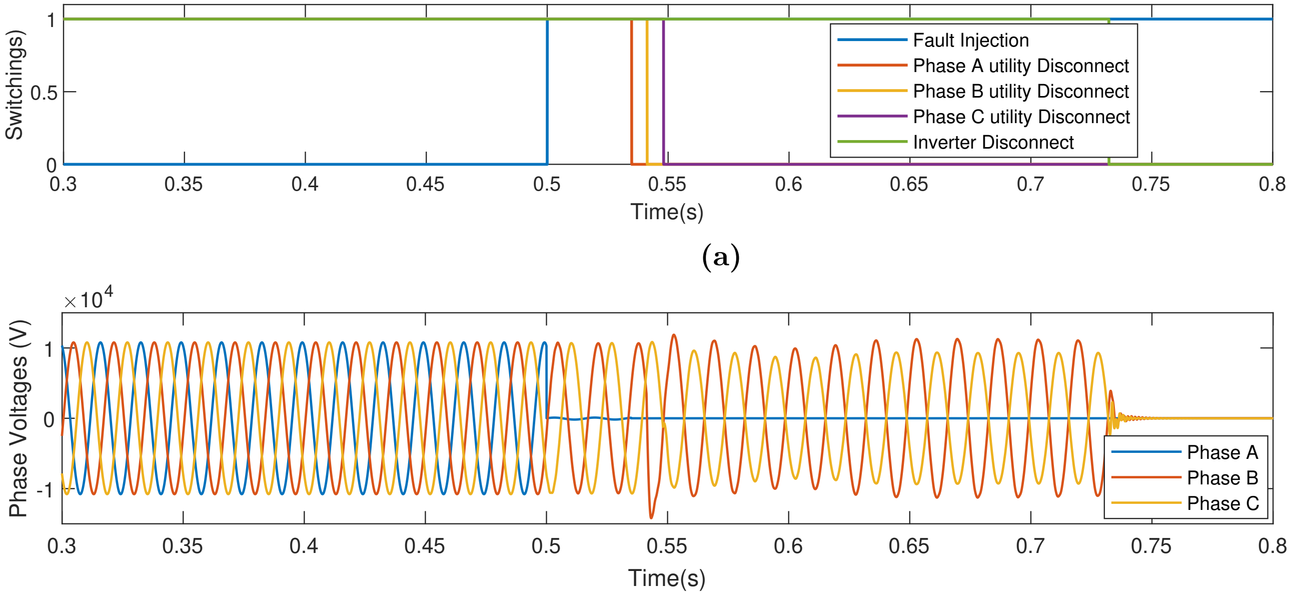


Figure 6.7: Variation of Overvoltage With Increasing GLR

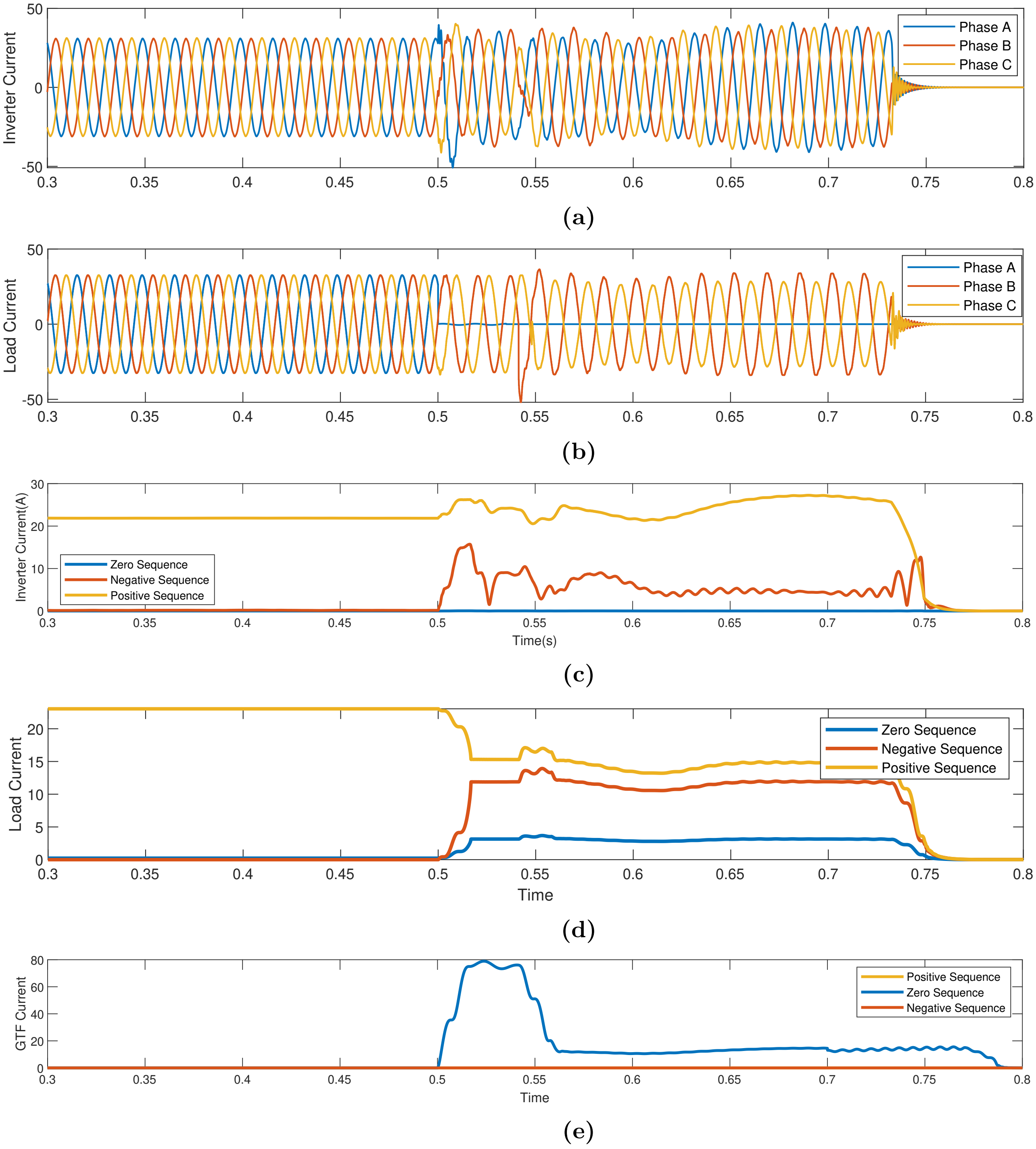
the load pf is 0.9 and the apparent power is 463 kVA. This yields an active power output of 416.6 kW, and a GLR of 1.2 if the inverter output is kept constant at 500kW. The observed maximum overvoltage in these experiment was close to 29%. These results are consistent with the results obtained with upf loads at GLR=1.0 and 1.2, which are presented in the



(a)

(b)

Figure 6.8: Overvoltage Simulation With GLR=1.0 and GTF: (a) Switching Sequence, (b) Phase Voltages



(a)

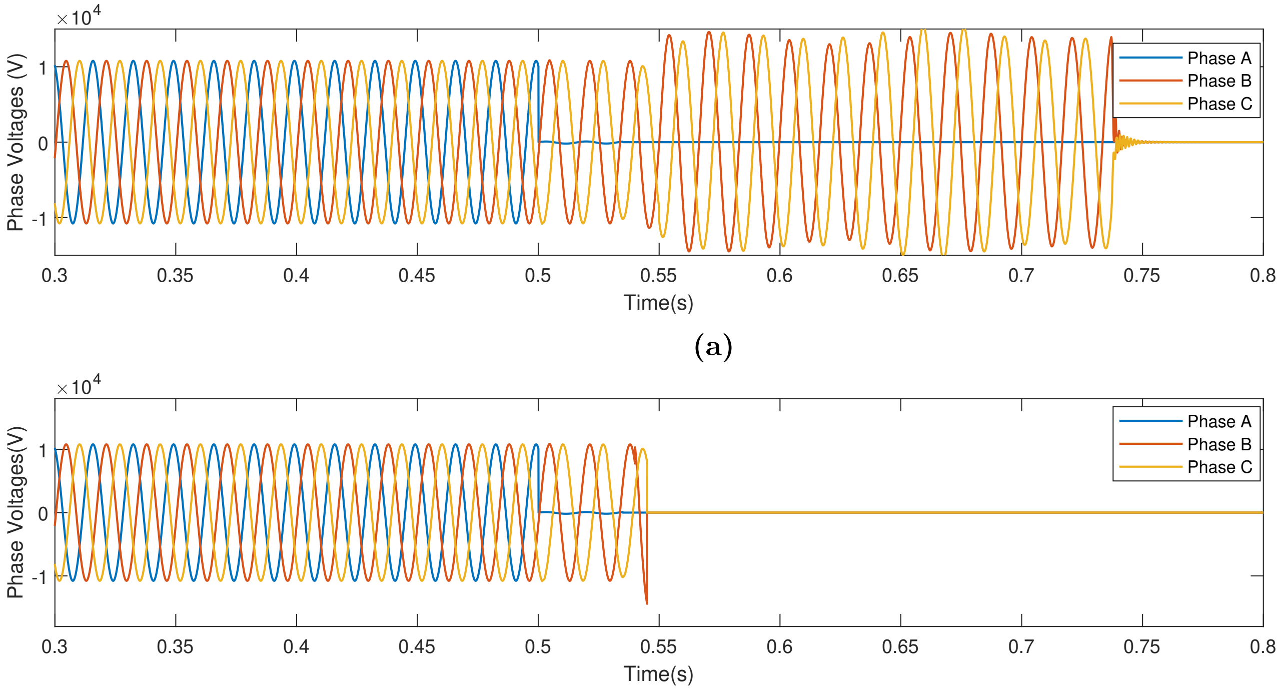
(b)

(c)

(d)

(e)

Figure 6.9: Currents With GTF in the Circuit: (a) 3-Phase Currents From PV Inverter, (b) 3-Phase Load Current, (c) Sequence Components of PV Inverter Currents (d) Sequence Components of Load Current (e) Sequence Components of the GTF Current



(a)

(b)

Figure 6.10: SPOV Interaction With the TrOV Simulations for GLR=1.30: (a) When SPOV is Disabled at 1.3 (b) When SPOV is Enabled at 1.3

second column of table 6.6. Thus, it can be concluded that, the inductive reactive power component of the load does **not** effect the maximum GFOV as long as the active power is maintained constant.

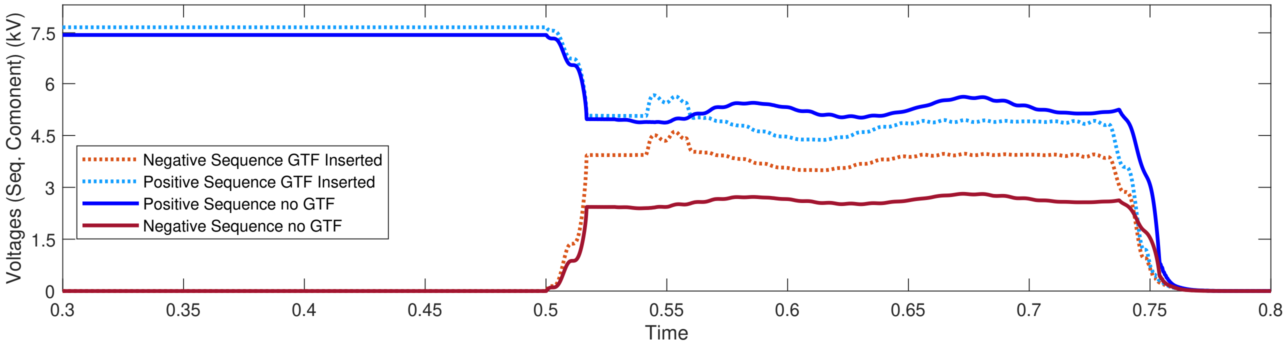
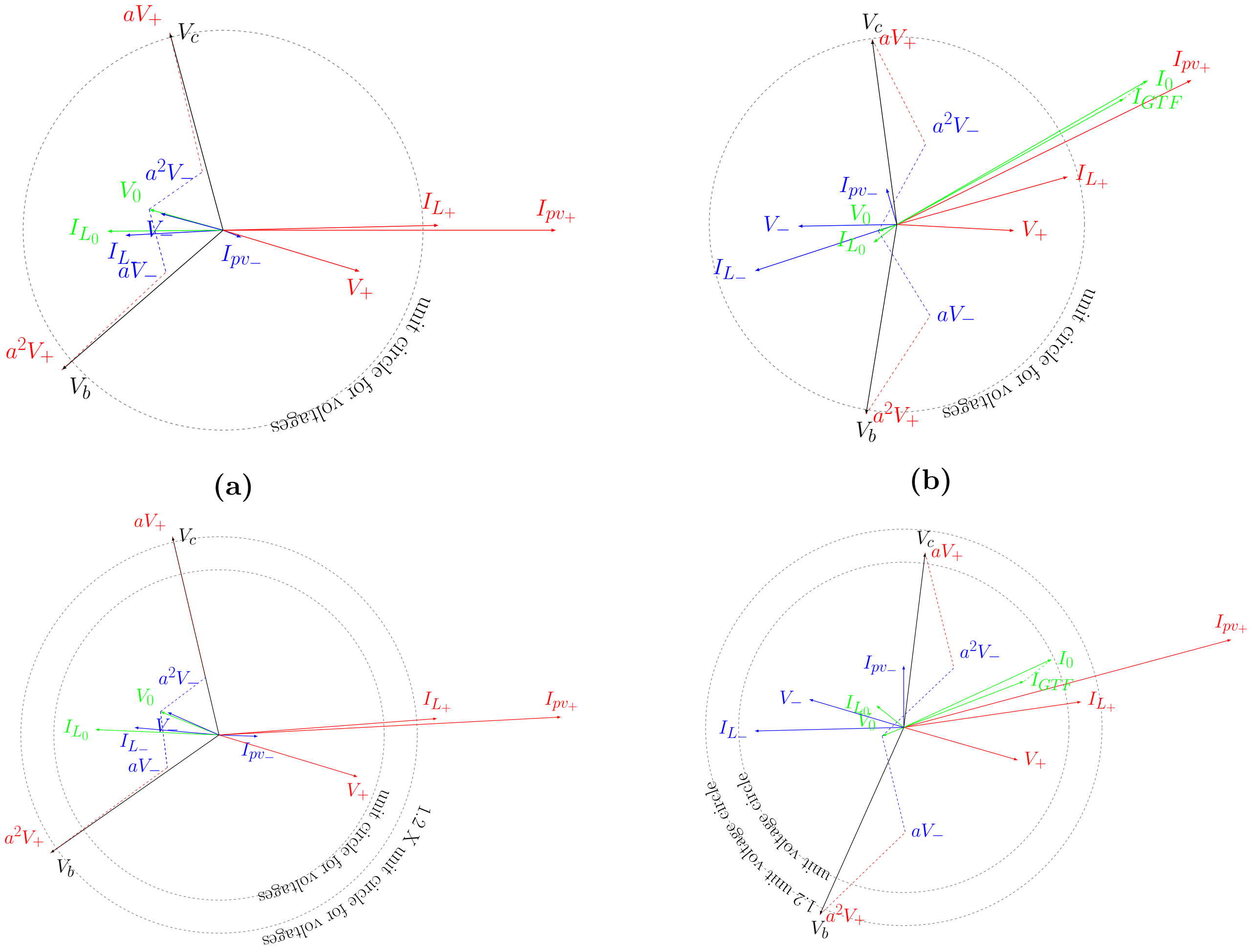


Figure 6.11: Comparison of Sequence Components of the Voltages With and Without the Grounding Transformer

6.3.5 Impact of Δ Connected Loads

While section 6.3.3 explores the utilization of grounding transformers for mitigating overvoltage in wye connected loads subjected to SLG, this section investigates whether Δ connected loads exhibit similar overvoltages and whether grounding transformers can be used to mitigate them. Thus, the existing wye connected load is re-configured to include considerable shares of Δ connected loads.

This phenomenon can be explained well by simple sequence analysis. When Δ connected loads are introduced in the load, the zero sequence circuit of the system gets modified. This leads to overvoltage in the zero sequence circuit. This zero sequence overvoltage



(a)

(b)

(c)

(d)

Figure 6.12: Phasor Diagrams: (a) No GTF Inserted, GLR=1, (b) GTF Inserted, GLR=1, (c) No GTF Inserted, GLR=1.2, (d) GTF Inserted, GLR=1.2. (Red=Positive Sequence, Blue=Negative Sequence , Green=Zero Sequence, Black= Phase Voltages)

Table 6.4: Sequence Components of Figure 6.5(b) at t=0.65 Sec

Eqn	Numerical Validation
(6.4)	$I_{PV(-)} - I_{load(-)} = 2.99\angle 106^\circ - 11.8\angle 199^\circ = 12.31\angle 33.2^\circ$ $I_{PV(+)} - I_{load(+)} = 25.9\angle 26.1^\circ - 14.02\angle 15.6^\circ = 12.37\angle 37^\circ$ $I_{load(0)} - I_{GTF} = I_0 = -10.25\angle 29^\circ + 2.35\angle -142^\circ = -12.37\angle 31^\circ$
(6.2)	$V_- = Z_{ph} \times I_{load(-)} = 331.1\angle -18.1^\circ \times 11.8\angle 199^\circ = 3906\angle 180.9^\circ$ $V_+ = Z_{ph} \times I_{load(+)} = 331.1\angle -18.1^\circ \times 14.02\angle 15.6^\circ = 4642\angle -2.5^\circ$ $V_0 = Z_{ph} \times I_{load(0)} = 331.1\angle -18.1^\circ \times 2.35\angle -142^\circ = 778.1\angle -160.1^\circ$
(6.3)	$V_b = a^2V_+ + aV_- + V_0 = 7841\angle -99.8^\circ = 1.03pu$ $V_c = aV_+ + a^2V_- + V_0 = 7151\angle 97.31^\circ = 0.93pu$

Table 6.5: Effect of GTFs on TrOV With Varying Share of Δ Connected Load

Max TrOV at GLR=1.0	Y=100% Δ=0%	Y=75% Δ=25%	Y=50% Δ=50%	Y=25% Δ=75%
Without GTF	1.089	1.19	1.32	1.48
With GTF	1.05	1.07	1.07	No good observations

translates into an overvoltage along the unfaulted phases. However, since the grounding transformer adjusts the zero sequence circuit of the system, it manages to compensate for the overvoltage stemming from the zero sequence voltage component. It is important to note that, for each test-case in table 6.5 (with increasing share of Δ connected load) the capacitance bank needs to be re-tuned, and this procedure is very time-consuming. This re-tuning ensures that the inverter manages to sustain the voltages for a few cycles after the utility disconnects. The comparison between figures 6.13 and 6.14 shows that the introduction of a grounding transformer reduces the overvoltages in both phases B and C. However, a closer observation reveals that the voltage reduction is more in phase C than it is in phase B. This is similar to what was observed in a system without any Δ connected load and is supported theoretically by equation (C.4). Overall, it was observed that the maximum overvoltage reduced from 19% to 7%, upon the utilization of the grounding transformer.

Similar observations were recorded when the share of Δ connected load is increased from 25% to 50%. Without the grounding transformer the overvoltage was observed to be around 32% (Figure 6.14.(b)). The grounding transformer reduced this overvoltage from 32% to 7% (Figure 6.15.(b)). Clearly, for systems with Δ connected loads the usage of grounding transformers leads to significant reduction in overvoltages.

Figures 6.14.(c) demonstrates that, when the share of Δ connected load is increased to 75%, it was observed that the overvoltage under SLG condition was close to 50%. Upon the addition of the grounding transformer the inverter observes that Vbn.max, Ib.max, Ic.max flags are set from the ASGC controller side when the SLG fault is applied and the breaker

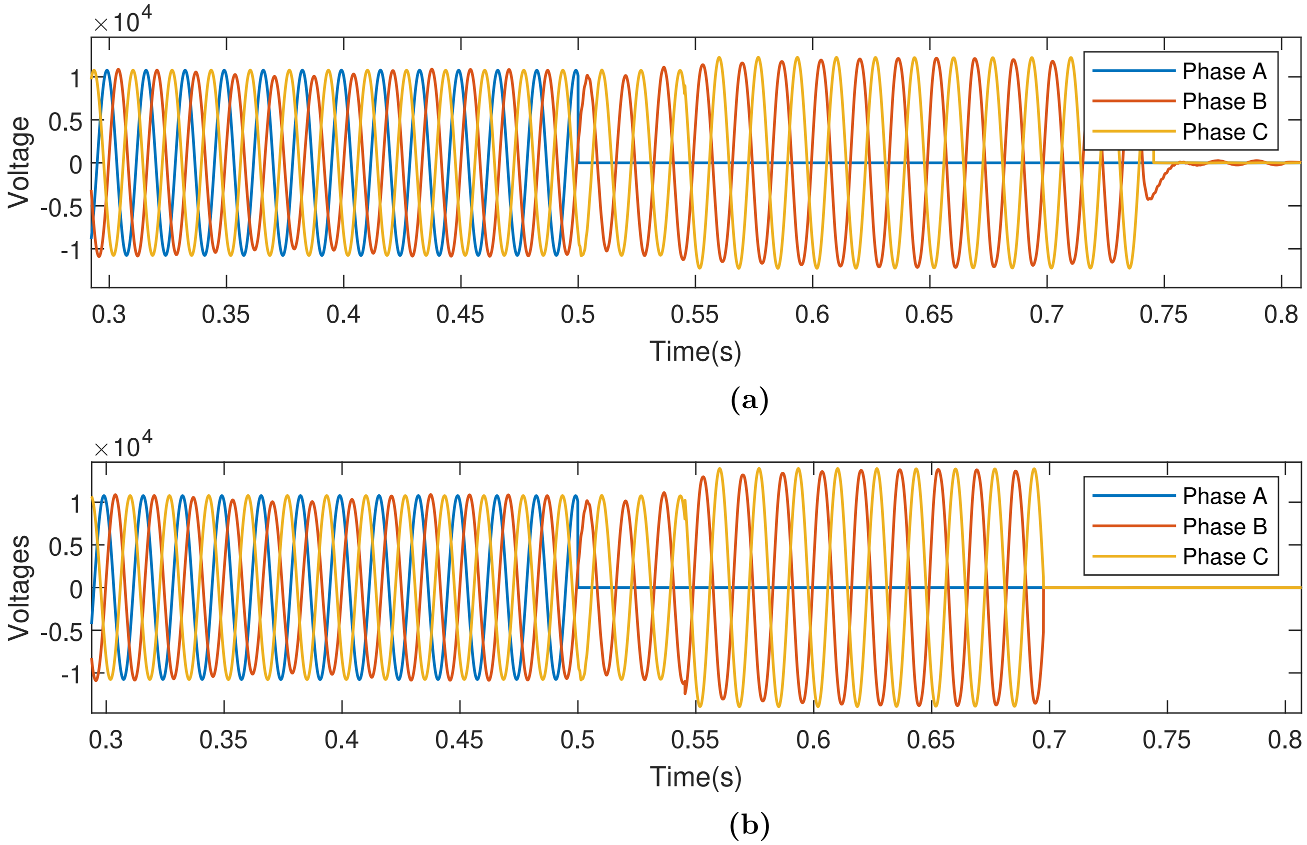
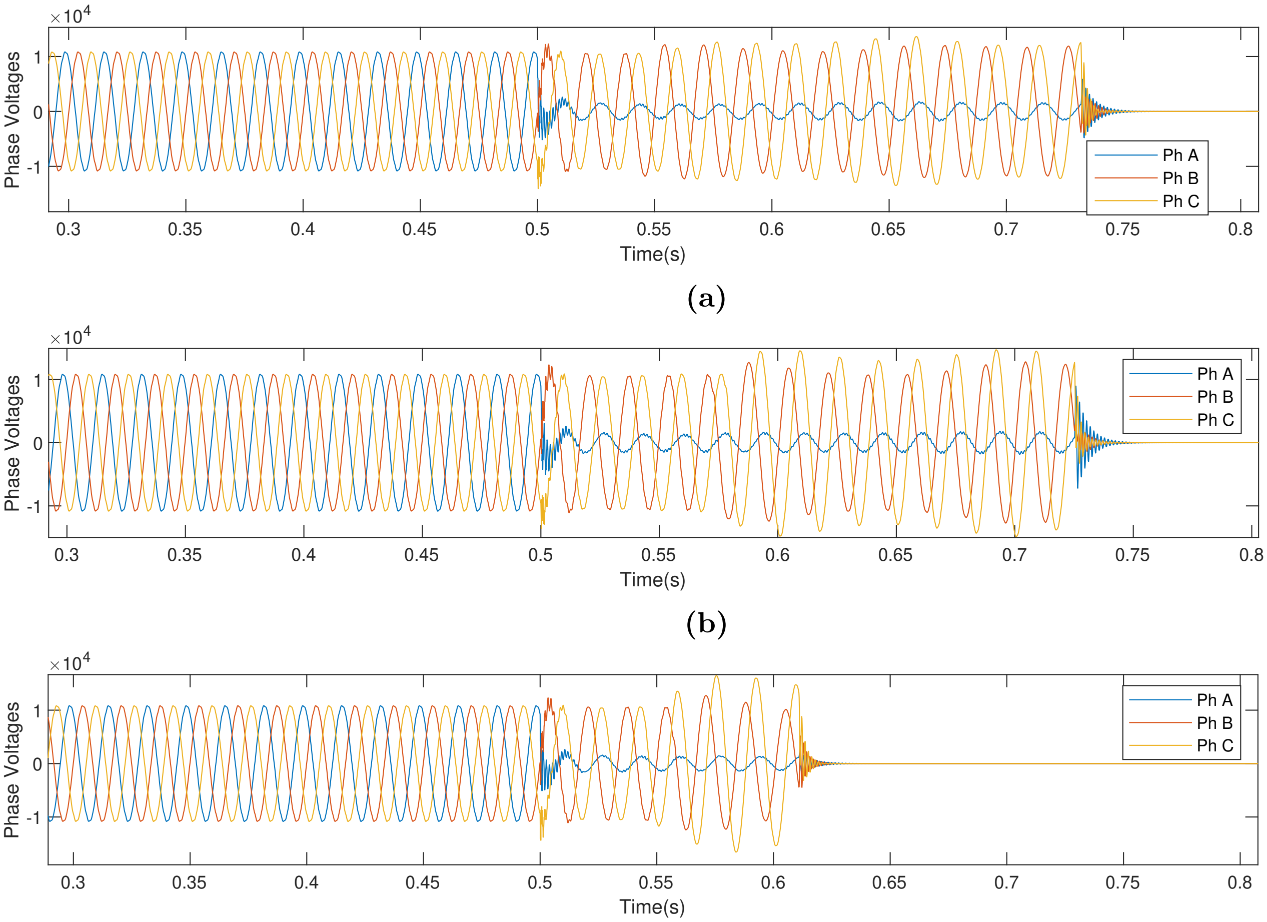


Figure 6.13: TrOV Experimentation With Non-UPF (0.9 Lag) Loads: (a) While GLR=1.0, (b) While GLR=1.2



(a)

(b)

(c)

Figure 6.14: Overvoltage Simulation With GLR=1.0 Under UPF, With No GTF Present: (a) While Δ-Share =25%, (b)While Δ-Share =50%, (c) While Δ-Share =75%

disconnects the utility from the rest of the system. These flags initiate the subroutine that disconnects the ASGC controller from the Typhoon HIL 604 simulator running the inverter model. It was not possible to bypass this protective operation from the controller, and thus no overvoltage computation is reported for this case in table 6.5. The corresponding phase voltages can be observed in Figure 6.15.(c).

The objective of the current research is to investigate whether or not grounding transformers are suitable to mitigate the ground-fault overvoltages in inverter based DERs. The experimental results demonstrated in this section establishes that grounding transformers can be effective in mitigating ground-fault overvoltages in inverter based DERs only when there is some significant share of Δ connected load being provided by the DERs.

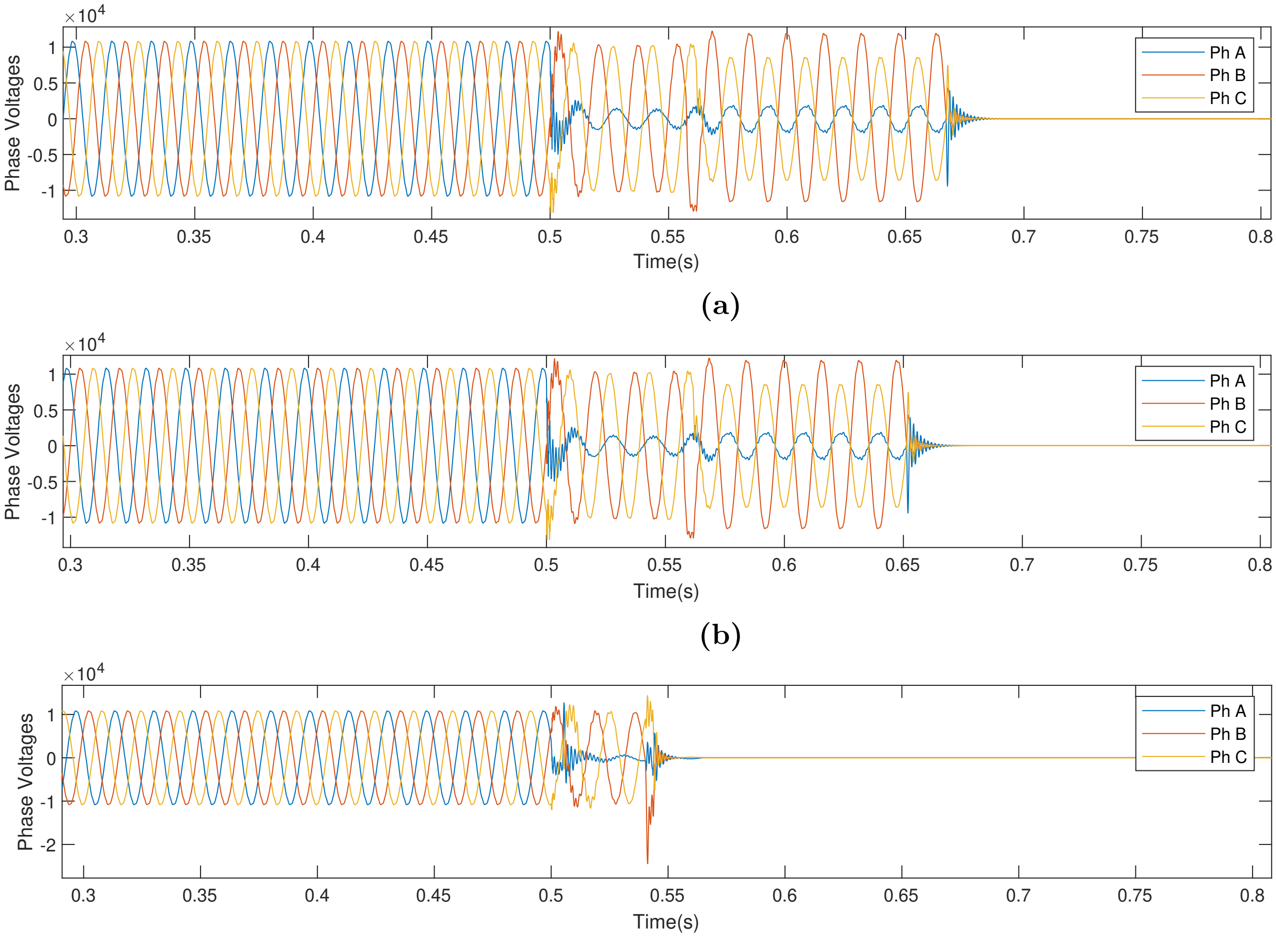
6.3.6 Impact of SPOV

It needs to be kept in mind that most modern inverters have a Self-Protection Over-voltage (SPOV) function that rapidly stops switch gating in case of an over-voltage. This operation is extremely fast and and is commonly based on instantaneous values of the voltages. This SPOV mechanism is very helpful in mitigating TrOV while the IBR operates at a GLR significantly over 1.0. Many inverters have the SPOV setting fixed at 1.3 pu or 1.4 pu, i.e. the inverter will disconnect in few hundred microseconds if it detects an over-voltage over 30% or 40%, respectively. The ASGC has a default SPOV setting **fixed at 1.3**. Because the TrOV varies almost linearly with the GLR, the SPOV for the inverter was automatically triggered whenever simulations were run for GLRs over 1.3. This makes experimenting with higher GLRs impossible without making modifications to the hardware. It also demonstrated that the internal SPOV settings of the ASGC controller hardware was very effective in mitigating the TrOV.

In order to run experiments with higher GLRs, the controller firmware was modified. However, direct modification of the SPOV is not possible, i.e. it is not possible to change the value from 1.3 to a higher value directly. Instead, it was necessary to change the *nominal voltage* parameter from 480 V to 600 V. This makes the new set-point for SPOV trigger at $600 \times 1.3 = 780$ V, which is 1.625 times of the previous nominal voltage of 480 V. Thus, under this new firmware settings, it is possible to run experiments which might result in higher over-voltages up to 1.625 pu. In Table 6.6, the results presented in blue colored rows are obtained from experiments performed with this new controller-firmware.

Table 6.6: TrOV Variation With and Without GTF

GLR	Max TrOV without GTF	Max TrOV with GTF
0.6	0.67	0.62
0.7	0.78	0.73
0.8	0.88	0.84
0.9	0.99	0.935
1	1.09	1.067
1.1	1.178	1.15
1.2	1.298	1.265
1.3	1.412	1.384
1.4	1.518	1.475
1.5	1.60	1.559
1.6	Unsustained	Unsustained



(a)

(b)

(c)

Figure 6.15: Overvoltage Simulation With GLR=1.0 Under UPF, With GTF Present: (a) While Δ-Share =25%, (b)While Δ-Share =50%, (c) While Δ-Share =75%

For the purpose of simplicity, this paper refers to the SPOV value of 1.3 as SPOV `enabled`, and the SPOV value of 1.625 as SPOV effectively `disabled`.

For a GLR of 1.3, two experiments were performed with the SPOV settings `disabled` and `enabled` respectively, in order to show how the SPOV interacts with the GFOV experiments. While the SPOV setting is kept as `disabled`, the overvoltage simulations were run and the results are archived in Figure 6.10.(a). It was observed that the overvoltage was close to 40% and the system experienced this level of overvoltage for over 10 cycles. However, when the firmware is changed to the version with lower voltage rating, i.e. SPOV `enabled`, the system disconnects instantaneously after it experience the 30% overvoltage the first time. This response is shown in Figure 6.10.(b). The SPOV operation can also be confirmed by investigating the flags `Protection.OC` and `Vbn.max` of the controller hardware, both of which were observed to be changed to 1, after running this test with SPOV `enabled`.

6.4 Summary and Analysis of Experimental Results

6.4.1 Impact of TrOV in IBRs

The experimental results summarized in section 3 establishes that inverter based DER systems are less prone to TrOV compared to synchronous generators. However, for systems with higher GLR, the overvoltage (more specifically, LROV) can still be significant. The overvoltage effect would also be prominent for IBRs supplying loads of lower power factors.

Crucially, it was observed that conventional grounding transformers can **not** mitigate these overvoltages properly. This is because, the introduction of GTFs increase the negative sequence voltage slightly, in spite of reducing the zero sequence voltage by manipulating the zero sequence impedance. This is illustrated in Figure 6.11 and Figure 6.12. Additionally, with the rising GLR, there is significant increase in TrOV due to the LROV phenomenon (which originates from the positive sequence circuit). To construct the phasor diagrams presented in Figure 6.12, the instantaneous values of currents and voltages at t=0.65s were used.

6.4.2 Importance of SPOV Settings in IBRs

Building upon the previous observation, it can be stated that an inverter operating under a high GLR, remains vulnerable to TrOV (due to LROV and GFOV) under SLG conditions even if a grounding transformer is utilized in the system to mitigate overvoltages. However, based on the results shown in Figure 6.10.(a)-(b) it can be hypthesized that the IEEE 1547-2018 recommended SPOV settings are extremely useful in protecting inverter based DERs from such TrOV. Table 6.7 illustrates the effectiveness of appropriate SPOV settings to mitigate maximum overvoltages in IBRs. It demonstrates the effect of setting a lower (i.e. 1.3) value of SPOV while the IBRs are providing (i) a load with 50% share of Yg load and (ii) a load with 100% share of Yg load. In this table, the observations in 'red' exceed the acceptable limit of maximum overvoltage according to the IEEE 1547-2018 std. These overvoltages were mitigated by simply setting a more restrictive SPOV value of 1.3.

Apart from the maximum overvoltage constraints, IEEE std 1547-2018 also proposes predetermined upper-limits for the *cumulative overvoltage durations* for certain discrete over-

Table 6.7: **Impact of (i) SPOV Settings and (ii) Percentage of Y-connected Loads- on the Maximum Overvoltage**

GLR	Overvoltage with 100% Yg load		Overvoltage with 50% Yg load	
	SPOV=1.3	SPOV=1.625	SPOV=1.3	SPOV=1.625
0.6	0.67	0.67	0.96	0.96
0.7	0.78	0.78	1.14	1.15
0.8	0.88	0.88	1.23	1.23
0.9	0.99	0.99	Unsustained	1.28
1.0	1.09	1.09	Unsustained	1.32
1.1	1.2	1.18	Unsustained	1.49
1.2	1.3	1.3	Unsustained	Unsustained
1.3	Unsustained	1.412	Unsustained	Unsustained
1.4	Unsustained	1.518	Unsustained	Unsustained
1.5	Unsustained	1.6	Unsustained	Unsustained
1.6	Unsustained	Unsustained	Unsustained	Unsustained

voltage levels (from 1.3 pu to 2 pu). Figure 6.16 demonstrates that without the SPOV settings enabled, the cumulative overvoltage-duration for the observed voltages fall outside the acceptable operating region when the system is subjected to SLG. The estimation for cumulative overvoltage duration can be carried out by using the relationship $T_{cumulative_{1.3}} = N_{cycles} \times T_{1.3}$ where N_{cycles} is the number of cycles the inverter can sustain itself post-breaker operation, and $T_{1.3}$ is the amount of time in each cycle during which the instantaneous voltage of the unfaulted phase is over 1.3 pu. It is to be noted that, for experiments where the maximum overvoltage is over 1.4 pu, both $T_{cumulative_{1.3}}$ and $T_{cumulative_{1.4}}$ need to be computed. Similarly, for experiments where the maximum voltage was over 1.5 pu, three cumulative times $T_{cumulative_{1.3}}$, $T_{cumulative_{1.4}}$ and $T_{cumulative_{1.5}}$ need to be calculated. The observations presented in Figure 6.16 establishes that a restrictive SPOV setting is crucial to protect IBRs from *cumulative overvoltage duration* violations.

6.5 Conclusions and Future Works

Real-time CHIL simulation facilitates understanding of how modern inverter controls interact with conventional power grids without the need of traditional modeling assumptions used in traditional off-line studies or to perform field experiments. This paper presented CHIL simulation results for SLG faults in a distribution network with grid connected PV-inverters, and demonstrated that the GFOV in such systems is much less severe than

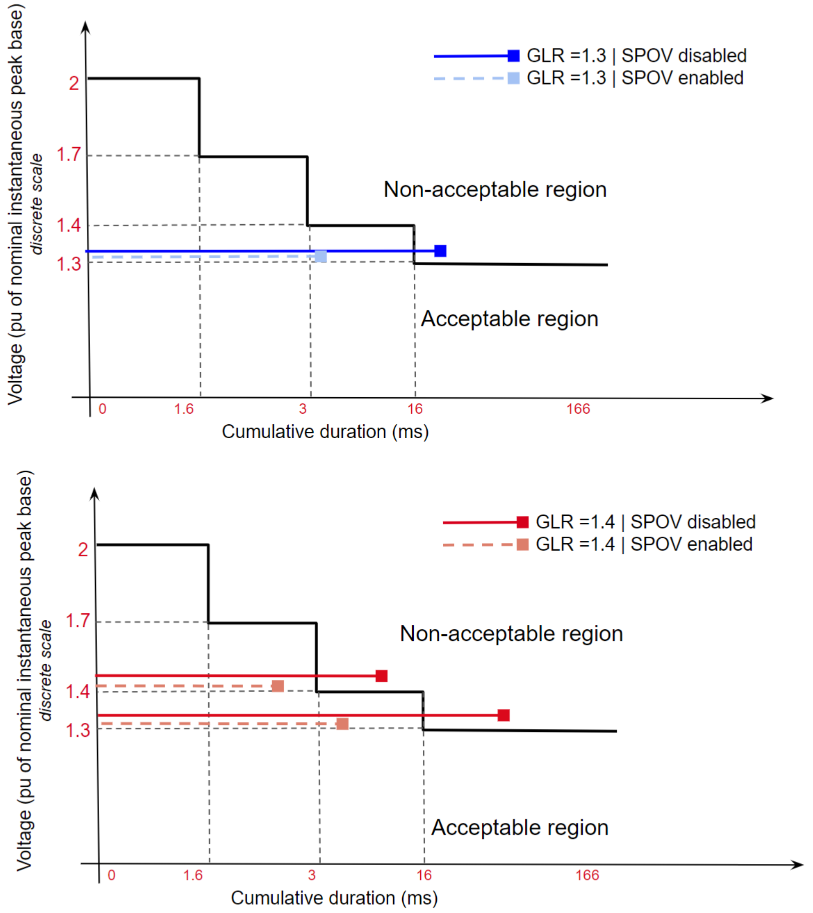


Figure 6.16: Cumulative Overvoltage Duration Calculation (a) GLR=1.3 (b) GLR=1.4

they are when conventional synchronous generators are used. It was also demonstrated that overvoltage becomes more severe at higher generation to load (GLR) ratios. A standard GFOV mitigation technique, i.e. grounding transformers (GTF), was applied to the system and only a marginal improvement was observed in terms of overvoltage reduction. Detailed sequence component analysis was performed and it was observed that the resulting overvoltage observed was generated predominantly from the negative sequence network. Since, GTFs change the zero sequence network, they were unable to completely mitigate the overvoltages. However, since the magnitude of the over-voltage was small for standard operating conditions, the grounding needs are much lower than those of synchronous generators. Finally, the impact of the controller's SPOV was analyzed, showing that this mechanism will disconnect the inverter when an over-voltage occurs.

The most important conclusion of this research is that, grounding transformers can

not prevent transient overvoltages for IBRs under SLG fault. Utilities are still reluctant to abandon this practice due to a lack of sufficiently-convincing information, which the authors believe is leading to increased capital expenditures in the interconnection of new inverter-based plants. The analysis identified how a supplemental ground source e.g. GTF, will not completely eliminate a GFOV condition as they can only reduce the zero sequence impedance of the system. In systems with higher GLRs, both the zero sequence and the negative sequence network contributes to the overvoltages on the unfaulted phases. The contribution from the negative sequence circuit is not compensated by the GTF, and thus it dominates the overvoltage phenomenon, especially at the higher GLRs. Grounding transformers can also lead to negative impacts on distribution circuits, such as desensitizing the utility ground relaying, increased arc flash energy and blinding of the DERs to single phase open circuits. In addition, without the supplemental ground source (i.e. the GTF) on the utility system, the negative impacts will not be present which include issues with single phase open detection and the desensitization of utility ground relaying. Thus, there are tangible positive consequences of not using the GTFs within the network.

The results of this paper support the hypothesis that the SPOV function will mitigate excessive overvoltages during a ground fault on a distribution circuit when installed with a wye-ground / wye-ground interconnection transformer. The same function will also mitigate load rejection overvoltages (LROV).

The experiments reported in this paper were performed only in a controller-hardware-in-the-loop (CHIL) configuration. Thus, it is advisable to perform similar experiments in **power**-hardware-in-the-loop setting with industry-grade inverters before proposing actual modifications to the IEEE standards. These experiments would require a safe laboratory space and personnel experienced in operating such high-voltage equipments. It is important to take into account that, the current paper only reports experiments with IBRs feeding Y-connected loads. In reality, Δ configurations are occasionally used in distribution level networks. Hence, it is important to validate the protective features of inverters for such connections. These experiments were beyond the scope of the current paper, and they require further exploration.

CHAPTER 7
ADVANCED COMPONENT MODELING: LI-ION BATTERIES

Today's power generation fleet is becoming more diversified, including both traditional energy sources and a variety of renewable energy sources. To help operate such a wide variety of energy sources reliably and in harmony, Energy Storage Systems (ESS) provide an alternative that offers a high degree of flexibility. To understand their behaviour and exploit their flexibility, modeling and simulation of ESSs is crucial in using the ESSs to solve the challenges introduced by the increased usage of photo-voltaic, wind, and other renewable energy sources. Most industrial energy storage systems use batteries as the primary energy storage device. This chapter reports a simplified model for the Li-ion batteries, which can calculate the state-of-charge (SOC) and output terminal voltage, while still meeting real-time modeling constraints of different hardware platforms. To test the model's validity, its outputs for both charging and discharging modes were compared with those of an existing battery model. The accuracy and performance of the proposed model was analyzed in two different real-time hardware architectures- (i) OPAL-RT OP4520/5030 and (ii) Typhoon HIL 603.

7.1 Introduction

7.1.1 Motivation

The proliferation of solar, wind, fuel-cell and other non-conventional energy sources has given rise to new challenges in power systems operation. Energy storage systems are emerging as a potential solution to such challenges. To understand their potential and limitations, it is crucial to model and simulate such energy storage systems with accuracy. In order to perform experiments to control the charging and discharging of energy storage systems a detailed and accurate mathematical model of energy storage systems (ESS) is necessary. It has been reported in [4] that the cost of ESS have decreased drastically, making their application

- Portions of this chapter previously appeared as: X. Jia, P. M. Adhikari, and L. Vanfretti, "Cross-platform real time simulation models for Li-ion batteries in Opal-RT and Typhoon HIL," in *IEEE Texas Power & Energy Conf.*, College Station, TX, USA, Feb. 2021, pp. 1-6, doi: 10.1109/TPEC51183.2021.9384928.

economically viable, both in grid-tied and remote-systems. It is also observed that, Li-ion batteries are the most widely used energy storage devices across all such industrially available ESS. This highlights the need for fast and accurate modeling of Li-ion batteries, both for their charging and discharging modes. Therefore, this chapter reports a simplified and efficient model for Li-ion batteries, suitable for real-time simulation.

To test the accuracy of the proposed model, the work in this chapter was compared with the battery model available in the *Simulink Simpowersystems* library, both in charging and discharging modes. Because, power system operations are time-critical in nature, it is important to test the real-time compatibility of the proposed models, so that they can be used in the development, implementation and testing of different control schemes. In order to test the real-time performance of the proposed model, it was implemented on Opal-RT 4520/5030 real-time simulator, and its performance was compared with an existing model. In addition, to address the differences arising from the modeling environments of different real-time simulators, the proposed battery model has been tested on two real-time simulators in order to ensure its cross-platform compatibility. One of the critical features of the proposed model is that, it is absolutely reproducible across different hardware platforms.

7.1.2 Related Works

Authors in [90] developed models and calculation methods for extracting the parameters of an electrical battery. Then, authors of [91] implemented those equations and built a complex lithium-ion model containing five subsystems in *Matlab/Simulink*. [92] presents LiFePO4 dynamic battery modeling for a battery simulation. *Mathworks* also provides an accurate and detailed generic battery model [93], which can model four types of rechargeable batteries. This model is suitable for off-line simulations and was used as a benchmark the model in this chapter. However, the existing battery models mentioned pose some difficulties when attempting to achieve real-time simulation. The authors of [94] described a real-time battery management model of a resistance-capacitance battery, however, this model is not generally compatible with the Li-ion batteries generally used in power systems. Furthermore, none of the above models have been verified for cross-platform real-time simulation. Thus, this chapter reports a simplified cross-platform lithium-ion battery model that is suitable for real-time simulation. In [86], it was reported that the real-time simulation results can be made similar across two entirely different real time simulators (Opal RT-4520/5030 and Typhoon HIL) if meticulous care is taken in model implementation. However, minor but visible mismatches were still observed between the transient responses when the same model was simulated across two different real-time simulator hardware architectures. To make the model's response consistent in different environments, the proposed Li-ion battery model was implemented and simulated across the two different hardware architectures: **Opal RT 4520/5030 and Typhoon HIL 603**.

7.1.3 Contributions

- A model to emulate the behaviour of a Li-ion battery system is proposed. The model was implemented in Simulink and the simulation results are compared with the benchmark battery model archived by the *Mathworks*. The proposed model can also compute the state-of-charge of the battery in real time

- Once the model was verified, it was prepared for real-time simulation on an Opal-RT target. Its performance was compared with the benchmark model, when run in real time. For this part, RT-LAB software was used.

- The proposed model was carefully migrated to Typhoon HIL family of real time simulators to ensure reproducibility and cross-platform compatibility. A sample test-case was implemented in Typhoon, where the battery operates as part of a microgrid. This implementation ensures the usability of the proposed battery model for simulating real-life applications.

- The models are made available as open source software in the GitHub repository : `github.com/alsetlab`

7.2 Theory

7.2.1 Review of the Real-time Hardware Platforms

The two Real Time simulator hardwares used for this analysis were (i) Opal-RT OP4520/5030 and (ii) Typhoon HIL 603.

The OP4520/5030 uses the OPAL-RT RT-LAB software and the Artemis real-time solver, along with the standard *Simulink* library. The Typhoon HIL platform uses a separate software package, *Typhoon HIL Control Center*, which has its own component libraries and can compile standard C programs to generate block level functionalities. A short summary of the two hardware platforms is given in table 7.1.

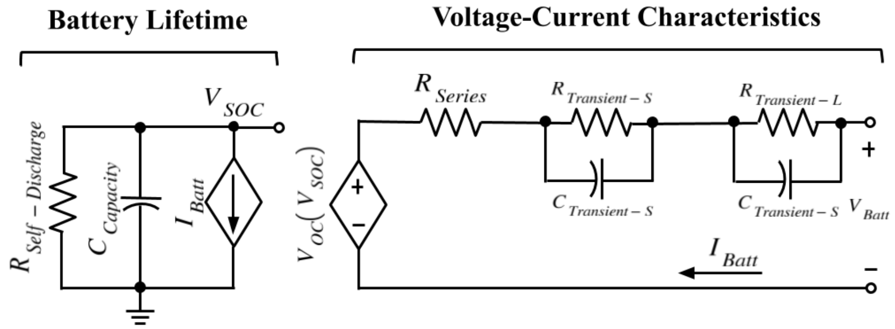


Figure 7.1: The Existing Comprehensive Battery-Model

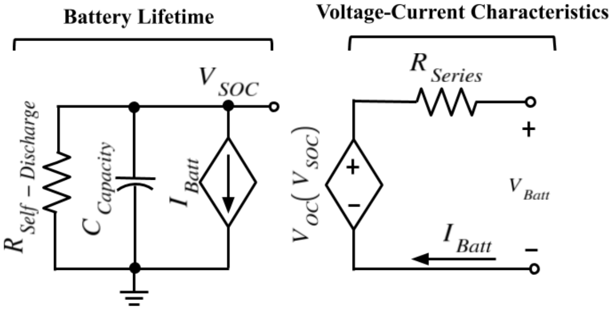


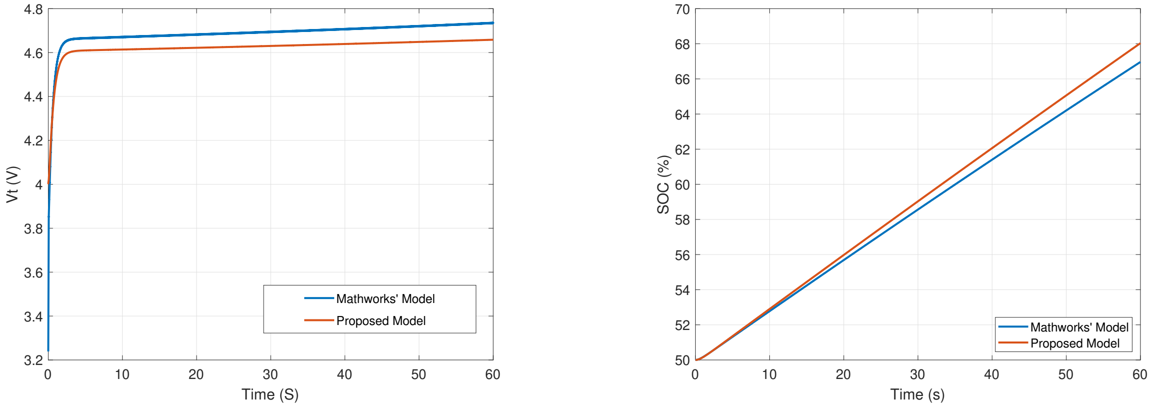
Figure 7.2: Proposed Simplified Equivalent Circuit Model for a Battery

Table 7.1: Real Time Hardwares: A Comparison of Features

RT Hardware	Opal RT 4520/5030	Typhoon HIL 603
Processor	Intel Xeon Quadcore 32 cores	ARM R 8x2 cores
FPGA	Xilinx Kintex 7	Xilinx Virtex 6
PCIe Connection	1-7 PCIe	8-Lane; 2 PCIe 4x

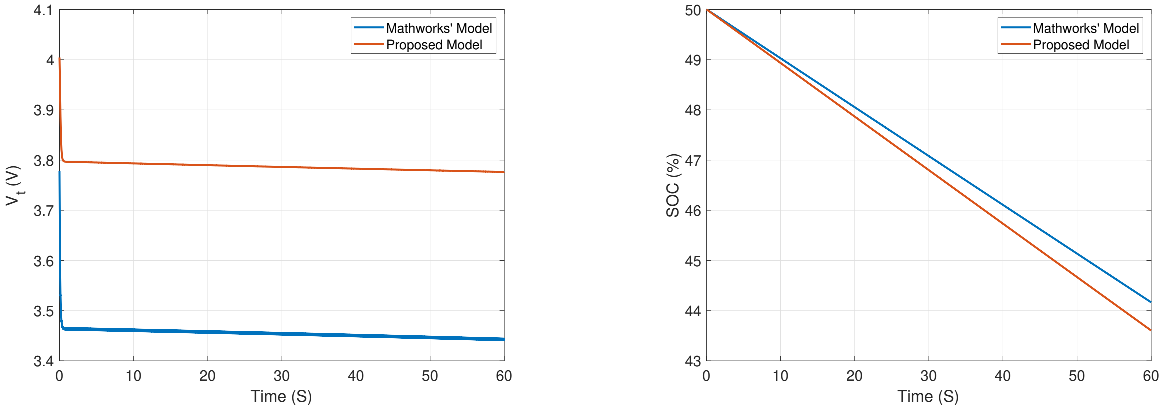
7.2.2 Description of the Proposed Model

As mentioned earlier, [91] reports an accurate and comprehensive model, which constructs a very detailed circuit to represent the behaviour of a battery. The model includes a capacitor $C_{capacity}$ and a discharge resistor $R_{self-discharge}$ in parallel with a current controlled current source. This part of the circuit effectively models the behaviour of the State-of-charge (battery lifetime), and the resultant voltage-parameter V_{SOC} is used to construct the remainder of the model. The remainder of the model contains an RC network that follows the same modeling rationale as the models reported before in [91], [92]. This RC network is parameterized based on the transient V-I response of the battery-system. The overall model is shown in Figure 7.1. A voltage controlled voltage source which relates an equivalent *Thevenin Voltage* to the State-Of-Charge is connected in series with this RC network. However, it was observed that the contribution of this RC network is negligible for Li-ion



(a) Terminal Voltage Variation **(b) SOC Variation**

Figure 7.3: Charging Mode: Performance Comparison Between the Proposed Model and the Benchmark Model



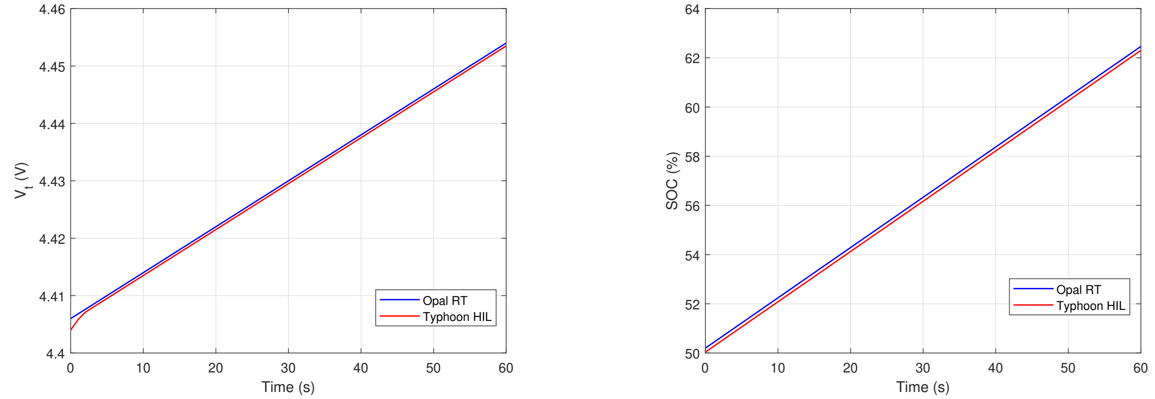
(a) Terminal Voltage Variation **(b) SOC Variation**

Figure 7.4: Discharging Mode: Performance Comparison Between the Proposed Model and the Benchmark Model

([90],[91]) batteries.

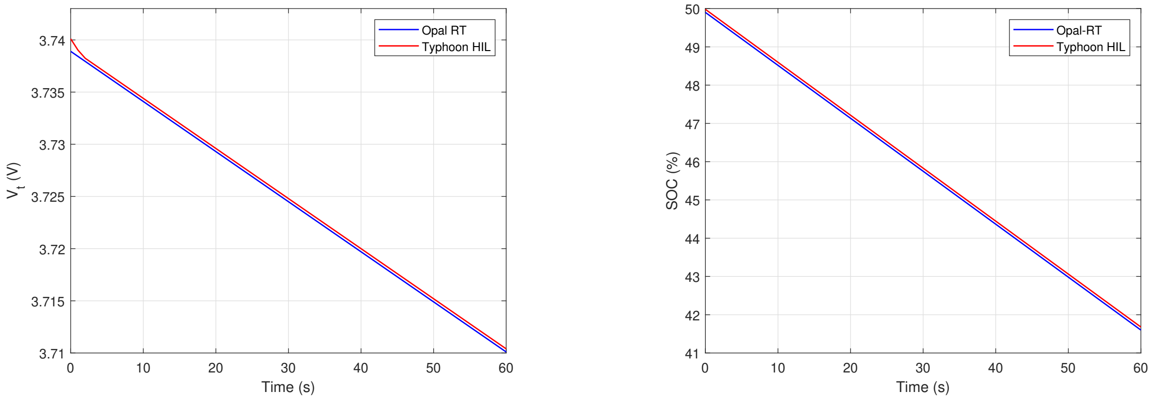
The observation above indicates that any potential transients due to the RC constant will be of ultra-fast nature, which would inherently limit the time-step for real-time simulation. Hence, assuming that their effect is minimal, the model proposed in this work replaces the RC components with a single series resistance R_{series}. This simplified model is shown in Figure 7.2.

This model is expected to emulate three important dynamic characteristics of a battery system.



(a) Terminal Voltage Variation **(b) SOC Variation**

Figure 7.5: Charging Mode: Terminal Voltage and SOC Variation For the Proposed and Benchmark Models



(a) Terminal Voltage Variation **(b) SOC Variation**

Figure 7.6: Discharging Mode: Terminal Voltage and SOC Variation For the Proposed and Benchmark Models

- Capacity: Usable capacity is the total extracted energy from the battery. It decreases with increasing discharge current, storage time and number of cycles under operation. It is also observed to increase when the ambient temperature decreases. The capacitor $C_{capacity}$ together with the resistors $R_{Self-discharge}$ and R_{series}. $C_{capacity}$ does not depend on current variations. However, the effective SOC depends on current variations, because a variation in current leads to a variation in the voltage drop across the R_{Series}. The $R_{Self-discharge}$ resistor, on the other hand, models the discharge of the battery when it is stored idly for a long time.

- Open Circuit Voltage: $V_{OC}(V_{SOC})$ is a function of V_{SOC} as it depends on the state-of-

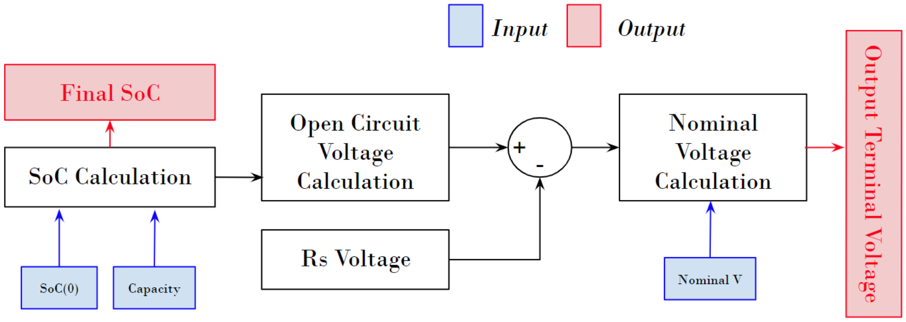


Figure 7.7: Flowchart of the Proposed Model Implementation

the-charge. Open-circuit voltage V_{OC} is the steady-state terminal voltage at a certain SOC value.

- Transient V-I Response: The circuit parameter that determines the transient response is R_{Series}. This assumes that the impact of very short time constants on the response of the battery-system is negligible as per [91].

The battery model proposed in this chapter was implemented in *Matlab/Simulink* consists of several parts, as shown in Figure 4, the main ones being the State-Of-Charge (SOC) calculation and the output terminal voltage calculation. The three input parameters of the model are Initial SOC (SOC_0), Rated Capacity, and Nominal Voltage. The two outputs are Final SOC (SOC_n) and Terminal Voltage (V_t). The SOC calculation algorithm is based on the *Matlab-Simulink* model as reported in [93]. After updating the final state-of-charge from (SOC_0), the new value of SOC_n is used for computing the value of open-circuit voltage (V_{OC}). This computation is based on the following mathematical expression, first reported in [90]:

$$V_{OC}(SOC) = -1.031e^{-35\times SOC}+3.685+0.2156\times SOC-0.1178\times SOC^2+0.3201\times SOC^3 \quad (7.1)$$

The value of SOC is also used to estimate the value of the resistance R_{Series} using the following expression:

$$R_{Series}(SOC) = 0.1562 \times e^{-24.37\times SOC} + 0.07446 \quad (7.2)$$

Finally,the terminal voltage $V_t = V_{OC} - R_{Series} \times I_{Battery}$, as seen from the circuit in

Table 7.2: Performance Characterization of the Proposed Battery Model in Opal-RT Platform

	Mathworks' Benchmark Model	Proposed Model	Change
Charging Mode			
SSN: State Space Operation Count	63	71	+12.6%
SSN: Memory Usage	0.00202 Mb	0.00228 Mb	+12.8%
Computation Time	0.7096 us	0.9496 us	+33.8%
Discharging Mode			
SSN: State Space Operation Count	45	51	+13.3%
SSN: Memory Usage	0.001448 Mb	0.00164 Mb	+13.1%
Computation Time	0.6609 us	0.9565 us	+44.7%

Figure 7.2.

7.3 Results

7.3.1 Model Verification and Characterization

It has already been discussed that, in order to validate the proposed model, it must be shown that:

- The proposed battery model performs satisfactorily both under charging and discharging scenarios.
- Both the responses of the terminal voltage V_t and the state of charge SOC are accurate.

Hence, the responses of the proposed model for V_t and SOC under both charging and discharging modes were compared with the respective responses of the *Mathworks*' benchmark model. These results are illustrated in Figure 7.3.

Observe that the terminal voltage suffers the largest discrepancy, which is due to a high value of R_{Series}, this would need to be calibrated to better match the benchmark's results. Meanwhile other responses expose similar behaviour with minor deviations from the benchmark.

Having determined that the proposed model has sufficient accuracy, its efficiency (in terms of hardware consumption and speed) needs to be assessed. For this, its hardware consumption for real time implementation (on Opal RT) is compared to that of the *Mathworks*' model. This comparison is summarized in table 7.2. It can be seen that the proposed model

Table 7.3: **Memory Utilization to Simulate the Battery Model in Typhoon HIL 603**

Memory Type	Used Memory	Available Memory (kb)	Utilization
Internal Memory	48	256	19.04%
Code Segment Size	38	256	14.94%
Data Segment Size	10	256	4.10%

consumes a small additional amount of hardware resources and its speed is slower than the *Mathworks*' model. However, it needs to be noted that the proposed model does not use any closed and/or proprietary library-specific functions and depends only on simple mathematical operations. This makes this model suitable for migration to a different platform, as shown next for case of the Typhoon HIL-603 simulator.

7.3.2 Simulation in Typhoon HIL Platform

In order to test the cross-platform compatibility of the proposed model, it was re-implemented in the Typhoon HIL real time simulator hardware. Details of this implementation including a detailed flow chart is provided in the `github.com/alsetlab` repository.

In this case, a Typhoon HIL 603 real-time simulator was used. Because only fundamental mathematical functions and numerical operations were used to construct the model, it was completely independent of specialized *Matlab/Simulink* library functions. Thus, it was expected that the Typhoon HIL real-time simulation results would be identical to those

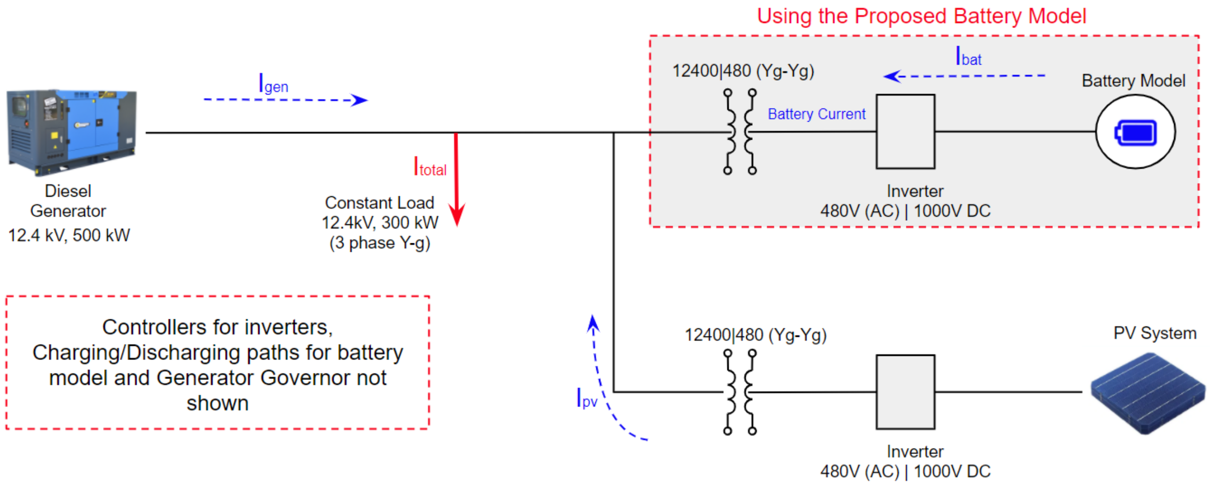

Using the Proposed Battery Model
I_{gen}
I_{bat}
12400|480 (Yg-Yg)
Battery Model
Battery Current
Diesel Generator 12.4 kV, 500 kW
I_{total}
Constant Load 12.4kV, 300 kW (3 phase Y-g)
Inverter 480V (AC) | 1000V DC
Controllers for inverters, Charging/Discharging paths for battery model and Generator Governor not shown
12400|480 (Yg-Yg)
PV System
I_{pv}
Inverter 480V (AC) | 1000V DC

Figure 7.8: **Microgrid Implementation to Validate the Battery Model**

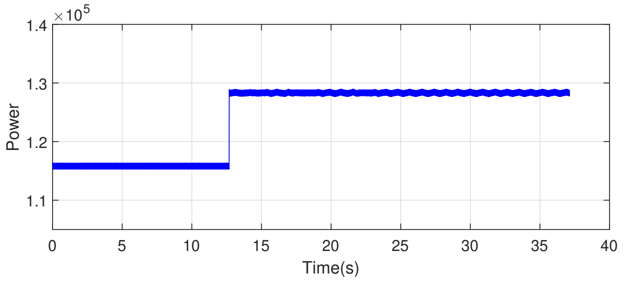


Figure 7.9: Step Increase in the Reference Active Power for the Battery Unit

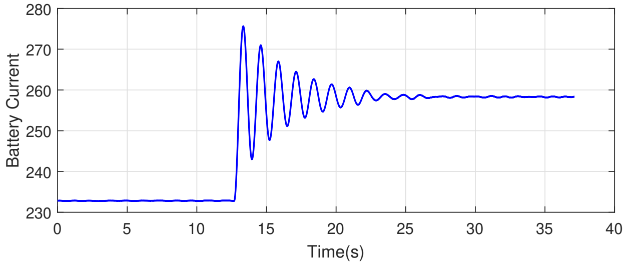


Figure 7.10: Output Current From the Battery Unit

observed in the Opal-RT platform. The responses that are summarized in Figure 7.5 affirm this hypothesis. It can also be concluded that for both in the charging and discharging modes, the proposed model is consistent in both platforms, i.e. compatible between Opal-RT and Typhoon. During compilation and execution on the Typhoon platform, the memory consumption by this model that are summarized in table 7.3.

7.3.3 Test Case: Application of the Battery Model

A sample case where the proposed battery model was utilized in a microgrid application is presented here. This microgrid contains a diesel generator (which is part of Typhoon's proprietary library [95]) and a PV system utilizing the model reported in [96] alongside the battery model presented in this chapter. The simulation was carried out in the Typhoon HIL 603 real-time simulator. The proposed battery model was linked to an inverter designed for connecting Li-Ion battery units to AC-grids. The details of this inverter is reported in [97]. The block-level representation of this microgrid is shown in Figure 7.6. However, a discussion on the detailed modeling of the individual components of this microgrid is beyond the scope of this chapter.

While simulating the microgrid, the battery was subjected to a load increase of 12.5 kW as shown in Figure 7.7. The battery responds to this step-change by increasing the output

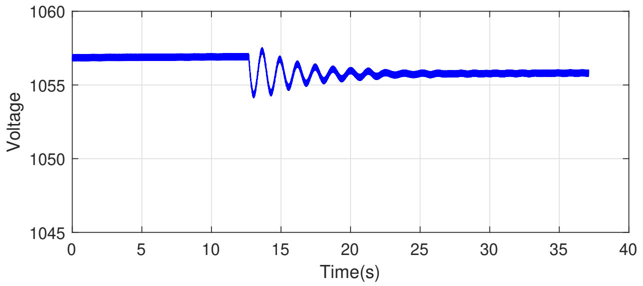


Figure 7.11: Terminal Voltage of the Battery

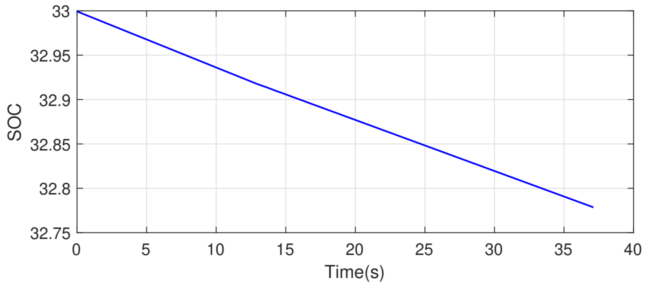


Figure 7.12: Response of the Battery-SOC

current as shown in Figure 7.8. It needs to be noted that, further studies of the *transient response* of the battery output current during this step-change, as observed in Figure 8, is beyond the scope of current work. As demonstrated in Figure 7.9, the voltage output of the battery remains relatively constant through the entire duration of this simulation. Since the battery is providing power to the microgrid, it is expected that the SOC of the battery would decline steadily, which is confirmed by the simulation results demonstrated in Figure 7.10.

7.4 Discussions and Future Works

It is desirable to ensure cross-platform compatibility and model response consistency in order to minimize the uncertainty in the simulations across different hardware acrhitectures. In other domains of science and engineering, modern interoperability standards such as *Modelica* and the *FMI* (Functional Mock-up Interfaces) are already in place. However, these standards are only adopted in very few power system tools [86]. The work in this chapter helps to motivate further the need for model portability using the Modelica and FMI standards for power system simulation.

It is reported in [86] that the library coverage for Simulink and Opal-RT is larger than that of other simulation platforms e.g. Typhoon HIL. It was also observed in [86] that the existing library models for similar components across these two different infrastructures can give different results. Thus, in order to obtain compatible simulation models for batteries, it is necessary to build them from the scratch. It is also highly recommended that, only simple mathematical and numerical functions are to be used to construct new models, so to limit platform dependencies.

The current work establishes that simplified simulation models for Li-ion batteries are portable across different real-time simulation platforms if proper implementation is made. figure 7.5 exhibits that the proposed model was able to generate identical simulation results across the two different platforms. To ensure this, the charging and discharging circuits used for experimentation was kept simple and minimal. This enabled a identical testing environment for the proposed model, across the two hardwares. It is expected that when the charging/discharging paths get more complicated, the proposed model's cross-platform performance may deteriorate.

A preliminary demonstration of the usage of this battery model is shown in the context of a simple microgrid model. However, this part of the work needs a lot of further analysis and studies to understand how the battery model responds to changes applied to different blocks of the microgrid. This analysis is crucial for the validation of the proposed battery model.

7.5 Conclusions

This chapter reported a simplified cross-platform compatible real-time simulation model for Li-ion batteries. The model, when implemented on Opal RT real-time simulator, is competitive in terms of accuracy and efficiency, while compared to a closed-source proprietary battery model. The proposed model was then migrated to the Typhoon HIL real time simulator. The real-time simulation results for this model on Typhoon and Opal RT were found to be identical thanks to the implementation approach adopted in this work. The preliminary testing of this model was successfully carried out by connecting it to a microgrid system.

CHAPTER 8
PRECISION TIMING AND COMMUNICATION NETWORKING EXPERIMENTS IN A REAL-TIME POWER GRID HARDWARE-IN-THE-LOOP LABORATORY

The growing wave of digitization in power grids is bringing increased reliance on information and communication technologies (ICT) for the operation of the grid. Such cyber-physical power systems (CPPS) involve the interaction of the physics of the power grid with the performance of other engineered systems, such as communications and time-synchronization. To understand the interplay of such interactions, experimentation is required. However, there are very limited opportunities to perform experiments on *actual* CPPS systems, due to safety and security concerns. Nevertheless, the demand for more functionalities on cyber components will continue to rise, and thus, other means to understand CPPS behavior for design, implementation and testing are needed.

To address this gap, a real-time simulator-based power system laboratory was implemented with the objective of facilitating experiments involving precision timing and communication networking systems which are coupled with power grid models running in real-time. These are integrated in three layers: (a) Precise Timing Layer, (b) Communication/Network Layer, and (c) Electrical Component Layer. This chapter reports on detailed experiments performed on the precision timing and communication layers of the laboratory. It shows how to couple the different layers together, and how to conduct experiments to tamper with both the precision timing and communication layers, along with their interactions with the simulated grid. Finally, the chapter shows how to validate the Quality of Service (QoS) rules implemented in virtual local area networks (VLANs) in the laboratory environment when

• Portions of this chapter previously appeared as: P. M. Adhikari, H. Hooshyar, R. J. Fitsik, and L. Vanfretti, “Precision timing and communication networking experiments in a real-time power grid hardware-in-the-loop laboratory,” *Sustain. Energy Grids & Netw.*, vol. 28, Dec. 2021, doi: 10.1016/j.segan.2021.100549.

• Portions of this chapter previously appeared as: H. Hooshyar, L. Vanfretti, J. H. Chow, R. Fitsik, P. M. Adhikari, “ALSET lab: Designing precise timing and communications for a digital power grid laboratory,” in *IEEE Power & Energy Soc. General Meeting*, Montreal, QC, Canada, Aug. 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281808.

using different power system communication protocols.

8.1 Introduction

This section discusses the motivation behind this research and presents a brief literature review. Additionally, it enlists the specific contributions reported in this chapter. Section 8.2 reviews the architectural details of the laboratory. Section 8.3, describes the experiments performed to demonstrate the cyber-physical interactions in the different layers of the laboratory, and summarizes the observations and results obtained from those experiments. The chapter concludes with a summary of the current and prospective experimental research suitable for this kind of laboratory infrastructure.

8.1.1 Motivation

The digitization of power grid coupled with the penetration of various distributed non-conventional energy resources has led to a different paradigm of how a power system operates. This new paradigm reveals new challenges in the fields of monitoring, protection and communication in a power network. This has required the development of new capabilities on the "cyber" assets to address these challenges, for example, networking and controlling inverter-based resources (IBRs). These infrastructures has been standardized in [98]. However, there are limited opportunities to test these new functionalities before they are deployed into the power grid, which in turn limits the understanding of their interaction and impact on the grid's operation.

To verify and validate new technologies requires to perform tests in laboratory conditions, researchers and engineers need elaborate testing tools that would be able to mimic the behaviour of a power system in real-time. This requirement makes the utilization of various real-time simulators (e.g. Opal-RT, RTDS, Typhoon HIL) relevant in context of power systems research as discussed in detail in [57]. While these high-end computing devices emulate the physical behaviour of the power system, the extensive data communication network of a modern power system needs to be implemented or emulated separately [99]. The analog or digital signals generated through the real-time simulators need to be transmitted through this network following specific protocols, like IEEE C37.118[40] and IEC 61850[100], in order to run experiments that would emulate a modern interconnected digitized power system. This makes the network configuration associated with power system experimentation crucial for the validity of the experiments.

With the electrical layer (consisting of real-time simulators) emulating power systems in real-time, it is important to illustrate how those simulation results can be interfaced, monitored and analyzed throughout the network. Moreover, in order to validate industry-grade physical equipment (e.g. substation equipment such as digital relays [101]) within a laboratory environment, all simulations, analog signals and measurement data need to carry precise timing information as demonstrated in [102]. This chapter aims to illustrate these different infrastructures are brought together within a fully functional digital power grid laboratory, and more importantly, how the precision timing and communication networking systems can be safely and lawfully "tampered" with by proposing and performing real-time experiments.

8.1.2 Related Works

Simulation labs for power system research are becoming ubiquitous among major research institutions. Some important examples of such laboratories were reported in [99], [103], [104], [105], [106], [107]. However, the focus has largely emphasized the capabilities of these implementations on grid simulation with some communication networking, leaving the precise timing infrastructure outside of their design considerations. Precision timing becomes critical when dealing with applications that require time-synchronization, such synchrophasor data applications and protective relaying. As reported in [108], the ALSET laboratory implementation features a separate timing layer into its hierarchy, making the implementation suitable for real-time experimentation on time-critical applications. This approach is similar to the one proposed in [109], where the authors hypothesized a functional block based architecture for validating and evaluating smart grid functionalities. On a similar note, the authors in [110] proposed a software driven flexible testbed for these applications. The architecture explored in the current chapter was directly influenced by the Smart Grid Architecture Model (SGAM) proposed in [111], and it can be visualized along three distinct layers: (a) Precise Timing Layer, (b) Communication/Network Layer, and (c) Electrical Component Layer. More details on the specifications and implementations of these layers are illustrated on Section 8.2.

In related works, authors of [106] reported a full-scale laboratory implementation featuring synchronous generators, physical loads (both passive loads and active loads) and converters. This implementation also incorporated a high-power hybrid DC-AC type power system, for testing various smart grid applications. While this research made significant contributions towards the development of a standalone power systems laboratory with a fully operational networked supervisory control and data acquisition (SCADA) system, the network implementation was not tested for various relevant power system communication protocols and precision timing. Built upon the implementation reported in [108], the current work specifically focuses on the protocols, networking and timing infrastructures of a power system research laboratory. Thus, most of the efforts were targeted towards the development and verification of fully operational time-sensitive network suited for smart-grid instrumentation.

As reported in [112], the future smart control centers would be able to carry out data-driven analysis over networked devices to apply control actions on various power system components. To realize such an infrastructure in a laboratory environment, the network implementation must be robust and it must support various networking protocols without any hardware reconfiguration. The current work reports such an implementation and presents some relevant test-cases which utilize those networking protocols. Additionally, with the introduction of Phasor Measurement Units (PMUs) and synchrophasor technologies in the power system, incorporation of a precise timing source have become a critical infrastructure for power system monitoring and protection. To incorporate this into any cyber-physical power system laboratory, the devices and network within the laboratory environment must support the IEEE 1588 compliant Precise Time Protocol (PTP) as recommended in [113]. The survivability and compliance-testing performed in [114] also reported a time-synchronized interconnection. The GPS facilities incorporated within a power network needs to be *secured* in order to ensure immunity against malicious external attacks as reported by the authors in [115]. Such malicious attacks were studied in details, in the context of a real-time CPS, by the authors in [116] and [117]. To tackle these issues, the research presented in [118] proposed spectrum-sharing for wireless communication enabled smart grid functionalities. However, because the proposed implementation in ALSET lab utilizes wired ethernet connections instead of wireless networks, it has a comparatively safer communication infrastructure. Apart from PTP and synchrophasor (C37.118), the power system specific data transmission protocols, that are of crucial importance for this current chapter are IEC 61850 (GOOSE and SV).

As reported by the authors in [119], the IEC 61850 can be utilized to transmit seven different types of messages, with varying speed. The fastest `type` of message-Generic Object Oriented Substation Events (GOOSE) messages, which utilizes the IEC 61850 protocol for message transmission, suitable for reporting substation events such as faults. On the other hand, Sampled Value (SV) data transmits continuous voltage and current measurements through the same IEC 61850 protocol, but at a lower speed. The C37.118 protocol, on the other hand, is dedicated for transmitting time-stamped data generated through phasor-estimation algorithms at a fixed reporting rate (i.e. 50/60, 100/120 packets per second). The proposed laboratory implementation supports all these protocols, including PTP, GOOSE, SV, Synchrophasor and still keeps a provision for generic TCP/IP based data transfer operations within the network. Authors in [120] and [121] presented significant architectural details of similar setups from the perspective of communication engineering. The study presented in [122] explored the quality-of-service rules for the communications setups under similar configurations.

In terms of experimentation, the large majority of the software used in real-time simulation labs consists of proprietary tools such as the ones reported by the authors in [57] and [99], both for device configuration as for modeling and simulation. In this work, devices are configured with software provided by SEL (*Quickset* and *Architect*), Opal-RT (*RT-LAB*), and National Instruments (*LabVIEW*). However, to conduct the research in this work, other developed within the lab were necessary. One such tool was *Khorjin* [123] which provides a IEEE C37.118.2 to IEC 61850-90-5 mapping and transformation for real-time synchrophasor data transfer. This software can be used to implement a time-synchronized hardware-based *Synchrophasor Synchronization Gateway* that can ingest and process real-time synchrophasor data from multiple PMU-streams as reported by the authors in [124]. In this research *Khorjin* and *SSG* were utilized to trace the latency of real-time synchrophasor data streamed by various protection devices connected in the ALSET lab communication network. All these software requirements are also summarized in Table 8.2. This summary provides a useful benchmark which can be compared against the similar surveys published in [125].

8.1.3 Contributions

The main contributions of this chapter are:

- To present how an SGAM-defined architecture has been implemented for three fundamental layers.

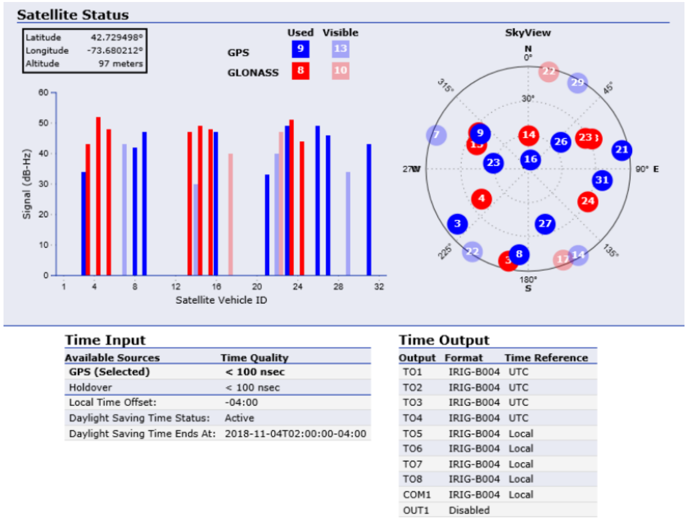
- To provide examples of experiments designed to illustrate how the electrical layer, the network layer and the timing layer interact with each other while running tests on a power system model. The results of those experiments are presented therein.
- To demonstrate how a precision timing network can be safely and lawfully impaired and how the IEDs respond to timing tampering.
- To demonstrate how a data communication-network was impaired and how the IEDs respond to such impairments.
- To show how delay tracing can be performed using a time-synchronized hardware-based *Synchrophasor Synchronization Gateway* which utilizes a GPS source present on-board for timing analysis.
- To report on experimental tests performed to demonstrate the different virtual LAN (VLAN) networks' operation.
- To demonstrate how Quality of Service (QoS) rules are implemented on VLANs and to propose experiments to validate them.

8.2 Review of the ALSET Infrastructure

The electrical component layer of the ALSET lab infrastructure contains IEDs, real-time simulators and controllers with Ethernet connectivity. This layer is self-explanatory as there were no additional ALSET-specific configurations required for this layer. Thus, this section only reviews and summarizes the precise timing layer and the communication/network layer.

8.2.1 Review of the Precise Timing Layer

The architecture for the timing layer of the ALSET lab is illustrated in Figure 8.2. The four antennas are placed on the roof of the building, which receive both Global Positioning System (GPS) and Global National Navigation System (GNNS) signals and forward it them to Satellite Synchronized network clocks (SEL-2488), and to a 1-to-16 GPS splitter placed within the laboratory. The SEL-2488 clocks have the capability to extract precise timing information from the GNSS signal and make it available in several formats, including pulse-per-second (PPS), modulated and unmodulated Inter-range Instrumentation Group (IRIG)-B signals for the IEDs in the lab to utilize. These IEDs include the Opal-RT simulators, different SEL Relays, a SEL RTAC, and the server computer. Devices which have in-built GPS receivers (e.g. NI-cRIOs) require the raw GPS signals instead of the IRIG-B signal. Thus, they are fed direct GPS signals from the GPS-splitter. LMR-400 coax cables were used for transmitting GPS signals and RG-58 coax cables were used to transmit IRIG-B signals. All the IEDs in the laboratory were configured to receive timing information externally via their IRIG-B inputs. It may be necessary to use additional DC blockers (e.g. MCL 15542) and attenuators (e.g. BW-VX-1W54) to keep the signal level of the GPS signal within the permissible limit specified by the IED that is receiving the GPS signal. These hardware adjustments were specifically utilized for connecting NI-cRIO devices to the ALSET timing network. On the other hand, Opal-RT simulators are synchronized through Oregano Systems syn1588® PCIe network cards. Figure 8.1 demonstrates the user interface for the SEL-2488 substation clock, which displays its reception of the GPS and GLONASS signals from the satelites via the antennas. GLONASS signals were not used in the experiments reported in this chapter.



Satellite Status

Latitude	42.729498°
Longitude	-73.680212°
Altitude	97 meters

	Used	Visible
GPS	9	13
GLONASS	8	10

Time Input

Available Sources	Time Quality
GPS (Selected)	**< 100 nsec**
Holdover	< 100 nsec
Local Time Offset:	-04:00
Daylight Saving Time Status:	Active
Daylight Saving Time Ends At:	2018-11-04T02:00:00-04:00

Time Output

Output	Format	Time Reference
TO1	IRIG-B004	UTC
TO2	IRIG-B004	UTC
TO3	IRIG-B004	UTC
TO4	IRIG-B004	UTC
TO5	IRIG-B004	Local
TO6	IRIG-B004	Local
TO7	IRIG-B004	Local
TO8	IRIG-B004	Local
COM1	IRIG-B004	Local
OUT1	Disabled	

Figure 8.1: The Source of Timing Information: As Seen on the Substation Clock GUI

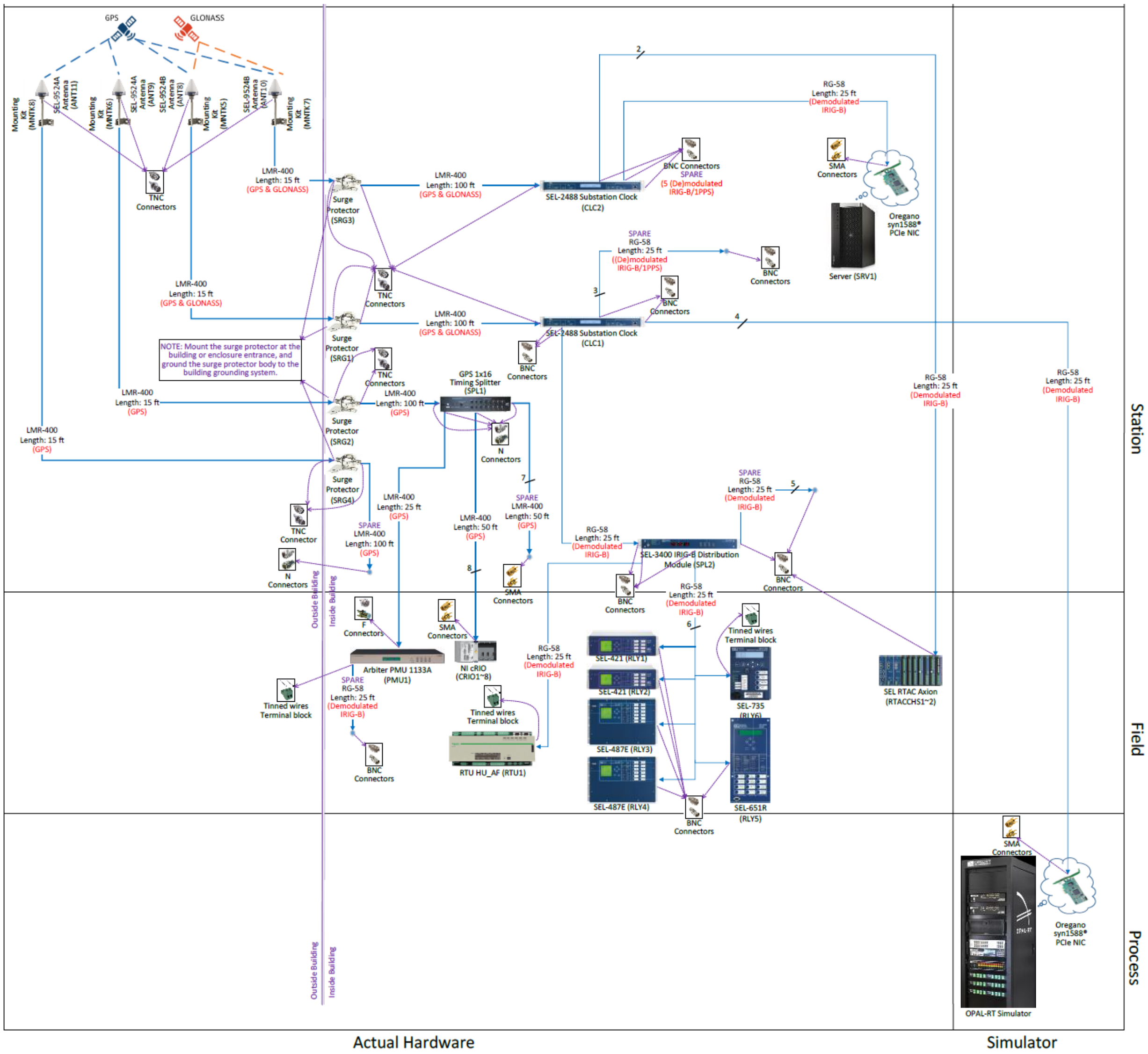

GPS
GLONASS
Mounting Kit (MNTK8)
SEL-9524A Antenna (ANT11)
Mounting Kit (MNTK6)
SEL-9524A Antenna (ANT9)
SEL-9524B Antenna (ANT8)
Mounting Kit (MNTK5)
SEL-9524B Antenna (ANT10)
Mounting Kit (MNTK7)
TNC Connectors
LMR-400 Length: 15 ft (GPS & GLONASS)
Surge Protector (SRG3)
LMR-400 Length: 100 ft (GPS & GLONASS)
SEL-2488 Substation Clock (CLC2)
BNC Connectors SPARE (5 (De)modulated IRIG-B/1PPS)
2
RG-58 Length: 25 ft (Demodulated IRIG-B)
SMA Connectors
Oregano syn1588® PCIe NIC
Server (SRV1)
TNC Connectors
SPARE RG-58 Length: 25 ft ((De)modulated IRIG-B/1PPS)
BNC Connectors
3
BNC Connectors
LMR-400 Length: 15 ft (GPS & GLONASS)
LMR-400 Length: 100 ft (GPS & GLONASS)
SEL-2488 Substation Clock (CLC1)
4
Surge Protector (SRG1)
NOTE: Mount the surge protector at the building or enclosure entrance, and ground the surge protector body to the building grounding system.
TNC Connectors
BNC Connectors
GPS 1x16 Timing Splitter (SPL1)
LMR-400 Length: 15 ft (GPS)
LMR-400 Length: 100 ft (GPS)
Surge Protector (SRG2)
N Connectors
RG-58 Length: 25 ft (Demodulated IRIG-B)
RG-58 Length: 25 ft (Demodulated IRIG-B)
LMR-400 Length: 15 ft (GPS)
Surge Protector (SRG4)
7
SPARE RG-58 Length: 25 ft (Demodulated IRIG-B)
5
TNC Connector
LMR-400 Length: 25 ft (GPS)
LMR-400 Length: 50 ft (GPS)
SPARE LMR-400 Length: 50 ft (GPS)
SPARE LMR-400 Length: 100 ft (GPS)
RG-58 Length: 25 ft (Demodulated IRIG-B)
SEL-3400 IRIG-B Distribution Module (SPL2)
N Connectors
8
SMA Connectors
BNC Connectors
RG-58 Length: 25 ft (Demodulated IRIG-B)
BNC Connectors
Outside Building
Inside Building
F Connectors
SMA Connectors
6
Tinned wires Terminal block
Arbiter PMU 1133A (PMU1)
NI cRIO (CRIO1~8)
RG-58 Length: 25 ft (Demodulated IRIG-B)
SEL-421 (RLY1)
SEL-421 (RLY2)
SEL-735 (RLY6)
SEL RTAC Axion (RTACCHS1~2)
Tinned wires Terminal block
SPARE RG-58 Length: 25 ft (Demodulated IRIG-B)
Tinned wires Terminal block
SEL-487E (RLY3)
BNC Connectors
RTU HU_AF (RTU1)
SEL-487E (RLY4)
SEL-651R (RLY5)
BNC Connectors
SMA Connectors
Oregano syn1588® PCIe NIC
OPAL-RT Simulator
Outside Building
Inside Building
Station
Field
Process
Actual Hardware
Simulator

Figure 8.2: Architecture of Timing Layer for the ALSET Lab

8.2.2 Review of the Communication Network Layer

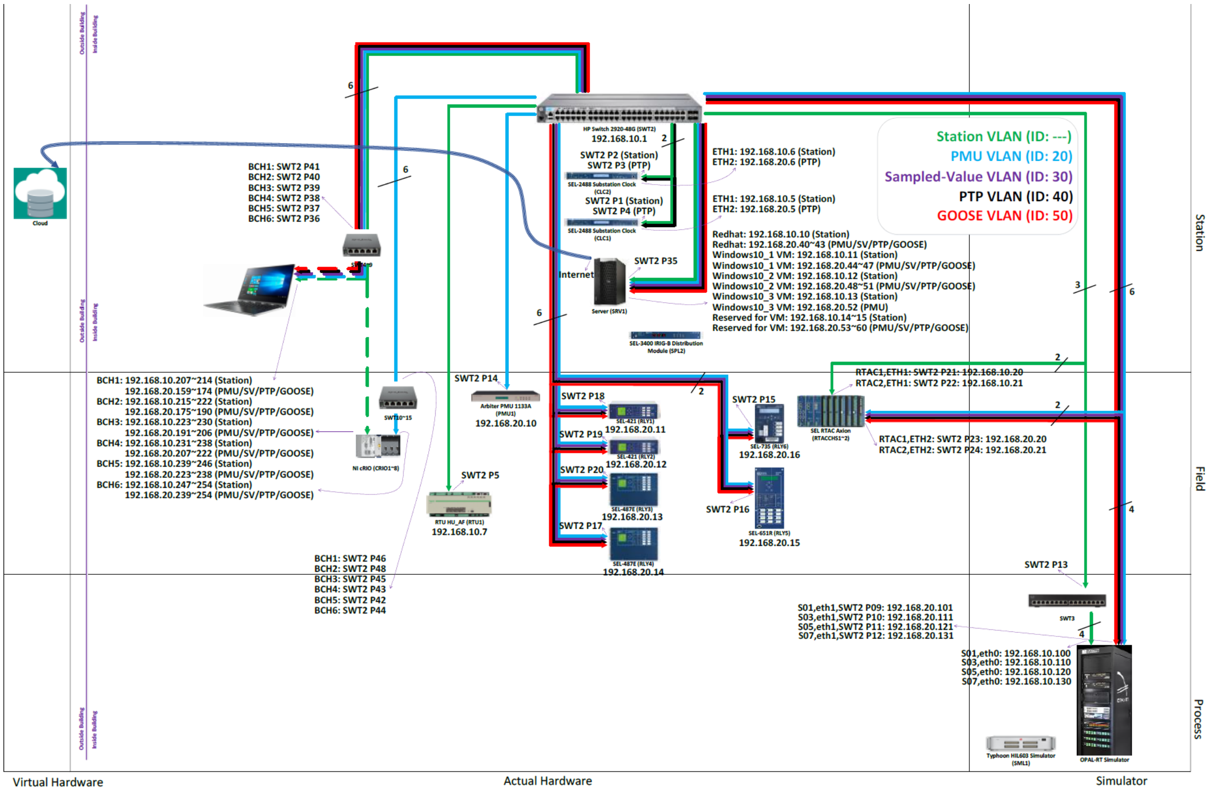
The communication network layer consists of multiple virtual LAN (VLAN) implementations which are targeted for specific types of data-transmission. This configuration ensures that the path utilized by each type of data are *virtually separated* even though they share the same physical LAN hardware (consisting ethernet cables and network-switches). Since most of these data are time-critical and the latency needs to be minimized, the VLANs were configured with two pre-defined Quality-of-Service (QoS) rules. These two rules defined the priority with which a certain data type is issued a queue and the guaranteed minimum bandwidth (GMB) of that issued queue. A summary of the specifications for the five existing

VLANs is presented in Table 8.1. The architecture of this layer is illustrated in Figure 8.3.

More detailed descriptions of the timing layer and the network layer can be found in [108].

Table 8.1: Specifications of the Existing ALSET VLANs

VLAN	VLAN ID	Priority	GMB	GMB of Switch	Application
GOOSE	50	7	1 Mbps	strict	Substation Events (Breaker/Alarm)
PTP	40	6	5.2 kbps	1%	Precise Timing Information
SV	30	4	480 Mbps	50%	Instantaneous Transmission of Current/Voltage Measurements
PMU	20	3	5 Mbps	1%	Synchrophasor Measurements following C37.118
Station	10	0	100 Mbps	10%	Generic file operations (print-jobs, model-transfer)



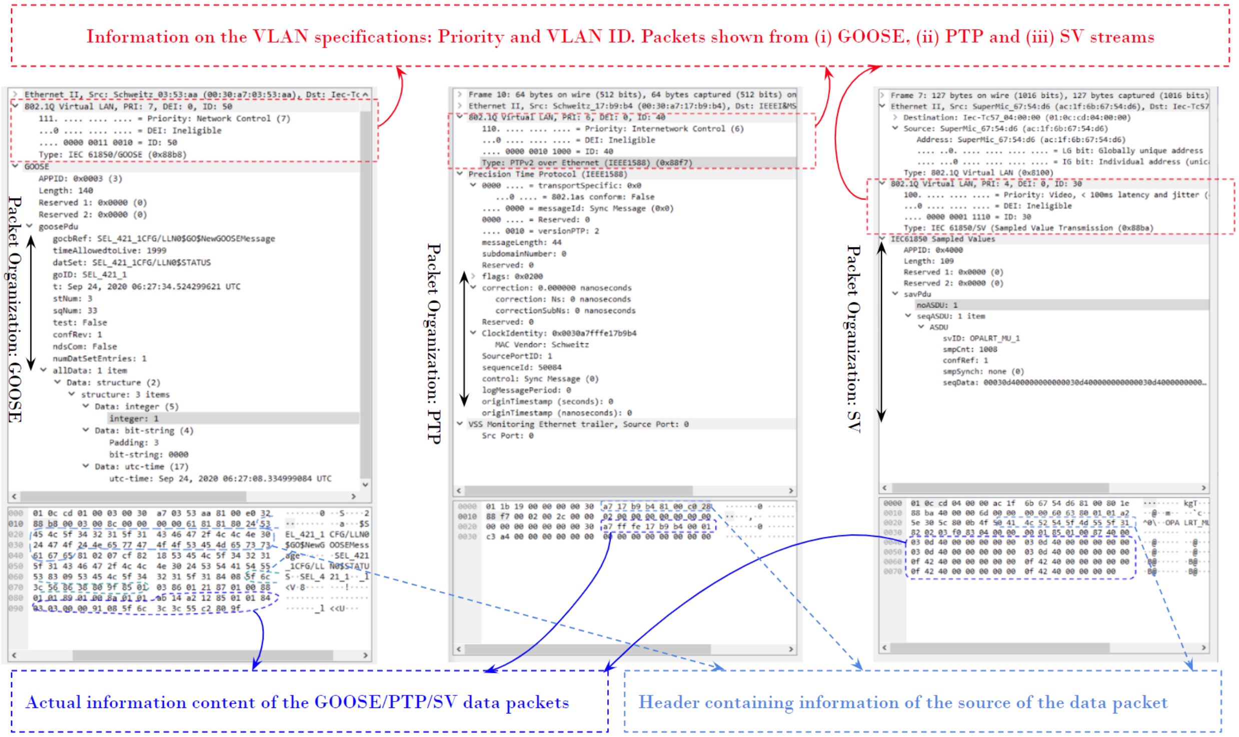
Outside Building
Inside Building
Cloud
BCH1: SWT2 P41
BCH2: SWT2 P40
BCH3: SWT2 P39
BCH4: SWT2 P38
BCH5: SWT2 P37
BCH6: SWT2 P36
SWT4~9
6
6
6
HP Switch 2920-48G (SWT2)
192.168.10.1
2
SWT2 P2 (Station)
SWT2 P3 (PTP)
SEL-2488 Substation Clock (CLC2)
SWT2 P1 (Station)
SWT2 P4 (PTP)
SEL-2488 Substation Clock (CLC1)
ETH1: 192.168.10.6 (Station)
ETH2: 192.168.20.6 (PTP)
ETH1: 192.168.10.5 (Station)
ETH2: 192.168.20.5 (PTP)
Station VLAN (ID: ---)
PMU VLAN (ID: 20)
Sampled-Value VLAN (ID: 30)
PTP VLAN (ID: 40)
GOOSE VLAN (ID: 50)
Internet
SWT2 P35
Server (SRV1)
Redhat: 192.168.10.10 (Station)
Redhat: 192.168.20.40~43 (PMU/SV/PTP/GOOSE)
Windows10_1 VM: 192.168.10.11 (Station)
Windows10_1 VM: 192.168.20.44~47 (PMU/SV/PTP/GOOSE)
Windows10_2 VM: 192.168.10.12 (Station)
Windows10_2 VM: 192.168.20.48~51 (PMU/SV/PTP/GOOSE)
Windows10_3 VM: 192.168.10.13 (Station)
Windows10_3 VM: 192.168.20.52 (PMU)
Reserved for VM: 192.168.10.14~15 (Station)
Reserved for VM: 192.168.20.53~60 (PMU/SV/PTP/GOOSE)
SEL-3400 IRIG-B Distribution Module (SPL2)
3
6
Station
BCH1: 192.168.10.207~214 (Station)
192.168.20.159~174 (PMU/SV/PTP/GOOSE)
BCH2: 192.168.10.215~222 (Station)
192.168.20.175~190 (PMU/SV/PTP/GOOSE)
BCH3: 192.168.10.223~230 (Station)
192.168.20.191~206 (PMU/SV/PTP/GOOSE)
BCH4: 192.168.10.231~238 (Station)
192.168.20.207~222 (PMU/SV/PTP/GOOSE)
BCH5: 192.168.10.239~246 (Station)
192.168.20.223~238 (PMU/SV/PTP/GOOSE)
BCH6: 192.168.10.247~254 (Station)
192.168.20.239~254 (PMU/SV/PTP/GOOSE)
SWT10~15
NI cRIO (CRIO1~8)
SWT2 P14
Arbiter PMU 1133A (PMU1)
192.168.20.10
SWT2 P5
RTU HU_AF (RTU1)
192.168.10.7
SWT2 P18
SEL-421 (RLY1)
192.168.20.11
SWT2 P19
SEL-421 (RLY2)
192.168.20.12
SWT2 P20
SEL-487E (RLY3)
192.168.20.13
SWT2 P17
SEL-487E (RLY4)
192.168.20.14
2
SWT2 P15
SEL-735 (RLY6)
192.168.20.16
SWT2 P16
SEL-651R (RLY5)
192.168.20.15
SEL RTAC Axion (RTACCHS1~2)
2
RTAC1,ETH1: SWT2 P21: 192.168.10.20
RTAC2,ETH1: SWT2 P22: 192.168.10.21
2
RTAC1,ETH2: SWT2 P23: 192.168.20.20
RTAC2,ETH2: SWT2 P24: 192.168.20.21
4
Field
BCH1: SWT2 P46
BCH2: SWT2 P48
BCH3: SWT2 P45
BCH4: SWT2 P43
BCH5: SWT2 P42
BCH6: SWT2 P44
SWT2 P13
SWT3
4
S01,eth1,SWT2 P09: 192.168.20.101
S03,eth1,SWT2 P10: 192.168.20.111
S05,eth1,SWT2 P11: 192.168.20.121
S07,eth1,SWT2 P12: 192.168.20.131
S01,eth0: 192.168.10.100
S03,eth0: 192.168.10.110
S05,eth0: 192.168.10.120
S07,eth0: 192.168.10.130
Typhoon HIL603 Simulator (SML1)
OPAL-RT Simulator
Process
Virtual Hardware
Actual Hardware
Simulator

Figure 8.3: Architecture of the Network Layer of the ALSET Lab Illustrating all the Virtual LANs

8.3 Experiments

8.3.1 Experiment 1: Demonstration of the Different VLAN Data Types

In this experiment various data types were transmitted within the ALSET-network and the streamed packets were captured and monitored through *Wireshark*. The actual


Information on the VLAN specifications: Priority and VLAN ID. Packets shown from (i) GOOSE, (ii) PTP and (iii) SV streams
Packet Organization: GOOSE
Packet Organization: PTP
Packet Organization: SV
Actual information content of the GOOSE/PTP/SV data packets
Header containing information of the source of the data packet

```
> Ethernet II, Src: Schweitz_03:53:aa (00:30:a7:03:53:aa), Dst: Iec-Tc
v 802.1Q Virtual LAN, PRI: 7, DEI: 0, ID: 50
    111. .... .... .... = Priority: Network Control (7)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0011 0010 = ID: 50
    Type: IEC 61850/GOOSE (0x88b8)
v GOOSE
    APPID: 0x0003 (3)
    Length: 140
    Reserved 1: 0x0000 (0)
    Reserved 2: 0x0000 (0)
  v goosePdu
      gocbRef: SEL_421_1CFG/LLN0$GO$NewGOOSEMessage
      timeAllowedtoLive: 1999
      datSet: SEL_421_1CFG/LLN0$STATUS
      goID: SEL_421_1
      t: Sep 24, 2020 06:27:34.524299621 UTC
      stNum: 3
      sqNum: 33
      test: False
      confRev: 1
      ndsCom: False
      numDatSetEntries: 1
    v allData: 1 item
      v Data: structure (2)
        v structure: 3 items
          v Data: integer (5)
              integer: 1
          v Data: bit-string (4)
              Padding: 3
              bit-string: 0000
          v Data: utc-time (17)
              utc-time: Sep 24, 2020 06:27:08.334999084 UTC
```

```
000  01 0c cd 01 00 03 00 30  a7 03 53 aa 81 00 e0 32   ·······0 ··S····2
010  88 b8 00 03 00 8c 00 00  00 00 61 81 81 80 24 53   ········ ··a···$S
020  45 4c 5f 34 32 31 5f 31  43 46 47 2f 4c 4c 4e 30   EL_421_1 CFG/LLN0
030  24 47 4f 24 4e 65 77 47  4f 4f 53 45 4d 65 73 73   $GO$NewG OOSEMess
040  61 67 65 81 02 07 cf 82  18 53 45 4c 5f 34 32 31   age····· ·SEL_421
050  5f 31 43 46 47 2f 4c 4c  4e 30 24 53 54 41 54 55   _1CFG/LL N0$STATU
060  53 83 09 53 45 4c 5f 34  32 31 5f 31 84 08 5f 6c   S··SEL_4 21_1··_l
070  3c 56 86 38 80 9f 85 01  03 86 01 21 87 01 00 88   <V·8···· ···!····
080  01 01 89 01 00 8a 01 01  ab 14 a2 12 85 01 01 84   ········ ········
090  03 03 00 00 91 08 5f 6c  3c 3c 55 c2 80 9f         ······_l <<U···
```

```
> Frame 10: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on
> Ethernet II, Src: Schweitz_17:b9:b4 (00:30:a7:17:b9:b4), Dst: IEEEI&MS
v 802.1Q Virtual LAN, PRI: 6, DEI: 0, ID: 40
    110. .... .... .... = Priority: Internetwork Control (6)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0010 1000 = ID: 40
    Type: PTPv2 over Ethernet (IEEE1588) (0x88f7)
v Precision Time Protocol (IEEE1588)
  v 0000 .... = transportSpecific: 0x0
      ...0 .... = 802.1as conform: False
    .... 0000 = messageId: Sync Message (0x0)
    0000 .... = Reserved: 0
    .... 0010 = versionPTP: 2
    messageLength: 44
    subdomainNumber: 0
    Reserved: 0
  > flags: 0x0200
  v correction: 0.000000 nanoseconds
      correction: Ns: 0 nanoseconds
      correctionSubNs: 0 nanoseconds
    Reserved: 0
  v ClockIdentity: 0x0030a7fffe17b9b4
      MAC Vendor: Schweitz
    SourcePortID: 1
    sequenceId: 50084
    control: Sync Message (0)
    logMessagePeriod: 0
    originTimestamp (seconds): 0
    originTimestamp (nanoseconds): 0
v VSS Monitoring Ethernet trailer, Source Port: 0
    Src Port: 0
```

```
0000  01 1b 19 00 00 00 00 30  a7 17 b9 b4 81 00 c0 28   ·······0 ······
0010  88 f7 00 02 00 2c 00 00  02 00 00 00 00 00 00 00   ·····,·· ·······
0020  00 00 00 00 00 00 00 30  a7 ff fe 17 b9 b4 00 01   ·······0 ·······
0030  c3 a4 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ·······
```

```
> Frame 7: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits)
v Ethernet II, Src: SuperMic_67:54:d6 (ac:1f:6b:67:54:d6), Dst: Iec-Tc57
  > Destination: Iec-Tc57_04:00:00 (01:0c:cd:04:00:00)
  v Source: SuperMic_67:54:d6 (ac:1f:6b:67:54:d6)
      Address: SuperMic_67:54:d6 (ac:1f:6b:67:54:d6)
      .... ..0. .... .... .... .... = LG bit: Globally unique address
      .... ...0 .... .... .... .... = IG bit: Individual address (unic
    Type: 802.1Q Virtual LAN (0x8100)
v 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 30
    100. .... .... .... = Priority: Video, < 100ms latency and jitter (
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0001 1110 = ID: 30
    Type: IEC 61850/SV (Sampled Value Transmission (0x88ba)
v IEC61850 Sampled Values
    APPID: 0x4000
    Length: 109
    Reserved 1: 0x0000 (0)
    Reserved 2: 0x0000 (0)
  v savPdu
      noASDU: 1
    v seqASDU: 1 item
      v ASDU
          svID: OPALRT_MU_1
          smpCnt: 1008
          confRef: 1
          smpSynch: none (0)
          seqData: 00030d40000000000030d40000000000030d4000000000…
```

```
0000  01 0c cd 04 00 00 ac 1f  6b 67 54 d6 81 00 80 1e   ········ kgT···
0010  88 ba 40 00 00 6d 00 00  00 00 60 63 80 01 01 a2   ··@··m·· ··`c···
0020  5e 30 5c 80 0b 4f 50 41  4c 52 54 5f 4d 55 5f 31   ^0\··OPA LRT_MU
0030  82 02 03 f0 83 04 00 00  00 01 85 01 00 87 40 00   ········ ·······
0040  03 0d 40 00 00 00 00 00  03 0d 40 00 00 00 00 00   ··@····· ··@····
0050  03 0d 40 00 00 00 00 00  03 0d 40 00 00 00 00 00   ··@····· ··@····
0060  0f 42 40 00 00 00 00 00  0f 42 40 00 00 00 00 00   ·B@····· ·B@····
0070  0f 42 40 00 00 00 00 00  0f 42 40 00 00 00 00      ·B@····· ·B@····
```

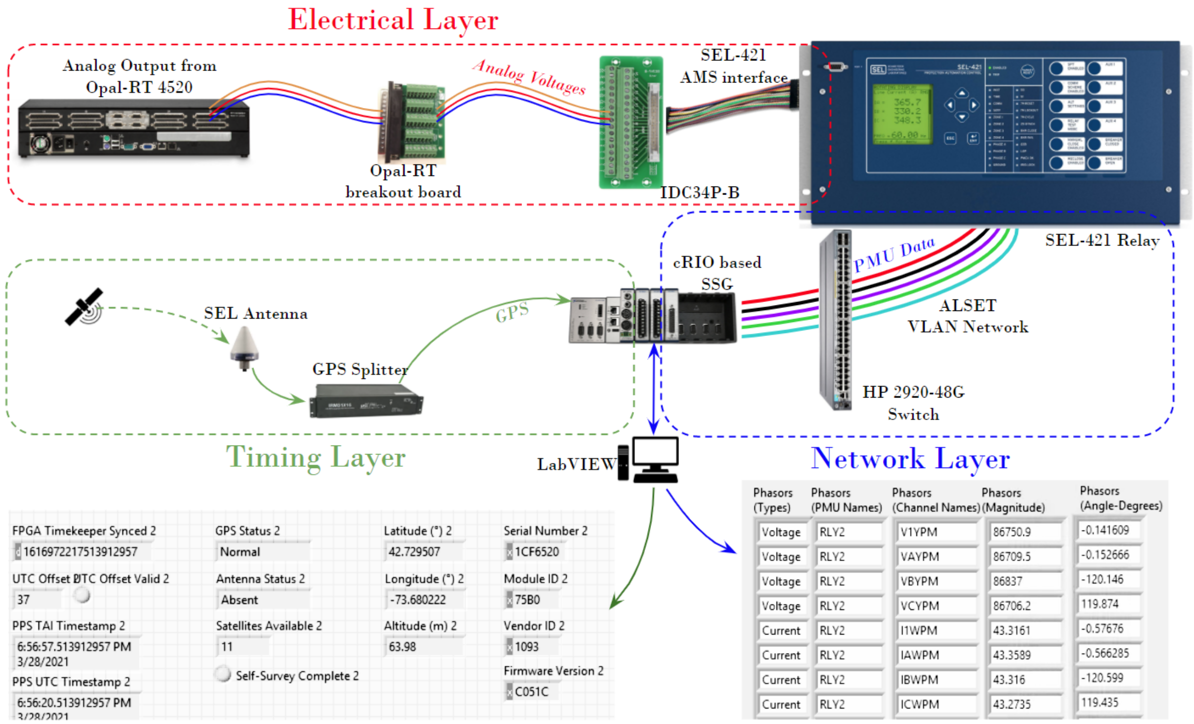
Figure 8.4: Wireshark Screen Captures of the Data Packets Through Different VLANs Implemented in ALSET Lab

data-content of the streams were kept minimal, so that visualization is easier and latency low. It can be observed in Figure 8.4 that *Wireshark captures* display the VLAN ID and the corresponding priority for each data type. It can also be seen that each protocol has some data-bytes dedicated for identifying the source which that data is coming from.

Figure 8.4 also illustrates that the GOOSE data was transmitted from SEL-421 relay with the appropriate VLAN tag. Since the available SEL-421 is not capable of transmitting SV-data, the representative SV-data were transmitted from OPAL-RT real-time simulator. The PTP-data represented in the Figure8.4 was generated from a SEL-2488 Satellite Synchronized Networked clock.

8.3.2 Experiment 2: Demonstrating the Three Layers of ALSET Lab Through an Experiment

This experiment illustrates how the three layers of the ALSET lab are interconnected and how they interact in the context of a simple experiment. The overall configuration, and a set of representative observations for this experiment is shown in Figure 8.5.



FPGA Timekeeper Synced 2: 1616972217513912957
UTC Offset 2: 37
UTC Offset Valid 2: ○
PPS TAI Timestamp 2: 6:56:57.513912957 PM 3/28/2021
PPS UTC Timestamp 2: 6:56:20.513912957 PM 3/28/2021
GPS Status 2: Normal
Antenna Status 2: Absent
Satellites Available 2: 11
○ Self-Survey Complete 2
Latitude (°) 2: 42.729507
Longitude (°) 2: -73.680222
Altitude (m) 2: 63.98
Serial Number 2: 1CF6520
Module ID 2: 75B0
Vendor ID 2: 1093
Firmware Version 2: C051C

Phasors (Types)	Phasors (PMU Names)	Phasors (Channel Names)	Phasors (Magnitude)	Phasors (Angle-Degrees)
Voltage	RLY2	V1YPM	86750.9	-0.141609
Voltage	RLY2	VAYPM	86709.5	-0.152666
Voltage	RLY2	VBYPM	86837	-120.146
Voltage	RLY2	VCYPM	86706.2	119.874
Current	RLY2	I1WPM	43.3161	-0.57676
Current	RLY2	IAWPM	43.3589	-0.566285
Current	RLY2	IBWPM	43.316	-120.599
Current	RLY2	ICWPM	43.2735	119.435

Figure 8.5: Illustration of a Simple Real-time Experiment that Utilizes all Three Layers of the ALSET Lab Hierarchy

Experiment Configuration:

The electrical layer consists of an OPAL-RT real-time simulator which hosts a program that generates three phase analog voltages (0-0.5V range) on its output pins. These analog signals are then connected to the SEL-421 relay's low voltage AMS interface through a IDC34P-B breakout board. The SEL-421 relay estimates the phasors corresponding to the analog voltages based on its internal CT and PT settings, and transmits those phasor measurements at a reporting rate of 30-60 packets per second utilizing C37.118 protocol through the ALSET lab VLAN network. The NI cRIO hardware (connected to the same VLAN) uses the *Khorjin* package to decipher the data enclosed in the C37.118 protocol in real-time, and displays them over the GUI designed on LabVIEW. The uniqueness of this setup is, unlike traditional PDCs the cRIO hardware utilizes an on-board receiver for GPS signal, making the architecture suitable for timing-analysis and timestamp-tracing.

Experiment Results:

This is how the network layer of the lab is utilized by an experiment in real-time. The cRIO, which runs the *Khorjin* package to unwrap the synchrophasor data, receives the GPS signal on the NI-9467 GPS module through the RMS-116 GPS splitter. The effective output of this experiment is dependent on the magnitude of the analog signals generated from the OPAL-RT Real-time simulator and the internal CT/PT settings of the SEL-421 relay. This output can be visualized on a LabVIEW based GUI proposed and designed by the authors in [123] and [124].

Possible Adjustments for cost-reduction:

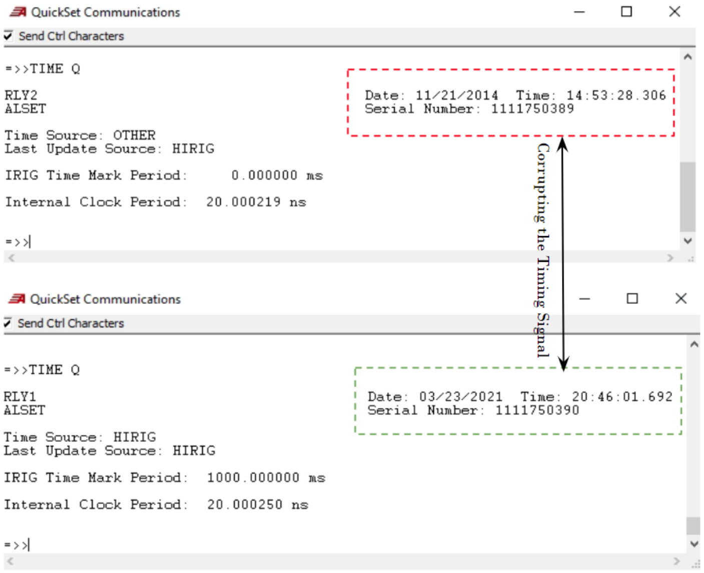
It is possible to run similar tests with a much simplified electrical layer. Instead of generating low-voltage analog signals from a real-time simulator, it can be possible to generate multiple sets of low-voltage analog signals using embedded microcomputer kits like *Raspberry Pi* or *Arduino*, and test the communication and timing infrastructures. Additionally, it is possible to replace the extensive Antenna-network of the reported laboratory, with low-cost USB GPS receivers. These adjustments will put a lot of geographical restraints on the setup, and make the system less flexible. However, they would reduce the cost of the system-development to a great extent.

8.3.3 Experiment 3: Tampering With the Network Layer

In this test the communication/network layer of the ALSET Lab is tampered with a communication network emulation device. However, in order to compare the data obtained through different synchrophasor streams (both through tampered and untampered networks) it is required to implement a device that can parse and aggregate and multiple streams of PMU data in real-time. To this end, a Synchrophasor Synchronization Gateway device was designed and implemented. The design and implementation of this device is presented in Section 9. A test using ALSET laboratory's infrastructure, which deals with receiving power system measurement data through a tampered network is presented in Section 9.3.3.

8.3.4 Experiment 4: Tampering With the Timing Layer

Impairing the timing layer is more challenging because it is unlawful to tamper with GPS signals, and thus, it is not possible to directly affect the signal received by the GPS



```
=>>TIME Q

RLY2                                          Date: 11/21/2014  Time: 14:53:28.306
ALSET                                         Serial Number: 1111750389

Time Source: OTHER
Last Update Source: HIRIG

IRIG Time Mark Period:      0.000000 ms

Internal Clock Period:   20.000219 ns

=>>
```

```
=>>TIME Q

RLY1                                          Date: 03/23/2021  Time: 20:46:01.692
ALSET                                         Serial Number: 1111750390

Time Source: HIRIG
Last Update Source: HIRIG

IRIG Time Mark Period:   1000.000000 ms

Internal Clock Period:   20.000250 ns

=>>
```

Figure 8.6: Observing the Corrupted Timestamp of SEL-421

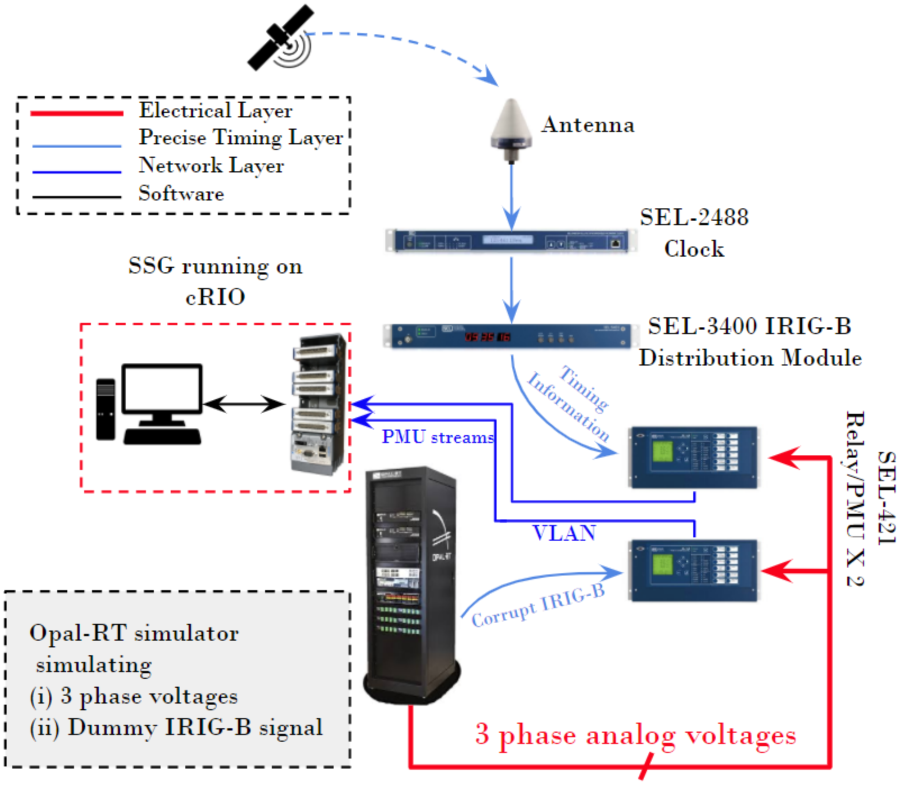
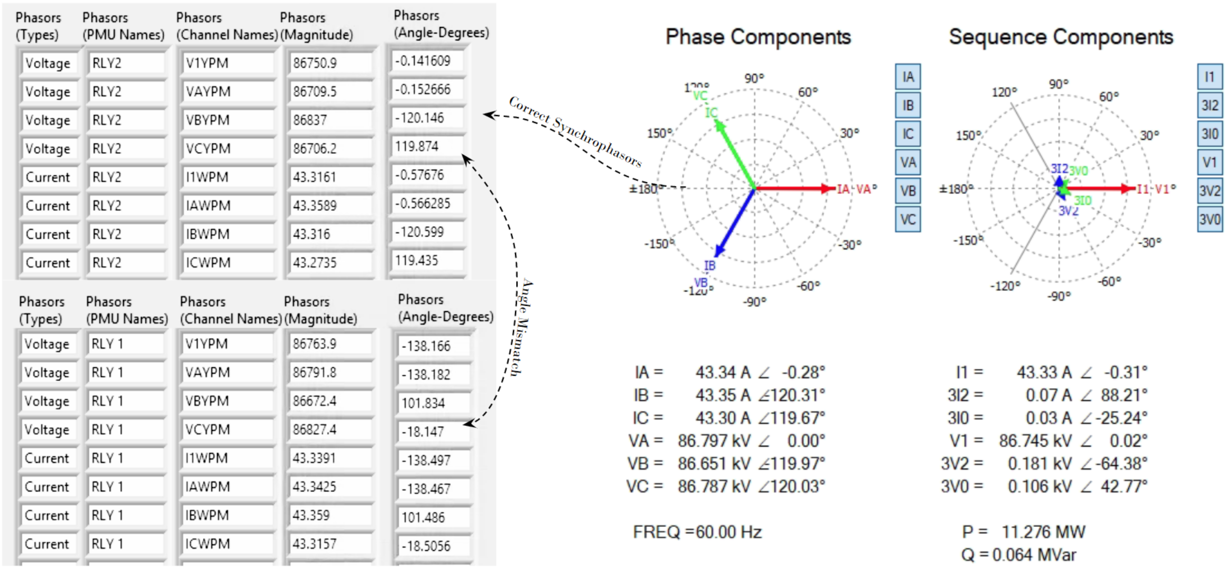


Figure 8.7: Experimental Setup to Tamper with the Precise Timing Layer of the ALSET Lab



Phasors (Types)	Phasors (PMU Names)	Phasors (Channel Names)	Phasors (Magnitude)	Phasors (Angle-Degrees)
Voltage	RLY2	V1YPM	86750.9	-0.141609
Voltage	RLY2	VAYPM	86709.5	-0.152666
Voltage	RLY2	VBYPM	86837	-120.146
Voltage	RLY2	VCYPM	86706.2	119.874
Current	RLY2	I1WPM	43.3161	-0.57676
Current	RLY2	IAWPM	43.3589	-0.566285
Current	RLY2	IBWPM	43.316	-120.599
Current	RLY2	ICWPM	43.2735	119.435

Phasors (Types)	Phasors (PMU Names)	Phasors (Channel Names)	Phasors (Magnitude)	Phasors (Angle-Degrees)
Voltage	RLY 1	V1YPM	86763.9	-138.166
Voltage	RLY 1	VAYPM	86791.8	-138.182
Voltage	RLY 1	VBYPM	86672.4	101.834
Voltage	RLY 1	VCYPM	86827.4	-18.147
Current	RLY 1	I1WPM	43.3391	-138.497
Current	RLY 1	IAWPM	43.3425	-138.467
Current	RLY 1	IBWPM	43.359	101.486
Current	RLY 1	ICWPM	43.3157	-18.5056

IA = 43.34 A ∠ -0.28°
IB = 43.35 A ∠120.31°
IC = 43.30 A ∠119.67°
VA = 86.797 kV ∠ 0.00°
VB = 86.651 kV ∠119.97°
VC = 86.787 kV ∠120.03°

FREQ =60.00 Hz

I1 = 43.33 A ∠ -0.31°
3I2 = 0.07 A ∠ 88.21°
3I0 = 0.03 A ∠-25.24°
V1 = 86.745 kV ∠ 0.02°
3V2 = 0.181 kV ∠-64.38°
3V0 = 0.106 kV ∠ 42.77°

P = 11.276 MW
Q = 0.064 MVar

(a) Correct(above) and corrupted Synchrophasor Measurement from the NI-cRIO

(b) Representative correct synchrophasor measurement from the SEL-421

Figure 8.8: Experimental Synchrophasor Observation After Tampering the Timing Layer

antennas. Hence, this section shows how timing can be safely and lawfully impaired for experimentation. The only lawful way to tamper with the precise timing layer is to generate an alternate source of timing signal in the IRIG-B format. The authors in [117] presented how mock IRIG-B signal can be generated through the output ports from the Opal-RT simulator. In this research, an IRIG-B signal was generated following the same technique.

Experiment Configuration:

In this experiment, two different models were simulated on the OPAL-RT hardware. The first one generated low voltage analog signals as described in Experiment 3. The second model is used to generate dummy IRIG-B signals to one of Opal-RT's digital output ports. This port was connected to one of the SEL-421's IRIG-B input. Thus, this SEL-421 PMU does not have access to the satellite synchronized precise clock, but it reads time from a synthetic IRIG-B signal which is configured to represent a time in the year 2014. An example of such tampering is shown in Figure 8.6. It can be seen in this figure that the tampered IRIG-B signal is translated into a wrong time from the year 2014, while the experiment was performed in March of 2021. The configuration of the overall experiment is shown in Figure 8.7. Both the actual and corrupt IRIG-B signals are represented in light blue lines,

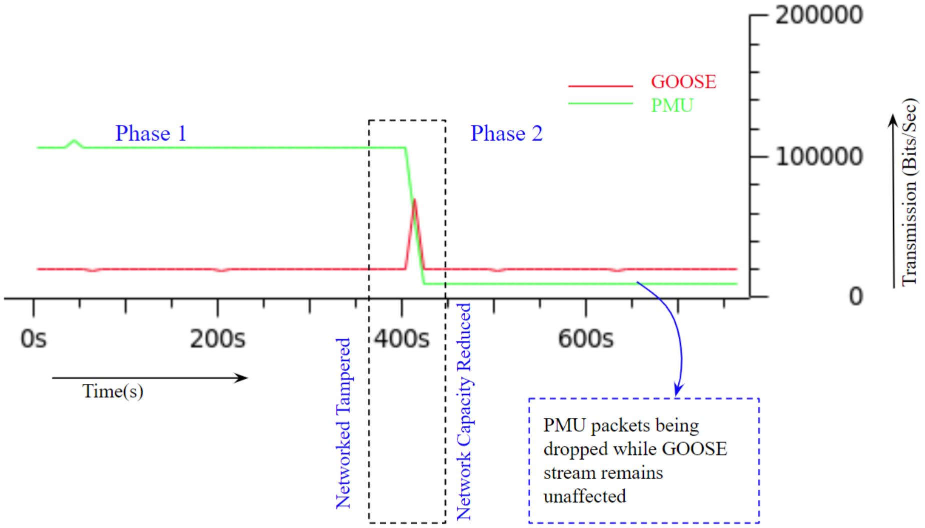


Figure 8.9: Effect of the QoS Priority Rule, Under Varying Network Bandwidth

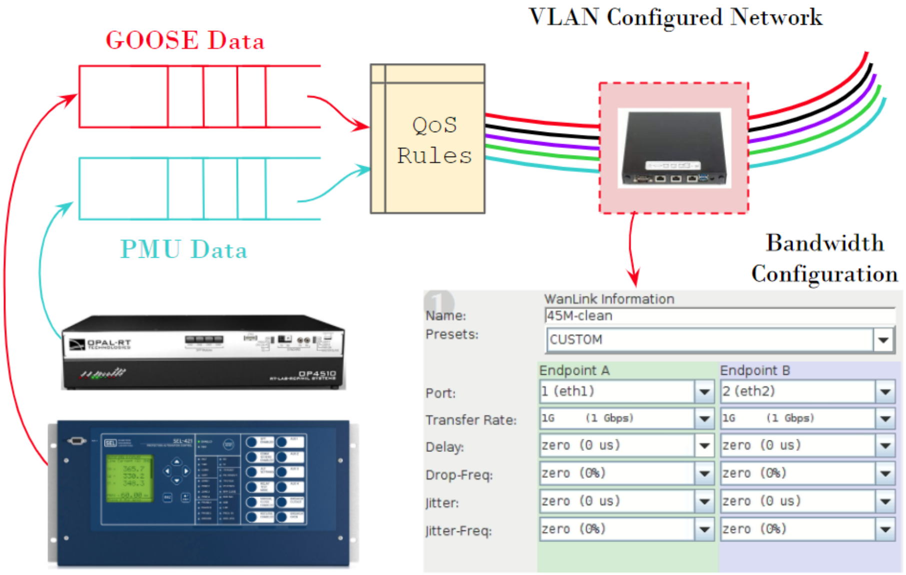
and the PMU streams travelling through VLAN-network is represented in dark blue lines. The electrical part of this experimental setup, which generates the analog voltage signals is represented by red lines.

Experiment Results:

The two synchrophasor streams would still read the same voltage signals with the exact same magnitude and frequency. However, since the timing-source of one of the streams is corrupted, it will fail to compute the phasor-angles correctly, as shown in Figure 8.8.(a) while Figure 8.8.(b) presents the correct phasor measurements from the SEL-421 relays.The GUI used for monitoring these measurements in real-time was implemented in LabVIEW. The phase angle mismatches can be observed directly on this GUI.

8.3.5 Experiment 5: Validating the QoS Rules for the Network

The QoS rules configured for the VLANs are verified in this experiment. These rules were first introduced in [108] and based on those rules the priorities and guaranteed minimum bandwidths (GMB) were specified. Two different experiments were performed to validate the QoS rule related to the priorities of the VLAN networks, and to validate the Guaranteed Minimum Bandwidth of those networks, respectively.



WanLink Information

Name: 45M-clean

Presets: CUSTOM

	Endpoint A	Endpoint B
Port:	1 (eth1)	2 (eth2)
Transfer Rate:	1G (1 Gbps)	1G (1 Gbps)
Delay:	zero (0 us)	zero (0 us)
Drop-Freq:	zero (0%)	zero (0%)
Jitter:	zero (0 us)	zero (0 us)
Jitter-Freq:	zero (0%)	zero (0%)

Figure 8.10: Experimental Setup to Demonstrate the QoS Rules

8.3.5.1 Experiment 5(a): Validating the QoS for Priorities of VLANs

It can be seen from Table 1, that GOOSE and PMU VLANs consume a similar range of bandwidth, but with different priorities. The objective of this experiment is to validate the priorities between these two VLAN networks.

Experiment Configuration:

The network is utilized to transmit both PMU and GOOSE data initially. By using the functionalities provided by *CandelaTech Lanforge*, the total bandwidth of the network was then reduced from 1 Gbps to 256 Kbps, and the priority with which the two data-types were handled was monitored. The experimental setup is shown in Figure 8.10

Experiment Results:

The experimental protocol for this test can be divided into 2 phases as described below, along with the obtained results.

- **Phase 1:** Phase 1 spans from the initiation, up to 400 seconds as shown in Figure 8.9. During this time, PMU and GOOSE data are streamed through the same channel utilizing different VLAN specifications. The bandwidth of the channel is the maximum possible bandwidth of the network, i.e. 1 Gbps. Under this operating condition, all the GOOSE and PMU packets will be successfully transmitted through the network.

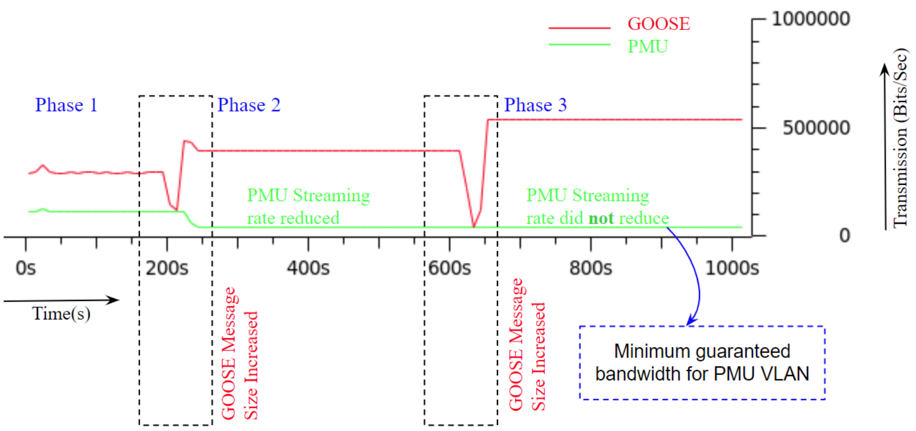


Figure 8.11: Effect of the QoS Guaranteed Minimum Bandwidth Rule Under Varying GOOSE-Data Stream

However, since the PMU message frames contain more bits than the GOOSE message frames, it can be seen that the majority of bandwidth is being utilized for transmitting PMU data (green plot).

- **Phase 2:** Phase 2 spans from the end of phase 1 (i.e. around 400 seconds) to 800 seconds as shown in Figure 8.9. At the end of Phase 1, the bandwidth of the network was reduced to 256 Kbps by utilizing the *Lanforge Manager* software package. It can be seen from Figure 8.9 that upon the imposition of this bandwidth restriction, the data-transfer through the PMU VLAN was drastically reduced, even though the data-transfer through the GOOSE VLAN was unaffected. This observation demonstrates that the GOOSE VLAN has a higher priority according the proposed QoS, and thus GOOSE transmissions are prioritized over PMU ones.

8.3.5.2 Experiment 5(b): Validating the Guaranteed Minimum Bandwidth for the VLANs

It is demonstrated in Table 1, that all VLAN networks have certain guaranteed minimum bandwidths. This means that even when there are higher priority data waiting in the queue, the data with lower priority VLAN would still be guaranteed a minimum bandwidth so that it does not get starved by the higher priority data. This experiment demonstrates the effect of this *minimum* bandwidth in data transmission through the ALSET network.

Experiment Configuration:

For this experiment, the SEL relay and the PMU are configured to stream GOOSE and PMU streams in parallel. In this situation, the network's maximum capacity is set at 64 Kbps. Initially, this bandwidth is large enough to accommodate both these streams. However, the GOOSE data-load is increased by modifying the GOOSE-data packet's configuration from the SEL *AcSELerator Architect* software, twice at t=200 seconds and t=600 seconds, and then the observations are noted.

Experiment Results:

The experimental observations for this test can be divided into 3 phases as described below, along with the obtained results.

- **Phase 1:** From t=0 second to 200 seconds in Figure 8.11, both PMU data and GOOSE data are streamed in parallel. The networks bandwidth of 64 Kbps was large enough to stream both these data-streams without any issues.
- **Phase 2:** This duration begins at 200 seconds and lasts till 600 seconds in Figure 8.11. At t=200 seconds, the load for GOOSE-data is increased. To incorporate this increased load of GOOSE stream within the same network of 64 Kbps capacity, the PMU data-stream reduced its transmission rate. This is because GOOSE VLAN has higher priority and the network has low bandwidth.
- **Phase 3:** This duration begins at 600 seconds and lasts till the end of this experiment as shown in Figure 8.11. At t=600 seconds, the GOOSE-data load is increased furthermore. Since, bandwidth is still kept constant at 64 Kbps, it was expected that the PMU transmission would reduce furthermore compared to what was observed in phase 2. However, the PMU VLAN transmission remained unaffected, even though the GOOSE transmission was increased. This can be explained by the fact that the PMU VLAN was operating at its *guaranteed minimum bandwidth*, so it would not operate at a lower bandwidth, even though there were higher-priority GOOSE data on queue.

8.3.6 Analysis and Synthesis of Experimental Complexity

The authors' experience in conducting these experiments reveals that while it is possible to conduct CPPS experiments safely and securely in the implemented laboratory setting, all

Table 8.2: Reviewing the Complexities of the Performed Experiments

Expt	Objective	Complexity	Software Requirement	Hardware Requirement	Comments
1	Validating the VLANs	Low	WireShark, SEL Quickset	SEL-421/Opal-RT	Source of the data varies based on VLAN choice
2	Introducing the 3 layers of the lab	Medium	RT-Lab, WireShark, LabVIEW	Opal-RT Simulator, NI-cRIO, Antenna	
3	Tampering Communication Network	Medium	RT-Lab, Lanforge, LabVIEW	Opal-RT, CT-910, NI-cRIO	
4	Tampering Timing Network	High	RT-Lab, SEL Quickset, LabVIEW	Opal-RT Simulator, SEL-421 NI-cRIO, Antenna	Additional GNC Cable needed
5(a)	Demonstrating the QoS rule-a	High	RT-Lab, SEL Quickset, Lanforge, LabVIEW	Opal-RT, CT-910, SEL-421, NI-cRIO	The network bandwidth needs to be carefully
5(b)	Demonstrating the QoS rule-b	High	RT-Lab, SEL Quickset, Lanforge, LabVIEW, SEL Architect	Opal-RT, CT-910, SEL-421, NI-cRIO	configured by Lanforge for each run.

technologies involved make the experimental configuration process time-consuming and hard to automate. In addition, there is a steep learning curve to master all the “cyber” know-how together with power grid understanding. This poses a major challenge for verification, testing and validation of cyber-components and systems that need to be deployed in the power grid to support the on-going energy transition. More efforts are needed in making experimental testing more efficient and reproducible to keep up with the rate of innovation in new functionalities being developed in the “cyber” side so to facilitate, not only their adoption without unwanted consequences, but also for the “physical” side to fully exploit the benefits of increased digitization. It needs to be noted that, the experiments demonstrated in this research required expensive equipment such as industrial relays, real-time simulators, and NI cRIOs. Additionally, the auxiliary infrastructures(e.g. arrangement of LMR cables, proper connection of GNC cable to attenuate/amplify the signal level, placement of antennas) take significant amount of time to procure, deploy and setup. In an attempt to formalize, synthesize and grade the challenges involved in the experiments reported in this chapter, the experiments were classified into three categories based on their complexity in Table 8.2. It is noteworthy to mention that, the proposed experiments are only designed to demonstrate the functionalities of the laboratory. An experiment to simulate and test real-world power system problems (as demonstrated in [107], [127]) increases complexity.

8.4 Conclusions

This chapter reported the design and results from experiments of the precision timing and communication network layers of a cyber-physical power system simulation lab implemented at Rensselaer Polytechnic Institute. The experiments demonstrate how these two layers can be safely and lawfully tampered with, which can allow to better understand the interactions between different engineered systems of a cyber-physical power grid. Additionally, the VLAN network and its corresponding QoS rules were demonstrated by streaming predetermined set of data.

The results of the experiments reported in this chapter validate the proposed SGAM-based architecture for digital power system simulation labs in [108]. While more extensive laboratories targeted for similar experiments exist in U.S. National Labs [99] and the industry [126], most of those implementations are too expensive and/or complex for most research/academic purposes. The implementation proposed in this work requires less resources and is suitable for teaching and training students and engineers alike. Section 3.2 proposed some major simplifications on the existing architecture, which can reduce the cost of the implementation drastically. The proposed simplifications however, are only valid for demonstrations at the lab scale because of the restrictive nature of the low-cost equipment. This implies that all the experimental demonstration on such low-cost systems would ultimately need to go through the expensive product development process before they can actually be utilized in a real-world grid setting.

CHAPTER 9
IMPLEMENTATION OF SYNCHROPHASOR SYNCHRONIZATION GATEWAY & CONTROL ARCHITECTURE

Phasor Data Concentrators (PDC) receive and time-synchronize phasor data from multiple phasor measurement units (PMUs) to produce a real-time, time-aligned output data stream. PDCs are expected to handle large sets of data and may consume substantial hardware resources in terms of memory. In this chapter, an alternative hardware architecture is proposed which can read & extract multiple PMU streams in real-time. This proposed hardware is referred to as the Synchrophasor Synchronization Gateway (SSG).

This chapter presents the development and preliminary applications of this hard real-time Synchrophasor Synchronization Gateway (SSG) hardware. The architecture of the SSG hardware is based on *National Instruments'* Compact Reconfigurable Input-Output (cRIO) platform. It utilizes the *Khorjin* library [123] which is able to receive and parse synchrophasor data from a PMU/PDC based on IEEE C37.118.2 protocol. A fully functional PDC is expected to store and publish PMU data. The proposed real-time hardware prototype, however, does not store data, as its goal is to provide essential synchronization and aggregation functions to be used by protection and control devices. Hence, this prototype is described as a Synchrophasor Synchronization Gateway (SSG) instead of a PDC. In this chapter, elementary tests related to timing, delay and reliability are performed on the SSG, and the results along with the observed issues are reported. The SSG was tested with PMU streams generated from Opal-RT real-time simulators, SEL-421 protective relays, and Na-

• Portions of this chapter previously appeared as: P. M. Adhikari, H. Hooshyar, R. J. Fitsik, and L. Vanfretti, "Precision timing and communication networking experiments in a real-time power grid hardware-in-the-loop laboratory," *Sustain. Energy Grids & Netw.*, vol. 28, Dec. 2021, doi: 10.1016/j.segan.2021.100549.

• Portions of this chapter previously appeared as: H. Hooshyar, L. Vanfretti, J. H. Chow, R. Fitsik, P. M. Adhikari, "ALSET lab: Designing precise timing and communications for a digital power grid laboratory," in *IEEE Power & Energy Soc. General Meeting*, Montreal, QC, Canada, Aug. 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281808.

• Portions of this chapter previously appeared as: P. M. Adhikari and L. Vanfretti, "Delay analysis of a real-time hard reconfigurable synchrophasor synchronization gateway," in *IASTED Int. Conf. Control & Optim. of Renewable Energy Syst.*, Annaheim, CA, USA, Dec. 2019, doi: 10.2316/P.2019.859-008.

tional Instruments' cRIO based PMUs. Once the SSG hardware was tested, it was extended to incorporate real-time control functionalities. This modified hardware is referred to as the Synchrophasor Synchronization Gateway and Controller (SSGC). The SSGC hardware was utilized to architect a partially decentralized control framework for microgrid applications. A microgrid model developed in Typhoon HIL 604 real-time simulator was used to validate the SSGC control architecture. A network-tampering device was introduced in the network to test the robustness and resilience of this proposed architecture when subjected to communication network impairments.

9.1 Introduction

9.1.1 Motivation

Phasor Data Concentrators (PDC) are integral part of modern synchrophasor measurement systems. PDCs are expected to receive, parse, allign, store and publish measurement data from field PMUs. They are expected to be compatible with synchrophasor transmission protocols such as IEEE C37.118.1/ 2 and substation automation protocols such as IEC 61850-90-5. However, existing PDC hardware architectures, fail to comply with hard real-time control requirements. As reported in literature [128]-[130], most implementations are purely on the software level, and needs extensive testing in terms of real time performance. In [131] the authors have proposed a similar platform for WACS applications, for a single PMU stream. This chapter reports implementation and testing of a synchrophasor synchronization gateway with multiple concurrent PMU/PDC streams, which makes it an attractive infrastructure to be used in WACS and WAMPAC applications. This chapter reports a hard Synchrophasor Synchronization Gateway (SSG) implemented in both **real-time software** and **real-time hardware** that performs the most essential functionalities of an industrial PDC, except data-publishing and storage as the main goal is to provide synchronization and aggregation services for the real-time controls and protection.

In recent years, the rapid development in the areas of distributed energy resources, such as PV and Li-ion Batteries has led to an increase in the research efforts towards the development of *microgrids*. 'Microgrids' - as defined by the International Electrotechnical Commission (IEC) in [132] are - *groups of interconnected loads and distributed energy resources with defined electrical boundaries forming a local electric power system at distribution voltage levels, that acts as a single controllable entity and is able to operate in either grid-*

connected or island mode. The monitoring, protection, and control of microgrids have been an interesting research area for the past decade. Because microgrids are typically supported by distributed energy resources (DERs), they involve a significant amount of power electronic hardware and the sophisticated control systems associated with those power electronic circuits. This is addressed by the literature in [8] and [12].

The central motivation of extending the SSG architecture into the paradigm of control functionalities is to utilize the time synchronized measurements obtained from the PMUs to seamlessly control the power electronics based distributed energy resources (DERs) that are used to construct the microgrid. The survey in [13] demonstrates that the design of a functional control system for microgrids needs to be hierarchical. This architecture allows for the proposed SSG to incorporate parts of the microgrid control functionalities to run on a synchrophasor synchronization gateway hardware, while the remaining control actions are implemented locally within the individual DERs. To this end, the SSG design is extended into a prototype Synchrophasor Synchronization Gateway & Controller (SSGC). This SSGC was tested with a real-time microgrid implementation on real-time simulator (Typhoon HIL-604).

The SSG/SSGC implementation is carried out based on the compact reconfigurable input-output (cRIO) devices. The underlying cRIO hardware is configurable by a graphical interface designed in the LabVIEW environment. This architecture is user friendly in terms of configuration, display and hardware management.

9.1.2 Related Works

Because, the proposed SSG performs the tasks traditionally performed by a PDC, the literature review consists of past research that dealt with PDC implementations. The work illustrated in this chapter extends the library presented in [123] for unwrapping PMU data in the proposed prototype SSG. The functions of this library are written in C and compiled into a dynamically linked library using National Instruments' CVI infrastructure. Concerning PDC design, [129] presented important proposals in terms of standardization of the measurement architecture. It also dealt with the problems of accurate time synchronization and management of data-loss scenarios. Even though this work was crucial in terms of standardization, it did not propose any feasible hardware implementation that can meet hard real-time control requirements at the sub-second level. These standardization efforts were further extended by the authors in [130]. Authors in [128] experimented with Raspberry Pi-based hardware architecture for PDC implementation. Their work was successful in implementing a functional PDC, but the performance analysis in terms of time-synchronization, latency, reliability were not reported. [133] presented comparisons between the existing open source PDC software systems. The authors in [134] proposed and compared PDC network architectures in a real life industrial scale networks in Switzerland and Netherlands.

In the domain of control system design for microgrids, the authors in [13] surveyed and classified the existing control strategies into three different classes depending on the priorities, time-scales and required speed of the various control actions. To elaborate further, the primary control class consists of the fastest control actions including voltage and current control algorithms for the individual DERs. The secondary control class evaluates the power flows to and from the different existing DERs, and helps the microgrid navigate between the islanded and the grid-tied modes. The secondary control class tackles slower dynamic responses (e.g. power flow) compared to the primary control class. Finally, the tertiary control class consists of supplementary control algorithms sitting on top of both primary and secondary classes of control, and enables the microgrid to operate in an financially optimized fashion. The research reported in [135] demonstrated significant efforts of standardization across these three classes of control systems in microgrids. The authors in [136] explored the utilization of synchrophasor data to monitor microgrids and to increase the reliability of measurement data. To this end, this research proposed an Advanced Phasor Data Concentrator (APDC) hardware which is capable of operating under a tampered network and estimate missing data points in the synchrophasor streams. However, this hardware was not time synchronized and the reported experiments were performed by a programmable voltage source, instead of RT-simulation models of microgrids. The experiments reported in [137] illustrated a synchrophasor based control architecture for microgrids, where the synchrophasor data is used to formulate reduced order dynamic models for the DERs within the microgrid, and used those models to seamlessly navigate the microgrid between the islanded mode and the grid tied mode. Researchers in [138] demonstrated the utilization of adaptive network management tools within the PDC to compensate the network delays between the PDC and the individual PMUs.

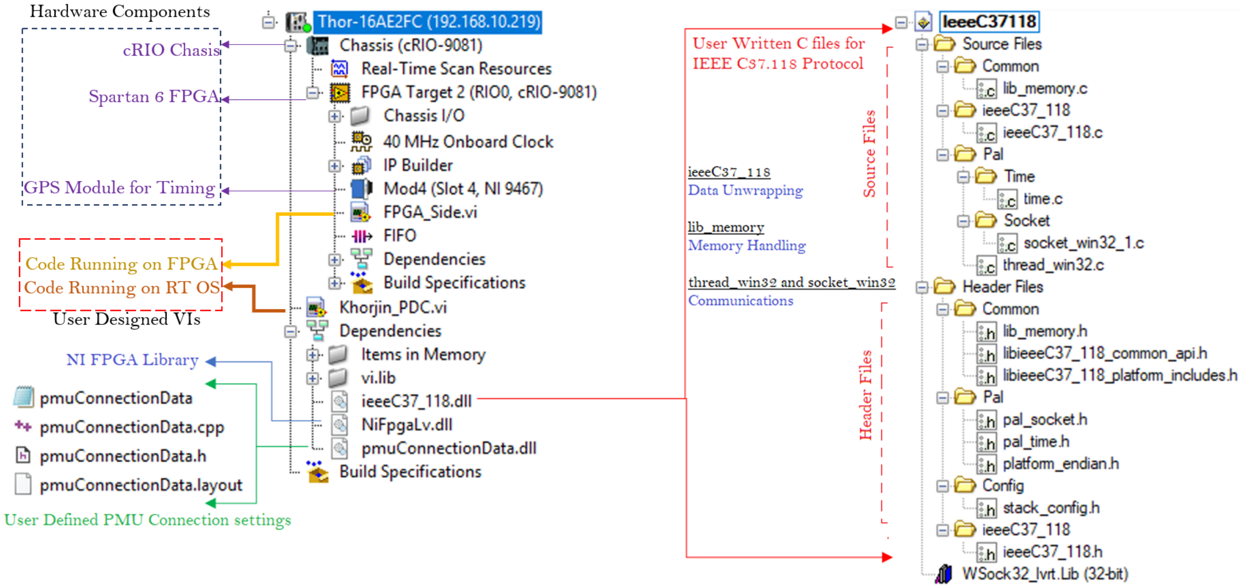


Figure 9.1: An Overview of the Hardware and Software Components for the SSG Implementation

9.1.3 Highlights of This Chapter

- The application and extension of a C-based library to receive and parse synchrophasor data from a PMU/PDC to construct multiple receiver threads inside the proposed prototype SSG.

- Implementation and testing of the proposed SSG prototype using multiple concurrent PMU streams, and characterizing its reliability and performance.

- Stress-testing of the SSG prototype by introducing network traffic and characterizing its performance under varying network conditions.

- Extension of the SSG hardware to incorporate rudimentary control actions for microgrids. This updated prototype is referred to as the SSG Control (SSGC) hardware.

- Utilization of the SSGC hardware to monitor and control a microgrid model running in real-time on the Typhoon HIL 604 simulator.

- Stress-testing the SSGC by introducing network delay and data-drops. In that process, the robustness of the SSGC based control system for the microgrid is evaluated.

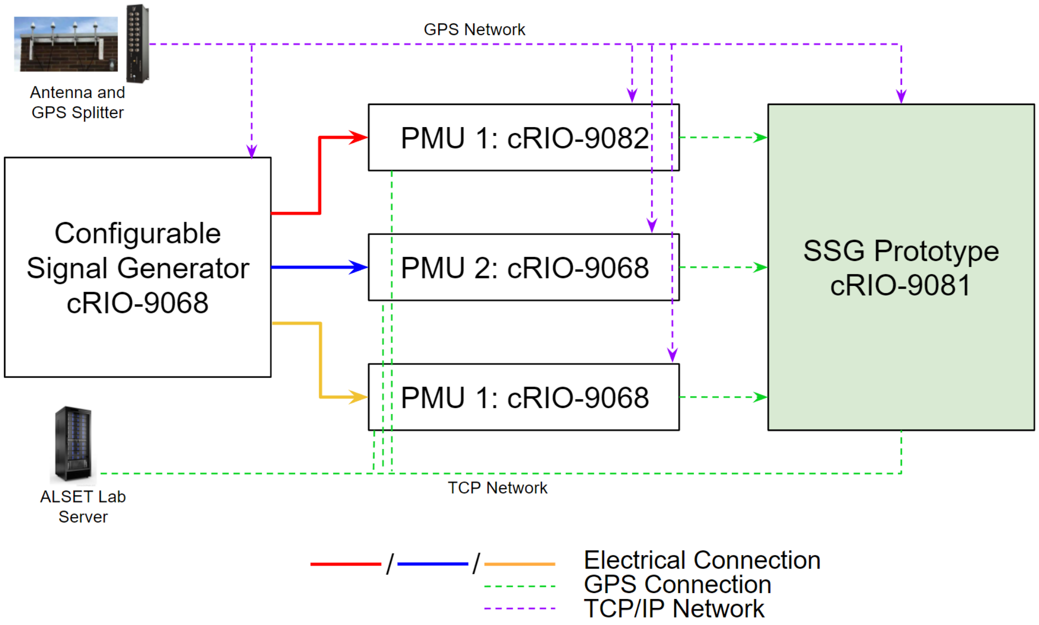


Figure 9.2: Block Diagram of the Experimental Setup for Test Case 1

9.2 Proposed Hardware & Software Framework for SSG/SSGC Architecture

This section introduces the components in the hardware and software layers of the proposed SSG prototype. These components are illustrated in Figure 9.1.

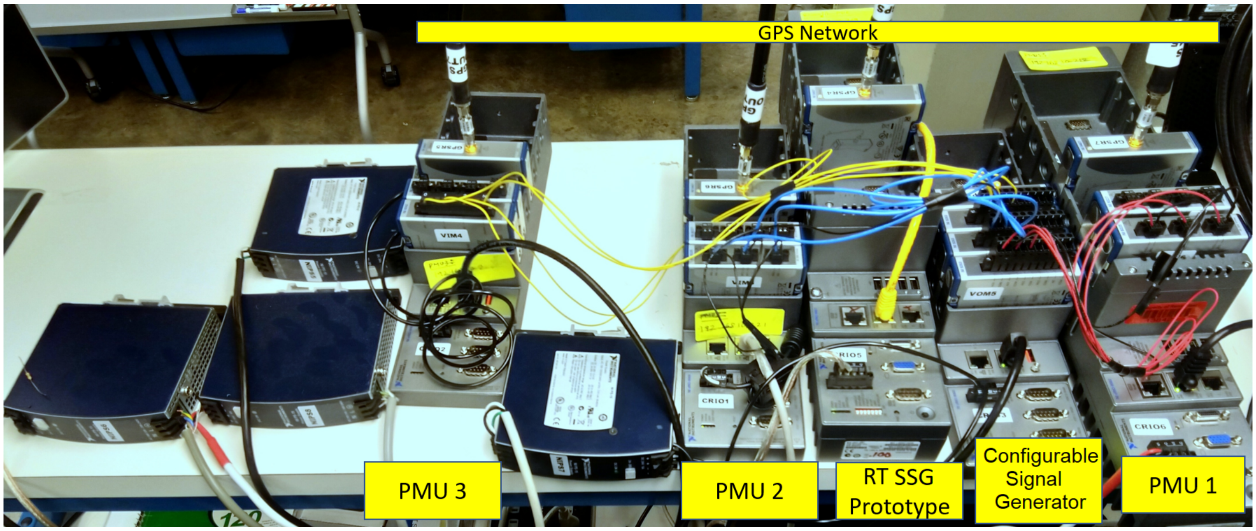


Figure 9.3: Photograph of the Experimental Setup

9.2.1 Software Components

The software architecture utilizes *National Instruments'* virtual instrumentation (VI) infrastructure. In general, there are two top level VIs. One of them runs on the real time processor, and the other one runs on a Spartan 6 FPGA inside the cRIO chassis. The VI that runs on the real time processor (`Khorjin_PDC.vi` in Figure 9.1) has all the functionalities for networking, parsing and processing. These functionalities are used to connect with different PMUs and to unwrap the PMU data being received. The extension of the code which provides control functionalities for the microgrid are also added to this part of the code. All the functionalities (except the control functions) were encoded in standard C, and were converted into a dynamic library using National Instruments *Windows CVI* tool-chain. The source code are also tested with *Microsoft Visual Studio* and *Eclipse*, however the use of the CVI tool-chain was observed to be the most reliable and was ultimately recommended by NI. An additional .cpp file was written to configure the connection settings for each of the PMUs that communicate with the gateway. This file, along with an appropriate header, were converted into dynamic libraries, which are used by the SSG VI running on the RT processor of the cRIO hardware. The VI running on the FPGA communicates with the C series NI 9467 GPS module and provides the time-stamp which will enable the delay computations within the hardware.

9.2.2 Hardware Components

In addition to the software components described above, the SSG has appropriate physical hardware to enable the functionalities of those software components. The two

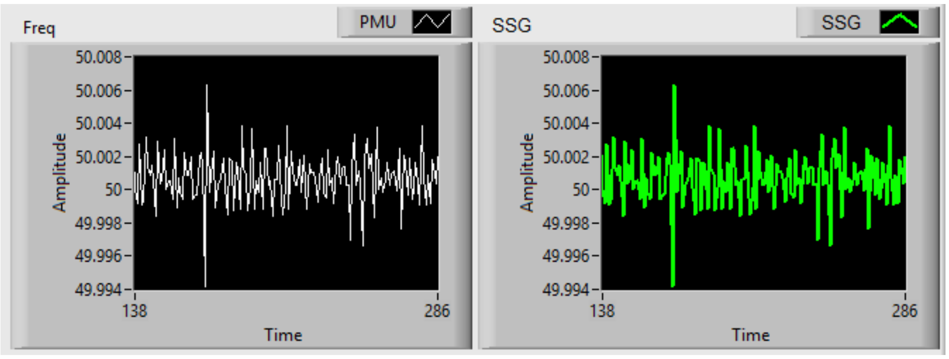


Figure 9.4: Frequency Measured in PMU (white) and the Measured Frequency Received by the SSG (green)

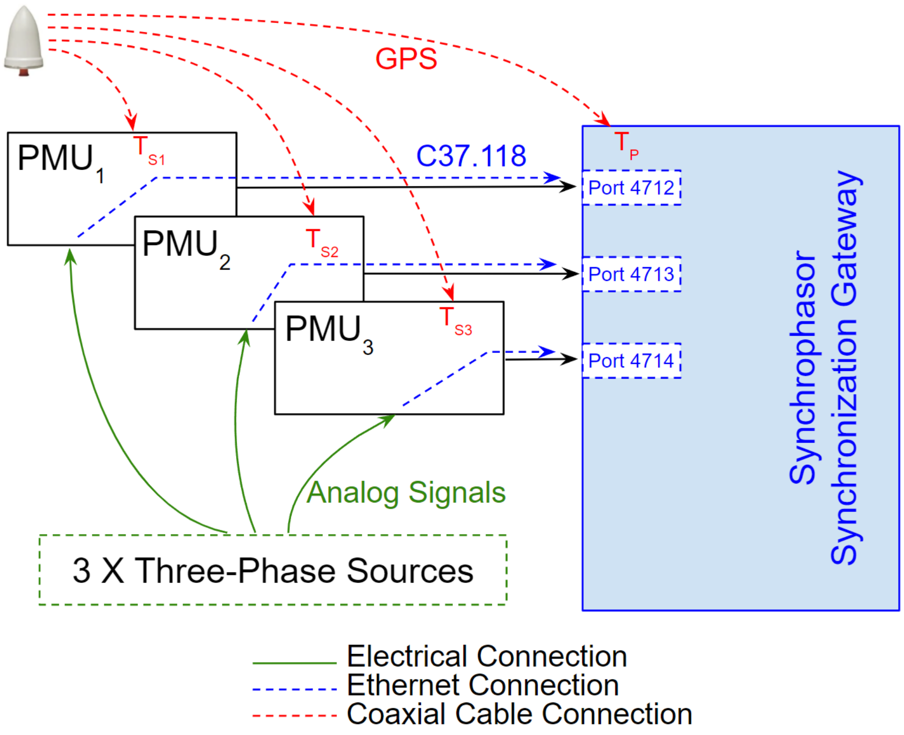


Figure 9.5: Timing Diagram of the Experimental Setup

major components of the hardware are (a) RT processor Intel Celeron U3405 of the cRIO 9081 chassis and (b) Xilinx Spartan 6 LX 75 FPGA. The RT processor runs Microsoft WES 7 Runtime OS. All the C37.118 functionalities along with the control algorithm, and the TCP communication interfaces run on this part of the hardware. On the other hand, the Spartan 6 FPGA interacts with the NI 9467 GPS module to acquire the latest time-stamp, and sends this time-stamp to the WES7 OS running on the real time processor for further use.

9.3 Experimental Case Studies and Analysis

In order to validate the operation of the SSG and the SSGC prototypes, real-time experiments were performed. In this section, four different test-cases are demonstrated. In the first test-case, the SSG hardware is utilized to monitor multiple PMU streams in real-time. These PMUs are electrically connected to a low-voltage configurable analog signal generator as introduced in chapter 3. The second test case introduces user-controlled delay in the network between the PMU and the SSG. The performance of the SSG is then evaluated under these conditions. The third test case demonstrates the real-time synchrophasor data aggregation through the network, and explores a possible application of this infrastructure. The fourth and final test utilizes the power system models described in chapters 5 and 6 to construct a real-time simulation model of a microgrid. Then, the SSG hardware is utilized to monitor this microgrid in real-time through multiple synchrophasor streams. The SSG hardware is updated to incorporate parts of the real-time control system for the microgrid, based on the measurement-data received through synchrophasor streams. This updated hardware is referred to as the SSGC hardware, which is tested under varying network conditions in this final experiment.

9.3.1 Test case 1: Usage of SSG Hardware to Parse PMU Streams

To analyse the performance of any prototype PDC/SSG, more than one PMU is required. In this test case, the PMUs used were implemented on three separate compact RIOs (one cRIO 9082 and two cRIO 9068). The basic design of these PMUs were obtained from National Instruments 'Advanced PMU Development System' [139] . Those designs were compiled and synthesized on NI's high performance computing server using Xilinx ISE synthesis toolchain. As seen from the diagram in Figure 9.2, a TCP network is used to interface these PMUs to the SSG hardware. The PMUs and the SSG running on the cRIO devices are expected to be discoverable by one another, since they are under the same subnet mask. It is important to note that, the cRIOs can talk to each other, via the physical TCP/IP network through their own respective ports. In this particular case, the SSG on the cRIO-9081 was interacting with the PMU_1 on cRIO-9082 via port 4712, with PMU_2 on cRIO-9068 via port 4713 and with PMU_3 on cRIO-9068 via port 4714.

With all of this network setup completed, some electrical disturbances (e.g. step changes, frequency change, voltage phase/magnitude modulation) can be easily provided inside any of the programmable signal generators, for the sake of experimentation. To perform such experiments, the PMUs need to be fed with legitimate 3 phase signals in order for them to produce any meaningful output stream. To this end, three sets of configurable time synchronized balanced 3 phase analog voltage/current signals were connected to the individual PMUs which feed the SSG running on the cRIO 9081 device. An additional cRIO 9068 hardware was used to generate these three programmable balanced three phase signals. This signal generation utilizes NI 9263 C series analog output modules and the analog signals were configured using a LabVIEW GUI. A picture of the hardware used for this experiment is shown in Figure 9.3.

For testing, the entire experiment was configured to run for several days, to make sure the TCP communication would not abruptly break or saturate. Upon rigorous testing, it was concluded that, the proposed architecture was stable, and can support multiple PMUs (up to 3 were tested), connected over a network. The first test for validating the SSG prototype was to verify whether the communication is continuous, real-time and accurate. It needs to be noted, that the data observed in the SSG may suffer a lack of accuracy because of occasional data drops over the TCP network from PMU to SSG. It is also important to note that, a reduction in the transmission rate can lead to a more reliable communication, however, it will sacrifice the resolution offered to the SSG. To illustrate, in Figure 9.4, the PMU (white) reports the frequency at a rate of 50 samples/second. However, the reporting rate is set to 20 samples/sec, thus the SSG reads only 20 samples every second, as shown by the green plot in Figure 9.4.

For delay analysis purposes, the SSG implementation requires to receive an accurate time-stamp from the GPS antenna through the NI 9467 GPS synchronizer, as well as time-stamps from each of the individual PMUs. The SSG communication functions run on the real time processor of the cRIO 9081, while the Spartan 6 FPGA inside the cRIO runs the program to receive the GPS data. This part of the hardware is shown in the block diagram of Figure 9.2. It can be seen that GPS antennas also provide the 3 PMUs running on the cRIO 9082 and cRIO 9068 devices with their own time references via three NI 9467 GPS acquisition modules.

The reference timing diagram for the experimental setup is shown in Figure 9.5. PMU_1 measures data at time t_1, whose phasors have a corresponding time-stamp T_{S1}. PMU_2 measures data at time t_2, with corresponding time-stamp T_{S2}. PMU_3 measures data at time t_3, with corresponding time-stamp T_{S3}. All these measurements are sent to the SSG. The SSG will read any data at time t_P with corresponding time-stamp T_P. T_P will be *greater* than T_1, T_2 and T_3, because the SSG receives data that will be imminently delayed. For a given data snapshot, three new variables are defined as follows.

$$\Delta_1 = T_P - T_{S1}, \Delta_2 = T_P - T_{S2}, \Delta_3 = T_P - T_{S3} \tag{9.1}$$

In the sequel, Δ_1 Δ_2 and Δ_3 are analyzed to characterize the performance of the

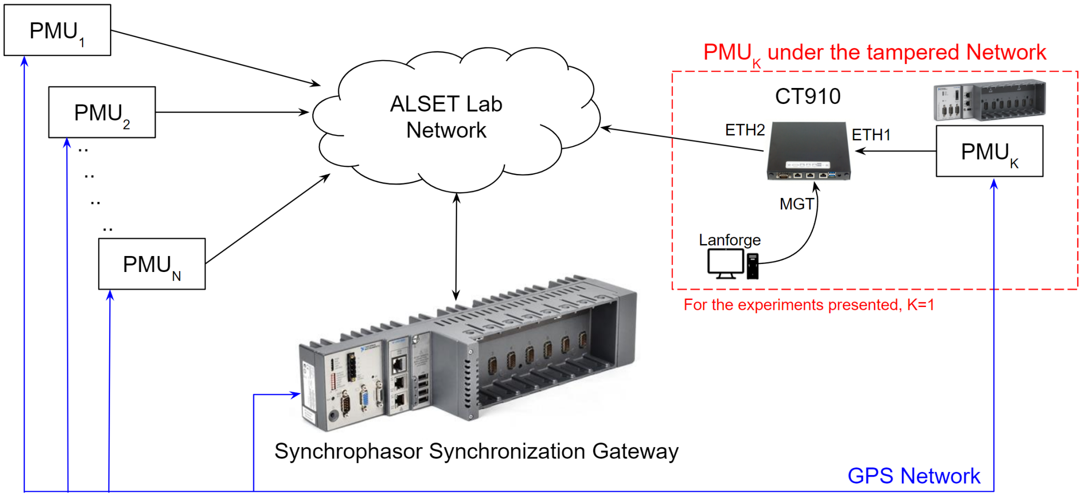

PMU$_1$
PMU$_2$
PMU$_N$
ALSET Lab Network
PMU$_K$ under the tampered Network
CT910
ETH2
ETH1
PMU$_K$
MGT
Lanforge
For the experiments presented, K=1
Synchrophasor Synchronization Gateway
GPS Network

Figure 9.6: Networking Infrastructure to Tamper a PMU-SSG Network Utilizing the CT-910 Network Traffic Generator

SSG setup. The PMU with the shortest delay can be found very easily. When more than three PMUs connected to the SSG, it would be necessary to implement a sorting algorithm to find the shortest-delay PMU connected to the SSG. In neutral test conditions Δ_1, Δ_2 and Δ_3 exhibited similar characteristics. NI's Network Published Shared Variable (NPSV) library was used to measure and compute these delays. With this test conditions, the three aforementioned delays were characterized by the statistic reported in table 9.1. Afterwards, the experiments were performed without the NPSV infrastructure as well.

Table 9.1 clearly shows that the network is largely uniform across all three communication links. Hence, the next experiments on delay were performed only for PMU 1, as all three PMU-SSG links will behave similarly under the studied network conditions.

9.3.2 Test case 2: Evaluating the SSG Under Network Delay

To test the proposed SSG prototype's reliability, a configurable network traffic generator CandelaTech CT-910 [140] was used. Using the CT-910 device, additional delays were

Table 9.1: Delay Statistics for All the PMUs (NI Network Published Shared Variable Used for Measurements)

	Δ_1	Δ_2	Δ_3
Mean (s)	0.0481	0.04726	0.04824
Standard Deviation (s)	0.0199	0.02065	0.01945

introduced in between the PMU and the prototype SSG. In figure 9.6, the insertion of the CT-910 network traffic generator within the existing network is graphically shown. The network traffic generator is capable of introducing a user-configurable delay between the two networks. The CT-910 runs a standard Linux operating system, and is capable of modifying its own internal communication network architecture dynamically. This can be performed from a networked GUI, or by directly running shell-commands on the operating system of the network traffic generator.

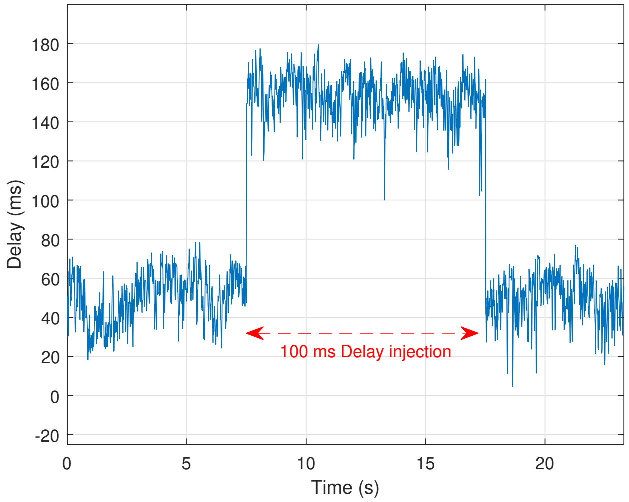


Figure 9.7: Example of 100 ms Delay Injection in the Communication Link

In this particular test, artificial delays were introduced in between PMU_1 and SSG prototype. To introduce artificial delays, CT910 Network Traffic Generator was reconfigured. The CT910 has a configurable GUI which can tamper with the network between the two ports where PMU_1 and SSG prototype were connected, respectively. In Figure 9.8, the GUI is for the network-reconfiguration software is shown. It can be seen that the network between the two ends are closed (black) and the network is functional (45M-clean). It can incorporate user specified network traffic, as shown by the green lines (`45M-random` and `45M-impair`). In Figure 9.7, a sample network impairment (by a delay injection of 100 ms) for a duration of 10 seconds is shown.

Figure 9.9a shows the plots with variable amounts of network delay injected in between the PMU_1 and the SSG prototype. It is interesting to note that, with 50 ms delay injected (blue plot), the actual average PMU_1 to SSG delay is close to 90 ms. This gives an indication that the network itself has an intrinsic delay of about 40ms. To validate this hypothesis, from

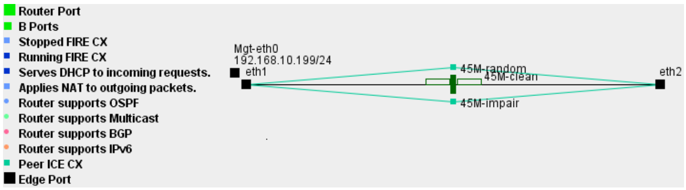
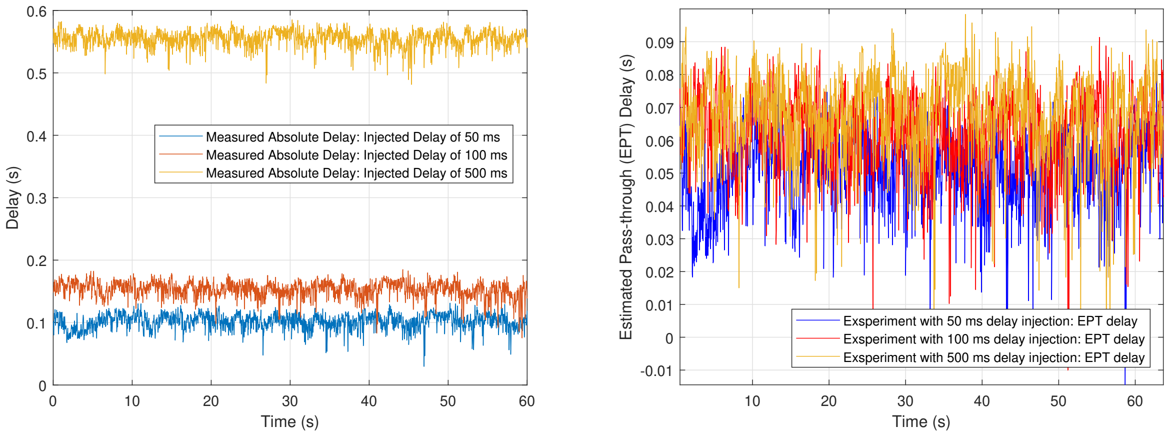


Figure 9.8: Tampering the Network Between PMU_1 and the SSG



(a) **(b)**

Figure 9.9: PMU_1 to SSG Delay Under Varying Injected Delay: (a) Measured Absolute Delay Between PMU_1 & SSG After Variable Additional Network Traffic Injection (b) Estimated Pass-through (EPT) Delay for PMU_1 to SSG Delay After Variable Additional Traffic Injection

all the delay plots of Figure 9.9a, the injected delay is subtracted (i.e. if the injected delay is 500 ms, 500 ms is subtracted from all observations). This newly obtained delay-parameter is refered as the Estimated Pass-through (EPT) delay from here onward.

The EPT delays for three different values of fixed injected delay (50ms, 100ms, and 500ms) are plotted in Figure 9.9b. It can be seen that, the intrinsic delay of the path is fairly similar for all three configurations. Upon additional experimentation, it was also observed, that even without any injected delay, the network exhibited an average PMU_1-to-SSG delay of 47 ms over a run-time of 4 hours.

To investigate further, histograms for all the delays (both *Measured Actual values* and *EPT* values) with varying injected delays are shown in Figure 9.10, along with the delay of the system when the network is set in pass-through mode (i.e. no injected delay). When normal distributions were fitted through the histograms of their corresponding *Estimated Pass-through (EPT)* delays, it was be observed that, the histogram gets wider as the injected

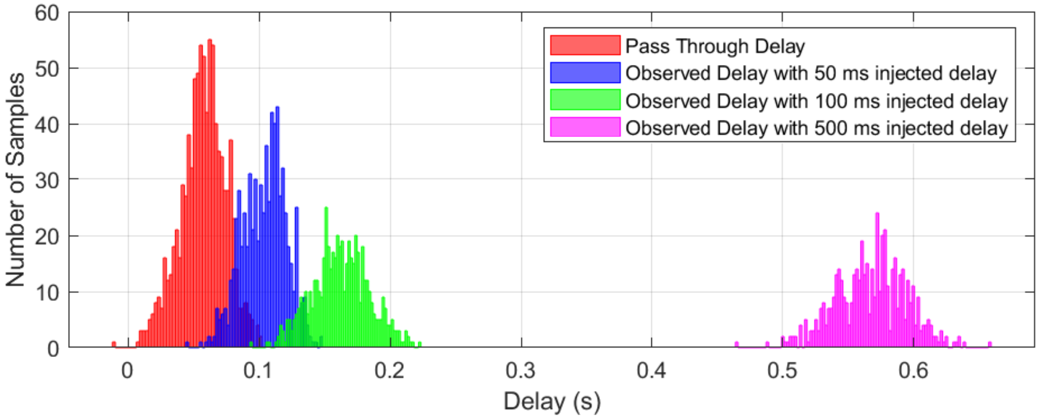


Figure 9.10: Histograms for Measured Delay in Different Network Traffic Injections

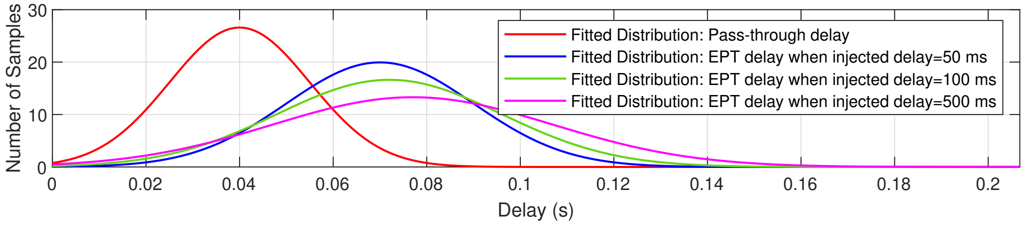
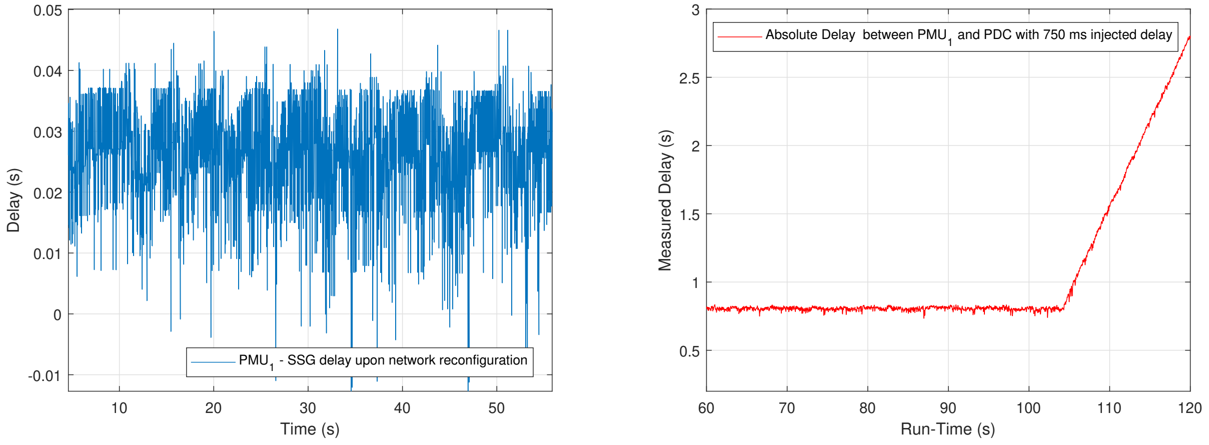


Figure 9.11: Fitted Distributions of the Delay-Histogram for Different Values of Injected Delays

delay increases. Figure 9.11 shows the fitted distributions of the EPT delay-histograms for these observations.

In order to reduce this delay, two changes were made to the network (i) the PMU-SSG interconnect was reconnected with crossover cables, (ii) all instances of NI Network Published Shared Variables were removed. Once, these changes were made, the actual time of execution of the SSG algorithm and data-transmission protocols should be the only component present in the total delay.

However, without the presence of shared variables, the measurement procedure becomes more difficult. To measure precise time-stamps in this new setting, time-stamps are converted to numbers and sent through the “analog” channels of the PMU which are part of the C37.118.2 protocol. However, it was observed that the “analog” field of the C37.118 protocol is not the most efficient and reliable method to transmit *timestamps*. In this new setting, the delay between the PMU_1 and the SSG prototype is observed. A standard plot of the reduced delay is shown in Figure 9.12a. Table 9.2 exhibits the improvements observed in



(a) (b)

Figure 9.12: Real-time Delay Measurement With No Injected Delay, and With Very High Injected Delay: (a) PMU_1 to SSG Delay Without Using NI Network Published Shared Variable (b) Absolute PMU_1 to SSG Delay With Very High Delay-injection (750 ms)

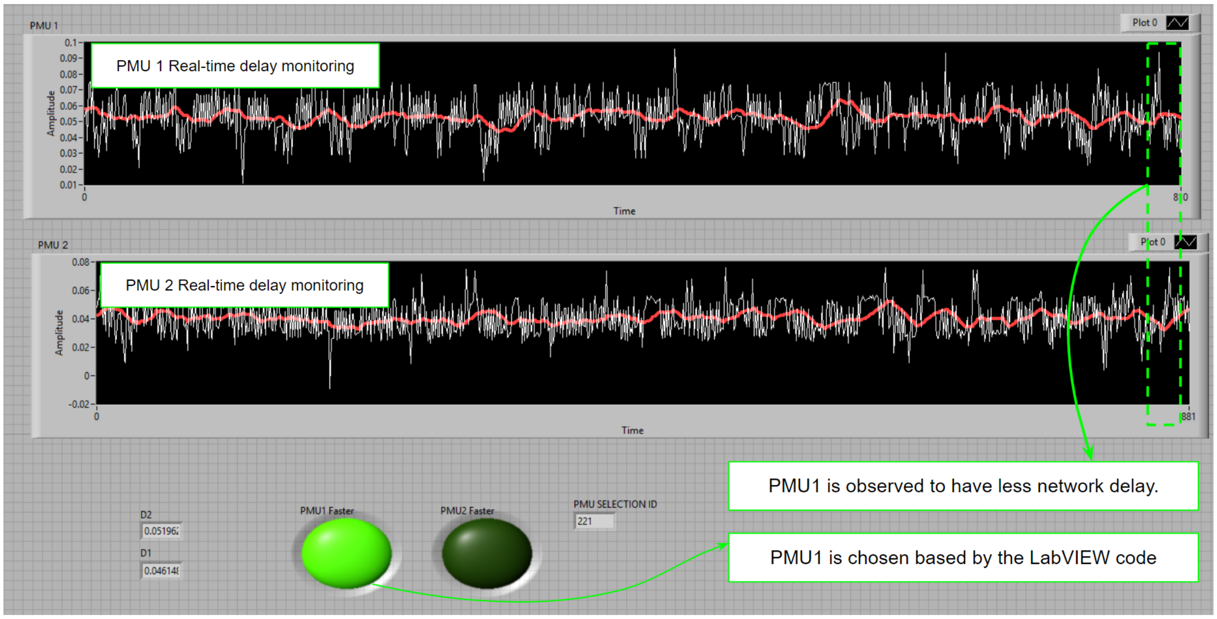
terms of timing when compared to the results in table 9.1.

In order to stress the system, the injected delay was increased from 500 ms to 750 ms. As shown in Figure 9.12b, it was observed that, with such a high injection of delay, the network becomes unstable and breaks the communication link between the PMU and SSG after a few seconds. In fact, this phenomenon was observed, whenever the injected delay was more than 600 ms. It can thus be concluded that 500 ms is the maximum possible delay that can be injected into the network without compromising the reliability of the communication system.

9.3.3 Test Case 3: Dynamic Measurements Through Tampered Networks

In real life, different PMUs in a power system will have different latencies and different link delays from the location of the SSG itself. In this section, the SSG identifies the **minimum-delay-PMU** in real time, and chooses to utilize the data received from it only. The idea behind such an application is to guarantee the action of a controller networked in the system for regulating the energy sources by using signals with the best delay distributions.

Figure 9.13 shows the basic LabVIEW based GUI for this real-time dynamic PMU selection framework for two PMUs. In order to display which PMU is chosen, two LEDs are added to the front end GUI. The two waveforms are the real-time SSG to PMU network delay measurement. The white lines represent the instantaneous network delay of that PMU from the SSG, while the red lines represent a smoothened trend of those delay measurements based on a simple averaging filter. This filter can be replaced by a more sophisticated filter in future. It can be observed from the GUI that even though the delays of PMU_1 and PMU_2 are within the same range, instantaneous delay is lower for PMU_1, and thus PMU_1 is chosen - which is demonstrated by glowing one of the two LEDs placed within the GUI.



PMU 1 Real-time delay monitoring

PMU 2 Real-time delay monitoring

PMU1 is observed to have less network delay.

PMU1 is chosen based by the LabVIEW code

D2 0.051962

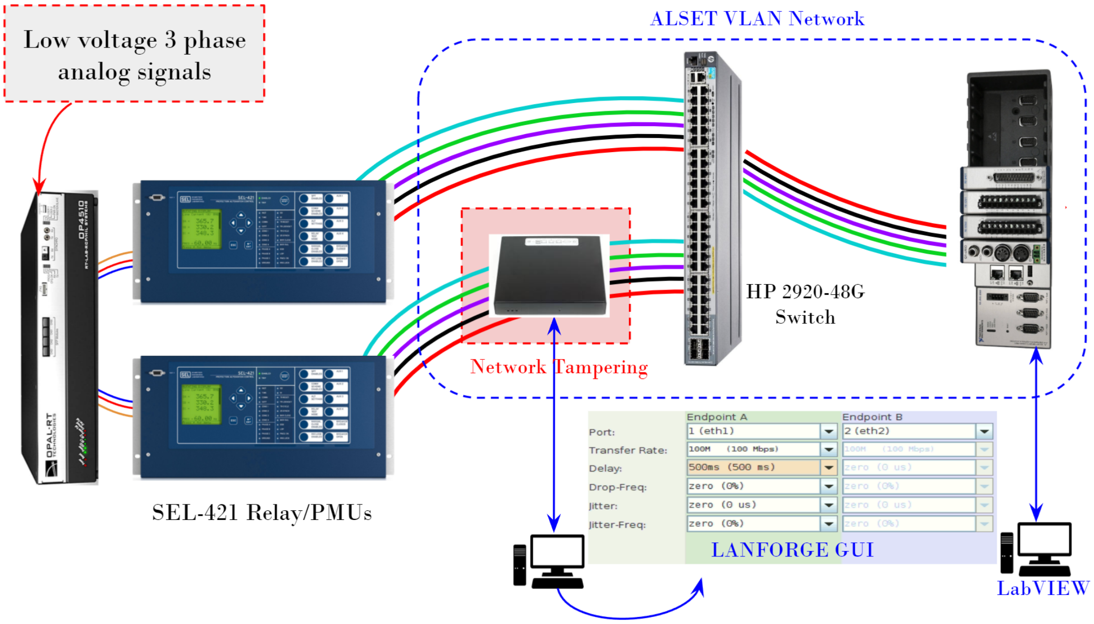
D1 0.046148

PMU SELECTION ID 221

Figure 9.13: Dynamic PMU Selection Based on the SSG

To make this experiment more realistic, the PMUs used were SEL-421 protective relays and the SEL-421 relays were fed analog signals through the Opal-RT 4520/5030 real-time simulator's analog output ports. A simple 3 phase electrical grid was being simulated on the real-time simulator and voltages and currents measured within this electrical grid was scaled down to 2V analog outputs. The SEL-421 relays are electrically connected to these outputs through their low voltage AMS (adaptive multichannel source) interface.

The network-tampering device (*CT910*) was placed between one SEL-421 relay (streaming synchrophasor data) and the SSG, and was configured to introduce a latency of 1000ms. Since, this value is very high, this experiment can reliably performed for a few seconds at a time only. The same electrical signals were fed to a second SEL-421 relay streaming synchrophasor data directly to the SSG **without** the *CT910* network-tampering device in between. Both these synchrophasor streams were read in real-time on the NI cRIO run-



	Endpoint A	Endpoint B
Port:	1 (eth1)	2 (eth2)
Transfer Rate:	100M (100 Mbps)	100M (100 Mbps)
Delay:	500ms (500 ms)	zero (0 us)
Drop-Freq:	zero (0%)	zero (0%)
Jitter:	zero (0 us)	zero (0 us)
Jitter-Freq:	zero (0%)	zero (0%)

Figure 9.14: Experimental Setup to Use the SSG For Reading Data From PMU Streams With Different Network Delays

ning the SSG model. The voltage source that feeds analog inputs to the SEL-421s is easily configurable through Simulink. The experimental setup is illustrated in Figure 9.14.

A step change of 0.5 Hz was applied to the frequency of this source for a window of 2 seconds. This step-change in frequency is reflected in both the synchrophasor streams. Observe that due to the tampering, the step-change is shown 1000 ms apart because of the injected latency via the CT910. This allows to observe and quantify the tampering the network for one of the PMUs. This experimental observation was monitored in real-time on LabVIEW GUI (similar to Figure 9.13) and the frequency measurement results are logged and presented in Figure 9.15. Notably, this setup enables the user to trace the network delays accurately by utilizing its on-board GPS module. The timestamp received through this GPS module can be compared with the time-stamp embedded in the PMU data-stream

Table 9.2: Delay Reduction After Removal of NI Network Published Shared Variable Infrastructure

Test-setting	μ	σ
With NI Network Published Shared Variable	0.0461	0.0190
Without NI Network Published Shared Variable	0.0265	0.0208

to compute the actual network delay.

9.3.4 Test Case 4: Extending the Architecture in SSG and Control of a Micrgrid Model in Real-time

In this experiment, the SSG is modified and upgraded to incorporate part of the control system for regulating the power output from the battery energy storage system (BESS) based DER. This control system consists of a proportional-integral controller that determines the

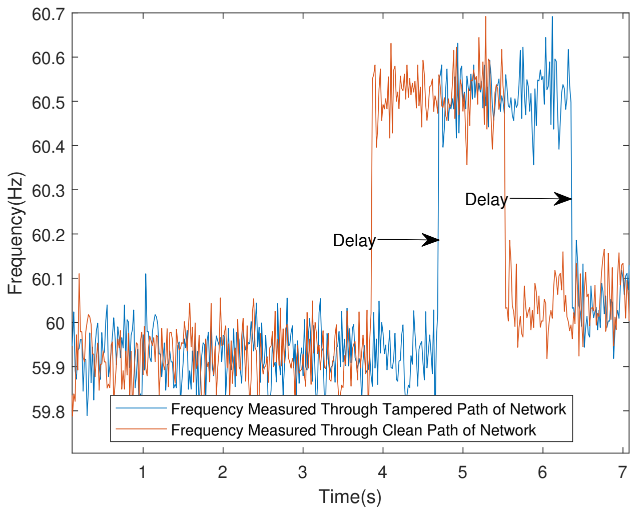


Figure 9.15: Tracking the Frequency of the Synchrophasor Data Received Through Two Networks With Different Network Delays

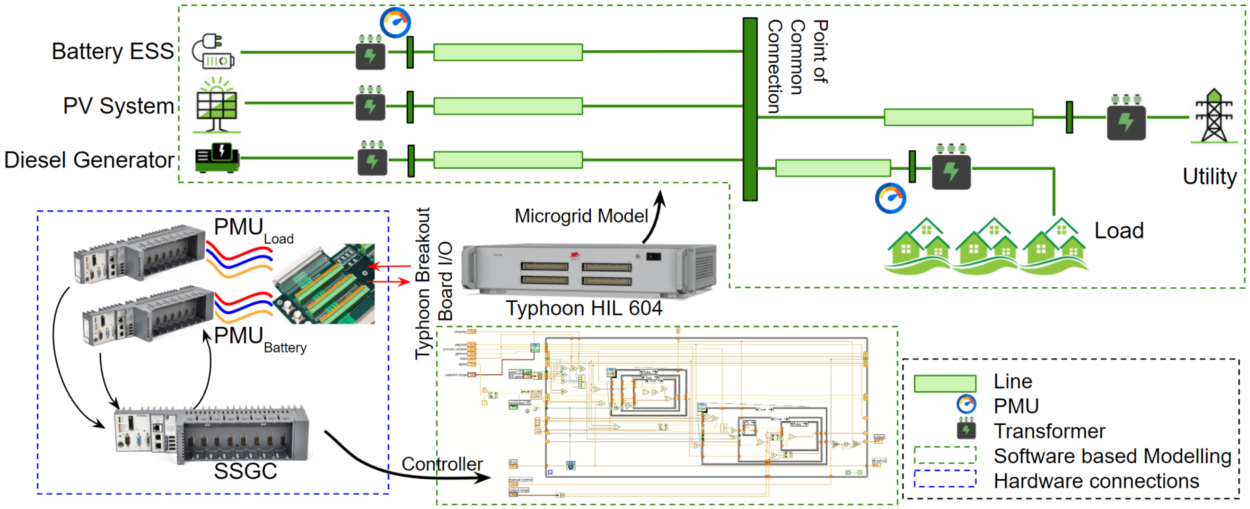
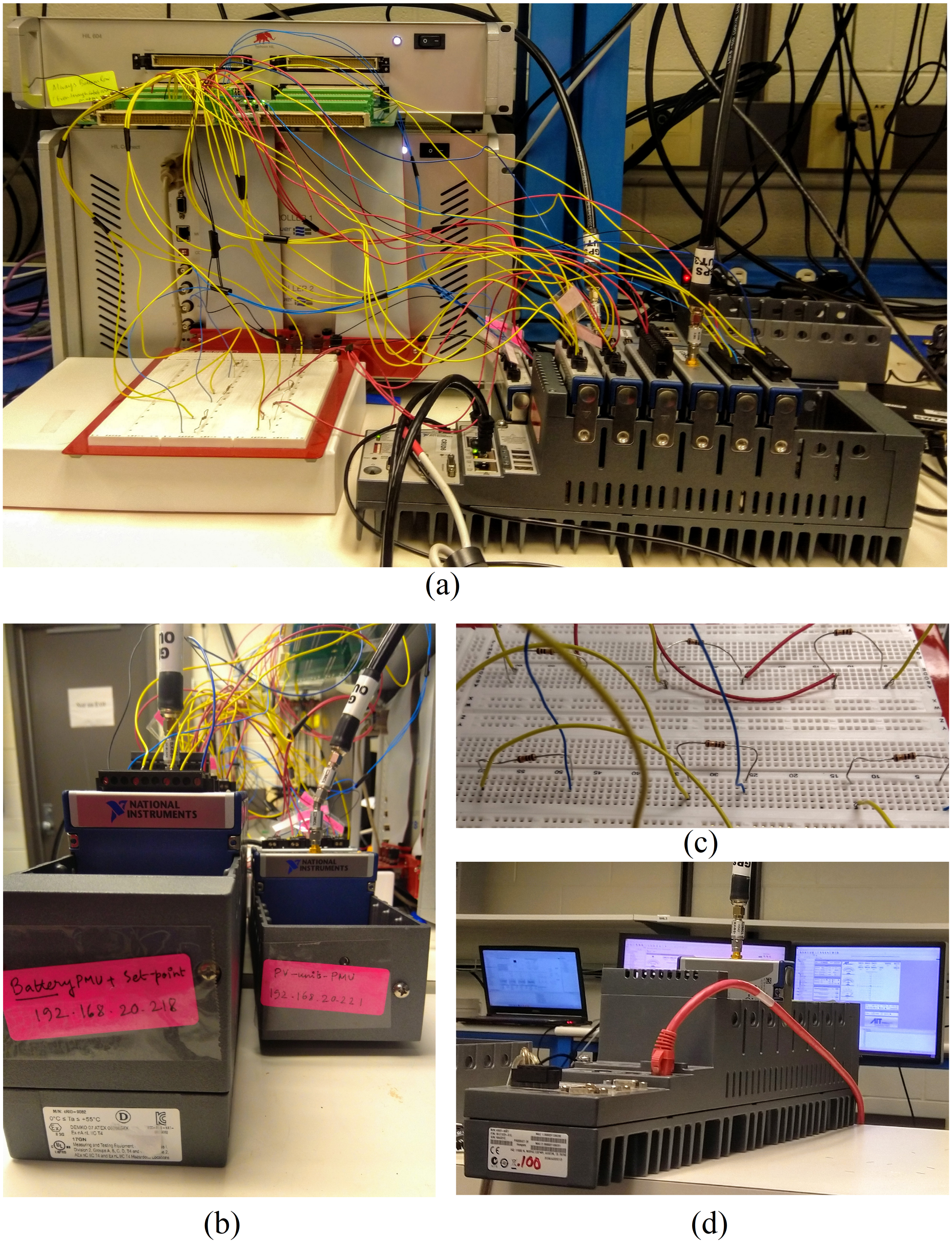


Figure 9.16: The Interaction Between the Microgrid Model on Typhoon HIL 604 and the SSGC Running on cRIO



(a)

(b)

(c)

(d)

Figure 9.17: Hardware Arrangement for Testing Microgrid With the SSGC Hardware: (a) Connection Between the Microgrid & the PMUs, (b) PMUs Receiving Timing Information, (c) Conversion of RT Low-voltage Signals into Current Signals, (d) SSGC Connected Remotely to the Microgrid

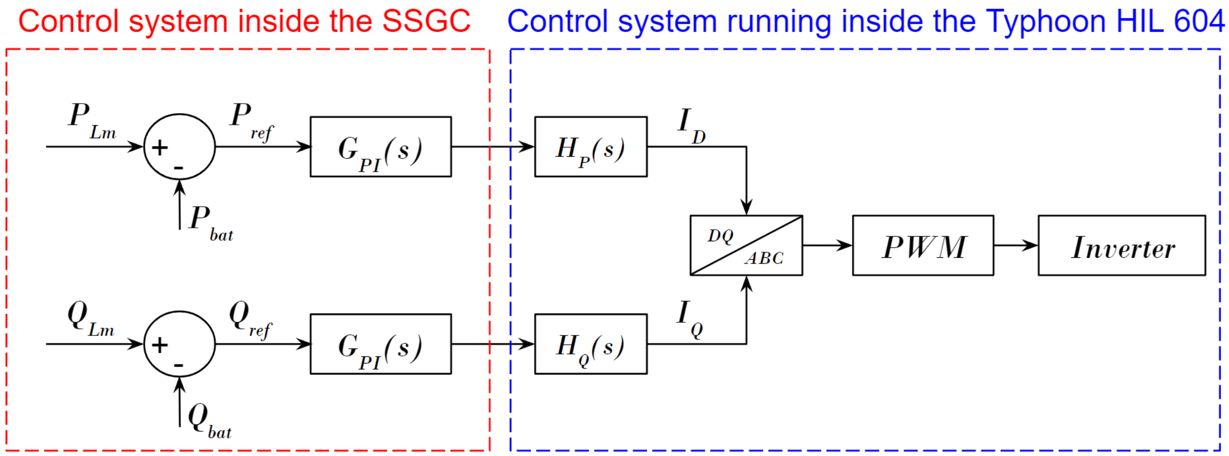


Figure 9.18: Microgrid Controller Utilized in the Experimental Setup

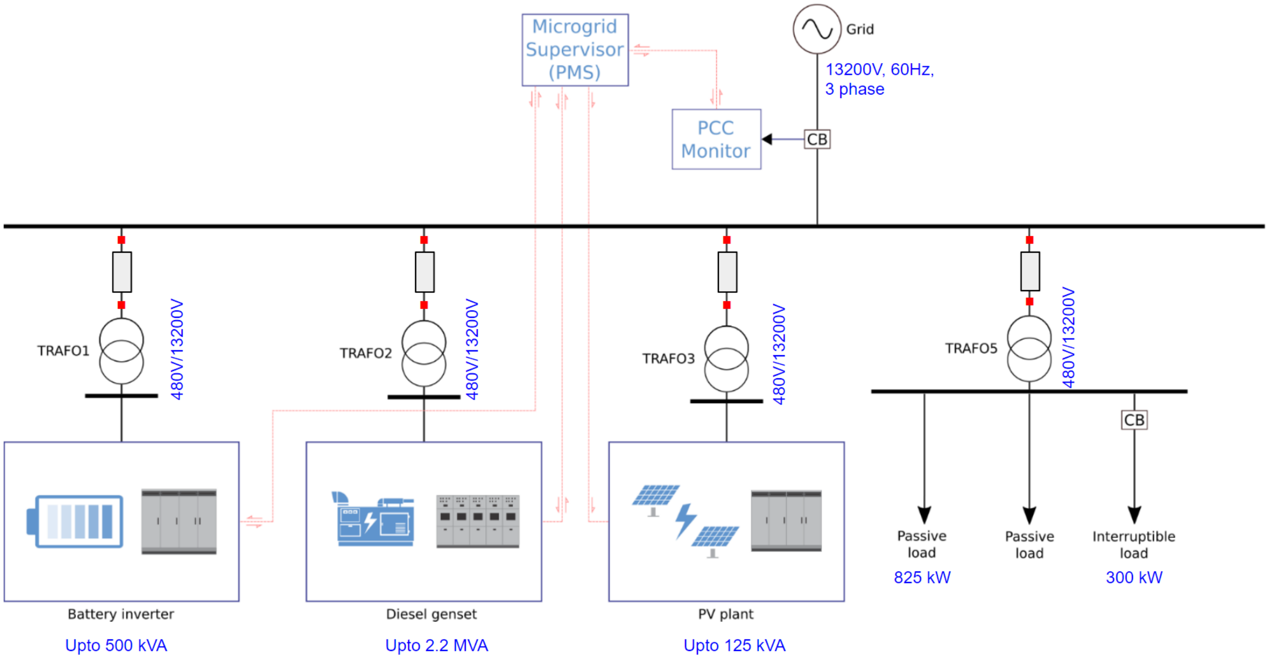


Figure 9.19: Simplified Microgrid Model Implemented in Real-time

set-point to be utilized in the local controller inside the BESS. The inputs to this controller were the active power P and reactive power Q computed based on the current and voltage phasors the SSGC receives from the incoming PMU streams. These values are then utilized to compute a control output which was fed back to the real-time simulator's input in order to control the Li-ion battery based energy storage system. The remainder of this section describes the software & hardware components of the experiment, explains the control action,

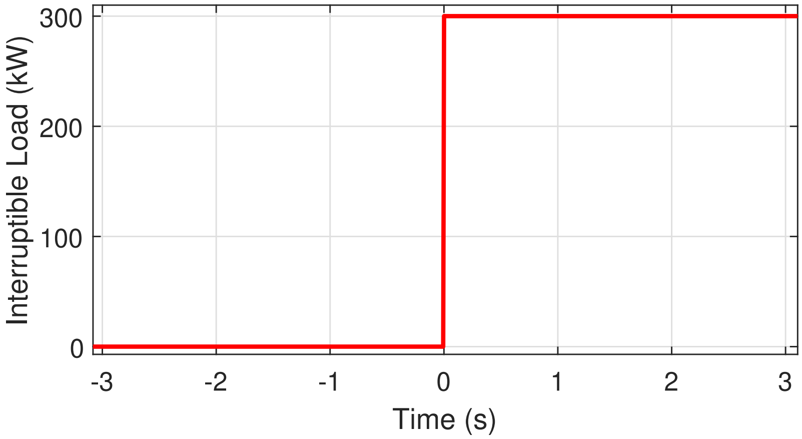


Figure 9.20: 300 kW Load Injection by Switching the Interruptible Load

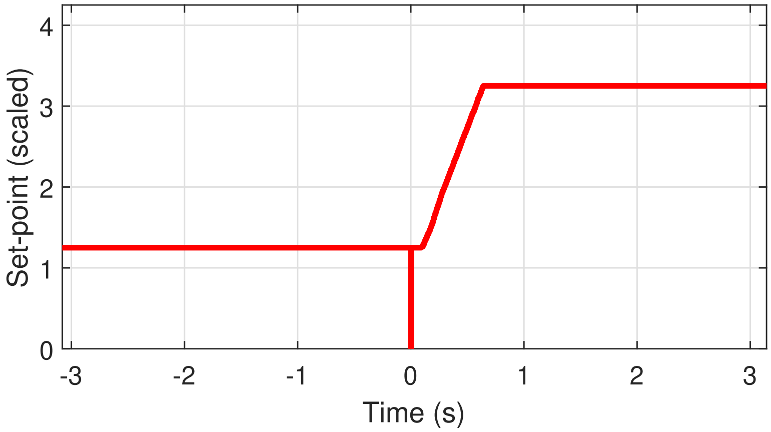


Figure 9.21: Response at the Output of the PI-controller Inside SSGC

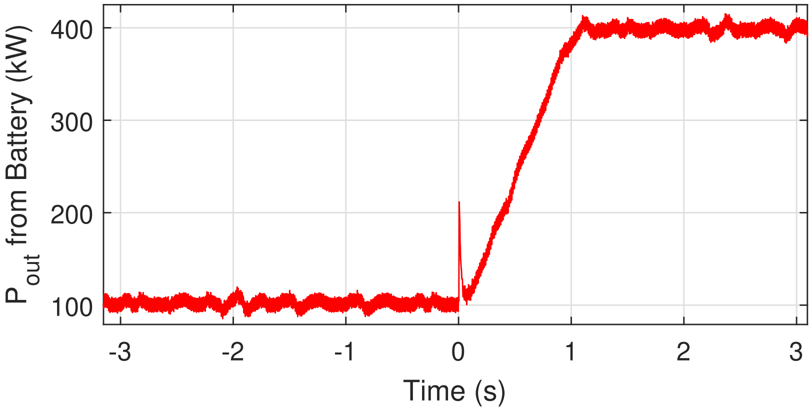


Figure 9.22: Active Power Output from the Battery Energy Storage System (BESS)

and evaluates the robustness in this architecture under varying network conditions.

9.3.4.1 Software and Hardware Components for Developing the SSGC and Testing it With a Microgrid Model

The framework of the SSGC is similar to that introduced in section 9.2. However, the host-side code running on the real-time operating system is modified to include a PI controller which would compute its set-point and process-variable by unwrapping the current and voltage phasors it receives. Notably, the controller implemented was a standard $P-Q$ type controller, and real-time retrieval of the current and voltage phasors are sufficient to compute the active and reactive powers in real-time.

The currents and voltages were generated by simulating a microgrid model developed in the Typhoon HIL 604 real-time simulator. This microgrid model contained a PV unit consisting of a custom PV module (as introduced in chapter 5), and a battery energy storage system (BESS) consisting a Li-ion model (as demonstrated in chapter 7), a diesel generator, utility source, and a configurable load which is shared by all these different DERs and the utility. By definition, a microgrid can operate in both grid-tied and islanded modes. In grid tied mode, the microgrid is connected to the utility and in the islanded mode the microgrid is disconnected from the utility and the load is only shared by the DERs within the microgrid.

For demonstration purposes, the Li-ion battery based energy storage system was controlled by the SSGC hardware in the current experiment. In some real-time simulators (e.g. Opal-RT) it is possible to stream synchrophasor data from within the simulator without connecting any physical PMUs to the hardware. However, even though the 2021.2 and 2021.3 releases of the *Typhoon HIL control center* toolkit have dedicated library components for streaming C37.118 data, it was discovered upon experimentation that the current implementations for such blocks are unstable and unreliable for communicating synchrophasor data to external hardwares such as the SSGC. Thus, additional PMUs (based on the design reported in chapter 2, and chapter 3) were connected to the low voltage analog outputs of the real-time simulator. Importantly, these PMU designs require both voltage and current inputs, whereas the Typhoon HIL 604 is capable of generating only voltage signals. Thus, it is required to design voltage to current conversion circuits consisting simple resistors as shown in Figure 9.17.(c). The connection between the real-time simulator and the PMUs is shown in Figure 9.17.(a). Figure 9.17.(b) demonstrates how the individual PMUs obtain GPS signals, and

Figure 9.17.(d) shows the SSGC operating on a remote location, connected to the network in order to receive real-time PMU data. The microgrid model used for simulation was based on the architecture reported in [141]. However, the individual components were simplified and the wind power plant was eliminated from the original model for simplicity. The circuit describing the model is shown in Figure 9.19.

9.3.4.2 Implementation of the Control Architecture for the Microgrid Model

The proposed control architecture is suitable for taking advantage of the hierarchical structure of standard microgrid control infrastructures described in [13]. It has been discussed that, the fastest control operations related to current and voltage controls of the DER inverters comprise of the primary class of control actions. The secondary class of control actions consist of algorithms to determine the power flows to and from the different DERs. This class of control, has much slower dynamics compared to the first class of control actions. The tertiary class of control actions consist of long-term optimizations, power system economics and other decision making functionalities which incorporate the long-term trends of the various environmental, and electrical parameters.

The hardware reported in this chapter is most suitable for incorporating control actions which fall into the second class and the third class. For demonstration, a second class control action (to control the power output of the battery energy storage system (BESS) based DER) is illustrated in details. The set-point for control action is based on two parameters P_{Lm} which represents the measurement from the load side, and P_{bat}. It is assumed that the P_{PV} (Active power output from PV system), P_{Uti} (Active power dispatch from the utility), and P_{DSG} (Active power output from the diesel generator) are kept constant. In this situation, the parameter P_{ref} will depend only on the total load consumption P_{load}. This can be explained from the block diagram shown in Figure 9.18.

For experimentation, the load is increased in a step by turning on the *interruptible load* as shown in Figure 9.19. Initially, a fixed load of 825 kW was supported by the PV unit (125 kW), Diesel Generator (500 kW), and the utility (100 kW). This makes the initial dispatch for the BESS-inverter to be fixed at 100 kW. With the system running in this condition, a step increase of 300 kW in load was provided externally. The control system must be designed in such a way that, this change in load is reflected in the BESS-inverter, and its dispatch should be increased from 100 kW to 400 kW. It is important to note that, the portion of

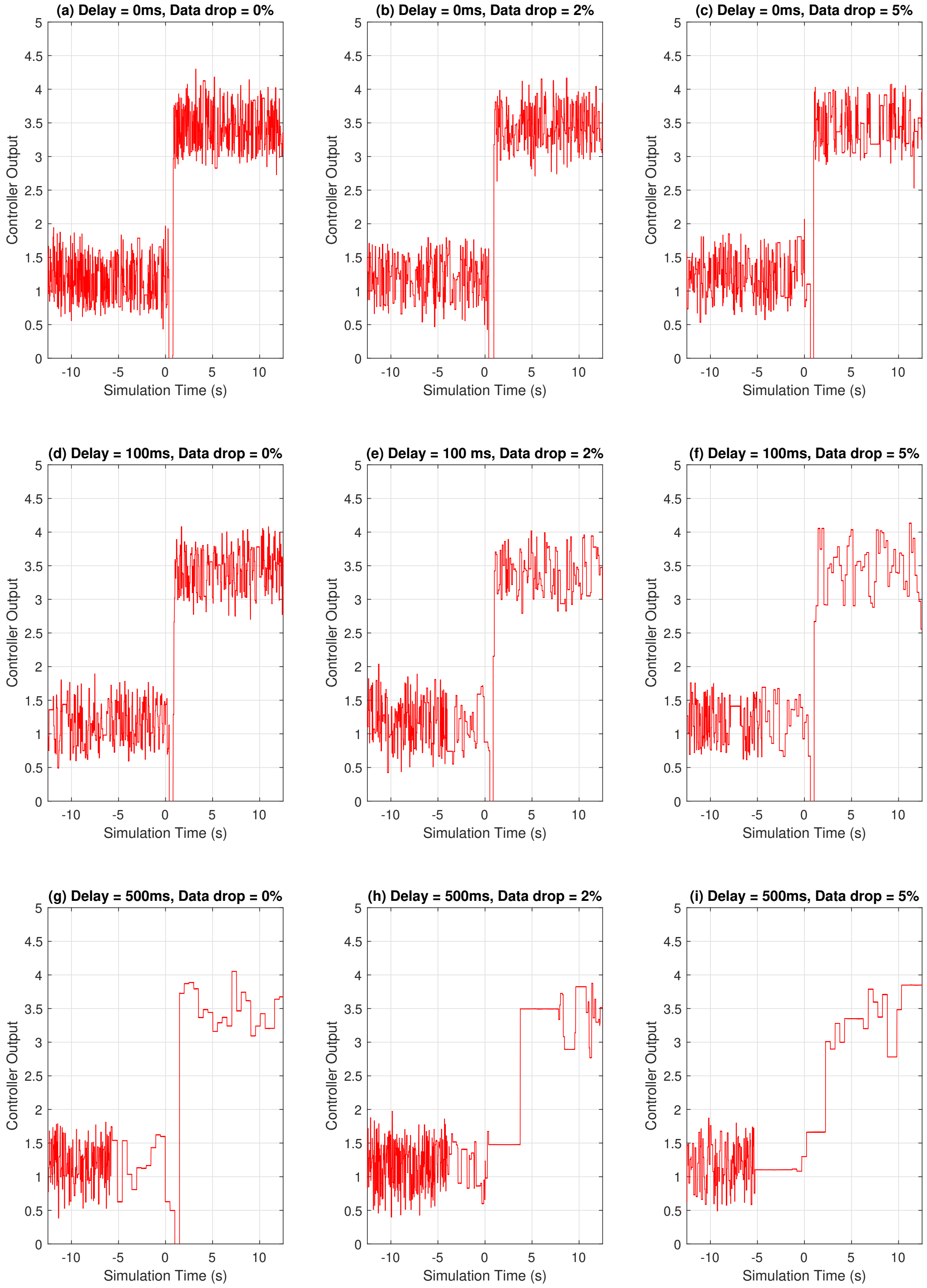


Figure 9.23: Control Signal Received in the Typhoon HIL Microgrid Model From the SSGC Under Varying Network Delay and Data-drop Rates

the controller (in Figure 9.18) within the dashed red-box is the only portion implemented within the SSGC hardware. This portion is capable of utilizing synchrophasor measurement data obtained from the PMUs placed at the load and at the BESS. The PMU data is utilized to compute the active and reactive powers (not demonstrated in this experiment), which are then used for calculating a new set-point (by using the PI controller block G_{PI}) for operating the BESS. This set-point is then utilized by the internal control algorithm (implemented locally inside the BESS) for controlling the individual current and voltage output of the inverter inside the BESS. This portion of the control system needs to be implemented locally within the BESS model of the real-time simulator, because it requires faster dynamic performance and needs to be capable of generating high-frequency switching sequences for the individual semiconductor switches in the inverters. Figure 9.20 shows the 300 kW manual load-injection in the system. The SSGC incorporates a PI controller onboard. The output of this PI controller - which provides a set-point for the local controller for the BESS to operate - is shown in Figure 9.21. This measurement is taken from the SSGC side. There maybe jitters, data-drops or imperfections if this same data is re-measured from the RT-simulator end where the simulator receives the set-point from the SSGC. In Figure 9.23, the BESS power output response is shown. It can be observed from this figure, that the power output of the battery increases from 100 kW to 400 kW to cover for the step increase in the load.

9.3.4.3 Evaluation of the Proposed Control Architecture Under Varied Network Conditions

In this part of experimentation, the CT910 device is utilized to tamper with the network between the SSGC and the PMUs. For testing the reliability of the control architecture under varying network conditions, the network delay was varied from 0 ms to 500 ms, and the data-drop rate was varied from 0% to 10%. Under these conditions, the microgrid was simulated alongside the SSGC based control infrastructure. Two sets of experiments were performed to investigate the robustness and reliability of the proposed control system.

- Controller performance under varied conditions: In this test, the experiment performed in section 9.3.4.2 is repeated for varying network delays and varying data-drop rates. The controller regulating the power output of the inverter inside the BESS, is set to react to a step-increase of 300 kW in the load end. However, its performance is

Table 9.3: **Testing the Resilience of the SSGC Framework Under Varying Network Conditions**

Drop Rate (%)	0%	0.5%	1%	2%	5%	10%
Delay						
0 ms	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$3_{sustained}/10$
50 ms	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$9_{sustained}/10$	$0_{sustained}/10$
100 ms	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$9_{sustained}/10$	$7_{sustained}/10$	$0_{sustained}/10$
200 ms	$10_{sustained}/10$	$10_{sustained}/10$	$10_{sustained}/10$	$6_{sustained}/10$	$2_{sustained}/10$	$0_{sustained}/10$
500 ms	$10_{sustained}/10$	$10_{sustained}/10$	$5_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$
750 ms	$1_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$	$0_{sustained}/10$

expected to deteriorate under stressed network conditions. The results of these tests are summarized in Figure 9.23. It can be observed that the analog output of the remote controller loses a lot of resolution under higher network delays and higher data-drop rates. However, as discussed earlier in section 9.3.4.2, the control-objective of the SSGC does not involve any management of the high frequency dynamics of the system. Thus, in short-term the SSGC-driven control architecture can sustain itself even while operating within a tampered network.

- Resilience test under varied conditions: For this test, the SSGC is set to operate freely under varying network conditions, and whether or not it can sustain itself for **longer periods** of time, is tested. The network was tampered by introducing network delay and data-drop. Under these conditions, the network was kept running for 10 minutes. After 10 minutes, notes were taken on whether the SSGC is still receiving all the PMU streams successfully, and if the real-time simulator is still receiving controller output. This observation is taken 10 times, for each network conditions. The summary of these results is shown in Table 9.3. It can be seen that the network delay and data-drop can both adversely effect the robustness of the SSGC framework. In fact, in a situation where the SSGC is subjected to both high network delay coupled with high data-drop rate, the SSGC is almost certain to be unable to sustain itself for a long period of time.

9.4 Directions for Future Experiments

It needs to be noted, that only one scenario for controlling the DERs within a microgrid is demonstrated in this chapter. In that scenario, the battery is covering for a step increase in the load. While, this experiment is an important 'proof of concept'-study, additional experimentation is crucial before implementing the proposed architecture for controlling real-world DERs. Following are some of the possible future directions for the current experiments.

9.4.1 Adjustments to Adhere to the Constraints of the Battery

In real world, the behaviour of a BESS based DER in a power system is very complex and the operation of the controllers are restricted by the physics of the battery. In fact, batteries utilized in energy storage systems are limited by their slow response times, both during charging and discharging operations. As reported in [142], the average response time for a battery energy storage system during charging is about 2.2 seconds and that for discharging it is about 0.6 second. Keeping these numbers in mind, the battery cannot be subjected to rapid movements or perturbations in the load. Such high frequency variations, if kept unfiltered, would rapidly increase the switching and would generate excessive heat, thus compromising the health of the batteries. Implementing these precautions within the control system, is well inside the dynamic range of operation for the SSGC hardware. Significant future effort is required in this domain.

9.4.2 Alternate PMU Placement and Utilizing Different System Variables for Controlling DERs

Because one of the main objectives for the current research is to minimize the number of PMUs in constructing a functional control system, it is crucial to explore different locations of the PMUs. One such direction of experimentation should be, instead of using measurements from the load utilizing the measurements from the utility side to design the BESS control system. This would require a operational PMU connected at the utility side. This configuration should also explore, how the phase angle measured by the PMU placed at the utility end can be used to implement a control system for controlling the active power flow of the DERs. This is a well-studied approach, and the usage of phase angle differences as the control variable for active power management has been known to be utilized for stable, reliable and robust control systems in traditional power systems featuring synchronous machines. It should be an important exercise to export that concept into the paradigm of networked control for DERs in microgrids. Since PMUs readily provide angle measurements, this approach can reduce the design-complexities of the overall control system.

9.4.3 Incorporation of the Battery State-of-charge (SOC) Within the Control Action

While managing BESSs, the state of charge (SOC) of the battery is a crucial parameter. Depending on SOC, the battery is often put into the charging or the discharging modes. In this chapter the SOC is assumed to be high enough, so that the battery can reliably operate in discharging mode, i.e. it can feed active loads. In real systems, this may not be the case. So, an additional control loop must be designed to determine the SOC state and ensure the safe and reliable operation of the BESS.

9.5 Conclusions

This chapter proposes a new architecture an implementation for a true real-time Synchrophasor Synchronization Gateway (SSG), and extends its functionalities to incorporate control-algorithms targeted for DERs in microgrids. The goal of this hardware is to replace a PDC in PMU-based real-time control applications, provide supplementary control functionalities, and decrease the latency of the overall networked control system. By incorporating a hierarchical decentralization, the proposed framework can increase the resilience and robustness for synchrophasor-driven control algorithms of DERs, especially in microgrids.

CHAPTER 10
SUMMARY AND FUTURE WORKS

This chapter aims to recapitulate the research presented in the previous chapters and outline some of the possible directions of future research based on the work presented in this thesis.

10.1 Summary

To summarize the contributions of the thesis effectively, the problem-statements introduced in section 1.2 are answered based on the results presented through chapters 2-9.

- *Are there definite hardware-specifications which must be met to implement PMUs on a certain FPGA?*

  For the Spartan family of Xilinx FPGAs, the expression $N_{LUT} \times N_{LUT-size}$ was observed to be a reliable metric to determine whether or not certain PMU designs can be compiled on a certain FPGA unit. For corss-verification purposes this metric was reevaluated for more standardized designs such as ALUs and FFT blocks.

• Portions of this chapter previously appeared as: P. M. Adhikari, H. Hooshyar, R. J. Fitsik, and L. Vanfretti, "Precision timing and communication networking experiments in a real-time power grid hardware-in-the-loop laboratory," *Sustain. Energy Grids & Netw.*, vol. 28, Dec. 2021, doi: 10.1016/j.segan.2021.100549.

• Portions of this chapter previously appeared as: P. M. Adhikari and L. Vanfretti, "Delay analysis of a real-time hard reconfigurable synchrophasor synchronization gateway," in *IASTED Int. Conf. Control & Optim. of Renewable Energy Syst.*, Annaheim, CA, USA, Dec. 2019, doi: 10.2316/P.2019.859-008.

• Portions of this chapter previously appeared as: H. Hooshyar, L. Vanfretti, J. H. Chow, R. Fitsik, P. M. Adhikari, "ALSET lab: Designing precise timing and communications for a digital power grid laboratory," in *IEEE Power & Energy Soc. General Meeting*, Montreal, QC, Canada, Aug. 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281808.

• Portions of this chapter previously appeared as: B. Azimian, P. M. Adhikari, L. Vanfretti, H. Hooshyar, "Cross-platform comparison of standard power system components used in real time simulation," *7th Workshop on Model. and Simul. of Cyber-Physical Energy Syst.*, Montreal, QC, Canada, Apr. 2019, pp. 1-6, doi: 10.1109/MSCPES.2019.8738789.

• Portions of this chapter previously appeared as: P. M. Adhikari, L. Vanfretti, M. Ruppert, M. Ropp, "Real-time controller hardware-in-the-loop (RT CHIL) analysis of ground fault overvoltages (GFOVs)," presented at the *CIGRE US Nat. Committee (USNC) Grid of the Future (GOTF) Symp.*, Providence, RI, USA, Oct. 17-20, 2021, Paper B5.

• Portions of this chapter previously appeared as: P. M. Adhikari, H. Hooshyar, and L. Vanfretti, "Experimental quantification of hardware requirements for FPGA-based reconfigurable PMUs," *IEEE Access*, vol. 7, pp. 57527-57538, Apr. 2019, doi: 10.1109/ACCESS.2019.2911916.

- *How to design a user-friendly and configurable hardware to perform the pre-compliance tests on a custom PMU?*

  It is possible to generate these test scenarios in a low-voltage hardware prototype by utilizing National Instruments' cRIO platform. Interestingly, the utilization of this proposed hardware revealed that most of the PMU designs available for academic and industrial purposes struggle to meet the required performance criterion when the frequency of the analog inputs are subjected to a ramp increase.

- *Is it possible to implement homogeneous power system components which can give identical results across different real-time (RT) simulation platforms?*

  It is only possible if certain design considerations are followed. One such crucial design consideration was to make sure that the implementations of individual components have no library dependencies. In order to ensure that, it is necessary to redesign and re-implement certain electrical, electronic, mechanical and physical systems starting from the first principle, i.e. by utilizing simple mathematical operators and numerical methods, without using any readily available library elements in the respective platforms. Cross-platform RT modeling is a time consuming process and in order to avoid it, real-time simulator hardware vendors should fully adopt the FMI standard.

- *How to ensure uniform performance, while complex components such as PV cells and Li-ion batteries are implemented across two different RT-simulation hardware?*

  This is only possible by simplifying the mathematical models for such systems, and re-implementing those simplified models by using elementary mathematical & signal processing operators, and numerical methods. This methodology might produce models which are inferior in terms of accuracy, but the results obtained by simulating those models will be reproducible in different RT simulation hardware.

- *Are inverter based DERs as susceptible to GFOV when subjected to SLG faults, as the traditional generators? Are traditional 'effective grounding' techniques suitable for application on inverter based DERs?*

  Under standard operating conditions, inverter based DERs are not as susceptible to GFOV as traditional generators. However, DERs operating in underloaded conditions, i.e. for higher GLRs, the impact of GFOV becomes severe. The most efficient method to mitigate those overvoltages is to utilize the inverter's internal SPOV settings, as the traditional overvoltage mitigation strategies such as grounding transformers are relatively ineffective when utilized on inverter based DERs. However, this requires utilities to use the SPOV settings in protection coordination and for PV inverter manufacturers to provide more information for its modeling in power system studies.

- *How to ensure the reliable transmission of power system data through the network, following different protocols within a digital real-time grid simulation based laboratory ecosystem? How does the timing network interact with the intelligent electronic devices (IEDs) within the laboratory?*

  The reliability of the laboratory ecosystem can be ensured by enforcing specific quality of service (QoS) rules. It was also observed that a tampered timing network would effectively compromise the accuracy of time critical computations e.g. phase angle and frequency estimations.

- *Is it possible to utilize synchrophasor streams to reduce the complexity of the traditional WAMS/WAMPAC architecture targeted for DERs and microgrids?*

  This thesis proposes a novel synchrophasor synchronization gateway and controller (SSGC) hardware, which aims to track the network latency and reduce complexity of the control system for DERs and microgrids. The proposed SSGC hardware was stress-tested under varying network conditions. Upon rigorous experimentation, it was concluded that the proposed SSGC framework is robust under standard conditions, but its reliability gets affected when the communication network is performance degrades, which is representative of cases when it becomes compromised.

10.2 Future Works

The future directions for the research presented in this thesis are broadly classified into three categories. These are described as follows.

10.2.1 Extension of the Experiments into the Paradigm of Power Hardware in the Loop (PHIL) Configurations

The research presented in this thesis only explores real-time (RT) simulation of power system models under controller-hardware-in-the-loop (CHIL) configurations. Such experimental techniques eliminate the high voltage, and high current interfaces from the setup, thus ensuring the safety of the end users. However, it is also necessary to incorporate power-hardware-in-the-loop (PHIL) experiments before proposing architectural modifications to the industry. To this end, there has been some progress towards the development of real-time PHIL simulation laboratory featuring a 150 kW PV-inverter manufactured by *SMA*. The communication and control frameworks explored in this research can be assessed using the PHIL paradigm. However, a significant experimental effort is required in those prospective developments.

10.2.2 Efforts Towards Model Standardization

This is because model exchange standards such as *Modelica* and *Functional Mock-up Interface (FMI)* can provide a homogeneous and reliable infrastructure for cross-platform simulations and experiments. Based on the simplified models proposed in the current thesis, FMI compliant models can be developed in the future. Once those FMI compliant models are successfully tested for accuracy, they can be made available (as open-source software) for power system researchers to utilize for their work.

10.2.3 SSGC Based Control System Development for Microgrids and DERs

The proposed SSGC based control architecture was only tested for facilitating a step increase in the load through the BESS. This ‘proof of concept’ design can be extended to

- Incorporate synchrophasor measurements from different DERs or from the utility end.
- Incorporate the impact of the battery SOC to control the BESS.
- Investigate other control variables (e.g phase angle difference) to control power flow to and from the DERs.

A more detailed description of such future experiments are presented in section 9.4. While the prototype shows that the concept is valuable, it is cost-prohibitive for real-world applications. With the availability of low-cost IoT platforms, a future implementation based on such hardware could help reducing the cost of the SSG/SSGC hardware.

REFERENCES

[1] C. P. Steinmetz, “Complex quantities and their use in electrical engineering,” in *Proc. Int. Elect. Congress*, Chicago, 1893, pp. 33–74.

[2] A. G. Phadke, “Synchronized phasor measurements-a historical overview,” in *IEEE/PES Transmiss. & Distribution Conf. & Exhib.,* Oct. 2002, pp. 476–479, doi: 10.1109/TDC.2002.1178427.

[3] “Factors affecting PMU installation costs,” US Department of Energy. Accessed: Nov. 30, 2021. [Online]. Available: https://www.smartgrid.gov/files/recovery_act/PMU-cost-study-final-10162014_1.pdf

[4] S. Mukhopadhyay, P. K. Agarwal, V. Menghani, D. B. Tripathi, P. Batra, R. Walawalkar, “Indian experience with smart grid applications- transmission sector,” presented at the *IEEE Power & Energy Soc. General Meeting — Conf. & Expo.*, National Harbor, MD, USA, July 27-31, 2014, Paper PESGM2014P-002691.

[5] M. Ballal, A. Kulkarni, and H. Suryawanshi, “Data compression techniques for phasor measurement unit (PMU) applications in smart transmission grid,” *Int. J. Emerg. Elect. Power Syst.*, vol. 21, no. 3, July 2020, doi: 10.1515/ijeeps-2019-0266.

[6] “Renewables: Global status report 2020.” Ren21.net., Paris, France. Accessed Nov. 30, 2021. [Online]. Available: https://www.ren21.net/wp-content/uploads/2020/05/gsr_2020_full_report_en.pdf

[7] “Renewables: Global status report 2019.” Ren21.net., Paris, France. Accessed Nov. 30, 2021. [Online]. Available: https://www.ren21.net/wp-content/uploads/2019/05/gsr_2019_full_report_en.pdf

[8] R. H. Lasseter, “MicroGrids,” in *IEEE Power Eng. Soc. Winter Meeting. Conf. Proc.*, New York, NY, USA, Jan. 27-31, 2002, pp. 305-308, vol.1, doi: 10.1109/PESW.2002.985003.

[9] “US microgrid market growing faster than previously thought: New GTM research.” greentechmedia.com., MA, USA. Accessed Nov. 30, 2021. [Online]. Available: https://microgridknowledge.com/us-microgrid-market-gtm/

[10] M. Monadi, H. Hooshyar, L. Vanfretti, F. Mahmood, J. I. Candela and P. Rodriguez, “Measurement-based network clustering for active distribution systems,” *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6714-6723, Nov. 2019, doi: 10.1109/TSG.2019.2910510.

[11] P. Asmus. “Microgrids ramp up in Latin America but Asia Pacific remains the global leader.” microgridknowledge, https://microgridknowledge.com/microgrids-navigant/ (accessed Nov. 30, 2021).

[12] N. Hatziargyriou, H. Asano, R. Iravani and C. Marnay, “Microgrids,” *IEEE Power & Energy Mag.*, vol. 5, no. 4, pp. 78-94, July-Aug. 2007, doi: 10.1109/MPAE.2007.376583.

[13] A. Bidram and A. Davoudi, “Hierarchical structure of microgrids control system,” *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1963-1976, Dec. 2012, doi: 10.1109/TSG.2012.2197425.

[14] D. Laverty. “The OpenPMU project.” OpenPMU. https://sites.google.com/site/openpmu/home (accessed Nov. 30, 2021).

[15] T. A. Bhatti, A. Raheem, T. Alam, M. O. Malik, and A. Munir, “Implementation of low cost non-DFT based phasor measurement unit for 50 Hz power system,” in *Proc. Int. Conf. Comput., Electron. & Elect. Eng.*, Quetta, Pakistan, Apr. 11-12, 2016, pp. 120-125, doi: 10.1109/ICECUBE.2016.7495208.

[16] M. Farmad, S. Farhangi, S. Afsharnia and G. B. Gharehpetian, “Application of voltage source converter in interphase power controller for power flow control and fault limitation,” in *Proc. 34th Annu. Conf. IEEE Ind. Electron.*, Orlando, FL, USA, Nov. 2008, pp. 2301-2306, doi: 10.1109/IECON.2008.4758316.

[17] S. D. Choudante and A. A. Bhole, “A review: Voltage stability and power flow improvement by using UPFC controller,” in *Proc. Int. Conf. Computation of Power, Energy, Info. and Commun.*, Chennai, India, Mar. 2018, pp. 462-465, doi: 10.1109/ICCPEIC.2018.8525161.

[18] “Smart grid data management solution enables utility companies to improve overall system operation,” Intel Inc., Santa Clara, CA, USA. Accessed: Nov. 30, 2021. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/energy-smart-grid-solution-blueprint.pdf

[19] “Ultrascale FPGAs: Product tables and product selection guides,” Xilinx Inc., San Jose, CA, USA. Accessed: Nov. 30, 2021. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/ultrascale-fpga-product-selection-guide.pdf

[20] “NI-9225,” National Instruments, Austin, TX, USA. Accessed: Nov. 30, 2021. [Online]. Available: http://www.ni.com/en-us/support/model.ni-9225.html

[21] “Xilinx Spartan family,” Xilinx Inc., San Jose, CA, USA. Accessed: Nov. 30, 2021. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html

[22] P. Romano, M. Paolone, T. Chau, B. Jeppesen and E. Ahmed, “A high-performance, low-cost PMU prototype for distribution networks based on FPGA,” in *IEEE Manchester PowerTech*, Manchester, June 2017, pp. 1-6, doi: 10.1109/PTC.2017.7981025.

[23] R. Tessier, K. Pocek and A. DeHon., “Reconfigurable computing architectures,” *Proc. IEEE, vol. 103, no. 3*, pp. 332-354, Mar. 2015, doi: 10.1109/JPROC.2014.2386883.

[24] A. Agarwal, N. Verma, H. Tiwari, J. Singh and V. Maheshwari, “Design and development of phasor measurement unit on FPGA,” in *Proc. IEEE Region 10 Humanitarian Technol. Conf.*, Agra, India, Dec. 2016, pp. 1-6, doi: 10.1109/R10-HTC.2016.7906790.

[25] S. Ramadhan, F. I. Hariadi and A. S. Achmad, “Development FPGA-based phasor measurement unit (PMU) for smartgrid applications,” in *Proc. Int. Symp. Electron. & Smart Devices*, Bandung, Indonesia, Nov. 2016, pp. 21-25, doi: 10.1109/ISESD.2016.7886685.

[26] A. L. Akhila, R. Micky, A. Mohemmed, E. P. Cheriyan, and S. Ashok, “Implementation of an algorithm for phasor estimation and system parameters for an FPGA based PMU,” in *Proc. IEEE Int. Conf. Signal Process., Inform., Commun. & Energy Syst.*, Kollam, India, Aug. 2017, pp. 1-6, doi: 10.1109/SPICES.2017.8091353.

[27] S. Karn, A. Malkhandi and T. Ghose, “Laboratory prototype of a phasor measurement unit using FPGA based controller,” in *Proc. Int. Conf. Elect., Electron., & Optim. Techn.*, Chennai, India, Mar. 2016, pp. 2029-2034, doi: 10.1109/ICEEOT.2016.7755044.

[28] M. Pignati *et al.* “Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs,” in *Proc. IEEE Power & Energy Soc. Innovative Smart Grid Technol. Conf.*, Washington, DC, USA, Feb. 2015, pp. 1-5, doi: 10.1109/ISGT.2015.7131877.

[29] A. Katsimichas, P. Brogan and D. Laverty, “XML data interface and PMU testing applications for synchrophasor estimation with a reconfigurable controller,” in *Proc. 49th Int. Univ. Power Eng. Conf.*, Cluj-Napoca, Romania, Sept. 2014, pp. 1-5, doi: 10.1109/UPEC.2014.6934779.

[30] P. Romano, M. Paolone, J. Arnold and R. Piacentini, “An interpolated-DFT synchrophasor estimation algorithm and its implementation in an FPGA-based PMU prototype,” in *Proc. IEEE Power & Energy Soc. Annu. Meeting*, Vancouver, BC, Canada, July 2013, pp 1-6, doi: 10.1109/PESMG.2013.6672906.

[31] P. Romano and M. Paolone,“Enhanced interpolated-DFT for synchrophasor estimation in FPGAs: theory, implementation, and validation of a PMU prototype,” *IEEE Trans. Instrum. & Meas.*, vol. 63, no. 12, pp. 2824-2836, Dec. 2014, doi: 10.1109/TIM.2014.2321463.

[32] G. Frigo, A. Derviskadic and M. Paolone, “Reduced leakage synchrophasor estimation: Hilbert transform plus interpolated DFT,” *IEEE Trans. Instrum. & Meas.*, vol. 68, no. 10, pp. 3468-3483, Oct. 2019, doi: 10.1109/TIM.2018.2879070.

[33] A. Mohemmed, R. Mickey, A. L. Akhila and S. Ashok, “Open source implementation of IEEE C37.118-2011 protocol and serial communication interface for an FPGA based phasor measurement unit,” in *Proc. 2nd IEEE Int. Conf. Recent Trends in Electron., Inf. & Commun. Technol.*, Bangalore, India, May 2017, pp. 308-313, doi: 10.1109/RTEICT.2017.8256606.

[34] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems— a review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824-1842, Aug. 2007, doi: 10.1109/TIE.2007.898281.

[35] B. C. Richards, C. Chang, J. Wowrzynek, R. W. Brodersen, “Programming streaming FPGA applications using block diagrams in simulink,” in *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*, S. Hauck and A. Dehon, Eds., Burlington, MA, USA: Morgan Kaufmann Publisher, 2008, pp. 183-202.

[36] A. Khatri, “Selecting right FPGA for the right application: A technical survey for Xilinx FPGAs,” in *Quaid-e-Awam Univ. Res. J. Eng., Sci. & Technol.*, vol. 15, no. 1, pp. 46-49, 2016.

[37] R. Selow, H. S. Lopes and C. R. E. Lima, “A comparison of FPGA and FPAA technologies for a signal processing application,” in *Proc. Int. Conf. Field Programmable Logic & Appl.*, Prague, Czech Republic, Aug-Sept. 2009, pp. 230-235, doi: 10.1109/FPL.2009.5272306.

[38] H. Quinn, K. Morgan, P. Graham, J. Krone and M. Caffrey, “A review of Xilinx FPGA architectural reliability concerns from Virtex to Virtex-5,” in *Proc. 9th European Conf. Radiat. & Its Effects on Components & Syst.*, Deauville, France, Sept. 2007, pp. 1-8, doi: 10.1109/RADECS.2007.5205533.

[39] M. S. Reza, M. Ciobotaru and V. G. Agelidis, “A recursive DFT based technique for accurate estimation of grid voltage frequency,” in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2013, pp. 6420-6425, doi: 10.1109/IECON.2013.6700193.

[40] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Std C37.118-2011, IEEE Power & Energy Soc., Dec. 2011.

[41] “Plan-ahead design and analysis tool,” Xilinx Inc., San Jose, CA, USA. Accessed: Nov. 30, 2021. [Online]. Available: https://www.xilinx.com/support/documentation/sw-manuals/xilinx11/PlanAheadUserGuide.pdf

[42] P. V. Brogan, D. M. Laverty, X. Zhao, J. Hastings, D. J. Morrow, and L. Vanfretti, “Technique for pre-compliance testing of phasor measurement units,” in *Int. J. Elect. Power & Energy Syst.,* vol. 99, July 2018, pp. 323-330, doi: 10.1016/j.ijepes.2018.01.031.

[43] M. S. Almas, J. Kilter and L. Vanfretti, “Experiences with steady-state PMU compliance testing using standard relay testing equipment,” in *Proc. Elect. Power Qual. & Supply Rel. Conf.*, Rakvere, Estonia, June 2014, pp. 103-110, doi: 10.1109/PQ.2014.6866793.

[44] M. Horinek, P. Bilik and T. Wittassek, "Modular PMU tester," in *Proc. 22nd Int. Conf. & Exhib. Electricity Distribution*, Stockholm, Sweden, June 2013, pp. 1-4, doi: 10.1049/cp.2013.1183.

[45] R. Ghiga, Q. Wu, K. Martin, W. Z. El-Khatib, L. Cheng and A. H. Nielsen, "Dynamic PMU compliance test under C37.118.1a™-2014," in *Proc. IEEE Power & Energy Soc. General Meeting*, Denver, CO, USA, July 2015, pp. 1-5, doi: 10.1109/PESGM.2015.7285970.

[46] P. Castello, C. Muscas, P. Attilio Pegoraro and S. Sulis, "Critical analysis of PMU testing procedures for step response evaluation," in *Proc. IEEE Int. Workshop Appl. Meas. for Power Syst. (AMPS)*, Aachen, Sept. 2015, pp. 37-42, doi: 10.1109/AMPS.2015.7312735.

[47] R. S. Singh, H. Hooshyar and L. Vanfretti, "Laboratory test set-up for the assessment of PMU time synchronization requirements," in *Proc. IEEE Eindhoven PowerTech*, Eindhoven, Netherlands, June-July 2015, pp. 1-5, doi: 10.1109/PTC.2015.7232731.

[48] K. Narendra, Zhiying Zhang, J. Lane, B. Lackey and E. Khan, "Calibration and testing of TESLA phasor measurement unit (PMU) using doble F6150 test instrument," in *Proc. iREP Symp. - Bulk Power Syst. Dyn. & Control - VII. Revitalizing Oper. Rel.*, Charleston, SC, USA, Aug. 2007, pp. 1-13, doi: 10.1109/IREP.2007.4410561.

[49] M. S. Almas, M. Baudette, L. Vanfretti, S. Løvlund and J. O. Gjerde, "Synchrophasor network, laboratory and software applications developed in the STRONg2rid project," in *Proc. IEEE Power & Energy Soc. General Meeting — Conf. & Expo.*, National Harbor, MD, USA, July 2014, pp. 1-5, doi: 10.1109/PESGM.2014.6938835.

[50] J. Kilter, I. Palu, M. S. Almas and L. Vanfretti, "Experiences with dynamic PMU compliance testing using standard relay testing equipment," in *Proc. IEEE Power & Energy Soc. Innovative Smart Grid Technol. Conf.*, Washington, DC, USA, June 2015, pp. 1-5, doi: 10.1109/ISGT.2015.7131833.

[51] A. A. van der Meer *et al.*, "Cyber-physical energy systems modeling, test specification, and co-simulation based testing," in *Proc. Workshop on Model. & Simul. of Cyber-Physical Energy Syst.*, Pittsburgh, PA, USA, Apr. 2017, pp. 1-9, doi: 10.1109/MSCPES.2017.8064528.

[52] Q. Wang, W. Tai, Y. Tang, Y. Liang, L. Huang and D. Wang, "Architecture and application of real-time co-simulation platform for cyber-physical power system," in *Proc. IEEE 7th Annu. Int. Conf. CYBER Technol. in Automat., Control, and Intell. Syst.*, Honolulu, HI, USA, July-Aug. 2017, pp. 81-85, doi: 10.1109/CYBER.2017.8446098.

[53] J. J. Sanchez-Gasca, R. D'Aquila, W. W. Price, and J. J. Paserba, "Variable time step, implicit integration for extended-term power system dynamic simulation," in

Proc. Power Industry Computer Appl. Conf., Salt Lake City, UT, USA, May 1995, pp. 183-189, doi: 10.1109/PICA.1995.515182.

[54] J. J. Sanchez-Gasca, R. D'Aquila, J. J. Paserba, W. W. Price, D. B. Klapper and I. Hu, "Extended-term dynamic simulation using variable time step integration," *IEEE Computer Appl. in Power*, vol. 6, no. 4, pp. 23-28, Oct. 1993, doi: 10.1109/67.238201.

[55] E. Weingartner, H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Proc. IEEE Int. Conf. Commun.*, Dresden, Germany, June 2009, pp. 1-5, doi: 10.1109/ICC.2009.5198657.

[56] H. Shen and R. W. P. King, "Experimental investigation of the Rhombic EMP simulator: Comparison with theory and parallel-plate simulator," *IEEE Trans. Electromagn. Compat.*, vol. EMC-24, no. 3, pp. 349-355, Aug. 1982, doi: 10.1109/TEMC.1982.304047.

[57] X. Guillaud *et al.*, "Applications of real-time simulation technologies in power and energy systems," *IEEE Power & Energy Technol. Syst. J.*, vol. 2, no. 3, pp. 103-115, Sept. 2015, doi: 10.1109/JPETS.2015.2445296.

[58] L. Vanfretti, R. Leelaruji, W. Li, T. Bogodorova, "Simulation studies and experiences of the preliminary CIGRE DC grid test system with EMTP-RV, PSCAD and Opal-RT," in *Tech. Exhib., Smart Transmiss. Syst. Lab.*, Stockholm, Sweden. Accessed: Nov. 30, 2021. [Online]. Available: https://documents.pub/reader/full/simulation-studies-and-experiences-of-the-preliminary-cigre-dc-grid-test-system

[59] P. Forsyth, R. Kuffel, R. Wierckx, J. Choo, Y. Yoon and T. Kim, "Comparison of transient stability analysis and large-scale real time digital simulation," in *IEEE Porto Power Tech Proc. (Cat. No.01EX502)*, Porto, Portugal, Sept. 2001, pp. 7, vol.4, doi: 10.1109/PTC.2001.964822.

[60] "Hydraulic Turbine and Governor." mathworks.com. https://www.mathworks.com/help/physmod/sps/powersys/ref/hydraulicturbineandgovernor.html (accessed Nov. 30, 2021).

[61] H. Hooshyar, L. Vanfretti and C. Dufour, "Delay-free parallelization for real-time simulation of a large active distribution grid model," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Florence, Italy, Oct. 2016, pp. 6278-6284, doi: 10.1109/IECON.2016.7793885.

[62] T. Blochwitz *et al.*, "The functional mockup interface for tool independent exchange of simulation models," in *Proc. 8th Int. Modelica Conf.*, Dresden, Germany, Mar. 2011, pp. 105-114, doi: 10.3384/ecp11063105.

[63] G. Schweiger *et al.*, "Functional mock-up interface: An empirical survey identifies research challenges and current barriers," in *Proc. 1st Amer. Modelica Conf.*, Cambridge, MA, USA, Oct. 2018, pp. 138-146, doi:10.3384/ecp18154138.

[64] L. Vanfretti, W. Li, T. Bogodorova and P. Panciatici, “Unambiguous power system dynamic modeling and simulation using modelica tools,” in *Proc. IEEE Power & Energy Soc. General Meeting*, Vancouver, BC, Canada, Nov. 2013, pp. 1-5, doi: 10.1109/PESMG.2013.6672476.

[65] “Opal-RT 4520 user guide,” Opal-RT Technologies., Montreal, QC, Canada. Accessed: Nov. 30 2021. [Online]. https://www.opal-rt.com/wp-content/themes/enfold-opal/pdf/L00161_0304.pdf

[66] “FMI options in EMTP,” emtp.com., Montreal, QC, Canada. Accessed Nov. 30 2021. [Online]. https://www.emtp.com/news/fmi-options

[67] H. Park and H. Kim, “PV cell modeling on single-diode equivalent circuit,” in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc.*, Vienna, Italy, Nov. 2013, pp. 1845-1849, doi: 10.1109/IECON.2013.6699412.

[68] M. T. Ahmed, T. Gonçalves and M. Tlemcani, “Single diode model parameters analysis of photovoltaic cell,” in *Proc. IEEE Int. Conf. Renewable Energy Res. & Appl.*, Birmingham, UK, Nov. 2016, pp. 396-400, doi: 10.1109/ICRERA.2016.7884368.

[69] E. I. Batzelis and S. A. Papathanassiou, “A method for the analytical extraction of the single-diode PV model parameters,” *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 504-512, Apr. 2016, doi: 10.1109/TSTE.2015.2503435.

[70] G. Petrone and G. Spagnuolo, “Parameters identification of the single-diode model for amorphous photovoltaic panels,” in *Proc. Int. Conf. Clean Elect. Power*, Taormina, Italy, June 2015, pp. 105-109, doi: 10.1109/ICCEP.2015.7177608.

[71] M. K. Dave, “Modeling of PV arrays based on datasheet”. in *Proc. IEEE 1st Int. Conf. on Power Electron., Intell. Control & Energy Syst.*, Delhi, India, July 2016, pp. 1-4, doi: 10.1109/ICPEICES.2016.7853617.

[72] V. Tamrakar, S. C. Gupta, and Y. Sawle, “Study of characteristics of single and double diode electrical equivalent circuit models of solar PV module,” *Proc. Int. Conf. Energy Systems & Appl.*, Pune, India, Oct.-Nov. 2015, pp. 312-317, doi: 10.1109/ICESA.2015.7503362.

[73] F. Mahmood, L. Vanfretti, and H. Hooshyar, “Modeling of a detailed photovoltaic generation system for EMT-type simulation,” in *Proc. IEEE Int. Energy Conf.*, Cavtat, Croatia, May 2014, pp. 916-921, doi: 10.1109/ENERGYCON.2014.6850535.

[74] “Model for photovoltaic cells and arrays.” mathworks.com. https://www.mathworks.com/matlabcentral/fileexchange/61480-matlab-simulink-model-of-photovoltaic-cell-panel-and-array (accessed Nov. 30, 2021).

[75] S. Hänninen and M. Lehtonen, “Characteristics of earth faults in electrical distribution networks with high impedance earthing,” *Electric Power Syst. Res.*, vol. 44, no. 3, Mar. 1998, pp 155-161, doi: 10.1016/S0378-7796(97)01193-0.

[76] H. B. Lee, S. E. Chase, R. C. Dugan, “Overvoltage considerations for interconnecting dispersed generators with wye-grounded distribution feeders,” *IEEE Power Eng. Rev.*, vol. PER-4, no. 12, Dec. 1984, pp. 44-45, doi: 10.1109/TPAS.1984.318411.

[77] P. Jonke *et al.*, “Integrated rapid prototyping of distributed energy resources in a real-time validation environment,” in *Proc. IEEE 25th Int. Symp. Ind. Electron.*, Santa Clara, CA, USA, June 2016, pp. 714-719, doi: 10.1109/ISIE.2016.7744977.

[78] P. Barker, “Overvoltage considerations in applying distributed resources on power systems,” in *Proc. IEEE Power Eng. Soc. Summer Meeting*, Chicago, IL, USA, July 2002, pp. 109-114, vol.1, doi: 10.1109/PESS.2002.1043188.

[79] A. Cerretti, F. M. Gatta, A. Geri, S. Lauria, M. Maccioni, and G. Valtorta, “Ground fault temporary overvoltages in MV networks: Evaluation and experimental tests,” in *IEEE Trans. Power Del.*, vol. 27, no. 3, pp. 1592-1600, July 2012, doi: 10.1109/TPWRD.2012.2192456.

[80] A. Cerretti, F. M. Gatta, A. Geri, S. Lauria, M. Maccioni, and G. Valtorta, “Temporary overvoltages due to ground faults in MV networks,” in *IEEE Bucharest PowerTech*, Bucharest, Romania, June-July 2009, pp. 1-8, doi: 10.1109/PTC.2009.5282140.

[81] J. R. Acharya, Y. Wang, and W. Xu, “Temporary overvoltage and GPR characteristics of distribution feeders with multigrounded neutral,” *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 1036-1044, Apr. 2010, doi: 10.1109/TPWRD.2009.2034456.

[82] M. Ropp *et al.*, “Ground fault overvoltage with inverter-interfaced distributed energy resources,” *IEEE Trans. Power Del.*, vol. 32, no. 2, pp. 890-899, Apr. 2017, doi: 10.1109/TPWRD.2016.2577884.

[83] *IEEE Guide for the Application of Neutral Grounding in Electrical Utility Systems, Part II - Synchronous Generator Systems*, IEEE C62.92.2-2017, IEEE Power & Energy Soc., Feb. 2017.

[84] M. Ropp, Y. Cui, M. Kahrobaee, D. Schutz and C. Mouw, “On the sizing and benefits of grounding transformers with distribution-connected inverters,” in *IEEE/PES Transmission & Distribution Conf. & Expo.*, Denver, CO, USA, Apr. 2018, pp. 1-5, doi: 10.1109/TDC.2018.8440448.

[85] M. D. Omar Faruque *et al.*, “Real-time simulation technologies for power systems design, testing, and analysis,” *IEEE Power & Energy Technol. Syst. J.*, vol. 2, no. 2, pp. 63-73, June 2015, doi: 10.1109/JPETS.2015.2427370.

[86] B. Azimian, P. M. Adhikari, L. Vanfretti and H. Hooshyar, “Cross-platform comparison of standard power system components used in real time simulation,” in *Proc. 7th Workshop Model. & Simul. of Cyber-Physical Energy Syst.*, Montreal, QC, Canada, Apr. 2019, pp. 1-6, doi: 10.1109/MSCPES.2019.8738789.

[87] B. Lundstrom, S. Chakraborty, G. Lauss, R. Bründlinger, and R. Conklin, "Evaluation of system-integrated smart grid devices using software- and hardware-in-the-loop," in *Proc. IEEE Power & Energy Soc. Innovative Smart Grid Technol. Conf.*, Minneapolis, MN, USA, Sept. 2016, pp. 1-5, doi: 10.1109/ISGT.2016.7781181.

[88] R. Bründlinger, R. Ablinger, Z. Miletic, "AIT smart grid converter(SGC) controller featuring SunSpec protocol support utilizing hardware-in-the-Loop (HIL) technology," in *SunSpec Meeting*, Vienna, Austria, Sept. 2016, Accessed: Nov. 30, 2021. [Online]. Available: https://sunspec.org/wp-content/uploads/2016/09/AITSunSpecSmartGrid InverterControllerReferenceDesignwithSunSpecSVPsupportBrundlinger -final-160912.pdf

[89] A. Pană, A. Băloi, and F. Molnar-Matei, "From the balancing reactive compensator to the balancing capacitive compensator," *Energies (MDPI)*, vol. 11, no. 8, July 2018, doi: 10.3390/en11081979.

[90] M. Chen and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504-511, June 2006, doi: 10.1109/TEC.2006.874229.

[91] L. W. Yao, J. A. Aziz, P. Y. Kong, and N. R. N. Idris, "Modeling of lithium-ion battery using MATLAB/simulink," in *Proc. 39th Annu. Conf. Ind. Electron. Soc.*, Nov. 2013, pp. 1729–1734, doi: 10.1109/IECON.2013.6699393.

[92] K. Bae, S. Choi, J. Kim, C. Won, and Y. Jung, "LiFePO4 dynamic battery modeling for battery simulator," in *Proc. IEEE Int. Conf. Ind. Technol.*, Feb. 2014, pp. 354–358, doi: 10.1109/ICIT.2014.6894892.

[93] "Generic battery model." mathworks.com. https://www.mathworks.com/help/physmod/sps/powersys/ref/battery.html (accessed Nov. 30, 2021).

[94] M. Sitterly, L. Y. Wang, G. G. Yin, and C. Wang, "Enhanced identification of battery models for real-time battery management," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 300 – 308, Feb. 2011, doi: 10.1109/TSTE.2011.2116813.

[95] "Typhoon HIL diesel genset." typhoon-hil.com. https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/dieselgenset.html (accessed Nov. 30, 2020).

[96] X. Jia, P. M. Adhikari, L. Vanfretti, "Real-time simulation models for PV cells and arrays in Opal-RT and Typhoon-HIL," in *Proc. IEEE Power & Energy Soc. General Meeting*, Aug. 2020, Montréal, QC, Canada, doi: 10.1109/PESGM41954.2020.9282171.

[97] "Typhoon HIL battery inverter." typhoon-hil.com. https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/battery-inverter.html (accessed Nov. 30, 2020).

[98] *IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems With Electric Power Systems*, IEEE 1547.4-2011, July, 2011.

[99] M. H. Syed *et al.*, “Real-time coupling of geographically distributed research infrastructures: Taxonomy, overview, and real-world smart grid applications,” *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1747-1760, Mar. 2021, doi: 10.1109/TSG.2020.3033070.

[100] *Communication Networks and Systems For Power Utility Automation – Part 1: Introduction and Overview*, IEC 61850, Mar. 2013.

[101] M. S. Almas and L. Vanfretti, “RT-HIL implementation of the hybrid synchrophasor and GOOSE-based passive islanding schemes,” *IEEE Trans. Power Del.*, vol. 31, no. 3, pp. 1299-1309, June 2016, doi: 10.1109/TPWRD.2015.2473669.

[102] A. Silverstein *et al.*, “Time Synchronization in Electric Power Systems”, NASPI Technical Report, Mar. 2017. [Online]. Available: https://www.naspi.org/sites/default/files/reference_documents/ tstf_electric_power_system_report_pnnl_26331_march_2017_0.pdf

[103] M. J. Turner, “Design and development of a smart grid laboratory for an energy and power engineering technology program,” *The Int. J. Elect. Eng. & Educ.*, Oct. 2017; vol. 54, no. 4, pp. 299-318, doi: 10.1177/0020720916687315.

[104] “Distributed electrical systems laboratory.” EPFL. [Online]. Available: https://www.epfl.ch/labs/desl-pwrs/ (accessed Feb. 28, 2021).

[105] A. M. Gaouda, A. Abd-Rabou, and A. Dahir, “Developing educational smart grid laboratory,” in *Proc. IEEE Int. Conf. Teaching, Assessment and Learn. for Eng.*, Bali, Indonesia, Aug. 2013, pp. 404-409, doi: 10.1109/TALE.2013.6654471.

[106] V. Salehi, A. Mohamed, A. Mazloomzadeh, and O. A. Mohammed, “Laboratory-based smart power system, part I: Design and system development,” *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1394-1404, Sept. 2012, doi: 10.1109/TSG.2012.2194518.

[107] F. Huerta, J. K. Gruber, M. Prodanovic, P. Matatagui and T. Gafurov, “A laboratory environment for real-time testing of energy management scenarios: Smart energy integration lab (SEIL),” in *Proc. Int. Conf. Renewable Energy Res. & Appl.*, Oct. 2014, pp. 514-519, doi: 10.1109/ICRERA.2014.7016438.

[108] H. Hooshyar, L. Vanfretti, J. H. Chow, R. Fitsik, and P. M. Adhikari, “ALSET lab: Designing precise timing and communications for a digital power grid laboratory,” in *Proc. IEEE Power & Energy Soc. General Meeting*, Montreal, QC, Canada, Aug. 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281808.

[109] C. Landsteiner, F. Andren, and T. Strasser, “Evaluation and test environment for automation concepts in smart grids applications,” in *Proc. IEEE 1st Int. Workshop*

Smart Grid Model. & Simul., Brussels, Belgium, Oct. 2011, pp. 67-72, doi: 10.1109/SGMS.2011.6089200.

[110] H. Tong, M. Ni, L. Zhao, and M. Li, "Flexible hardware-in-the-loop testbed for cyber physical power system simulation," *IET Cyber-Physical Syst.: Theory & Appl.*, vol. 4, no. 1, pp. 374-381, July 2019, doi: 10.1049/iet-cps.2019.0001.

[111] J. Bruinenberg *et al.*, "Smart Grid Coordination Group Smart Grid Reference Architecture," CEN-CENELEC-ETSI Smart Grid Coordination Group, Jan. 2012. Accessed Nov. 30, 2021. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf

[112] P. Zhang, F. Li and N. Bhatt, "Next-generation monitoring, analysis, and control for the future smart control center," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 186-192, Sept. 2010, doi: 10.1109/TSG.2010.2053855.

[113] A. Carta, N. Locci, C. Muscas, F. Pinna, S. Sulis, "GPS and IEEE 1588 synchronization for the measurement of synchrophasors in electric power systems," *Computer Std. & Interfaces*, vol. 33, no. 2, pp 176-181, Feb. 2011, doi: 10.1016/j.csi.2010.06.009.

[114] M. Correia *et al.*, "Implementation of IEEE-1588 timing and synchronization for ATCA control and data acquisition systems," *Fusion Eng. & Design*, vol. 87, no. 12, pp 2178-2181, Dec. 2012, doi: 10.1016/j.fusengdes.2012.07.012.

[115] C. Bonebrake and L. R. O'Neil, "Attacks on GPS time reliability," *IEEE Secur. & Privacy*, vol. 12, no. 3, pp. 82-84, May-June 2014, doi: 10.1109/MSP.2014.40.

[116] C. Konstantinou, M. Sazos, A. S. Musleh, A. Keliris, A. Al-Durra, and M. Maniatakos, "GPS spoofing effect on phase angle monitoring and control in a real-time digital simulator-based hardware-in-the-loop environment," *IET Cyber-Physical Syst.: Theory & Appl.*, vol. 2, no. 4, pp. 180-187, June 2017, doi: 10.1049/iet-cps.2017.0033.

[117] M. S. Almas, L. Vanfretti, R. S. Singh, and G. M. Jonsdottir, "Vulnerability of synchrophasor-based WAMPAC applications' to time synchronization spoofing," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4601-4612, Sept. 2018, doi: 10.1109/TSG.2017.2665461.

[118] M. You, Q Liu, and H. Sun, "New communication strategy for spectrum sharing enabled smart grid cyber-physical system," *IET Cyber-Physical Syst.: Theory & Appl.*, vol. 2, no. 3, pp. 136-142, July 2017, doi: 10.1049/iet-cps.2017.0051.

[119] M. A. Aftab, S. M. S. Hussain, I. Ali, T. S. Ustun, "IEC 61850 based substation automation system: A survey," *Int. J. Elect. Power & Energy Syst.*, vol. 120, Sept. 2020, doi: 10.1016/j.ijepes.2020.106008.

[120] Y. Wu, and L. Nordström,"Adaptive data link configuration for WAMC applications using a stateful data delivery service platform," *Sustain. Energy, Grids, & Netw.*, vol. 6, pp. 166–175, June 2016, doi: 10.1016/j.segan.2016.04.001.

[121] E. Tebekaemi, D. Wijesekera, "Secure overlay communication and control model for decentralized autonomous control of smart micro-grids," *Sustain. Energy, Grids, & Netw.*, vol. 18, June 2019, doi: 10.1016/j.segan.2019.100222.

[122] A. A. de Sotomayor, D. D. Giustina, G. Massa, A. Dedè, F. Ramos, F., and A. Barbato, "IEC 61850-based adaptive protection system for the MV distribution smart grid," *Sustain. Energy, Grids, & Netw.*, vol. 15, pp. 26-33, June 2018, doi: 10.1016/j.segan.2017.09.003.

[123] S. R. Firouzi, L. Vanfretti, A. Ruiz-Alvarez, F. Mahmood, H. Hooshyar, and I. Cairo, "An IEC 61850-90-5 gateway for IEEE C37.118.2 synchrophasor data transfer," in *Proc. IEEE Power & Energy Soc. General Meeting*, Boston, MA, USA, July 2016, pp. 1-5, doi: 10.1109/PESGM.2016.774139.

[124] P. M. Adhikari and L. Vanfretti, "Delay analysis of a real-time hard reconfigurable synchrophasor synchronization gateway," in *Proc. IASTED Int. Conf. Control & Optim. of Renewable Energy Syst.*, Annaheim, CA, USA, Dec. 2019, doi: 10.2316/P.2019.859-008.

[125] K. Heussen, O. Gehrke, H. Kley, A. M. Kosek, A. Thavlov, "Use cases for laboratory software infrastructure - outline of smart grid lab software requirements," Technical Univ. of Denmark, Rep. D2.1, Jan. 2014, doi:10.13140/RG.2.2.16507.11041.

[126] *AGILe lab*, New York Power Authority. White Plains, NY, USA. Accessed: Mar. 30, 2021. [Online]. Available: https://www.nypa.gov/innovation/digital-utility/agile-lab

[127] T. Becejac, C. Eppinger, A. Ashok, U. Agrawal, and J. O'Brien, "PRIME: a real-time cyber-physical systems testbed: From wide-area monitoring, protection, and control prototyping to operator training and beyond," *IET Cyber-Physical Syst.: Theory & Appl.*, vol. 5, no. 2, pp. 186-195, June 2020, doi: 10.1049/iet-cps.2019.0049.

[128] P. Castello, C. Muscas, P. Attilio Pegoraro and S. Sulis, "Low-cost implementation of an active phasor data concentrator for smart grid," in *Proc. Workshop Metrology for Industry 4.0 & IoT*, Brescia, Italy, Apr. 2018, pp. 78-82, doi: 10.1109/METROI4.2018.8428312.

[129] M. G. Adamiak, M. Kanabar, J. Rodriquez, and M. D. Zadeh, "Design and implementation of a synchrophasor data concentrator," in *Proc. IEEE Power & Energy Soc. Conf. on Innovative Smart Grid Technol. - Middle East*, Jeddah, Saudi Arabia, Dec. 2011, pp. 1-5, doi: 10.1109/ISGT-MidEast.2011.6220778.

[130] H. Retty, J. Delport and V. Centeno, "Development of tests and procedures for evaluating phasor data concentrators," in *Proc. IEEE Grenoble Conf.*, Grenoble, France, June, 2013, pp. 1-5, doi: 10.1109/PTC.2013.6652210.

[131] L. Vanfretti, G. M. Jónsdóttir, M. S. Almas, E. Rebello, S .R. Firouzi, M. Baudette, "Audur: A platform for synchrophasor-based power system wide-area control system implementation," *SoftwareX*, vol.7, pp. 294-301, Jan.-Jun 2018, doi: 10.1016/j.softx.2018.08.003.

[132] "Definition of microgrid," International Electrotechnical Commission. [Online]. Available: https://www.electropedia.org/iev/iev.nsf/display?openformievref=617-04-22 (accessed: Nov. 30, 2021).

[133] J. Pongnark, S. Tanachutiwat, "Performance and reliability benchmarking of phasor data concentrator software systems and preliminary designing of wide-area monitoring system," in *Proc. 14th Int. Conf. Elect. Eng./Electron., Comput., Telecomm. & Info. Technol.*, Phuket, Thailand, June 2017, pp. 326-329, doi: 10.1109/ECTICon.2017.8096239.

[134] A. Derviškadić, P. Romano, M. Pignati, and M. Paolone, "Architecture and experimental validation of a low-latency phasor data concentrator," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2885-2893, July 2018, 10.1109/TSG.2016.2622725.

[135] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna and M. Castilla, "Hierarchical control of droop-controlled AC and DC microgrids: A general approach toward standardization," *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 158-172, Jan. 2011, doi: 10.1109/TIE.2010.2066534.

[136] R. Pourramezan, Y. Seyedi, H. Karimi, G. Zhu, and M. Mont-Briant, "Design of an advanced phasor data concentrator for monitoring of distributed energy resources in smart microgrids," *IEEE Trans. Ind. Inform.*, vol. 13, no. 6, pp. 3027-3036, Dec. 2017, doi: 10.1109/TII.2017.2697438.

[137] S. A. R. Konakalla, A. Valibeygi, and R. A. de Callafon, "Microgrid dynamic modeling and islanding control with synchrophasor data," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 905-915, Jan. 2020, doi: 10.1109/TSG.2019.2948815.

[138] P. Castello, C. Muscas, P. A. Pegoraro, S. Sulis, "Active phasor data concentrator performing adaptive management of latency," *Sustain. Energy, Grids, & Netw.*, vol. 16, pp. 270-277, Dec. 2018, doi: 10.1016/j.segan.2018.09.004.

[139] "NI expertise: Power quality analysis," National Instruments, Austin, TX, USA. Accessed: Nov. 30, 2021. [Online]. Available: http://www.ni.com/en-us/innovations/energy/power-quality-analysis.html

[140] "CT 910 product specifications," candelatech, Ferndale, WA, USA. Accessed: Nov. 30, 2021. [Online]. https://www.candelatech.com/ct910_product.php

[141] *Typhoon microgrid example*, typhoon-hil.com. available at: https://www.typhoon-hil.com/documentation/typhoon-hil-application-notes/References/microgrid-example.html (accessed: Mar. 30, 2021).

[142] L. Song, Z. Wang, W. Luo, "Analysis on the response time of the battery energy storage system," in *Proc. Int. Conf. Mater. Sci. & Energy Eng.*, Sanya, China, Dec. 2014, pp. 553-560, doi: 10.1142/9789814678971_0080.

APPENDIX A
CODE FOR PV CELL MODELING

Typhoon HIL has the *'Advanced C functionalities'* package that was utilized to compose complicated electrical models by incorporating C code segments within the RT simulation model. The single diode model (SDM) for PV cell was modeled this way. A sample C code for the SDM is archived below.

```
Initialization:
Istart=0.01; FI0=0; FpI0=0; I1=Istart; FI1=0;
FpI1=0; I2=I1;  FI2=0; FpI2=0; I3=I2; FI3=0;
FpI3=0; I4=I3; FI4=0; FpI4=0; I5=0.01;
Computation:
FI0=Istart-Isc+I0*(exp((V+Istart*Rs)/Vt)-1);
FpI0=1+I0*(exp((V+Istart*Rs)/Vt))*Rs/Vt;
I1=Istart-FI0/FpI0;
FI1=I1-Isc+I0*(exp((V+I1*Rs)/Vt)-1);
FpI1=1+I0*(exp((V+I1*Rs)/Vt))*Rs/Vt;
I2=I1-FI1/FpI1;
FI2=I2-Isc+I0*(exp((V+I2*Rs)/Vt)-1);
FpI2=1+I0*(exp((V+I2*Rs)/Vt))*Rs/Vt;
I3=I2-FI2/FpI2;
FI3=I3-Isc+I0*(exp((V+I3*Rs)/Vt)-1);
FpI3=1+I0*(exp((V+I3*Rs)/Vt))*Rs/Vt;
I4=I3-FI3/FpI3;
FI4=I4-Isc+I0*(exp((V+I4*Rs)/Vt)-1);
FpI4=1+I0*(exp((V+I4*Rs)/Vt))*Rs/Vt;
I5=I4-FI4/FpI4;
I_array=I5;
```

APPENDIX B
THEORETICAL ESTIMATION OF GFOV FOR SYNCHRONOUS MACHINES AND IBRS

In this appendix, the theoretical overvoltage computation for synchronous generators and IBRs under SLG faults is illustrated. It can be observed that, for synchronous generators the overvoltage is 73%, and for IBRs, it is much lower.

B.1 Theoretical GFOV Calculation in Synchronous Generators

It is known that during normal operating conditions, the phase voltages of the ungrounded synchronous generator as shown in Fig. B.1. are expressed as

$$E_a = E\angle 0^\circ, E_b = E\angle 240^\circ, E_c = E\angle 120^\circ \tag{B.1}$$

Thus, in unfaulted conditions $V_N = 0$. But, during an SLG fault (shown on phase C in Fig.B.1) the terminal voltage of phase C goes down to 0. In this situation, voltage of phase C V_{cn} along the load also becomes 0 as it is directly connected to the ground. In this scenario,

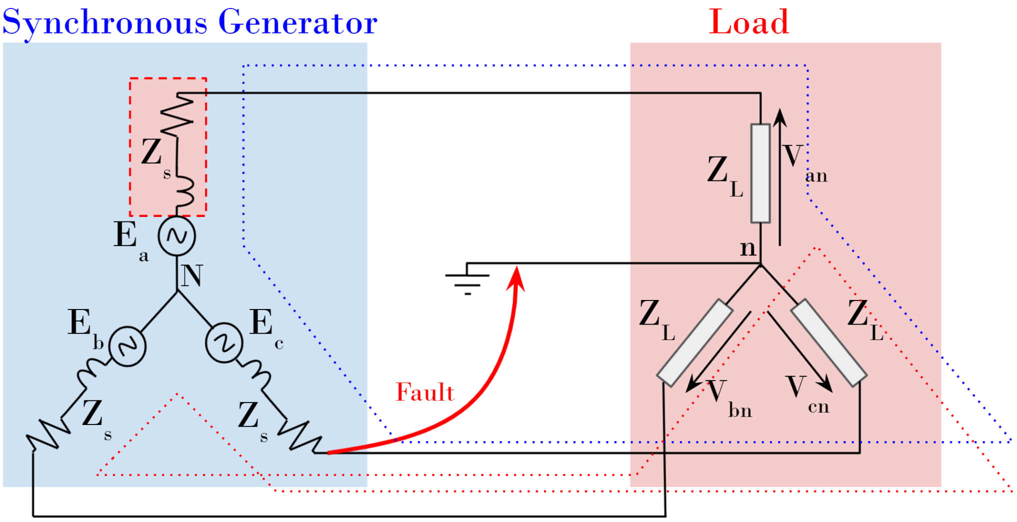


Figure B.1: Ground Fault in an Ungrounded Synchronous Generator Feeding a 3 Phase 4 Wire System

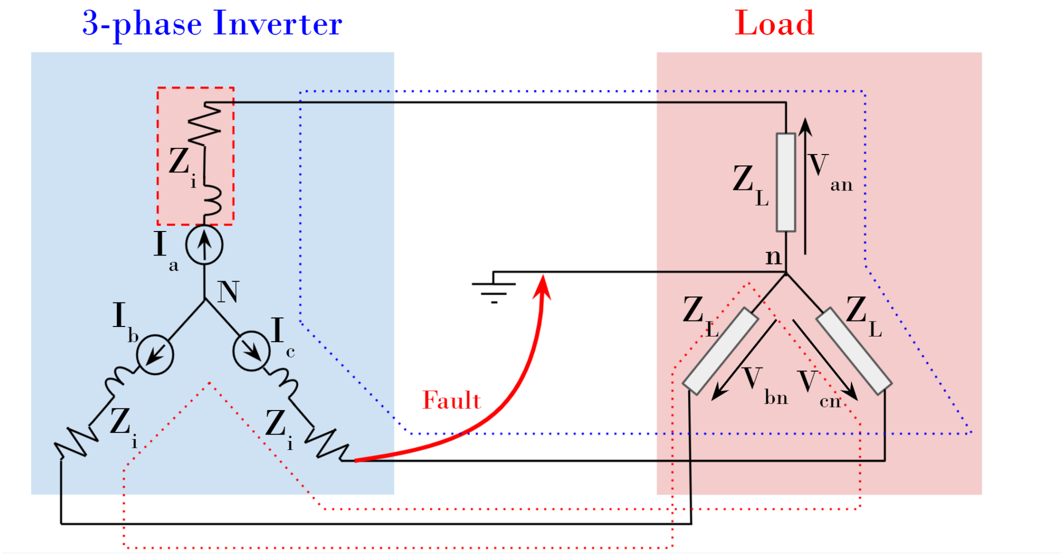


Figure B.2: Ground Fault in a Three Phase Inverter Feeding a 3 Phase 4 Wire System

by applying the KVL along the loop comprising phases C and B (represented by the dotted red line in Fig.B.1) the following expression can be obtained:

$$E_b - V_{bn} - E_c = 0 \Rightarrow V_{bn} = E\angle 240^\circ - E\angle 120^\circ = \sqrt{3}E\angle 270^\circ \tag{B.2}$$

Next, applying KVL through phases A and C (blue dotted line in Fig. B.1.) the following expression can be reached

$$E_a - V_{cn} - E_c = 0 \Rightarrow V_{an} = E\angle 0^\circ - E\angle 120^\circ = \sqrt{3}E\angle 330^\circ \tag{B.3}$$

Evidently, from (B.2) and (B.3) it can be concluded that the 173% (due to the $\sqrt{3}$ factor) ground-fault over-voltage is expected, thus it requires standardized measures to mitigate it.

B.2 Theoretical GFOV Calculation in Inverters

Next, consider the circuit in Fig. B.2. It is similar to Fig. B.1 but with a three-phase inverter in place of the synchronous generator. In the case of Fig. B.1, when the fault strikes on phase C of the load, the phase C of the synchronous generator maintains E_c constant. However, as shown in Fig.B.2, the inverter does not have such an externally excited internal voltage source, and it is modeled as a current source. Therefore, the voltage between terminal C and neutral N, most likely will not remain constant. In this scenario, the voltages of the unfaulted phases on the load side would be given by:

$$V_{an} = I_a \times Z_a, V_{bn} = I_b \times Z_b \tag{B.4}$$

Hence, if the current supplied by the inverter remains constant, no over-voltages should be observed on the unfaulted phases. However, in reality, to keep the overall *power output* at the pre-fault level, the inverter may be expected to increase the power output of each its two unfaulted phases. To keep the total power output (across three-phases) constant, the power output of the remaining unfaulted phases would have to be raised to 150%, yielding the following relationship:

$$P = 1.5 \times P_{old} = \frac{V_{an}^2}{Z} \Rightarrow 1.5 \times \frac{V_{old}^2}{Z} = \frac{V_{an}^2}{Z} \quad \Rightarrow 1.5 \times V_{old}^2 = V_{an}^2 \Rightarrow V_{an} = 1.22 \times V_{old} \tag{B.5}$$

Note that this value is theoretical and in reality it might be limited by the maximum current settings of the inverter and other aspects studied later in this paper.

Consequently, if the inverter is supposed to supply the same amount of load before and after the occurence of the SLG, the voltage on the load side for the unfaulted phases can typically increase up to 122% of the initial value. Clearly, this value is much lower than the value (173%) derived for synchronous generators. Note that this hypothesis is only valid if the PV-inverter operates as a perfect constant-power source, which only applies when the PV inverter controls are set to that operating mode.

APPENDIX C
THEORETICAL ESTIMATION FOR IMPEDANCE IN GROUNDING TRANSFORMERS

Figure C.1 demonstrates the sequence diagrams of a system fed by an inverter based DER with (b) and without (a) the grounding transformer. It can be seen that the grounding transformer modifies the zero sequence network of the system. The representative circuit can be further simplified for a balanced Yg connected system by putting $Z_{load_0} = Z_{load_+} = Z_{load_-} = Z_L$.

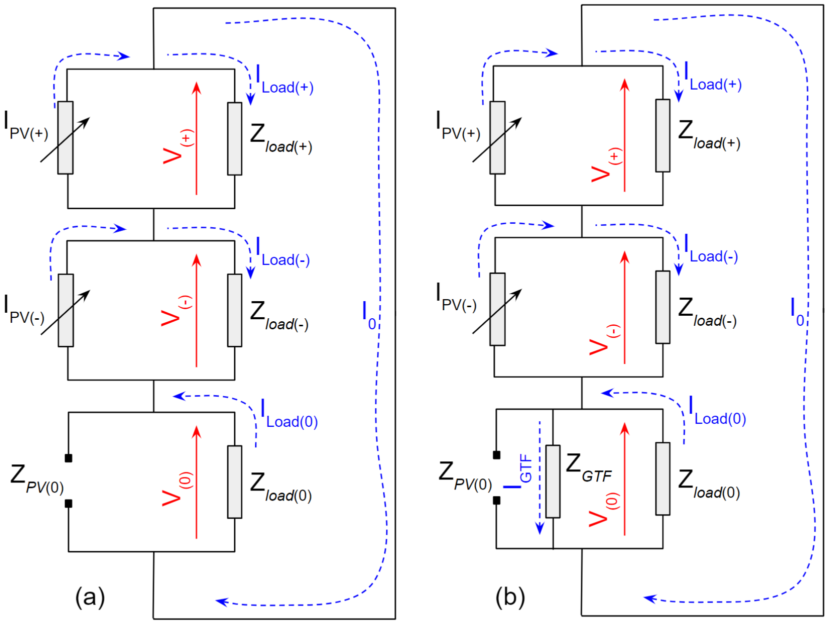


Figure C.1: Reference to the Sequence Diagram of IBRs Under SLG: (a) Without the GTF, (b) With the GTF, Reproduced From Figure 6.5

For the system in Figure C.1.(b), the equivalent combined series impedance of the negative and zero sequence networks would be given as,

$$Z_{0-} = \frac{Z_L Z_{GTF}}{Z_L + Z_{GTF}} + Z_L = \frac{Z_L^2 + 2Z_L Z_{GTF}}{Z_L + Z_{GTF}} \tag{C.1}$$

Where Z_L is the load impedance and Z_{GTF} is the impedance of the grounding transformer used in the system. Applying the current division rule, it can be obtained that the current through the positive sequence network can be expressed as

$$I_{-0} = \frac{Z_L + 2Z_{GTF}}{2Z_L + 3Z_{GTF}} I_{PV} \tag{C.2}$$

These expressions assume that I_{PV} is the fault current after the SLG fault. Using these expressions for currents, the sequence components of the three phase voltages are calculated as

$$V_0 = -\frac{Z_{GTF}}{2Z_L + Z_{3Z_{GTF}}} I_{PV} Z_L, V_+ = \frac{Z_L + 2Z_{GTF}}{2Z_L + Z_{3Z_{GTF}}} I_{PV} Z_L, V_- = -\frac{Z_{GTF}}{2Z_L + Z_{3Z_{GTF}}} I_{PV} Z_L \tag{C.3}$$

From these expressions, the phase voltages of the unfaulted B and C phases can be calculated as

$$V_b = \frac{I_{PV} Z_L}{2Z_L + 3Z_{GTF}} [2a^2 Z_L + 3a^2 Z_{GTF} + Z_L], V_c = \frac{I_{PV} Z_L}{2Z_L + 3Z_{GTF}} [2a Z_L + 3a Z_{GTF} + Z_L] \tag{C.4}$$

It is crucial to observe that the expressions for V_b and V_c are different, and there is no single value of Z_{GTF} which can minimize both their magnitudes at the same time. This analysis was supported by the analysis reported by the authors in [14] and is further validated through experimentation in section IV. The value of Z_{GTF} that minimizes $V_b(Z_{GTF-B})$ and the value of Z_{GTF} that minimizes $V_c(Z_{GTF-C})$ can be expressed as,

$$Z_{GTF-B} = -\frac{(2a^2+1)Z_L^2 I_{PV} - 2Z_L V_b}{3V_b - 3a^2 Z_L I_{PV}}, Z_{GTF-C} = -\frac{(2a+1)Z_L^2 I_{PV} - 2Z_L V_c}{3V_c - 3a^2 Z_L I_{PV}} \tag{C.5}$$

This demonstrates that there is no solution for Z_{GTF} which minimizes post-SLG V_b and V_c simultaneously.

APPENDIX D
HARDWARE AND SOFTWARE UTILIZED FOR CHIL EXPERIMENTS

D.1 Hardware Utilization

The hardware components used for experimentation are described in this section.

D.1.1 HIL-604 Real-Time Simulator

HIL-604 is an RTS manufactured by Typhoon HIL. It is predominantly used to simulate power-electronic systems and microgrids in real-time for different applications, including CHIL. The computational hardware of the HIL-604 consists of Xilinx Vertex FPGA cores alongside ARM real-time processors. The RTS has both digital and analog input and output channels that can be used to interface it with an external hardware. The simulator has been tested by the manufacturer for accuracy for a simulation speed up to to 2 MHz, while the PWM functionalities operate satisfactorily down to a resolution of 20 ns. The simulator is controlled by the user through the *Typhoon HIL SCADA* or *HIL Schematic Editor* software that run in a separate PC and communicate with the RTS via a USB 2.0 interface. While, this paper focuses on external hardware based control of the real-time model of the inverter, it is possible to simulate the controller models alongside the power electronic circuit of the inverter inside the real-time simulator.

D.1.2 ASGC CHIL

This controller hardware is the Austrian Institute of Technology Smart Grid Converter (ASGC) [77, 87]. It supports a broad range of communication protocols including IEC61850, ModBus TCP, SunSpec, etc., to interact with external hardware. Moreover, it provides analog and digital wired interfaces to couple it with sensor outputs, PWM signals, and additional capabilities to couple it to the HIL-604 RTS, The controller is capable of full four-quadrant bidirectional operation and has various in-built functionalities such as Volt-VAR control, pf control, Volt-Watt control, low and high-voltage ride-through, etc.

D.1.3 Interface Between HIL-604 and ASGC

The wired connection between HIL-604 and ASGC is based on a 192-pin on-board snap-in male-to-female configuration. This board makes all the 128 digital input and output pins available for easy access, and provides interfaces to couple with the analog channels as well.

D.2 Software Utilization

The various software tools used in this research are outlined in this section.

D.2.1 Schematic Editor

This software utilizes the Typhoon's component model library to construct the overall system model. While the library is smaller than those of other popular environments, it is possible to extend the library by incorporating user-defined models at the algorithmic level by using custom C functions or at the at the architectural level by assembling simpler blocks together [86].

D.2.2 HIL SCADA

This software provides a monitoring and control panel to interacts with the system model running on the real-time simulator. For the case of the ASGC, it also allows monitoring and adjustment of the controller. HIL SCADA provides data acquisition functions with capabilities to adjust the simulation model on run-time. It is possible to write macros that can help to automate simulation experiments, for example, applying a certain sequence of events. One such macro was developed for automating the GFOV testing. For the experiments reported below, two different implementations of the macro were developed to support the two different versions of HIL SCADA software that were used. HIL SCADA 8.4 (or above) was used for interacting with the model being simulated on HIL-604, while an older version - HIL SCADA 3.7 - was used for interacting with the ASGC controller.

D.2.3 aBoot Flasher

aBootFlasher is a software under MIT license provided by AIT. This is used to reconfigure the ASGC controller via USB COM port from the host PC. This reconfiguration can enable the ASGC controller to operate under different current and voltage ratings. This was an important requirement to analyze the SPOV functions of the controller.

D.2.4 Firmware Manager

This tool manages the configuration of the HIL-604. Typhoon simulators can be configured to execute models using a number of parallel processor-cores. The configuration needs to be chosen judiciously and the model's schematic diagram must be carefully partitioned into those cores. This is required so that the model runs within the desired computation time per core and without over-runs. Presently Typhoon HIL-604 can operate in 5 different configurations, among which *Configuration 4* was found to suit the proposed experiments. This configuration partitions the simulator into three sizable processor cores that are large enough to house complex components such as inverters.

APPENDIX E
SUPPLEMENTAL MATERIALS

Figures 1.2, 1.3, 1.4 were taken from external sources. Among these, figures 1.2 and 1.3 were taken from the surveys performed by *Ren21*. A permission was obtained via email to reuse their figures. The permission is attached as Ren21.pdf.

Figure 1.4 was obtained from a published article in newspaper reports in electronic media, i.e. *microgridknowledge*. Additional permission was obtained from *microgridknowledge*'s representative via email. The permission email is attached as MGK.pdf.