Pacific Northwest
NATIONAL LABORATORY

PNNL-38264

Integration and Validation of Multi-Layer Mitigation Strategies for Cyber Physical Systems Resilience (RD2C Capstone I Project)

Technical Report

October 2, 2025

Sai Pushpak Nandanoori
Ramij Raja Hossain
Samrat Acharya
Soumya Kundu
Laurentiu Marinovici
Thomas Edgar
Jan Westman
Alexander Schad
Burhan Hyder
Kyung-bin Kwon
Celina Wilkerson
Veronica Adetola
Shwetha Niddodi
Manisha Maharjan
Oceane Bel
Sayak Mukherjee
Thanh Long Vu

U.S. DEPARTMENT of ENERGY

Prepared for the U.S. Department of Energy
Under contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-38264

Integration and Validation of Multi-Layer Mitigation Strategies for Cyber Physical Systems Resilience (RD2C Capstone I Project)

Technical Report

October 2, 2025

Sai Pushpak Nandanoori
Ramij Raja Hossain
Samrat Acharya
Soumya Kundu
Laurentiu Marinovici
Thomas Edgar
Jan Westman
Alexander Schad
Burhan Hyder
Kyung-bin Kwon
Celina Wilkerson
Veronica Adetola
Shwetha Niddodi
Manisha Maharjan
Oceane Bel
Sayak Mukherjee
Thanh Long Vu

Prepared for
the U.S. Department of Energy
Under Contract DE-AC05-76RL01830
Pacific Northwest National Laboratory
Richland, Washington 99352

Executive Summary

This technical report documents the RD2C Capstone I project focused on integrating and validating multi-layer mitigation strategies to enhance cyber-physical systems resilience. The work centers on developing a cohesive, hierarchical control framework and validating it on a high-fidelity, real-time cyber-physical testbed.

- **Objective:** Design, integrate, and validate a hierarchical control system that combines multiple mitigation strategies to improve resilience of power distribution systems under diverse disturbances.
- **Modeling and Simulation:** Developed and configured detailed HYPERSIM models for distributed energy resources (DERs), including grid-forming (GFM) and grid-following (GFL) inverters. Requirements, core functions, and model parameterization workflows are specified, with automated DER configuration enabled via CSV and JSON templates.
- **Systems Under Study:** A large-scale 3000-node distribution feeder with 10 microgrids (comprehensive DER listings and decoupling/compensation locations) and an IEEE 123-node architecture with added DERs. These serve as platforms to exercise the mitigation strategies and control hierarchy across different scales and configurations.
- **Hierarchical Control Strategies:** A cohesive control architecture integrates multiple strategies operating at different timescales with a summary of required inputs/outputs and operational data for each control strategy. Under-frequency load shedding (UFLS) parameters and other strategy-specific settings are documented. The roles of each strategy within the hierarchy and their coordination are described.
- **Real-Time Cyber-Physical Testbed:** A virtualized environment couples HYPERSIM power system simulation with an NS3 communication network simulation. Controls are deployed across host virtual machines (with documented VM/IP mappings). Sensing and actuation use DNP3, with detailed analog input/output mappings and a PYDNP3 aggregator to orchestrate data exchange between simulation components and controllers.
- **Experimentation and Resilience Assessment:** Multiple scenarios and trial sequences are executed to evaluate the hierarchical control system. The report provides a structured scenario catalog, event sequences, and trial numbering, alongside resulting resilience assessments.
- **Tooling and Automation:** A flowchart-guided process enables automated updates of DER model parameters and configuration files, supporting repeatable, scalable experimentation across complex feeder models.
- **Outcomes and Future Work:** The project delivers an integrated, validated testbed and control framework, comprehensive model libraries and data pipelines, and a catalog of resilience experiments and assessments. The report concludes with major accomplishments and identifies opportunities for future work to deepen validation, expand scenario coverage, and further refine multi-layer mitigation strategies.

Acknowledgments

This research was supported by the Resilience through Data-Driven Controls (RD2C) Initiative, under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Acronyms and Abbreviations

PNNL	Pacific Northwest National Laboratory
SLAC3R	Self-Aware Local Autonomous and Semi-Cooperative Control for Cross-Layered Resilience
DAC	Decentralized Autonomous Controls
SLAC3R-L	Local Autonomous Controls part of SLAC3R or DACs
SLAC3R-C	Cooperative controls part of SLAC3R
ALERT	Adaptive Learning-Enabled Resilient Tuning Controls
P2PC	Peer-to-Peer Controls
PACP	Physical Aware Cyber Platform
CUT	Controls-Under-Test
EMT	electromagnetic transients
API	application programming interface
DER	Distributed Energy Resource
PV	Photovoltaic
ADMS	Advanced Distribution Management Systems
MG	Microgrid
NMG	Networked Microgrid
CPS	Cyber Physical System
DER	Distributed Energy Resource
GFM	Grid Forming
GFL	Grid Following
MiTM	Man–in–The–Middle
DDoS	Distributed–Denial–of–Service
RD2C	Resilient Data-Driven Controls

Contents

Figures

Tables

1.0 Introduction

Modern power grids are undergoing a transformation, shifting from large, fossil fuel-based power plants to systems incorporating numerous distributed energy resources (DERs) at the local level. Simultaneously, advanced communication networks are being developed to manage and control these DERs effectively. As a result, next-generation cyber-physical systems (CPS) will face significant challenges in continued, resilient operation of electric grid operations due to a dramatic increase in vulnerabilities to extreme weather and cyberattacks [1–3]. To preserve and even enhance the resilience of these future electric energy systems, new mitigation strategies that operate at each layer i.e., device, cyber and application layer of the CPS control architecture must be developed and tested against these blended scenarios in high fidelity simulation environments. Concurrently, rapid changes in grid architectures include the proliferation of multiple third-party energy entities such as system operators, aggregators, and transactive aggregators. This complexity significantly complicates control and operation, presenting challenges in analyzing the resilience and security of these systems across various conditions. Given that these third-party energy entities often pursue multiple objectives, effective coordination and control are crucial to ensure safe, reliable, and resilient operation. This decentralization can lead to vulnerabilities, especially in the face of adversarial threats. To address these challenges, advanced algorithms are essential. These algorithms should enable multiple decision-making agents within the network to act autonomously, collaboratively and adaptively. This means that individual components of the network should be able to make decisions based on local information, but they should also coordinate with other components and adapt to the system conditions to maintain the overall system's resilience. In addition, interconnection between different cyber-physical systems involving various types of networks such as mesh, 5G, optical fiber networks, different types of communication protocols and different types of control would introduce heterogeneous security and quality of service (QoS) requirements which need to be addressed at the cyber layer.

In the scope of this project, we define resilience to be the ability of the system to detect and mitigate any adversarial actions that threaten the reliable operation while also ensuring to maintain the system operating state within the operational constraints even in the emergence of high-impact low probability natural disturbances. Current control strategies often fall short in ensuring resilience. Local controls, such as inverter controls in microgrids, are typically designed to operate myopically, focusing on immediate performance without considering the broader system's health. Centralized controls, while useful for overall system management, are often too slow to respond to rapid disturbances or adversarial events. Many adversarial events in critical cyber-physical networks can lead to cascading failures. These failures start with local disruptions that can quickly spread throughout the network, causing widespread damage. Current resilience practices either involve long-term, and expensive, investment decisions; or are reactive in nature, relying on predominantly ad-hoc (simulations-based) strategies that fail to provide resilience guarantees and/or requiring intervention of human operators who are prone to mistakes and introduce response delays. Moreover, it is often cost-prohibitive to redesign existing baseline controls for resilient performance.

To ensure the operational resilience of complex cyber-physical networks, advanced algorithms are needed that enable autonomous, collaborative, and proactive resilience strategies. These algorithms must be capable of handling both deterministic and stochastic disturbances, as well as adversarial events. It is desirable that such a resilient control design would render an add-on layer such as the ones considered in this capstone effort for a preferable, easily implementable, and reconfigurable solution. By addressing the limitations of existing control strategies and providing system theoretical guarantees of (quantitative) resilience performance measures, it would not only help develop trust in the automated control algorithms and accelerate real-world deployment, but such guarantees would also provide quantitative specifications of sensor/control performance for

resilient co-design efforts.

The RD2C initiative has developed several independent mitigation strategies over the past few years, each targeting different system layers and timescales. ALERT 1.0 is an application-layer system designed to address long-term (tertiary timescale) threats to cyber-physical systems [4]. It adapts existing control strategies in real-time to counter cyberattacks, ensuring a sufficient safety margin for resilience. Coordinating and networking multiple CPS provide the operational flexibility needed to support end-users and can achieve common goals which are impossible to obtain by independent systems, but the networked CPSs are also exposed to cyber physical attacks due to complicated interconnections. Similarly operating at the application layer, the peer-to-peer (P2P) project [5] coordinates networks of CPS to withstand cyber physical attacks and failures through P2P communication. PACP is a middleware solution [6] with in-built cyber-threat intelligence to detect and mitigate attacks at the cyber layer. In addition to detection and mitigation of cyber-attacks at cyber layer, it informs the control applications connected to it with detected threats, probability of event, confidence level etc. thereby aiding applications in applying cyber-risk informed control and coordinated strategies. In contrast, SLAC3R-L and SLAC3R-C are device-level control systems operating on shorter timescales (primary and secondary) [7]. SLAC3R-L controls are adaptive, lightweight algorithms that empower devices like grid forming inverters (GFM) inverters to autonomously maintain operations under unforeseen high-impact-low-probability events. SLAC3R-C, on the other end, coordinates power distribution and frequency restoration among inverters with minimal communication. SLAC3R controls have been rigorously tested on high fidelity simulation of IEEE 123-bus test system. These different mitigation strategies will be integrated and evaluated on high-fidelity simulation environments including both natural and adversarial scenarios, to assess their effectiveness in safeguarding critical systems. The integration of several detection and mitigation strategies across many RD2C projects will demonstrate that these strategies can be coordinated to work seamlessly at different timescales in the test environment to address the threats from complicated cyber physical attacks and failures to maintain and improve the resilience at different layers of CPS and networked CPS.

As part of this Capstone project effort, we aim to achieve resilience to various natural and adversarial events across multiple timescales with integrated mitigation strategies (SLAC3R, ALERT 1.0, P2PC, and PACP) that are autonomous, collaborative, and adaptive. These different mitigation strategies complement one another to promote resilience across various timescales. The SLAC3R-L controls are already integrated directly at the inverters in the HYPERSIM, and the SLAC3R-C controls are implemented using additional Python agents where the inverter data and the setpoints are exchanged. The PACP will serve as a platform to enable data and setpoint transfer between the inverters and controllers (SLAC3R-C, ALERT 1.0 and P2PC) which are implemented as Python scripts on virtual machines. A unified methodology is developed to integrate these different mitigation strategies that enable them to work coherently and implemented on high-fidelity HYPERSIM simulator. One of the challenge this integration may result in is the action of multiple mitigation strategies for a single disturbance event on the system. However, as these different mitigation strategies are integrated in a hierarchical manner (as they operate at different time scales), to ensure resilience, the final control setpoint action is assigned to the SLAC3R-L controls which ensures resilience by maintaining the operating limits of frequency and voltage. Finally, the test scenarios are chosen in such a way that all the mitigation strategies are enabled to demonstrate the resilience. Fig. 1 illustrates the hierarchy of controls in a networked microgrid operation.

To test the novel controller hierarchy and assess resilient performance improvements, the RD2C Capstone 1 project has built a high-fidelity, real-time cyber-physical testbed. This testbed is composed of building blocks from the prior efforts in the RD2C initiative by leveraging the work products and expertise developed in modeling communication systems and electric power distribution systems in real-time simulation platforms.

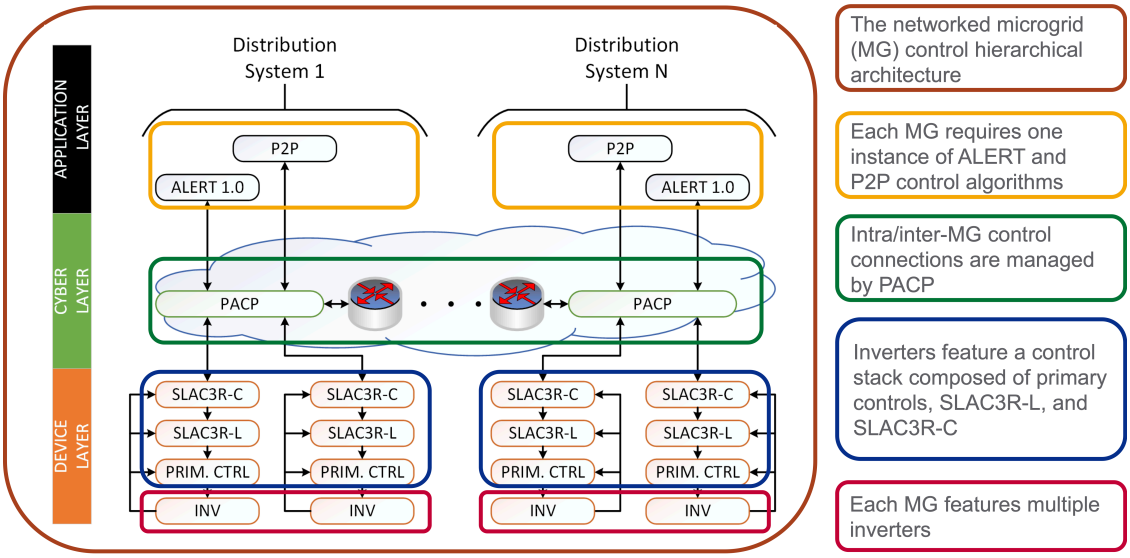


Figure 1: A control architecture for networked microgrids with mitigation strategies to improve resilience at each CPS layer.

For real-time power system simulation, we have built multiple distribution system models in OPAL-RT's electromagnetic transients (EMT) simulation platform called HYPERSIM. These models can be loaded onto PNNL's OPAL-RT real-time digital simulators, which allows controller devices to exchange signals with the simulation in real-time, which closely approximates real-world controller performance. HYPERSIM provides high-fidelity modeling of the distribution system network as well as the critical dynamic components, such as the distributed energy resources (DERs). The team has built upon the IEEE 123 Node Test Feeder [8] model developed in previous phases of the RD2C initiative by adding additional DER models, improving the benchmarking of the DER models, and adding interfaces for external DER and microgrid controls hosted on Cybernet virtual machines. Additionally, the team has created a HYPERSIM model of the third feeder in the IEEE 9500 Node Test Feeder [9], which the team is referring to as the 3000 Node Test Feeder. This model represents a substantial increase in the number of nodes, DERs, and microgrids. This model provides a new platform to rigorously test the scalability of resilient mitigation strategies in real-time. Given the dramatic increase in the scale of the HYPERSIM models, the team has developed new software tools in python to help with model parameter management.

For real-time communication network simulation, the team has developed a wide area network (WAN) model in NS3 [10]. The model is containerized and installed on a virtual machine hosted by PNNL's cloud computing infrastructure referred to as Cybernet. Cybernet enables flexible creation of virtual machines and virtual computer networks as well as connection of power system simulators and real controller hardware. The NS3 model contains network taps that are bridged to the network interfaces on the host virtual machine, allowing external devices to communicate across separate local area networks using routes that span the NS3 wide area network model. Therefore, the team was able to configure and control disruptions to the data flows between the various control systems integrated into our testbed experiments to simulate the effects of cyberattacks. The team integrated software tools to automate the process of configuring NS3 simulation disturbances and visualizing simulation results, leveraging other products from the RD2C initiative including the NATIG and GLIMPSE software.

To complete the cyber-physical testbed hosted with PNNL's Cybernet infrastructure, the NS3 host virtual machine and the OPAL-RT simulators were networked with the virtual machines

hosting the multi-layered, resilient, hierarchical control system. This testbed allowed the team to develop the processes for integrating and deploying the mitigation strategy software. Furthermore, using the newly developed testbed, the team conducted multiple experiments composed of several trials to determine the incremental change in system resilience by adding each layer of the controller hierarchy. The team also developed new software tools to automate the process of running the HYPERSIM simulations, which has improved the repeatability of executing the experiments. These tools have been integrated with the model parameter management tools to create a multi-functionality, scripting-based interface with HYPERSIM which will significantly improve the feasibility of managing large, real-time distribution system models in HYPERSIM. As a result, the team was able to generate many real-time simulation datasets from the various experiment trials that will be cataloged and available for future use.

Finally, the team made efforts to develop quantification of the resilience improvement of the system operation based on assessing the time series simulation data generated from HYPERSIM. A literature review was conducted and several metrics were identified that can assess the performance of the system using a single figure of merit (FOM) during different phases of a resilient response. The metrics can be aggregated into a single, overall score, which can be used to compare the system performance before and after introducing a new mitigation strategy.

This report has been organized into multiple sections describing the various aspects and outcomes of the RD2C Capstone 1 project. Following this introduction, Section 2.0 describes the HYPERSIM model development effort. Then Section 3.0 provides background on the mitigation strategies that were integrated into the cohesive, hierarchical control system as well as descriptions of what roles each mitigation strategy plays. Next, Section 4.0 describes the cyber-physical testbed used to test the hierarchical control system including: the testbed virtual network architecture; deployment of the controls onto to host virtual machines; integration of the HYPERSIM power system simulation; and integration of the NS3 communication network simulation. A summary of the experiments conducted with the cyber-physical testbed and the resulting resilience assessments are provided in Section 5.0. Finally, Section 6.0 concludes the report with the major outcomes, accomplishments, and opportunities for future work.

2.0 HYPERSIM Distribution System Models

This section presents approaches and outcomes of developing high-fidelity, real-time distribution system models for simulating physical power system disturbances and end-effects of the impacts of cyber-attack on power system operation. Section 2.1 contains the results of attempting to develop a 3000 Node Test Feeder model based on Feeder Three in the IEEE 9500 Node Test Feeder distribution system benchmark. Then Section 2.2 provides a brief discussion of modifications of the IEEE 123 Node Test Feeder model developed in a prior RD2C thrust, which helped create better use cases for the controls-under-test in this project. Finally, substantial effort was made to benchmark reduced order DER models, which is documented in Section 2.3.

2.1 IEEE 9500 Node Feeder Three Model

2.1.1 Model Description

The IEEE 9500 Node Distribution Test System is one of the largest available test feeders that facilitates wide variations in the adoption of advanced grid-technology as the model for scalability testing. It is a comprehensive and high-fidelity model designed to simulate modern electric distribution networks [9]. The model is an extension of the widely used IEEE 8500 Node Test Feeder, developed to support advanced power applications and real-time operational studies. It represents a realistic distribution system for large-scale simulations with multiple feeders, substations, distribution circuits, rooftop photovoltaics (PV), and multiple utility-scale DERs. This test system enables real-time assessment of advanced power applications, including simulations of operational platforms like Advanced Distribution Management Systems (ADMS) and Distributed Energy Resource Management Systems (DERMS) within a distribution control center environment. Fig. 2 incorporates modern feeders with enhanced smart grid technologies with legacy feeders comprising of 9500 nodes (unbalanced), 2 battery energy storage systems, 180 rooftop inverters, and 9 diesel generators.

2.1.2 Model Development

To create a large-scale, real-time distribution system model for generating a variety of cyber-physical scenarios and testing potential mitigation strategies, it was necessary to automate the model building activities. Thus, this project leveraged the PNNL GridLAB-D to HYPERSIM conversion tool, which was created during a prior RD2C project. The goal was to create a real-time, EMT model of the third feeder of the IEEE 9500 Node Test Feeder system, which represented an ambitious increase in the scale of distribution system models developed in HYPERSIM at PNNL. To reduce the model complexity and computational burden, single phase loads at the triplex level were aggregated up to the medium voltage nodes in this system. The resulting distribution model with 3000 nodes is large enough to investigate cyber-physical use cases. The next subsection discusses how the RD2C Capstone 1 project utilized the conversion tool to substantially reduce the labor required for the HYPERSIM model development.

2.1.2.1 Demonstration of the GridLAB-D to HYPERSIM Conversion Tool

Manually populating the 3000 Node Test Feeder model canvas with the passive network components would have presented an expensive, time consuming task. As a result, the project leveraged a GridLAB-D to HYPERSIM conversion tool that was developed in a previous RD2C

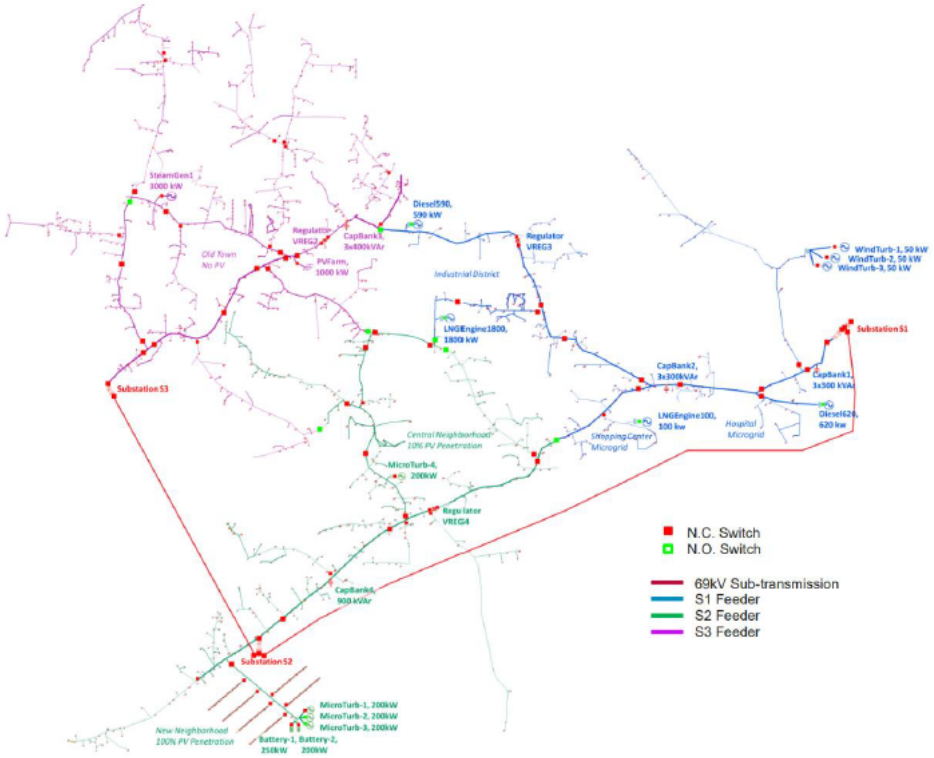


Figure 2: Topological map of the IEEE 9500 node test feeder illustrating the three main sub-feeders.

project [11]. Although no conversion tool development occurred as part of this capstone, we discuss the conversion tool functionality here to provide context for the first use of the tool to accomplish a modeling task.

Fig. 3 shows the flowchart for the detailed methodology used in the conversion tool. The tool utilizes the extensive library of large-scale feeder models developed in low-fidelity platforms like GridLAB-D. Thus, these low-fidelity models are used to convert to high-fidelity models that are capable of capturing EMT phenomena.

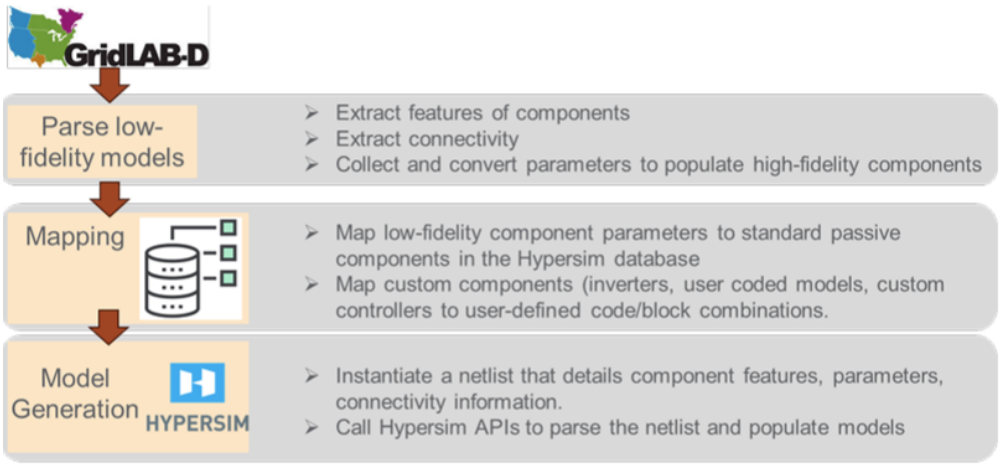


Figure 3: GridLAB-D to HYPERSIM conversion tool.

To automate generation of the 3000 Node Feeder model passive network, the GridLAB-D IEEE 9500 Node model is parsed to gather the component definitions and connectivity to create a basic graph of the feeder. Then the component parameters are retrieved and transformed into definitions that are compatible with HYPERSIM component parameter definitions. A netlist of components is created in a text-based format specifying dictionaries of data about buses, line parameters, transformer parameters, switch states, load definitions and other relevant parameters. The netlist of the component definitions is then mapped with the equivalent HYPERSIM built-in library components or with custom components built to match the features of low-fidelity components. Then, a python-based API interface is used to populate the components in the HYPERSIM environment and create the large-scale feeder model in HYPERSIM. Large models can now be translated to the high-fidelity domain using this tool, but simulating these models is still challenging due to its computational burden. Fig. 4 shows the HYPERSIM model populated from the 3000 Node test feeder in GridLAB-D using this conversion processes. The developed model was tested to benchmark and match steady state power flows as compared to its GridLAB-D counterpart

Using the conversion tool on the Capstone 1 project resulted in the first practical demonstration of the tool to build a model that was intended to accomplish simulation tasks for a project. Overall, the tool demonstration was a major success as substantial labor was avoided by automatically populating passive network components with fairly high accuracy (some very minor errors were identified and corrected). In-spite of the generally positive outcome, some limitations of the current form of the tool were identified, which provides valuable insight into next steps for development for next steps. These limitations included:

- Inability to populate line type decoupling components at pre-defined locations;
- Limited ability to control the spot load model type by node number;

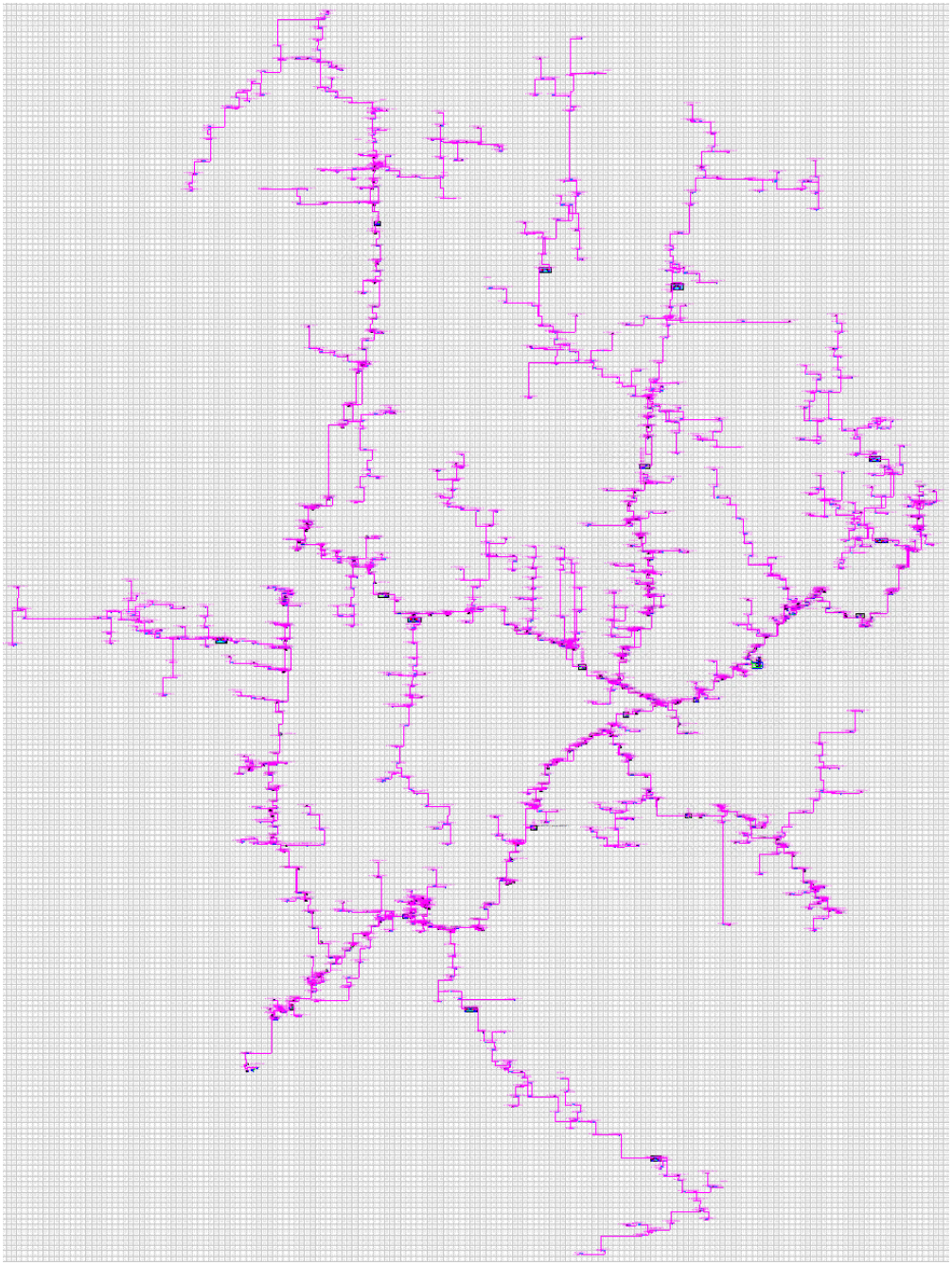


Figure 4: Real-time HYPERSIM model of 3000 Node Test Feeder.

- Limited ability to specify the DER model type and parameters by node number.

To address the limitation regarding the parameterization of a large number of DER models, a new python based tool using the HYPERSIM python API was developed and is discussed in Section 4.2.1. Addressing the other limitations is left for future work.

2.1.2.2 Placement of Decoupling Elements

The simulation of large scale models with associated controls in real-time simulators can be computationally burdensome. Thus, usually the real-time simulators decouple the system model into different sections to run in individual cores independently to achieve the real-time simulation of large-scale models. In prior RD2C work, a decoupling approach with appropriate compensation technique have been developed to enable real-time simulation of large-scale models without losing accuracy [12]. Several decoupling elements were added in the 3000 Node Test Feeder model to reduce the total computation times and to maintain a time-step of 50 μs for real-time simulation. These decoupling components enable rigorous real-time execution by distributing the computational load among several processors. Shunt reactive power compensation components were also added, which would adaptively adjust the reactive power compensation to provide based on the system voltages, to compensate for the reactive power errors caused by the decoupling elements. Table 1 highlights the decoupling locations with the quantities for decoupling elements and compensation values for the test feeder.

Table 1: Decoupling Locations and Compensation Table in 3000 Node Test Feeder

Name	Type	Location		Pi-Sections Replaced	Nodes Eliminated	Total Series R (Ω)	Total Series L (uH)	Total Shunt C Added (uF)	Total Equiv. Var Injection (Mvar)
		From Node	To Node						
Dec_line_ln5591284_2	Line	m1026655	m1026658	line_ln5591284_2		0.0195	132.0	19.32	1.1325
Dec_line_ln6077802_1	Line	l3254234	l3104114	line_ln6077802_1		0.137704	196.355	12.988	0.253788
Dec_line_ln5472403_3	Line	n1140828	n1140825	line_ln5472394_1; line_ln5472403_2; line_ln5472403_3	m1026708; n1140828	0.03603813	243.47	10.475	0.61404
Dec_line_ln6047566_1	Line	l2673313	m1027013	line_ln6047566_1; line_ln5683819_1	m1027011	0.285740888	407.422	6.259511	0.366948
Dec_line_ln6019479_1	Line	m1026851_C_1	d6440002_2_int	line_ln6019479_1; line_ln6440002_3	p827536	0.03281	45.513	56.033	3.2848
Dec_line_ln5593236_6	Line	m1047513	n1144665	line_ln5593236_6; line_ln6138610_2; line_ln6138610_3	n1144665	0.045582276	307.979	8.28057	0.485427
Dec_line_ln5739188_1	Line	l2726973	l3122821	line_ln5739188_1; line_ln5829831_1	m1047485	0.18855	647.9	3.936	0.23076
Dec_line_ln5479790_2	Line	l2916234	m1047420	line_ln5479790_2; line_ln5835142_1	m1047423	0.527607	764.771	3.335	0.0651619
Dec_line_ln5985355_1	Line	l2783231	l3254227	line_ln5985355_1		0.252521	360.076	7.082	0.138397
Dec_line_ln5803283_1	Line	m1069588	m1069590	line_ln5803283_1		0.188165	272.746	9.35	0.18271
Dec_line_ln6077798_1	Line	l2766738	n1134478	line_ln6077798_1; line_ln5653479_1; line_ln5863714_3	m1069509; m1069505	0.03007184	203.182	12.552	0.73581
Dec_line_ln6077781_3	Line	d6077791_3_int	m1047534	line_ln6077781_3; line_ln6077781_2	n1136663	0.108563835	373.103	6.835	0.400701
Dec_line_ln6077782_1	Line	m1026769	m1026780	line_ln6077782_1; line_6290215_1	m1026774	0.06171	416.9	6.117	0.3585
Dec_line_ln5865229_1	Line	m1026987	m1027001	line_ln5865229_1; line_ln5956456_1	m1026990	0.531439	770.327	3.311	0.194075
Dec_line_ln5744367_1	Line	l3215549	l2897800	line_ln5744367_1		0.244671	354.654	7.191	0.140516
Dec_line_ln5623405_1	Line	m1026660	l2729428	line_ln5623405_1; line_ln5655683_1; line_ln6322630_1	m1026657	0.497425	721.022	3.537	0.069115
Dec_line_ln6077784_1	Line	l2822871	d5502543_2_int	line_ln6077784_1; line_ln5502543_2	m1026872	0.1014022	348.41	7.32	0.42909
Dec_line_ln5742898_1	Line	m1047737	l2973155	line_ln5742898_1; line_ln6076346_1	l3027157	0.466762	668.18	3.817	0.22374
Dec_line_ln6163390_1	Line	l3103822	m1047592	line_ln6163390_1; line_ln5619489_1	m1047577	0.14744686	506.72	5.033	0.295038
Dec_line_ln6380809_2	Line	l2973153	m1047672	line_ln6380809_2; line_ln6380809_3; line_ln5956464_1	n1136998; m1047669	0.134917572	189.85	13.433	0.7875
Dec_line_ln5894192_1	Line	m1009698	m1009715	line_ln5894192_1; line_ln6380825_1	m1009705	0.085512929	293.883	8.678	0.50871

2.1.2.3 Placement of Microgrids and DERs

To understand the effects of high penetration of DERs on microgrids, we needed to logically place additional DERs and design a microgrid structures with respect to the feeder locations, switches and DER placements. The structuring of the feeder model into various microgrids was required to develop grid-connected and islanded modes of operation of microgrids and develop scenarios that allows microgrids to operate independently and understand the role of DERs during islanded operation of the microgrids.

To develop the microgrid topology for 3000 Node Test Feeder, the feeder was partitioned into segments based on the existing three phase breaker locations. The segmented network as shown in Fig. 5 is later used to seggregate microgrids in the test feeder.

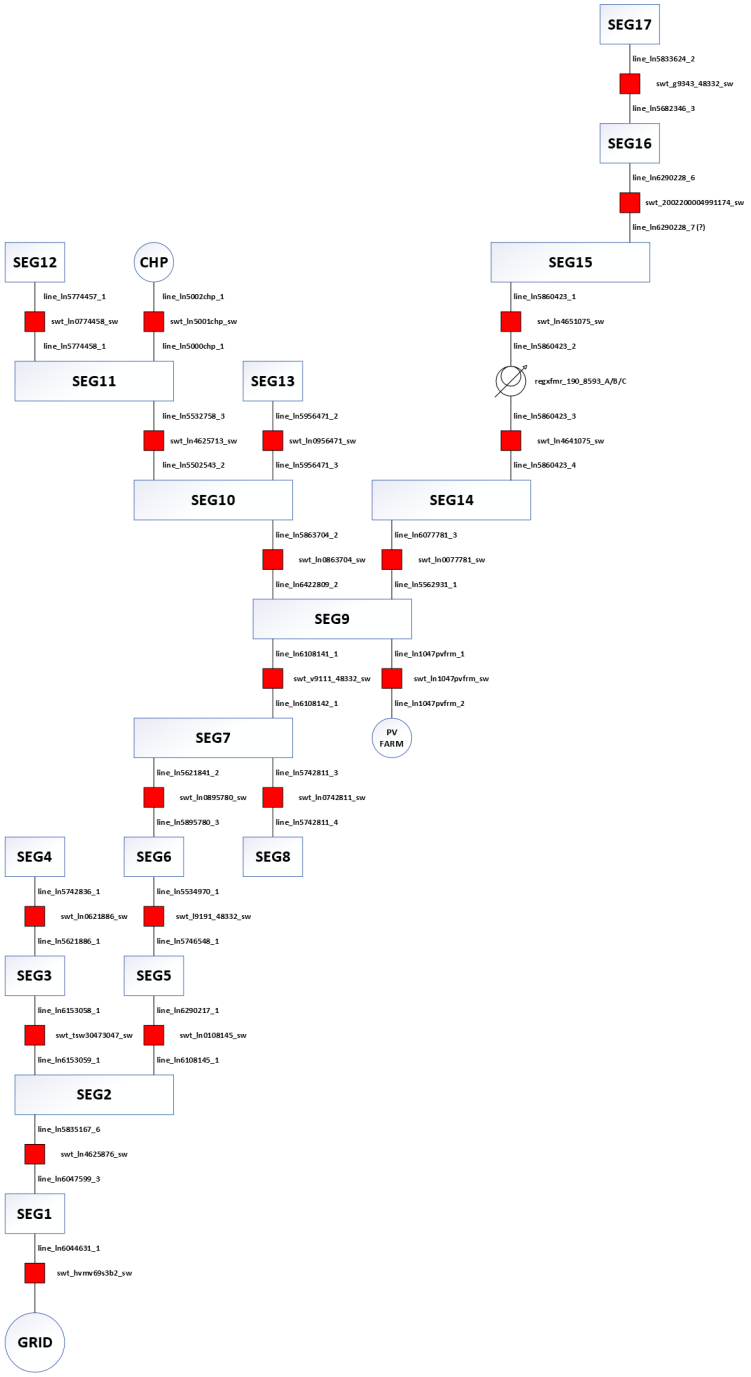


Figure 5: Preliminary segmentation of 3000 Node Feeder.

The existing DERs in Feeder Three of the IEEE 9500 Node Test Feeder were not specified to represent a feasible deployment allowing independent operation of a large number of small microgrids. There is effectively only one DER, the combined heat and power system (CHP) at node m1026, that provides a grid forming resource that could allow a microgrid to form. As a result, a large number of additional grid forming and grid following DERs were added to Feeder Three to accomplish the 3000 Node Test Feeder as illustrated in Fig. 6. The following process

describes how locations, sizes, and types of additional DER models were determined following the segmentation of Feeder Three into the 10 consituent microgrids.

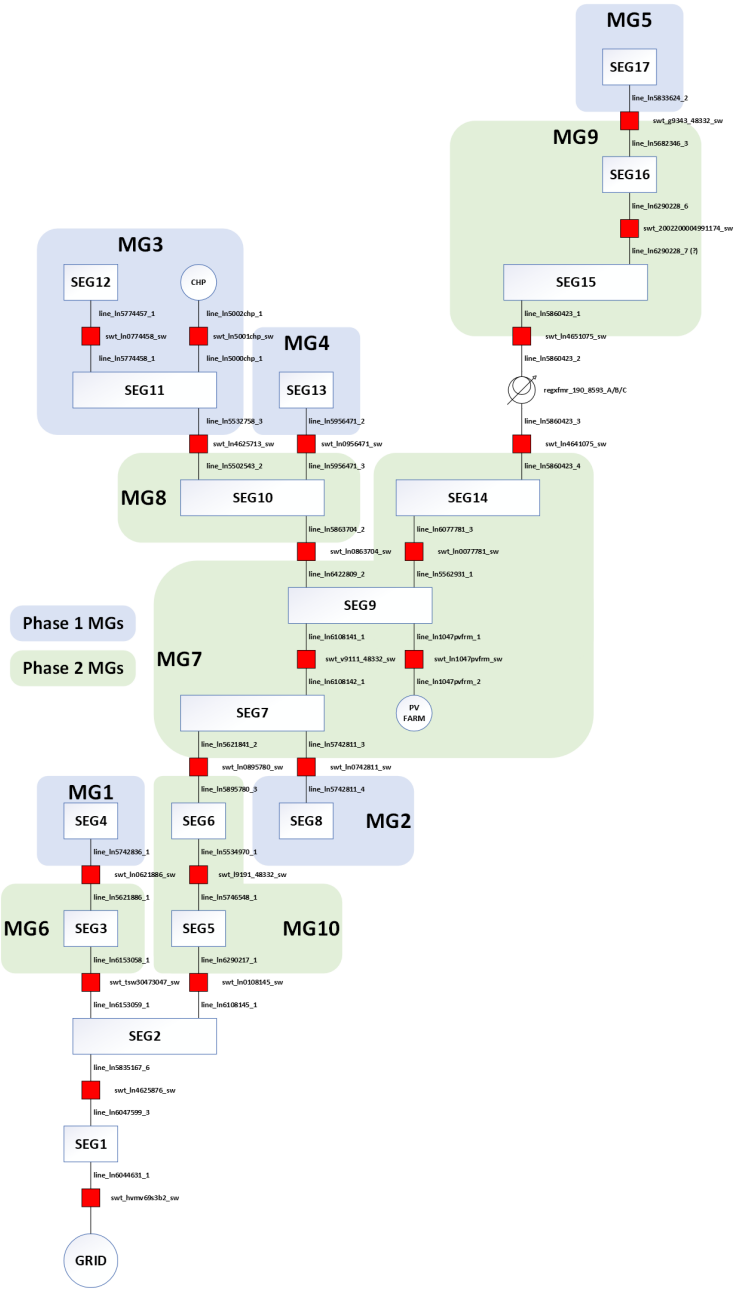


Figure 6: Proposed Microgrids for 3000 Node Feeder.

1. Identify the total load in each microgrid.
2. Classify each microgrid according to the total load:
   - Microgrids with less than 500 kW of load are considered small residential microgrids.
   - Microgrids with 500 kW of load or more are considered large commercial or industrial microgrids.

3. Propose DER sizes and types according to the individual microgrid classes:
    - Small residential microgrids have small battery energy storage resources (represented by single or three phase grid forming inverters) and small PV systems (represented by single phase, grid following inverters).
    - Large commercial or industrial microgrids have large, three phase battery energy storage resources, large three phase PV systems, and may have three phase diesel generators.
4. In general, make the combined DER capacity in each microgrid be 150% to 200% of the combined spot load in the microgrid.

The list of DERs modeled in the 3000 Node Test Feeder are provided in Table 2 through Table 4. The DERs were populated into the 3000 Node Test Feeder model by first developing a set of benchmarked models, which are contained in subcircuits, as described in Section 2.3. These subcircuits were then copy/pasted into the 3000 Node Test Feeder model, which was auto-generated using the conversion tool. Then each subcircuit was declared as a unique part type using the concatenation of the benchmarked DER model name and the DER instance name to create a unique part name. This was a critical step as any modifications made to contents of one subcircuit are automatically propagated to all subcircuits of the same part type. Therefore, in order to update the subcircuit parameters to reflect DERs of different ratings, each subcircuit needed a unique part type. To facilitate management of model parameters and streamline the process for updating the blocks inside the subcircuits (normally a very time consuming, manual process) a python automation tool was developed using the HYPERSIM python API. The tool allowed specification of models parameters in JSON format and then automatically updates the model parameters in a matter of seconds. More information about the python automation tool is provided in Section 4.2.1.

Table 2: List of DERs in 3000 Node Feeder Model: Microgrids 1 Through 4

Name	MG #	ParentNode	DERPhases	Rating(kVA)	ControlMode
inv_1	1	m1009805	ABCN	450	GFM IBR
inv_2	1	m1026844	BN	150	GFL IBR
inv_3	1	m1009807	BN	150	GFL IBR
inv_4	1	m1026829	CN	150	GFL IBR
dg_1	2	n1137992	ABCN	450	DG
inv_5	2	m1069468	ABCN	100	GFM IBR
inv_6	2	m1047486	ABCN	150	GFM IBR
inv_7	2	m1047484	ABCN	500	GFL IBR
inv_pv_pv_1115	2	l3728043	AN	20	GFL IBR
inv_pv_pv_1177	2	l2673308	BN	20	GFL IBR
inv_pv_pv_1081	2	l2897767	CN	20	GFL IBR
dg_steamgen1	3	m1026chp_3	ABCN	1,000	DG
inv_8	3	m1027013	ABCN	100	GFM IBR
inv_41	3	m1027023	ABCN	120	GFM IBR
inv_42	3	l3254213	ABCN	140	GFM IBR
inv_43	3	l3160103	ABCN	90	GFM IBR
inv_44	3	m1026954	ABCN	80	GFM IBR
inv_45	3	m1026977	ABCN	70	GFM IBR
inv_9	3	m1027039	ABCN	700	GFL IBR
dg_3	4	d59564712_int	ABCN	1,000	DG
inv_10	4	m1047713	ABCN	100	GFM IBR
inv_33	4	m1047572	ABCN	110	GFM IBR
inv_34	4	m1047615	ABCN	120	GFM IBR
inv_35	4	m1047688	ABCN	130	GFM IBR
inv_36	4	m1047724	ABCN	140	GFM IBR
inv_37	4	m1047732	ABCN	90	GFM IBR
inv_38	4	n1136028	ABCN	80	GFM IBR
inv_39	4	m1047580	ABCN	70	GFM IBR
inv_40	4	n1137986	ABCN	60	GFM IBR
inv_11	4	m1047720	ABCN	450	GFL IBR
inv_12	4	m1047763	ABCN	450	GFL IBR

Table 3: List of DERs in 3000 Node Feeder Model: Microgrids 5 Through 8

Name	MG #	ParentNode	DERPhases	Rating(kVA)	Type
dg_4	5	e206211	ABCN	450	DG
inv_13	5	n1136356	ABCN	300	GFM IBR
inv_14	5	m1069549	ABCN	300	GFL IBR
inv_pv_pv_1153	5	l2748143	AN	30	GFL IBR
inv_pv_pv_1159	5	l3104144	AN	30	GFL IBR
inv_pv_pv_1043	5	l2804272	BN	30	GFL IBR
inv_pv_pv_1003	5	l2673331	BN	30	GFL IBR
inv_pv_pv_1015	5	l3085410	BN	30	GFL IBR
inv_15	6	l3047059	ABCN	150	GFM IBR
inv_16	6	l2841623	AN	100	GFL IBR
inv_17	6	n1147857	BN	30	GFM IBR
inv_18	6	l2689691	BN	30	GFL IBR
inv_19	7	n1140519	AN	100	GFM IBR
inv_20	7	m1047513	AN	30	GFL IBR
inv_21	7	m1047550	AN	30	GFL IBR
inv_22	7	e206209	BN	20	GFM IBR
inv_23	7	m1047566	CN	150	GFM IBR
inv_24	7	m1047568	CN	60	GFL IBR
inv_pv_pv_1059	7	l2673318	CN	60	GFL IBR
inv_pv_pvfarm1	7	m1047pv3	ABCN	1,500	GFM IBR
dg_5	8	d58637042_int	ABCN	450	DG
inv_25	8	m3763619	ABCN	450	GFM IBR
inv_26	8	m1026830	ABCN	600	GFL IBR

Table 4: List of DERs in 3000 Node Feeder Model: Microgrids 9 Through 10

Name	MG #	ParentNode	DERPhases	Rating(kVA)	Type
inv_27	9	m1069513	ABCN	150	GFM IBR
inv_28	9	m1069514	AN	100	GFM IBR
inv_pv_pv_1167	9	l2897792	AN	90	GFL IBR
inv_29	9	l2841632	BN	30	GFL IBR
inv_30	9	m1069518	CN	30	GFL IBR
dg_6	10	m1026690	ABCN	600	DG
inv_31	10	m1026709	ABCN	450	GFM IBR
inv_32	10	m1026701	ABCN	300	GFL IBR
inv_pv_pv_1001	10	l3216348	AN	60	GFL IBR
inv_pv_pv_1169	10	l2673317	AN	60	GFL IBR
inv_pv_pv_1087	10	l3160106	AN	60	GFL IBR
inv_pv_pv_1071	10	l2841627	AN	60	GFL IBR
inv_pv_pv_1037	10	l2729428	AN	60	GFL IBR
inv_pv_pv_1049	10	l2814529	CN	30	GFL IBR
inv_pv_pv_1069	10	l3216368	CN	30	GFL IBR

2.1.3 GridLAB-D Modeling of the 3000 Node Test Feeder

In parallel with the HYPERSIM model development, a GridLAB-D model of the 3000 Node Test Feeder was also developed. The GridLAB-D model featured the same placement of microgrids and DERs as described in the proceeding sections (see Fig. 6 for the microgrid locations and Tables 2 through Table 4 for a list of DERs) and the goal was to create a replica of the HYPERSIM model in a phasor domain simulation platform. The purpose for developing a GridLAB-D model was to fast track steady state and dynamic testing of the new model so as to identify potential voltage and frequency stability issues. Although GridLAB-D performs simulation in the phasor domain and generally the DER models are significantly less detailed than EMT domain models, the dynamic testing of the GridLAB-D was useful because any stability issues identified in the phasor domain were likely to exist in the EMT domain as well since EMT is a higher fidelity modeling platform. Furthermore, as the conversion tool discussed in Section 2.1.2 is a relatively new tool, it was necessary to have a benchmark in a high trust simulation platform to validate the network automatically produced by the conversion tool.

To perform dynamic testing of the GridLAB-D model, we create a scenario in which the MGs are islanded by opening their respective point of interconnection (POI) switches. Upon initial dynamic testing, it was determined that Microgrid 6 (MG6) could not maintain stability during independent operation. As a result, MG6 is merged into MG1 and the revised 3000 Node Test Feeder model will contain just nine microgrids. The results of the dynamic simulation test are shown in Fig. 7.

Fig. 7 shows the frequency response and apparent power of the inverters in the MGs. A significant disturbance occurs at 5 seconds due to the microgrids transitioning to islanded operation, but all MGs remain stable thanks to the control of the inverters. This stability is further demonstrated in Fig. 7b, where the apparent power of the inverters changes to match the local microgrid load.

2.1.4 Model Comparison between GridLAB-D and HYPERSIM

Next, we perform steady state simulation of the IEEE 3000-bus system with the DERs specified in Table 2 through Table 4 in both GridLAB-D and HYPERSIM. The objective of the simulation is to compare the results between two simulation environments: GridLAB-D and HYPERSIM. Specifically, we measure the voltage and current flowing at a subset of buses and PI-Section lines during steady-state conditions. The comparison is conducted by analyzing the voltage and current differences between the two environments. The results, depicted in Fig. 8, show that the differences in voltage and current are within acceptable ranges. Specifically, the voltage difference is found to be within 3%, while the current difference remains within 5%. Notably, the simulation in GridLAB-D exhibits slightly higher voltage but smaller current compared to HYPERSIM results. Due to time constraints, no further modifications were made to the HYPERSIM model to bring the steady state network voltages and currents closer to the GridLAB-D model. However, some degree of error was expected due to the introduction of artificial shunt capacitance caused by the line type decoupling components, which allows the model to be simulated in real-time on parallel CPU cores.

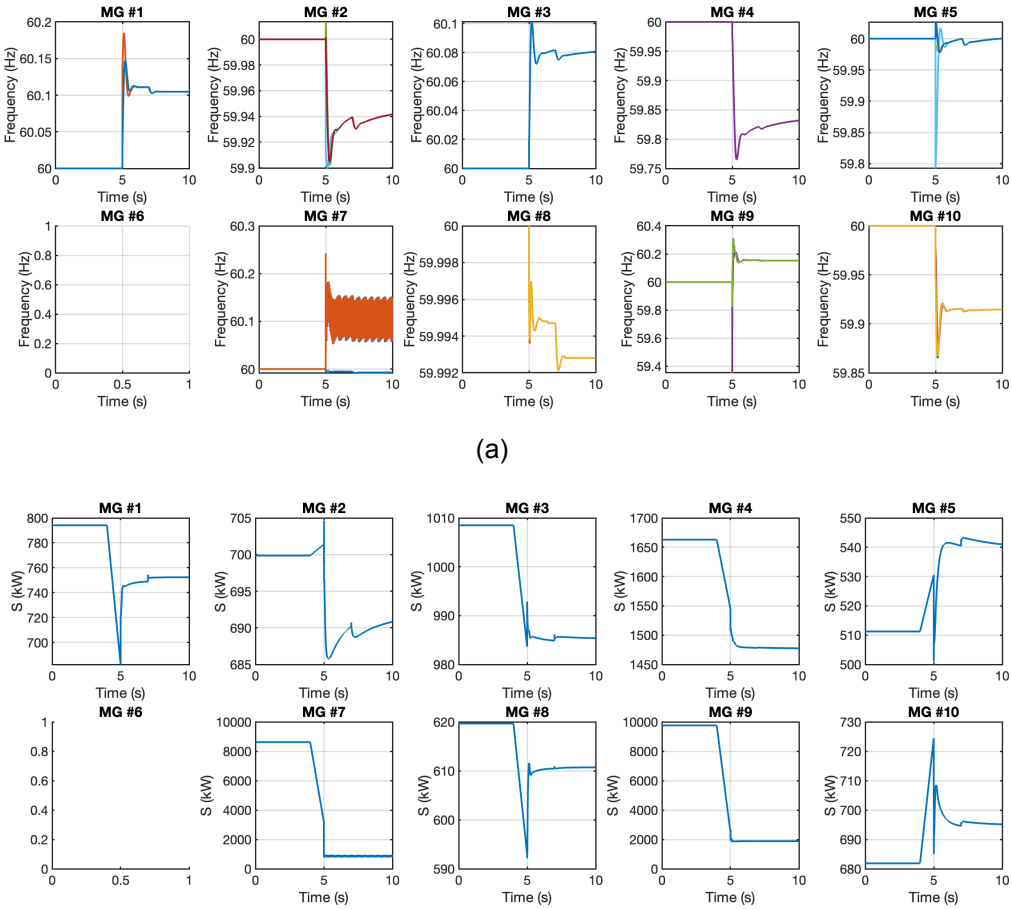
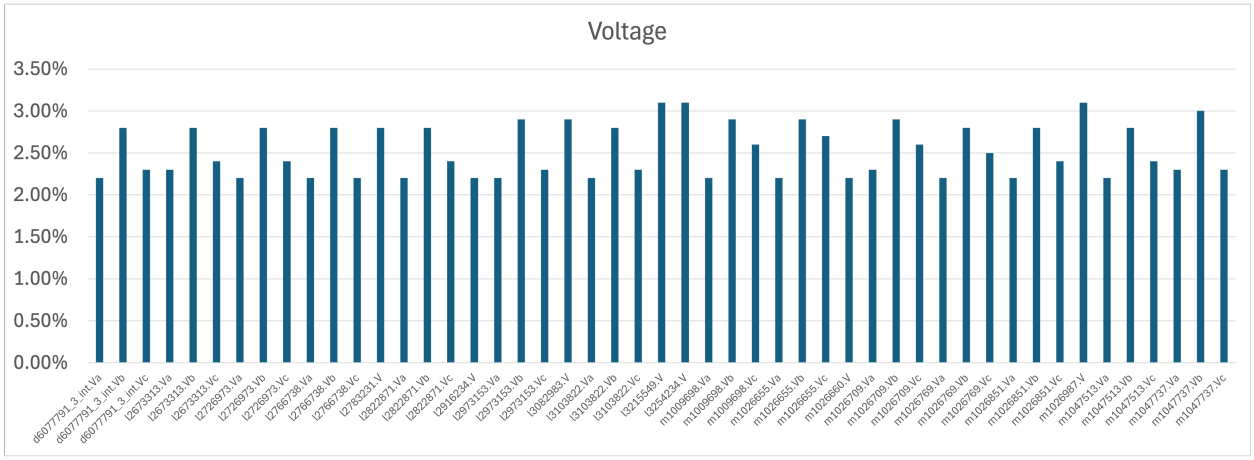
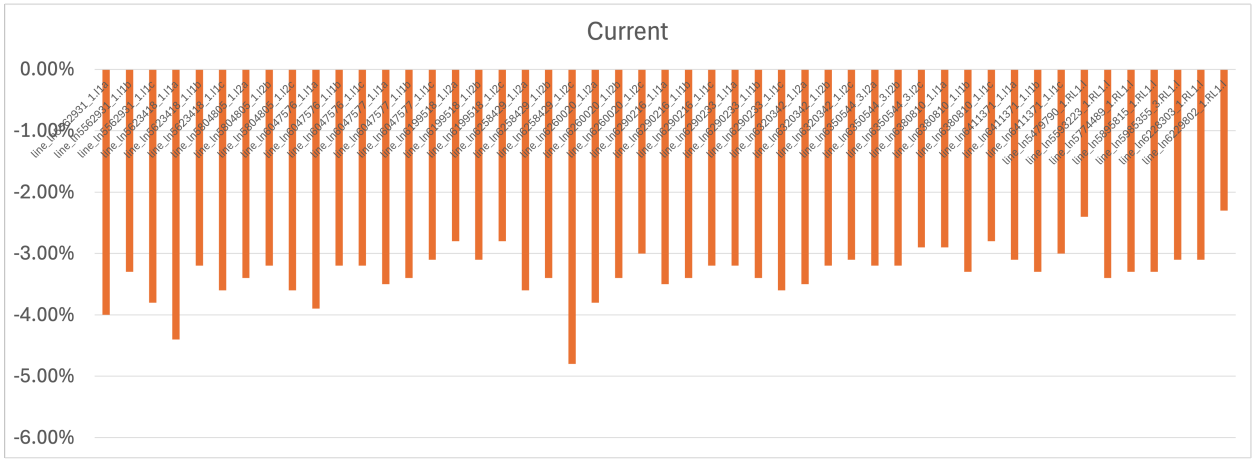


Figure 7: (a) Frequency response and (b) apparent power of inverters in MGs.



(a)



(b)

Figure 8: (a) Voltage and (b) current comparison between GridLAB-D and HYPERSIM.

2.1.5 Modeling Challenges and Solutions

The HYPERSIM modeling team encountered three major challenges when developing the real-time, EMT model of feeder three of the IEEE 9500 node distribution system. The challenges are listed below.

1. Model size and complexity
2. Model decoupling
3. Dynamic load model stability

2.1.5.1 Addressing Model Size and Complexity

The size and complexity of the model made using a graphical modeling software to perform model editing difficult and risky. Real-time, EMT modeling software often uses a graphical user interface where components are manually drawn and connected with wires on a drawing canvas. While the GridLAB-D to HYPERSIM conversion tool was able to populate the entire feeder three passive network, the wires representing connections between circuit elements were automatically drawn overlapping other wires despite there being no intent to connect the disparate wires. The process of drawing new components on the existing canvas created the risk that a wire could be incidentally connected to another overlapping wire, creating an unintentional conducting path that would be very difficult to visually detect. To address this risk, off-page connectors were added to each bus where additional models, such as DER models, would connect. This allowed the team to perform edits on separate canvas pages, where there was no risk of accidentally editing the network. One opportunity for improving the utility of the conversion tool would be to auto-populate off-page connectors at every node in the network. Any new series connections between existing nodes or shunt connections between a node and ground could be safely made on a separate page without having to edit the network diagram canvas page at all.

2.1.5.2 Decoupling the Model

In the EMT modeling domain, the process of decoupling a power system network model involves identifying separate regions of the network that can be simulated independently on separate CPU cores. A group of CPU cores on an OPAL-RT simulator can then essentially co-simulate the various regions in real-time by exchanging the voltages and currents measured at the interfaces of their respective regions over a message bus. The theory behind this arrangement is that an electrical impulse occurring at one end of a long transmission line takes a well defined amount of time to reach the other end, known as propagation delay. The propagation delay is a function of the lumped series inductance and shunt capacitance of the transmission line. The longer the line, the larger the inductance and capacitance and the longer the propagation delay. When the propagation delay is longer than the simulation time step, the time delay of exchange the voltage and current information over the message bus emulated the real-world time delay of the electromagnetic wave traveling down the transmission line. This allows transmission networks to be broken into many smaller regions that can be divided amongst a large number of CPU cores. Conversely, distribution systems lack power lines that are long enough to result in propagation times that are longer than the simulation time step. To address this issue, the team elected to use the following approach.

1. Identify where multiple primary feeder or single phase lateral lines are connected in series, without any models connected in shunt to their adjoining nodes.
2. Replace the multiple series connected lines with a single line type decoupling component having a series resistance and inductance equal to the total lumped series resistance and inductance of the multiple lines.
3. Add sufficient lumped shunt capacitance to the line type decoupling component to result in a propagation delay equal to the simulation time step.

The advantage of this approach is that by combining multiple lines, the added shunt capacitance is reduced, which helps reduce errors in the steady state powerflow results by the artificial insertion of capacitance that does not otherwise exist in the model. The drawbacks are that one or more nodes are eliminated from the model and the node voltages are unavailable during the simulation. This was an acceptable tradeoff as the voltages at the nodes where there were no load or DER models connected were less important in terms of capturing the dynamic behavior of the model. Another drawback is that this approach is heavily model dependent as other feeder models may not have nodes lacking shunt connected components.

An alternative approach in earlier phases of the RD2C initiative was to replace single line segments with the line type decoupling component. This avoided eliminating nodes, but required significantly more artificial shunt capacitance to be added. To compensate, large shunt inductances were added at either terminal of the decoupling components. The reactive power injection and absorption of the artificial capacitance and inductance would effectively cancel out. However, this is a solution oriented towards quasi-static simulations based on powerflow analysis. In EMT simulation, the addition of the shunt inductance creates new modes in the model by adding new state variables where there formerly were none (the current in the shunt inductance branches). These new modes are likely to have lower resonant frequencies than typical for distribution system networks as both the artificial capacitances and compensating inductances are large due to the lumped series inductance of the single feeder segment being small. As a result, the risk of coupling these newly introduced models onto fast controller dynamics, such as inverter based control dynamics, is higher. Furthermore, the switching transient behavior of the network will be significantly altered. Conversely, by replacing multiple series connected line segments, the artificial capacitance is reduced and no new modes are introduced. The resonant frequencies of the existing modes and the power flow results will be altered, but this may only introduce a few percent of error in the steady state and transient response of the network state variables.

2.1.5.3 Dynamic Load Model Stability

For decades, power electronics have continued to become cheaper and thus more ubiquitous in electricity end-use. They provide greater flexibility and control over the processes that they supply. For this reason, distribution system loads increasingly exhibit dynamic behaviors inconsistent with constant impedance load model behavior. As a result, benchmark test feeder models contain constant power spot loads in order to more accurately represent real-world systems. Constant power load models can be implemented in different ways, but one of the most common is by measurement of the terminal voltage and adjustment of the shunt resistance and inductance (for a lagging power factor load) or capacitance (for a leading power factor load) to yield the same terminal active power and reactive power consumption. This behavior is potentially unstable as the dynamics of these loads creates a negative impedance whereby a decrease in voltage is compensated by a decrease in the equivalent shunt impedance, causing greater current flow

from the source, which then leads to greater voltage drop across the source impedance and a decrease in the terminal voltage, thus creating the positive feedback loop. The likelihood of unstable model behavior increases as the equivalent source impedance at the terminals of the load models increases, however predicting whether this dynamic actually causes model instability is difficult, particularly in a large network with hundreds of dynamic load models.

The HYPERSIM modeling team encountered this phenomenon when attempting to simulate islanded operation of microgrids in the IEEE 9500 Node Feeder Three model. The GridLAB-D to HYPERSIM conversion tool discussed in Section 2.1.2 auto-populates the network with dynamic, single phase PQ load models, which exhibit the problematic behavior described above. During grid connected operation, the grid voltage source is sufficiently small such that the voltages at the various nodes in the network are relatively insensitive to the dynamic load model changes in impedance. However, when the microgrids islanded, the equivalent source impedance is determined by the AC filter impedance of the local DERs, which results in a larger source impedance. The result was that the dynamic PQ load model impedances never reached equilibrium and one phase voltage would typically collapse to almost zero while the other two phase voltages would swell to almost double their nominal value. Fig. 9 illustrates this behavior.

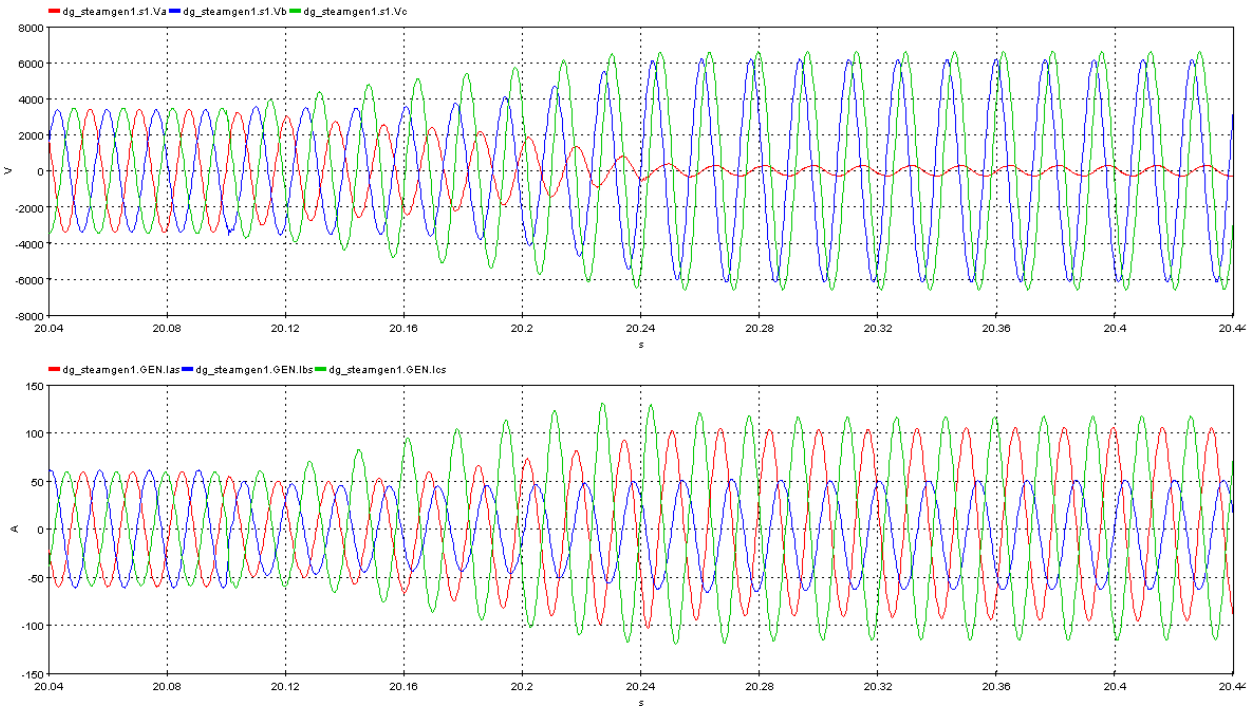


Figure 9: Results of islanding a small microgrid composed of a synchronous generator and single phase, dynamic PQ loads.

The figure shows that the line-neutral phase A voltage collapses to near zero (red trace), while the phase B and C voltages increase by almost 100%. Resolving this issue in the IEEE 9500 Node Feeder Three test model would have required replacing some fraction of the 500 dynamic load models with constant impedance load models, which could only be determined by lengthy experimentation. The project timeline could not accommodate further investigation and resolution of this issue and therefore the decision was made to use the existing IEEE 123 Node Feeder model for testing the controller hierarchy.

2.1.5.4 Positive Outcomes from Addressing Modeling Challenges

In-spite of not resolving this challenge within the timeline of the project, the project team engaged with OPAL-RT R&D staff to determine a path forward and explore future opportunities for collaboration. The PNNL team performed a series of simple tests involving constant impedance loads, the PNNL internally developed single phase dynamic loads, and a built-in HYPERSIM single phase dynamic load. The PNNL team provided the test results to the OPAL-RT R&D team. Using PNNL's simple test model, the OPAL-RT R&D team discovered a bug in the initialization of the HYPERSIM built-in model. As a result, OPAL-RT R&D will be rolling a bug fix into the next major release of HYPERSIM, which will benefit PNNL researchers as well as the HYPERSIM user community at large.

2.2 IEEE 123 Node Test Feeder Model

2.2.1 Model Description

The IEEE 123 node test feeder was developed in HYPERSIM to study the behavior of microgrids with increased penetration of DERs. The topology of the feeder, loads and parameters of various components are based on the original feeder specification, various DERs were added in different locations to increase the DER penetration level. Three microgrid structures were created in this model based on the location of the switches to create the islanded and grid-connected scenarios for different cyber-physical use-cases of interest. Fig. 10 shows the IEEE 123 node test feeder which three microgrids segregated through grey boxes and finally connected to the distribution feeder. Each microgrid has three DERs to support islanded operation. An excitation system and a speed governor are modeled for every synchronous machine. In particular, a typical AC1A exciter and DEGOV1 governor are modeled, and can be modified if necessary. The generator is configured for active power/frequency droop, and the exciters and governors have dynamically adjustable parameters.

2.2.2 Model Development

Moreover, additional DERs to the specified microgrids are added to ensure that the microgrids can operate independently after islanding, represent high penetration of DERs and create a good use case for the mitigation strategies. Each microgrid has three DERs to support islanded operation. There are multiple inverters (grid following and grid forming) and one diesel generator in each of the microgrids as shown in Fig. 11. Blue dots show the position of the diesel generators, the red dots show the positions of grid-forming inverters, and the gray dots show the grid-following inverters. Red (closed) and green(open) boxes show the positions of breakers/sectionalizing switches in the model.

Table 5 consists of the grid-following and grid-forming inverters and DGs in three microgrids and their capacity in the IEEE 123 node full architecture model used for this project. Microgrid 3 is the most populated microgrid with seven grid-forming inverters and one grid-following inverter in this model. The mix of DERs combined with the inertia constant used for DG was found to be ideal for demonstrating the mitigation control strategies.

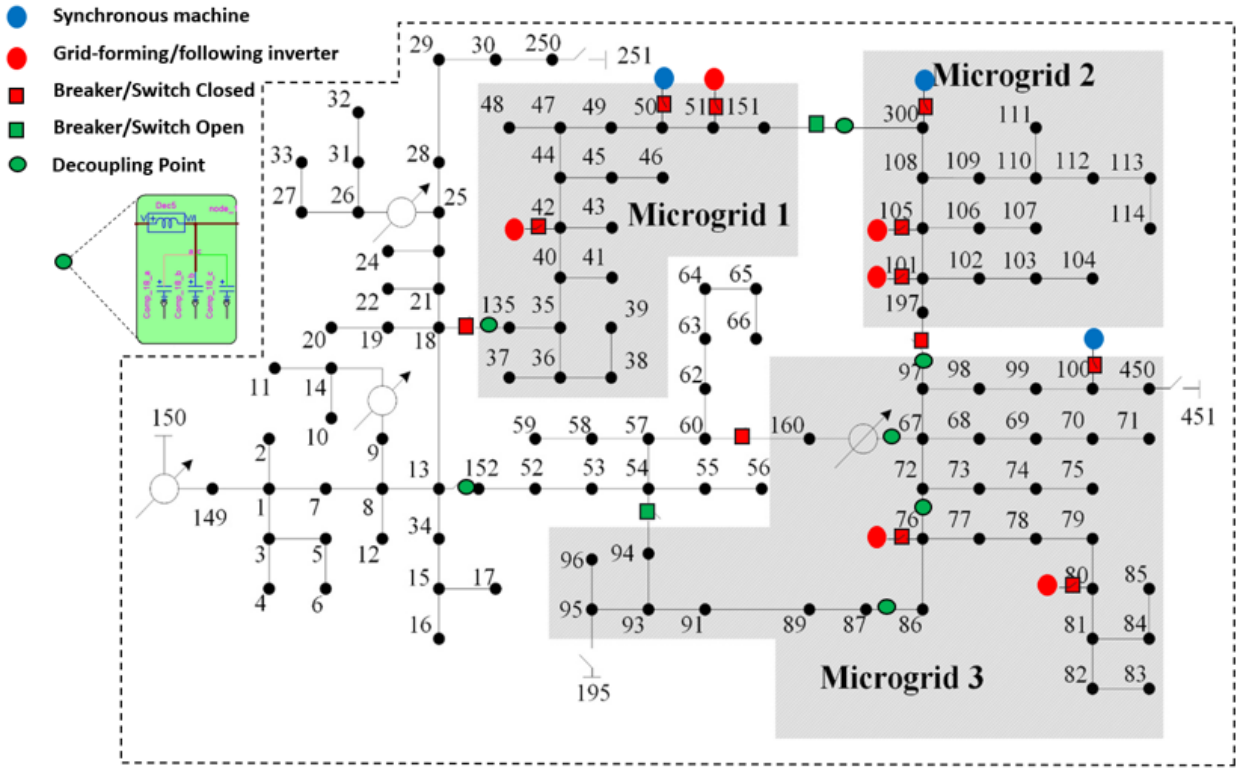


Figure 10: One-line diagram of modified IEEE 123 node test feeder with decoupling points.

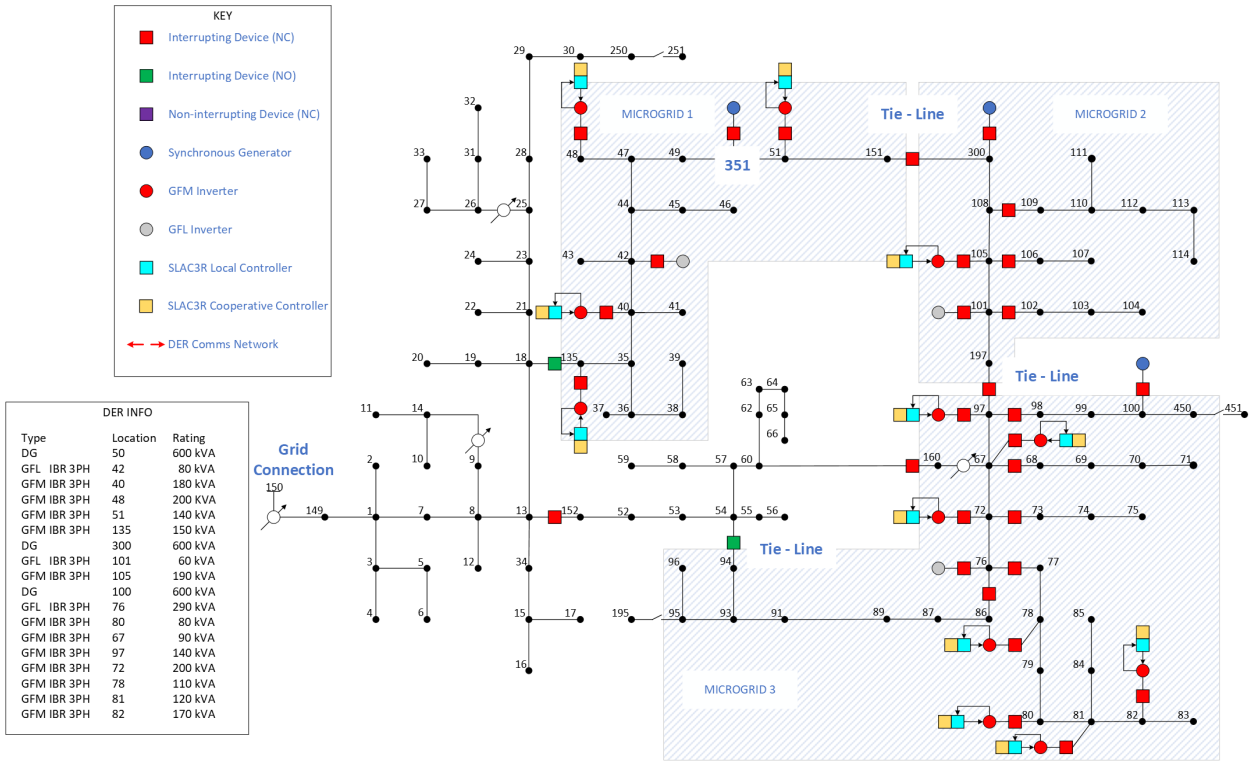


Figure 11: One-line diagram of IEEE 123 Node Full Architecture with additional DERs.

2.3 Benchmarked DER Models

2.3.1 DER Model Development

To accomplish the goals of the project, high-fidelity, real-time models of electric distribution systems with a large number of DERs would be required. The power system modeling and simulation platform used in this project is OPAL-RT HYPERSIM and an initial set of DER models had been developed in previous phases of the RD2C initiative. However, the size of the IEEE 9500 node test feeder created new requirements for the DER models that resulted in a significant model development effort. The new model requirements and the model development tasks performed to satisfy the requirements are summarized in Table 6. The team acknowledges that using average value instead of switching function or switching device models of inverters represents a loss of fidelity. However, this decision was deemed necessary to not exceed the process capacity of the OPAL-RT simulator given that over 50 inverter models would be populated. Additionally, it was seen as a prudent first step given that a model the size of the 3000 Node Feeder model had not been previously attempted in HYPERSIM at PNNL. Replacing the average value models with switching function models is left for future work.

Developing robustly stable DERs models was critical due to the size of the proposed test distribution system. In the event that the models exhibited unstable behavior, debugging the root cause would be extremely time consuming due to the very large number of potential physical and control system interactions.

In total, the following DER models were either updated from previous versions or built from scratch to meet the requirements.

Table 5: List of DERs in IEEE 123 Node Full Architecture

DER Name	Microgrid	Parent node	Phases	Rating (VA)	Type	Pset (W)	Qset (VAr)	Vset (pu)
DG50	1	meter_50	ABCN	600,000	DG	300,000	0	1.0
INV42	1	load_42	ABCN	80000	GFL IBR	40000	0	1.0
INV51	1	load_51	ABCN	140000	GFM IBR	70000	0	1.0
INV#1	1	node_135	ABCN	150000	GFM IBR	75000	0	1.0
INV#2	1	node_40	ABCN	180000	GFM IBR	90000	0	1.0
INV#3	1	load_48	ABCN	200000	GFM IBR	100000	0	1.0
DG300	2	node_300	ABCN	600,000	DG	300,000	0	1.0
INV101	2	load_101	ABCN	60000	GFL IBR	30000	0	1.0
INV105	2	node_105	ABCN	190000	GFM IBR	95000	0	1.0
DG100	3	load_100	ABCN	600,000	DG	300,000	0	1.0
INV76	3	load_76	ABCN	290000	GFL IBR	145000	0	1.0
INV80	3	load_80	ABCN	80000	GFM IBR	40000	0	1.0
INV#4	3	node_67	ABCN	90000	GFM IBR	45000	0	1.0
INV#5	3	node_97	ABCN	140000	GFM IBR	70000	0	1.0
INV#6	3	load_72	ABCN	200000	GFM IBR	100000	0	1.0
INV#7	3	node_78	ABCN	110000	GFM IBR	55000	0	1.0
INV#8	3	node_81	ABCN	120000	GFM IBR	60000	0	1.0
INV#9	3	load_82	ABCN	170000	GFM IBR	85000	0	1.0

- Three phase synchronous diesel generator.
- Three phase GFM inverter.
- Three phase GFL inverter.
- Single phase GFM inverter.
- Single phase GFL inverter.

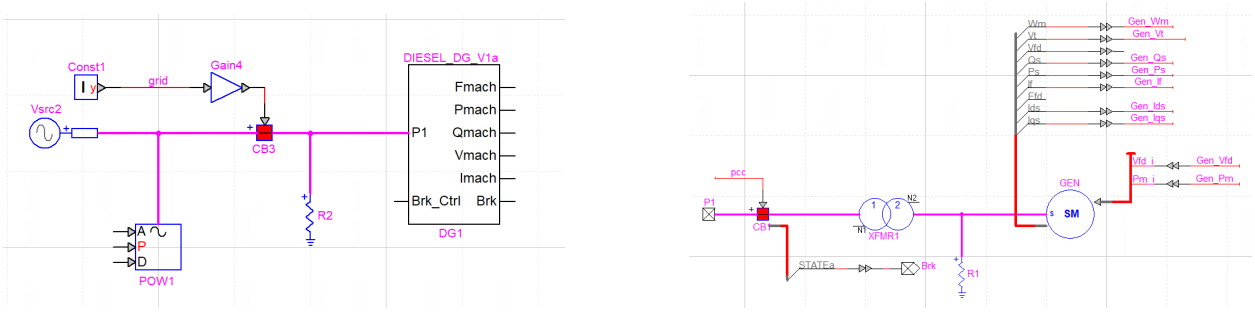
Each DER model was placed inside a subcircuit, which contained the physical system and controls system representations of the equipment. Organizing the models into subcircuits allowed for code reuse, which was crucial for modeling a large distribution system with many instances of each DER type. The subcircuits were connected to test circuits composed of a grid voltage source behind an impedance. Fig. 12, illustrates the diesel generator test circuit and the physical system model inside. The test circuits and subcircuit contents for each DER model are provided in the Appendix.

The grid voltage source impedance, which determines the Short Circuit Ratio (SCR) of the DER model, was selected with care to avoid masking model instability stemming from poor controller design and tuning. Each DER model is tested against multiple grid impedance values. In particular, the GFL DERs are subjected to SCR as low as 2.5 to ensure that they will exhibit acceptable dynamic performance is islanded scenarios. Conversely, the GFM DERs are subjected to SCR as high as 10.0 to ensure that acceptable performance will occur during grid connected scenarios, where the PCC voltage may be quite stiff.

A set of default physical and control system parameters was created for each DER model. The parameters are expressed in per unit quantities and fit within typical ranges observed in

Table 6: HYPERSIM DER Model Requirements and Development

Model Requirement	Model Development Task
Reduce Inverter Model Complexity	Replace Inverter Switching Function Representations with Average Value Model Representations [13] Neglect DC side dynamics
Reduce Generator Model Complexity	Replace built-in HYPERSIM exciter and governor models with simplified user created models
Capture behavior of DER protection schemes	Add simplified voltage, frequency, power and current protection schemes to DER models
Prioritize stable model behavior in both grid connected and islanded modes of operation	Rigorously test DER model dynamics before integration with network model
Model single phase inverters	Develop single phase GFL and GFM inverter models using basic control techniques available in the literature



(a) (b)

Figure 12: Diesel generator test circuit (left) and physical system model (right).

distribution system and DER modeling literature [14–17]. In order to streamline modeling DER with different voltage and power ratings, the same per unit default parameters are used, and only the system base quantities are modified. This guarantees reasonable parameter values in terms of real quantities that are input to the HYPERSIM model block parameters. The DER model default parameters are provided in the Appendix.

The test circuits are used to rigorously test each DER model to ensure that it exhibits stable behavior under dynamic conditions and achieves its core functional objectives consistent with how a DER vendor system would be expected to perform in the real world. We note that typical generator protection schemes and IEEE 1547 ride through controls were not part of the DER benchmarking tests and for that reason the voltage and frequency transients displayed in the test results are extreme. However, this was deliberate to show that the primary control systems were stable well beyond the practical operating ranges of real equipment, which are enforced by conventional protection systems. The DER core functionality is summarized in Table 7.

Table 7: HYPERSIM DER Model Core Functions

Model Requirement	Model Development Task
Three phase synchronous diesel generator	Provide voltage and frequency stability during islanded operation.
Three phase GFM inverter	Respond to active and reactive power dispatch during grid connected operation. Provide voltage and frequency stability during islanded operation.
Three phase GFL inverter	Respond to active and reactive power dispatch during both grid connected and islanded operation.
Single phase GFM inverter	Respond to active and reactive power dispatch during grid connected operation. Provide voltage and frequency stability on single phase radials during islanded operation.
Singe phase GFL inverter	Respond to active and reactive power dispatch during both grid connected and islanded operation.

2.3.2 Diesel Generator Model Validation

Conventionally, distributed diesel synchronous generators are not operated in parallel with the grid. Instead, they are started following a grid outage and provide a stable voltage source with firm generation capacity for islanded systems such as facility microgrids. As a result, the HYPERSIM diesel generator model is simulated in islanded mode operation and subjected to a rapid increase load as might occur if a large load is instantaneously brought online or an adjacent generation resource such as a PV system is lost. Additionally, the simplified diesel generator model response is compared to the response of the prior model version, which utilized higher order, built-in governor and exciter models. Fig. 13 presents the dynamic simulation results.

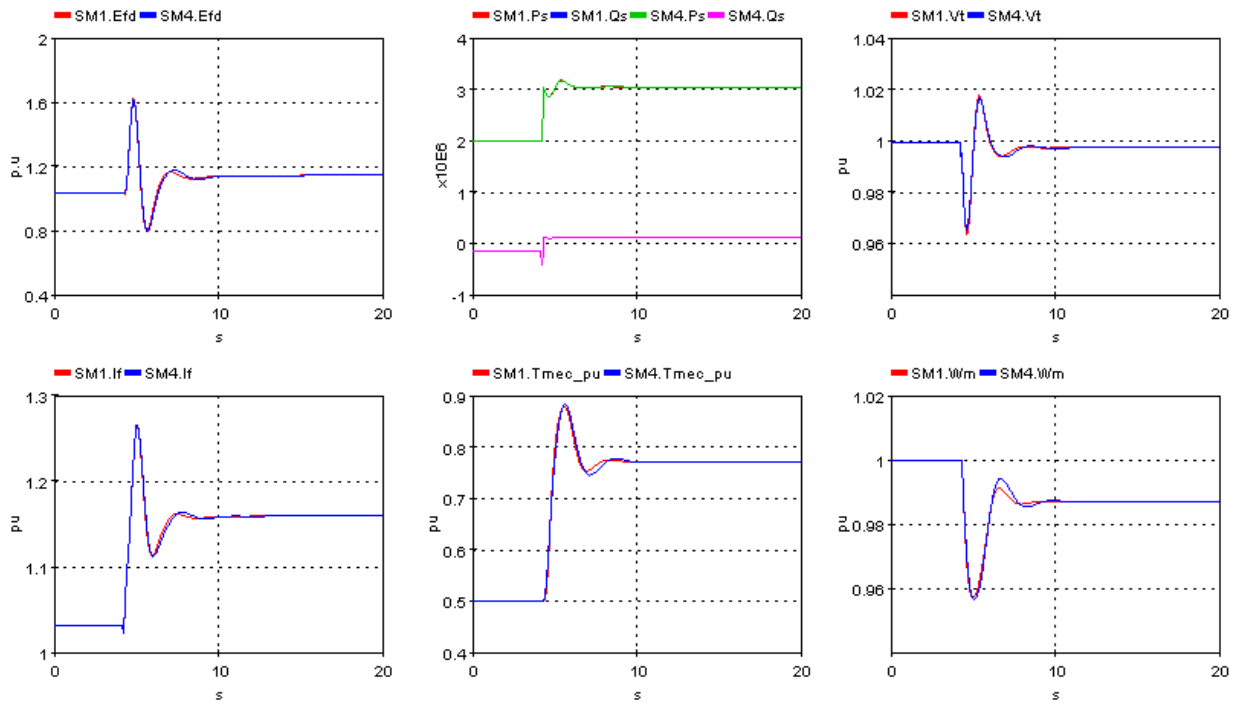


Figure 13: Islanded diesel generator load acceptance test. The simplified model response is indicated by the traces labeled ”SM4.[variable name]”. The higher order model response is indicated by the traces labeled ”SM1.[variable name]”. The model variables are: generator field voltage (top-left); generator real and reactive power (top-middle); generator terminal voltage (top-right); generator field current (bottom-left); generator prime mover torque (bottom-middle); and generator rotor speed (bottom-right).

The simplified generator model response closely approximates the higher order model response. Additionally, both generator models maintain terminal voltage and rotor speed stability despite an instantaneous 25% increase in load.

2.3.3 Three Phase GFM Inverter Validation

The three phase GFM inverter is representative of a bulk energy storage system that can be paralleled to the grid. The inverter is controlled using a typical GFM droop control as described in [17]. During grid connected operation, the inverter should automatically provide fast frequency and voltage support in the event of large disturbances in the transmission system due to the droop control dynamics. In the first pair of tests, the inverter operates in grid connected mode and is subjected to a 25% grid voltage sag and a 1.0 Hz grid frequency sag. The dynamic simulation results are documented in Fig. 14 and Fig. 15.

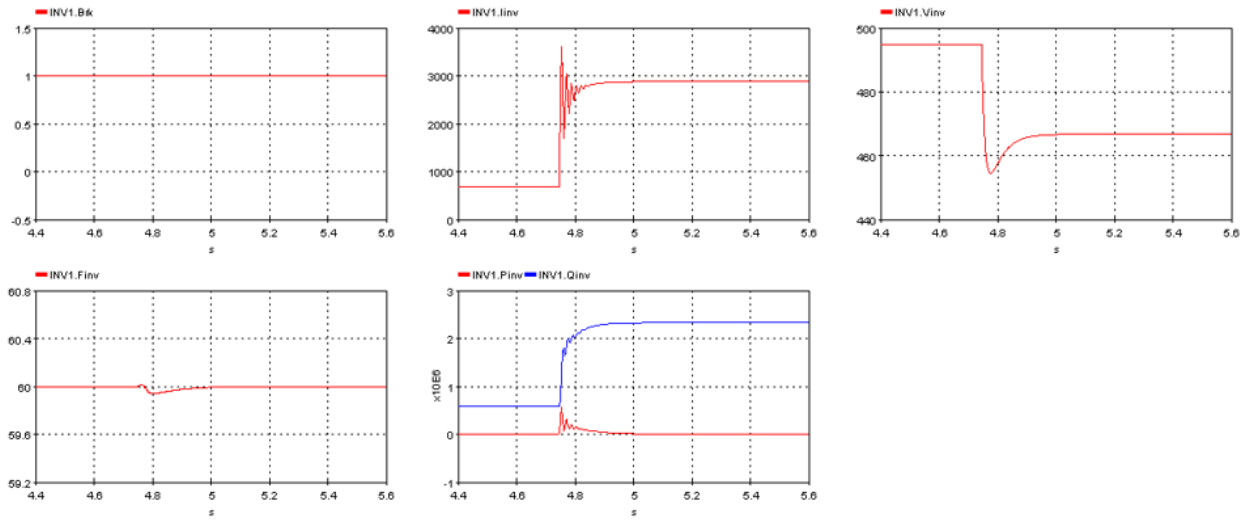


Figure 14: Three phase GFM inverter grid voltage sag test. The inverter model variables are: PCC breaker status (top-left); RMS average phase current (top-middle); RMS average phase voltage (top-right); measured frequency (bottom-left); and real and reactive power injection (bottom-middle).

The inverter model maintained grid synchronization in both dynamic tests. Furthermore, the inverter provides voltage support by increasing its reactive power injection in response to the grid voltage sag as expected. Similarly, the inverter also provides automatic frequency response by increasing its active power injection following the frequency dip.

The inverter is subjected to an instantaneous load increase while in islanded mode of operation to ensure that the model maintains stable terminal voltage magnitude and frequency. Fig. 16 gives the dynamics response of the inverter model.

The inverter successfully maintains voltage and frequency stability and exhibits the expected steady state frequency and voltage steady state error according to the droop control laws.

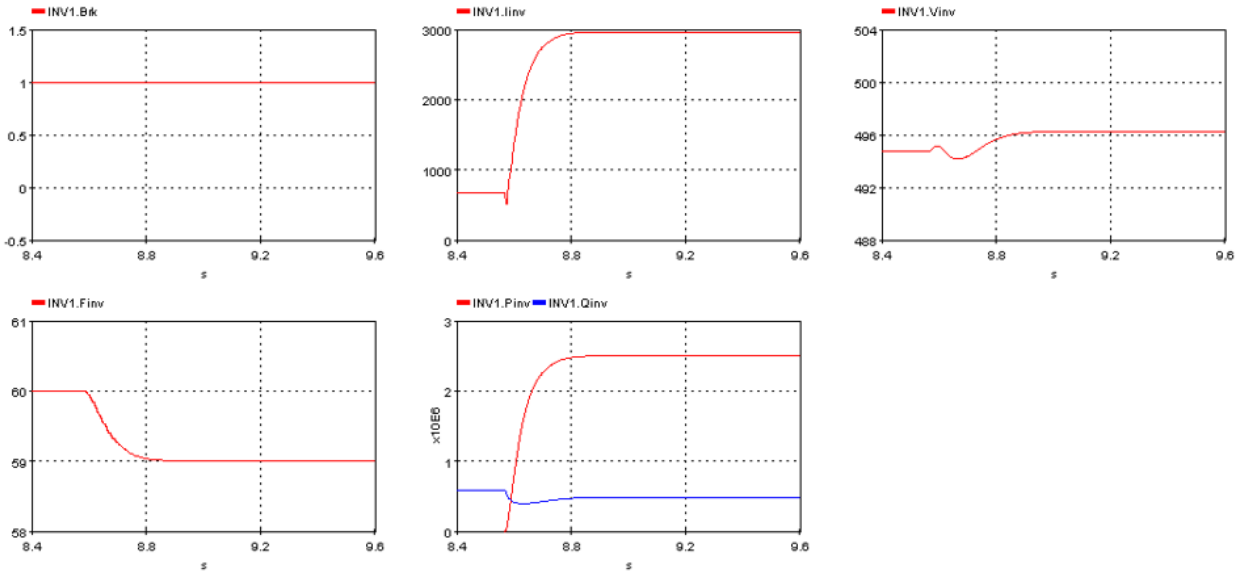


Figure 15: Three phase GFM inverter grid frequency sag test. The inverter model variables are: PCC breaker status (top-left); RMS average phase current (top-middle); RMS average phase voltage (top-right); measured frequency (bottom-left); and real and reactive power injection (bottom-middle).

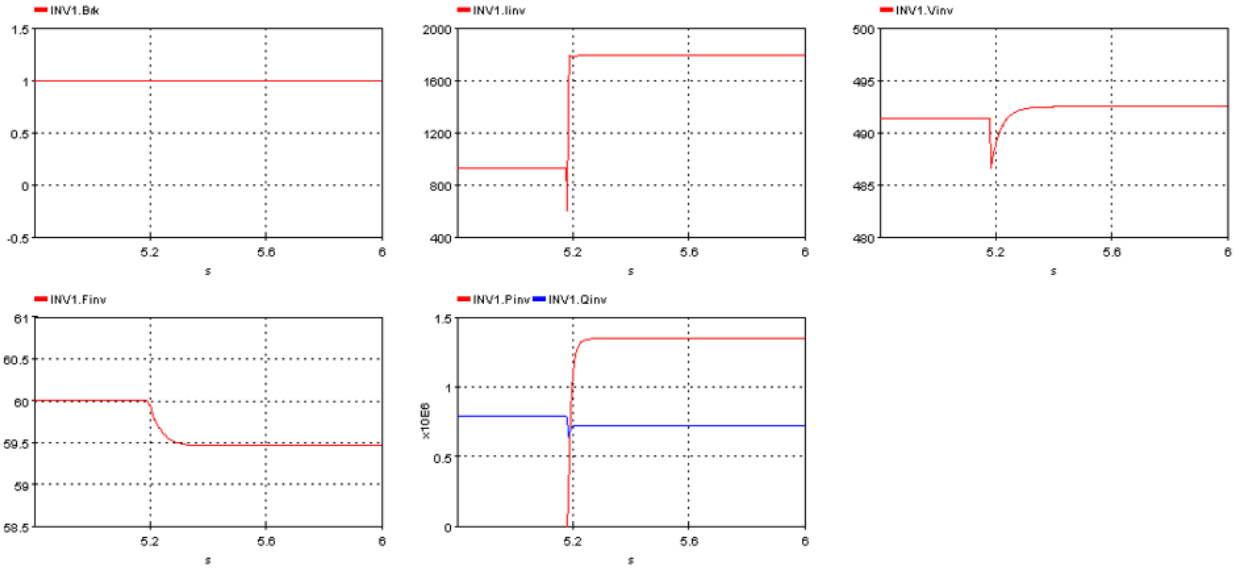


Figure 16: Three phase GFM inverter islanded load acceptance test. The inverter model variables are: PCC breaker status (top-left); RMS average phase current (top-middle); RMS average phase voltage (top-right); measured frequency (bottom-left); and real and reactive power injection (bottom-middle).

2.3.4 Three Phase GFL Inverter Validation

The three phase GFL inverter is representative of a utility scale PV system that generates revenue during grid connected operation and provides a local energy source when islanded. The GFL inverter is controlled using the typical rotating reference frame control, which can be found in [16]. The inverter is expected to regulate its active and reactive power injection according to external controller setpoints with minimal transient error and zero steady state error despite grid voltage and frequency disturbances. Several tests are performed to test the reference tracking performance and disturbance rejection performance of the active and reactive power controls. The results are presented in Fig. 17 through Fig. 20.

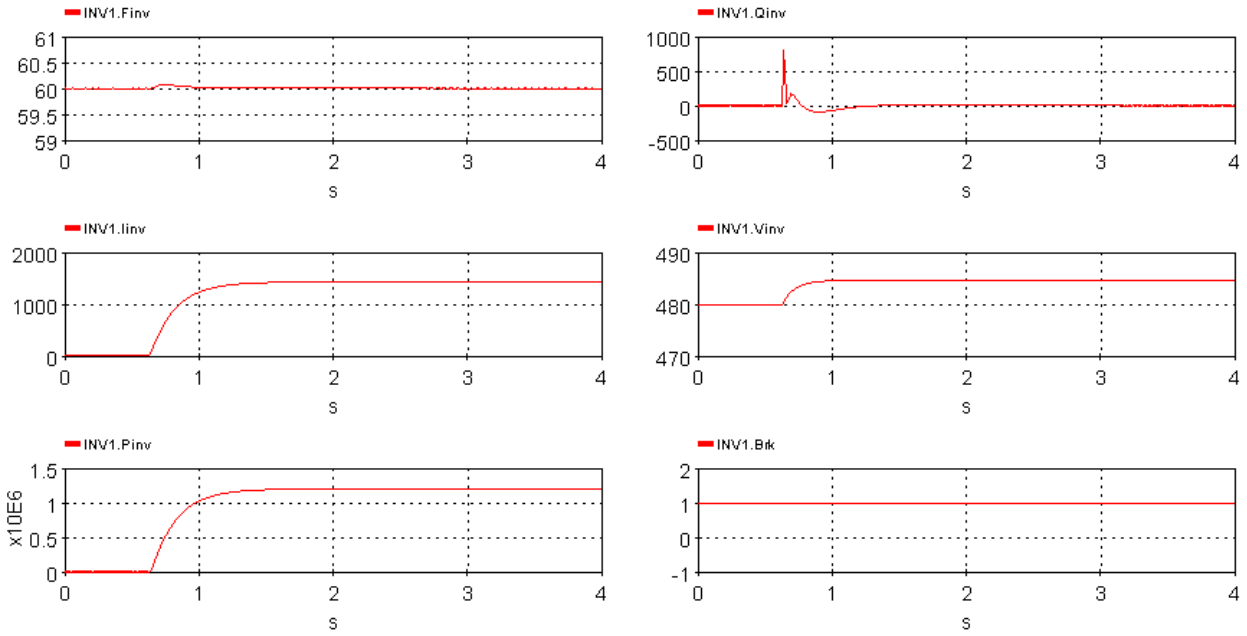


Figure 17: Three phase GFL inverter grid connected active power reference step response test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

From Fig. 17 through Fig. 20, it can be observed that the inverter responds to the changes in power dispatch within 1.0 s and rejects the frequency and voltage disturbances with reasonable performance - transient changes in active and reactive power injections last about 0.5 s and do not exceed 10% of the pre disturbance value. Note that control gain tuning was not performed to improve the transient performance of this inverter model or other DER models as finding an optimal set of gains that maximizes performance while ensuring stability for a range of SCRs would be time consuming. Instead, the gains are left slightly detuned to prioritize a large stability margin due to the uncertainty regarding the actual, effective grid impedance at the various PCCs of the numerous instances of this model.

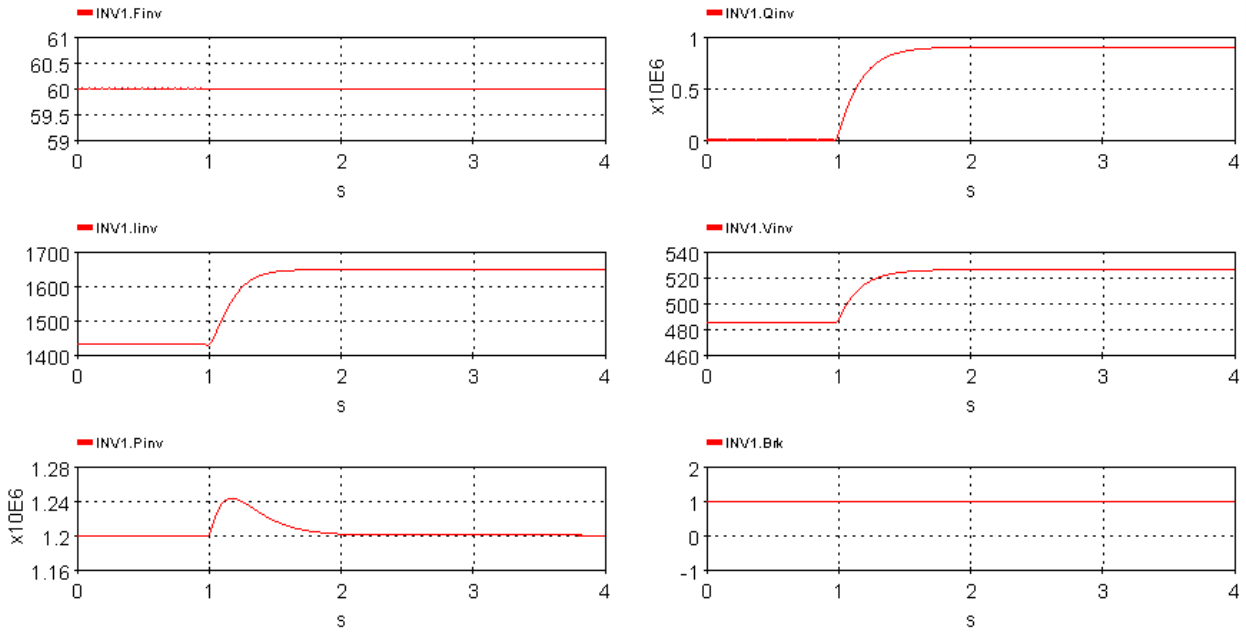


Figure 18: Three phase GFL inverter grid connected reactive power step response test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

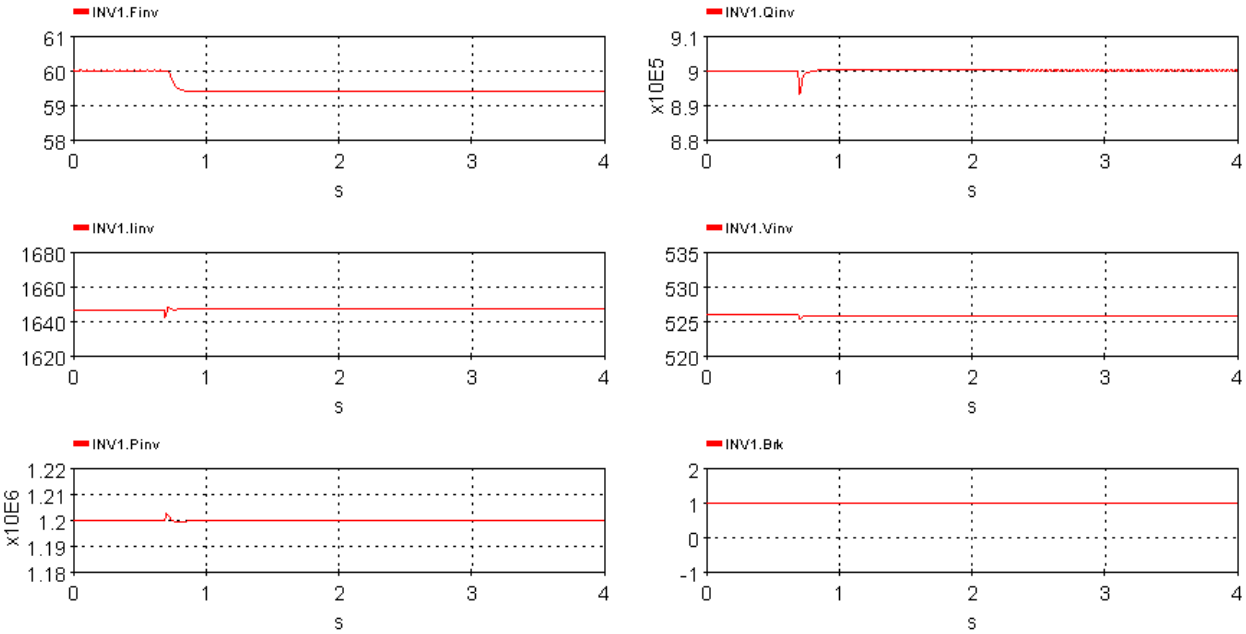


Figure 19: Three phase GFL inverter grid frequency sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

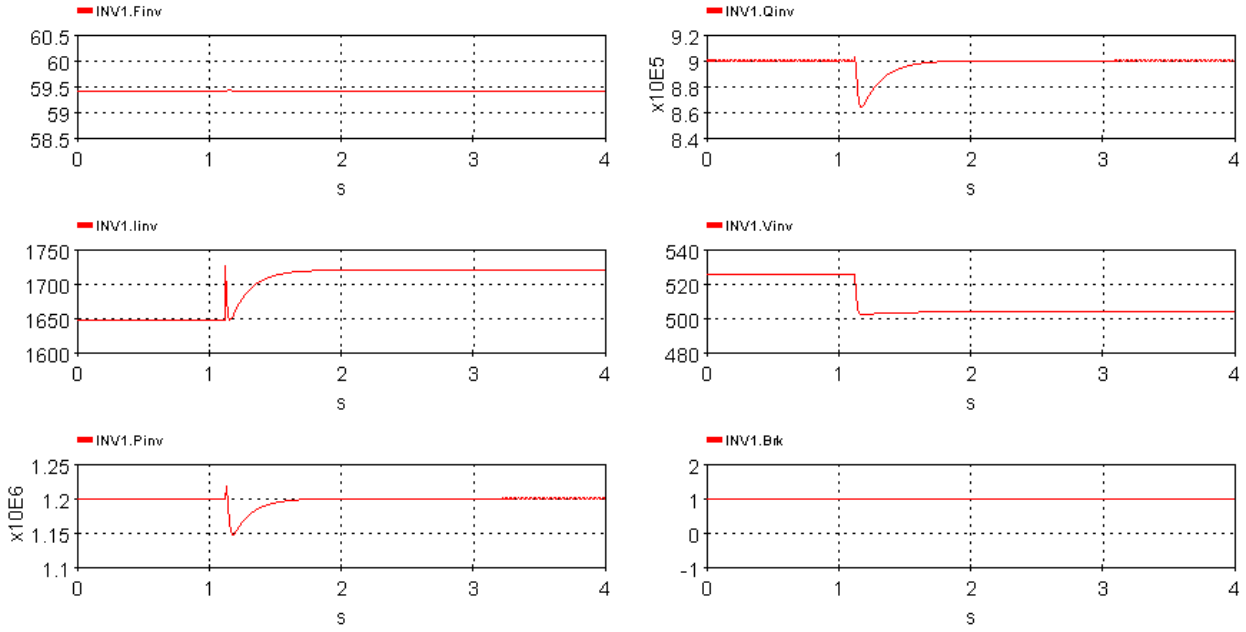


Figure 20: Three phase GFL inverter grid voltage sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

2.3.5 Single Phase GFM Inverter Validation

The single phase GFM inverter is representative of a small-scale energy storage system that could be installed behind-the-meter and connected to single phase distribution. This inverter model is controlled using typical GFM droop control. Similar to the three phase GFM inverter, this inverter is expected to provide automatic fast frequency and voltage support while connected to the grid. This is demonstrated in Fig. 21 and Fig. 22.

The figures demonstrate that the inverter provides a stable, well-damped response to the frequency and voltage disturbances. This implies that the inverter is able to rapidly resynchronize to changes in grid frequency and perform transient reactive power sharing with the grid voltage source. Additionally, the changes in active and reactive power injection are consistent with the expectation based upon the droop control laws.

The single GFM inverter is also expected to regulate frequency and single phase voltage when operating in islanded mode. To test this capability, the inverter is disconnected from the single phase grid voltage source and subjected to an instantaneous increase in load. Fig. 23 provides the results of the load step response test. The figure illustrates that the inverter maintains stable terminal voltage and frequency.

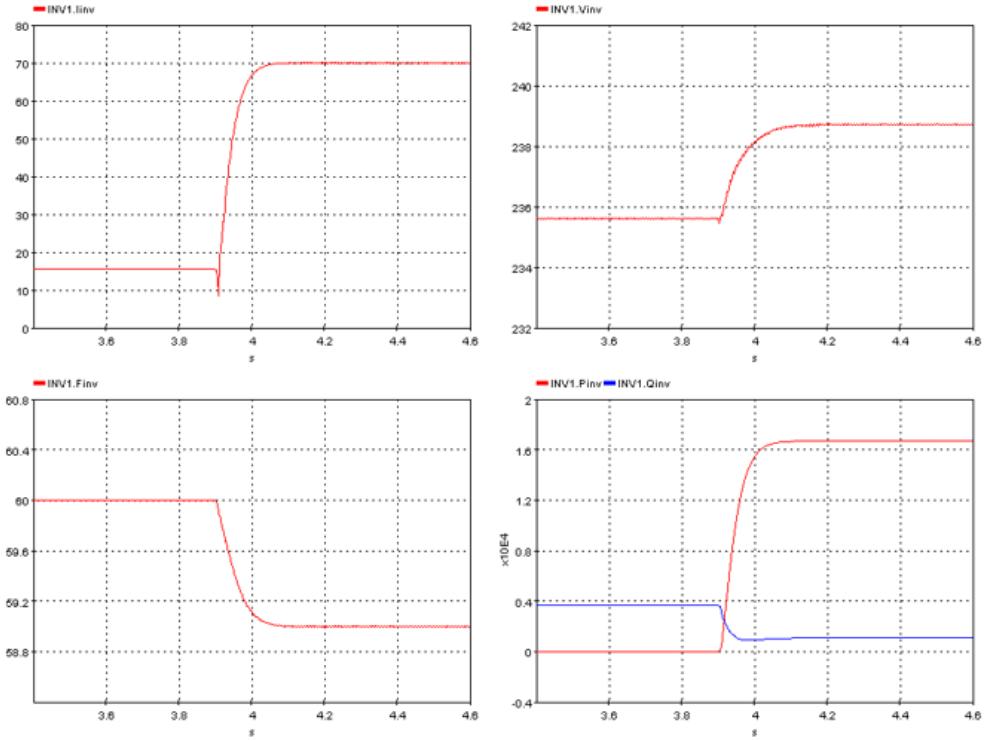


Figure 21: Single phase GFM inverter grid frequency sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

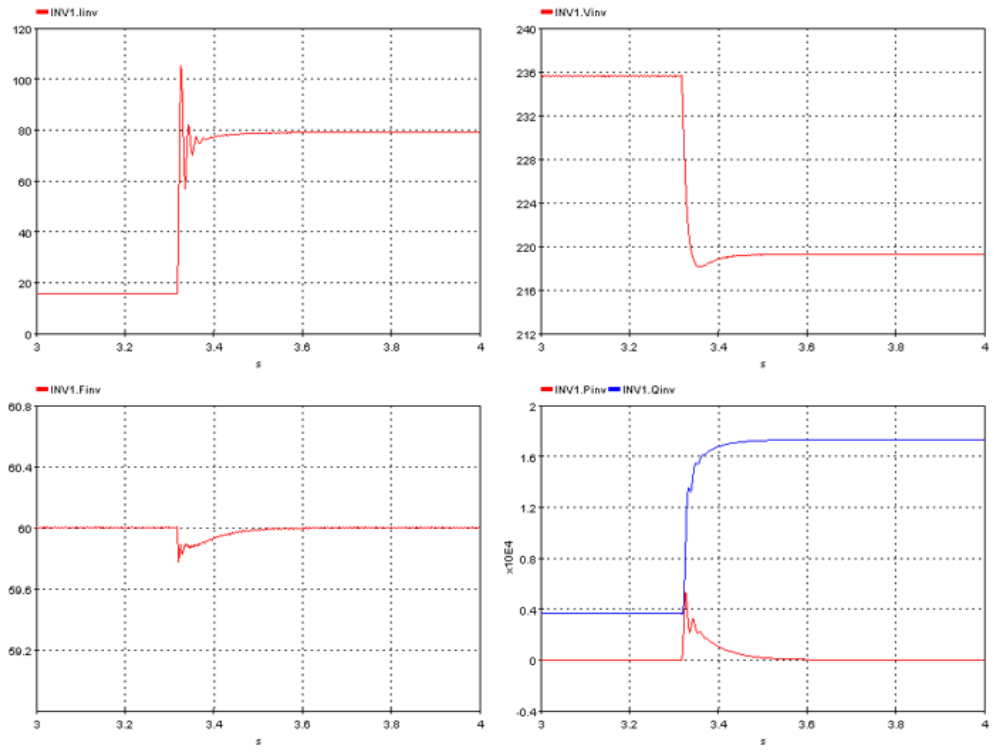


Figure 22: Single phase GFM inverter grid voltage sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

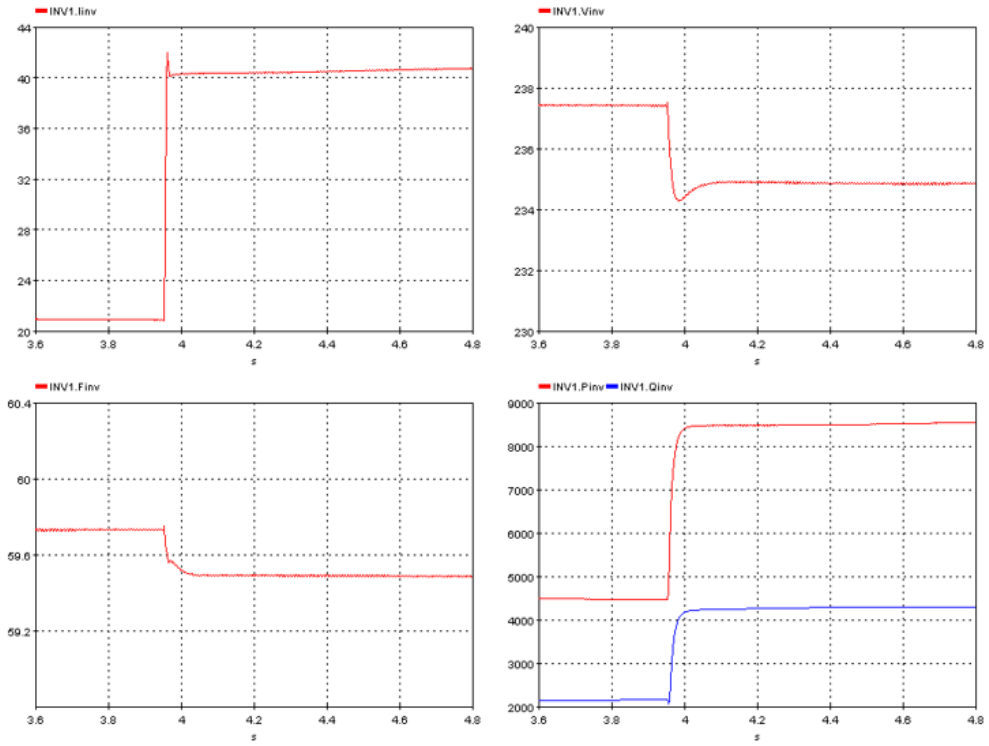


Figure 23: Single phase GFM inverter Islanded load step response. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

2.3.6 Single Phase GFL Inverter Validation

The single phase GFL inverter is representative of small-scale PV systems that are commonly installed behind-the-meter on rooftops. The inverter is expected to demonstrate adequate power reference tracking performance and disturbance rejection. A similar set of tests that were used to validate the three phase GFL inverter model are conducted for the single phase version. The test results are presented in Fig. 24 through Fig. 27.

From Fig. 24 and Fig. 25, it is observed that the inverter model exhibits satisfactory responses to active and reactive power dispatch signals respectively. Additionally, the inverter demonstrates adequate disturbance rejection in Fig. 26 and Fig. 27. This is noteworthy, because implementing rotating reference frame control for a variable frequency system is not a trivial task as it requires generating the orthogonal components of single phase quantities without the use of the abc-dq transform typically used in three phase systems or introducing a fixed time shift to achieve a 90 degree delayed version of the signal. The results in Fig. 26 indicate that the inverter successfully generates the orthogonal components despite the change in frequency and uses them to calculate active and reactive power measurements, which are then used in the closed-loop power control and current control schemes.

In conclusion, each of the five DER model types was robustly tested to determine satisfactory control performance. The default model parameters used in these tests are provided in the Appendix along with the HYPERSIM model block diagrams.

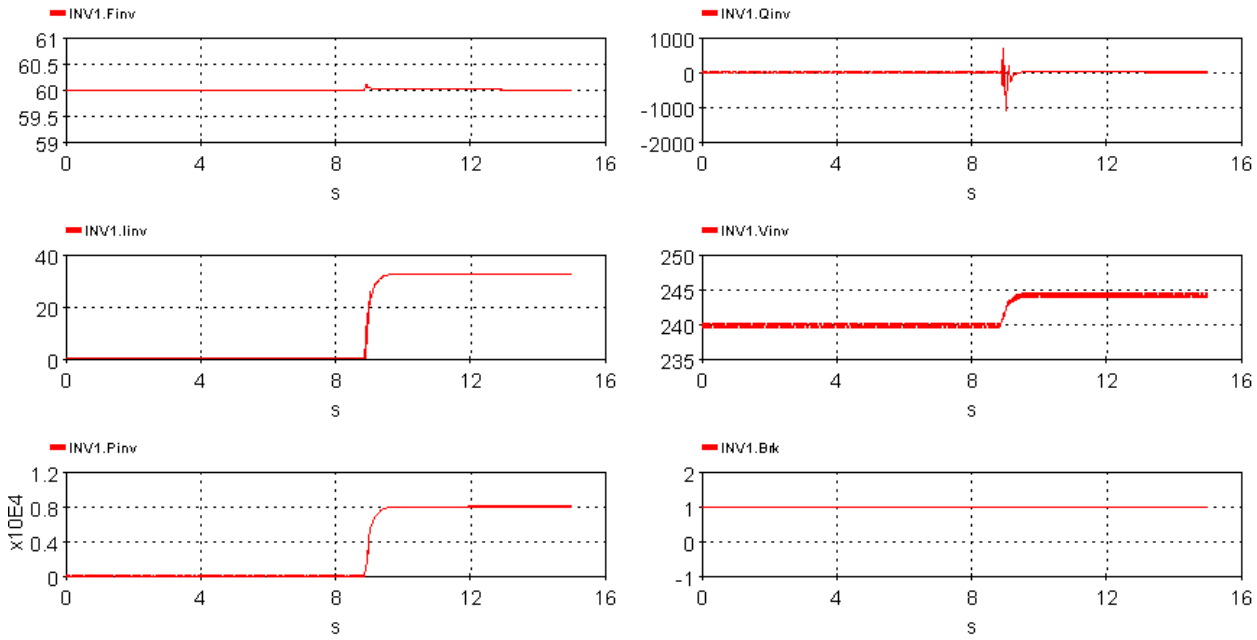


Figure 24: Single phase GFL inverter grid connected active power reference step response test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

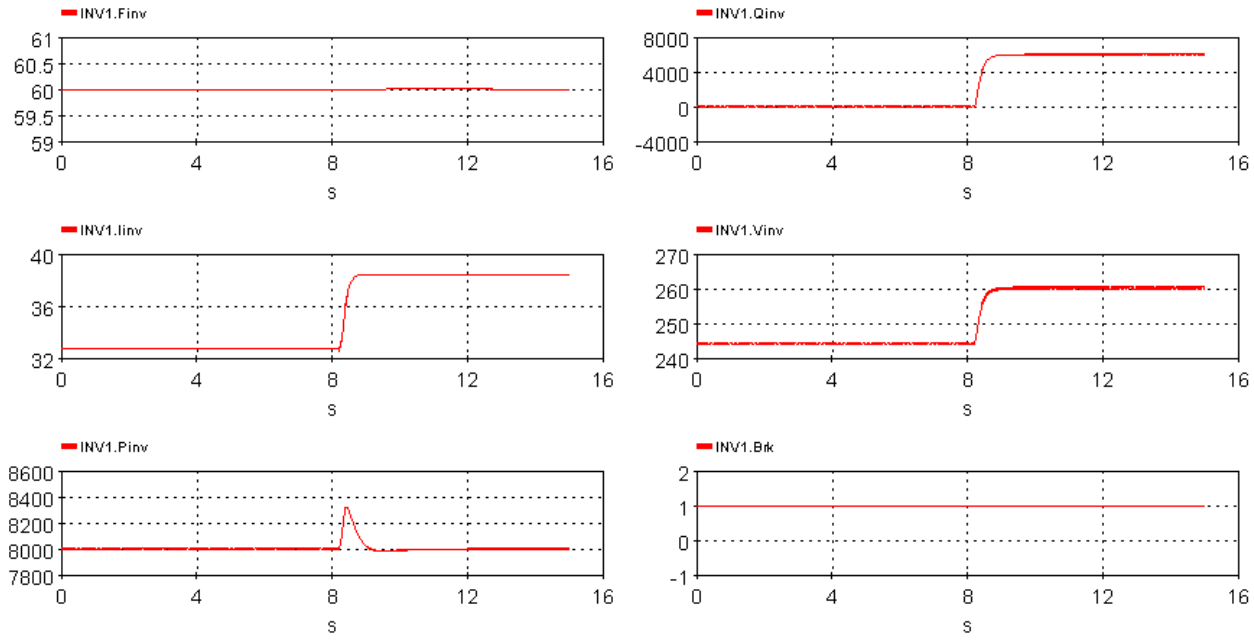


Figure 25: Single phase GFL inverter grid connected reactive power step response test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

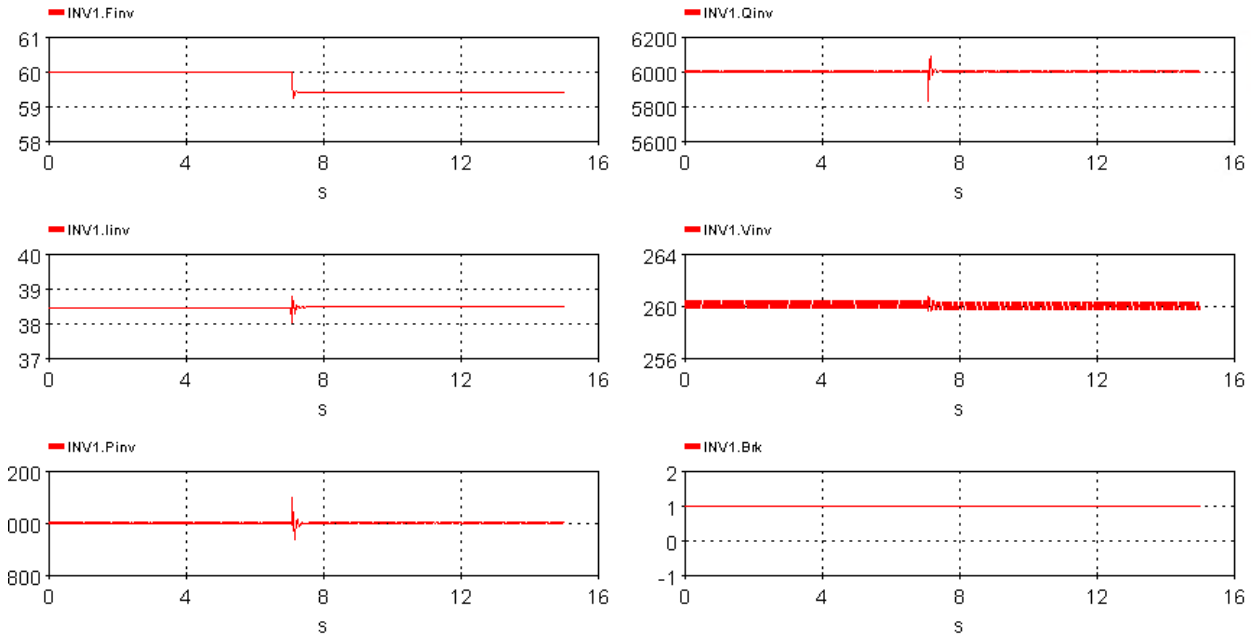


Figure 26: Single phase GFL inverter grid frequency sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

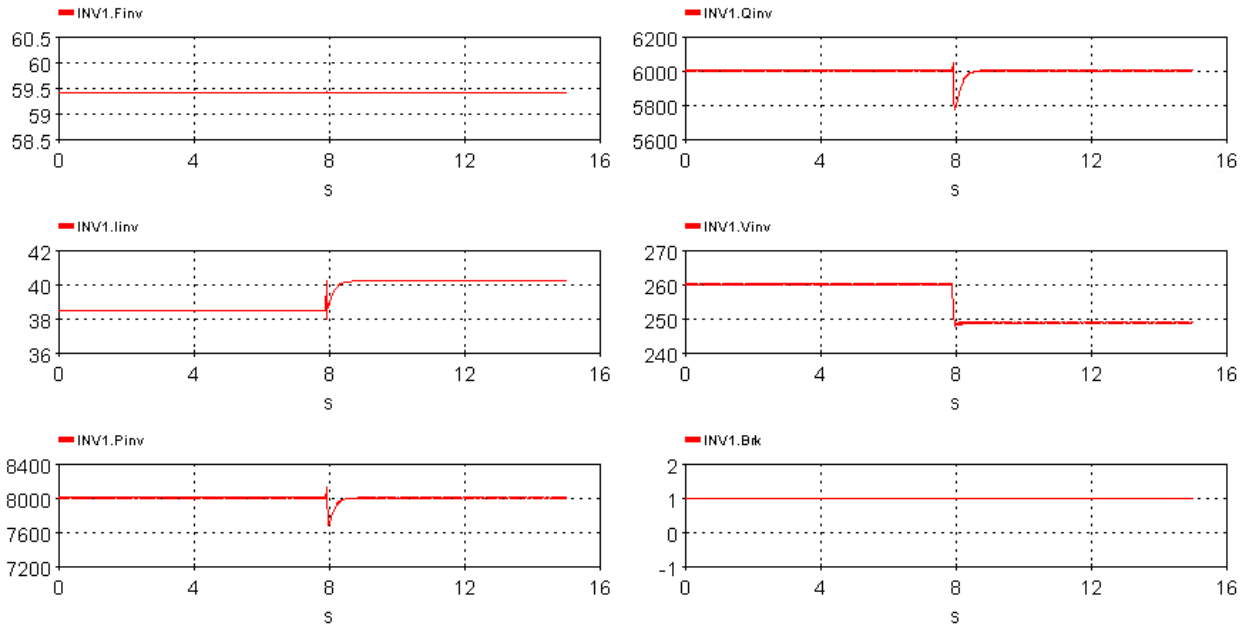


Figure 27: Single phase GFL inverter grid voltage sag test. The model variables are: measured frequency (top-left); reactive power injection (top-right); RMS average phase current (middle-left); RMS average phase voltage (middle-right); active power injection (bottom-left); and PCC breaker status.

3.0 Mitigation Strategies Overview

3.1 SLAC3R

This section discusses the several mitigation strategies developed under the RD2C initiative and used in the Capstone I project for integration with the HYPERSIM model. We first begin by recalling the Grid Forming Inverter (GFM) inverter and then follow up with SLAC3R controls, a two part controls, namely decentralized autonomous and coordinated.

3.1.1 GFM Model and Inverter Controls

3.1.1.1 Grid Forming Inverter

Consider the GFM inverter with CERTS droop-controlled GFM model from [18, 19] as shown in Fig. 28. The GFM inverter model consists of a P-f droop control, Q-V droop control as well as over- and under-load mitigation strategy.

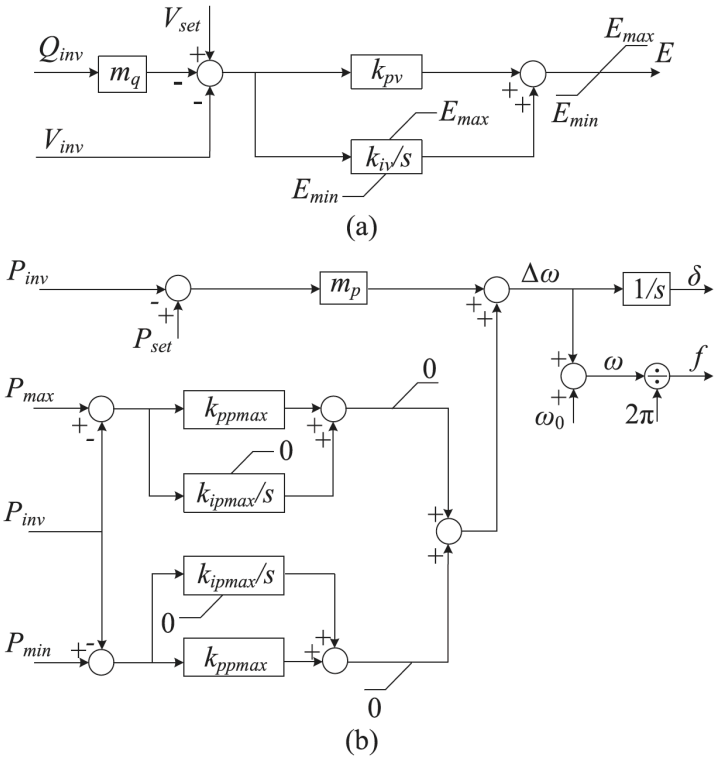


Figure 28: CERTS droop-controlled GFM inverter model, with (a) Q-V droop control, and (b) P-f droop control, including over- and under-load mitigation (adopted from [19]).

In addition, there are low-pass measurement filters that measure the active (P), reactive (Q) power, and voltage at the point of common coupling for the inverter. $P_{\text{grid},\phi}$ and $Q_{\text{grid},\phi}$ are the active and reactive power injections from the inverter into the grid, at phase $\phi \in \{a, b, c\}$ while $V_{\text{grid},\phi}$ is the voltage at phase $\phi \in \{a, b, c\}$. The inverters also honor power rating constraints;

therefore,

$$P_{\text{grid}} \in [P_{\text{min}}, P_{\text{max}}]$$

With the assumption that $\dot{P}_{\text{set}} = 0$ (i.e., the P_{set} is changed slowly compared to the inverter droop control timescales), we have the P-f droop equations given by

$$\dot{\delta} = \omega - \omega_0 \tag{1}$$

$$\tau\dot{\omega} = \omega_0 - \omega + m_p \left(P_{\text{set}} - P_{\text{inv}}\right) \tag{2}$$

where $P_{\text{inv}} := \sum_{\phi\in\{a,b,c\}} P_{\text{grid},\phi}$, ω_0 is the nominal frequency, m_p is the P-f droop gain, and $P_{\text{set}}, V_{\text{set}}$ are the real power and voltage inputs to the inverter dynamics. As mentioned earlier, the setpoint dynamics and the inverter droop control dynamics operate at different timescales, and as such there are separate control strategies for controlling the inverter dynamics (primary control) and setpoint dynamics (secondary control) [20].

3.1.1.2 Primary Controls

The P-f droop control at the GFM inverters acts as a primary control to the inverters and operates at millisecond (ms) timescales. The P-f droop controls (based on the internal inverter measurements) react quickly to any disturbance in the system to stabilize the frequency. The input to the droop controls is the reference setpoint (P_{set} or Q_{set}), which is usually provided by the secondary controls. The action of the droop controls in most cases will stabilize the frequency; however, there are no guarantees that the frequency will go back to the nominal frequency (60 Hz).

3.1.1.3 Secondary Controls

The real power setpoints to the inverters are updated by the secondary controls by accessing the inverter measurements. These controls could be either centralized or distributed and operate at slower timescales (in the order of a few seconds), compared to primary controls. The objective of this control is to change the real power setpoints in such a way that the frequency reaches nominal with equal power sharing among the inverters. The secondary controls discussed in [21] are implemented in this work. These controls are based on the leader-follower consensus framework and coordinate the GFM and grid-following (GFL) inverters to achieve frequency restoration to nominal frequency and accurate power sharing.

In the scope of this work, we define a safe operating region around the nominal frequency and define resilience as the ability of the system to maintain this safe region irrespective of the type of events happening on the system. Considering the timescale gap between the primary controls and secondary controls of operation there is a blind spot and if an event happens during this period the primary controls react but cannot ensure frequency resilience that could avoid any violations that lead to blackouts. To bridge the timescale gap between the primary and secondary controls and achieve resilience, we introduce reactive controls that are decentralized and maintain resilience.

3.1.2 Decentralized Autonomous Controls

The functioning of the decentralized autonomous controls (DACs) is illustrated in Fig. 29 and is described in detail as follows. The DAC framework that utilizes a few local measurements from

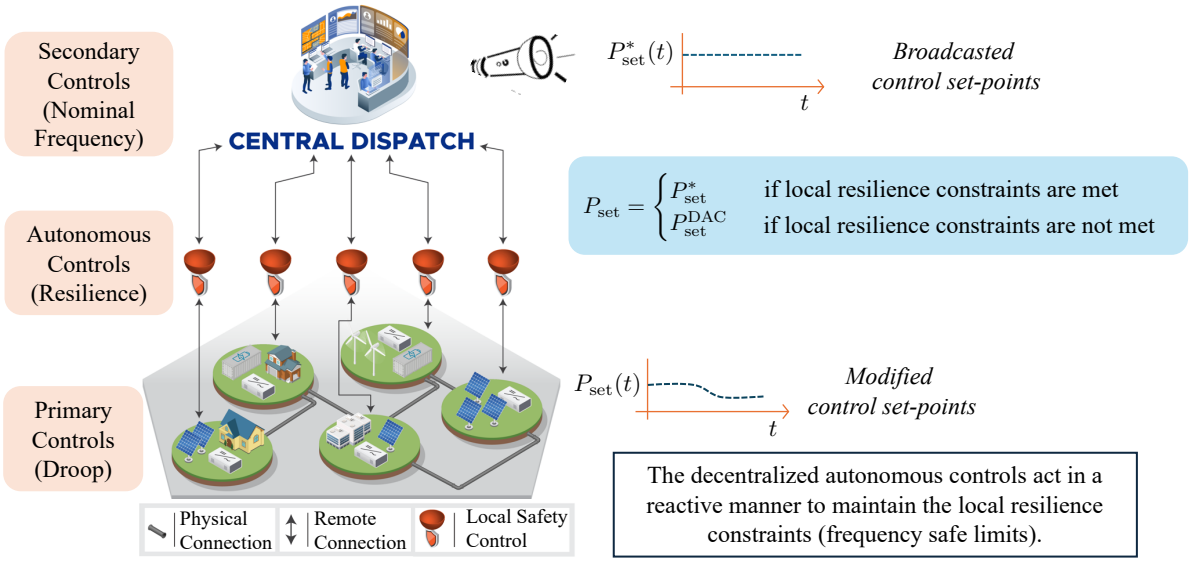


Figure 29: Overview of the functioning of the proposed decentralized autonomous controls (DACs): These inverter-based controls are located between primary and secondary controls. When the frequency resilience constraints are violated, these controls adjust the secondary control setpoints to ensure frequency resilience, otherwise, these controls will not intervene and pass the secondary control setpoints to the inverter. These autonomous controls are computationally efficient and rely solely on local measurements available at the inverters themselves.

the inverters to maintain a predefined safe region for frequency under disturbances. We begin by defining the local resilience constraints at each inverter.

At each inverter, suppose the frequency limits are defined as $\omega \in [\omega_{\text{min}}, \omega_{\text{max}}]$ where $\omega_{\text{min}}, \omega_{\text{max}}$ are the lower and upper-frequency limits respectively. These pre-defined safe limits for frequency at each inverter form the local resilience constraints that need to be maintained irrespective of the nature of the disturbance. Therefore, the goal is to modify the set points P_{set}, such that the local resilience constraints are respected. To this end, we introduce two barrier functions, corresponding to lower and upper frequency limits, that act when frequencies deviate from the safe limits either from below or above.

Consider the following barrier functions defined as,

$$B_{\text{min}}(\omega) = \omega - \omega_{\text{min}}, \text{ such that } \begin{cases} \geq 0, & \text{Safe} \\ < 0, & \text{Unsafe} \end{cases} \tag{3}$$

$$B_{\text{max}}(\omega) = \omega - \omega_{\text{max}}, \text{ such that } \begin{cases} \leq 0, & \text{Safe} \\ > 0, & \text{Unsafe} \end{cases} \tag{4}$$

We refer to B_{min} and B_{max} as the lower and upper barrier functions respectively. When the lower (upper) barrier function is positive (negative), the lower (upper) frequency safe limit is not violated. However, in the presence of a disturbance, if the lower frequency limit is violated, we want the derivative of the lower barrier function to change its sign and become non-decreasing. Similarly, due to a disturbance, if the upper frequency limit is violated, we want the derivative of the upper barrier function to change its sign and become non-increasing. Thus we need the

following conditions on the time-derivative of the barrier functions.

$$\dot{B}_{\text{min}} \geq -\alpha B_{\text{min}}^{q}, \tag{5}$$

$$\dot{B}_{\text{max}} \leq -\alpha B_{\text{max}}^{q}, \tag{6}$$

where q is required to be an odd number to preserve the sign of the right hand side of Eqs. (5)-(6) and α can be considered as control performance gain. The addition of the term, $-\alpha B_{\text{min}}^{q}$ ($-\alpha B_{\text{max}}^{q}$) on the right side of the barrier function derivative requires explanation. Whenever the frequencies are inside the safe region, the lower barrier function is nonnegative and we can allow the lower barrier function to decrease if the operating frequency is well inside the safe region. However, when the operating frequency goes outside the safe region from below, we want the lower barrier function to increase and the corresponding time-derivative to be non-negative (that is, $\dot{B}_{\text{min}} \geq 0$).

Similarly, the non-positive values of the upper barrier function indicate the frequency is within the safety limits, and the upper barrier function can increase inside the safe region as long as the operating frequency is lower than ω_{max}. However, when the operating frequency increases beyond, ω_{max}, the upper barrier function should decrease and the corresponding time-derivative should be non-positive (that is, $\dot{B}_{\text{max}} \leq 0$).

When the frequencies deviate outside the safe region due to an event, it is expected to steer the system to the safe region in finite time and to achieve this, we modify, $\dot{B}_{\text{min}} \geq 0$ to $\dot{B}_{\text{min}} \geq -\alpha B_{\text{min}}^{q}$ and similarly, $\dot{B}_{\text{max}} \leq 0$ to $\dot{B}_{\text{max}} \leq -\alpha B_{\text{max}}^{q}$. To achieve the desired performance to maintain safety while ensuring the smoothness of the resulting control adjustment, we chose $q = 3$. Therefore, from Eq. (5), we have,

$$\begin{aligned} P_{\text{set}} &\geq P_{\text{inv}} + \frac{1}{m_p}\left(\omega - \omega_0 - \alpha(\omega - \omega_{\text{min}})^3\right) \\ \implies \omega &\geq \omega_{\text{min}} \quad (\text{lower limit}) \end{aligned} \tag{7}$$

Similarly, from Eq. (6), we have,

$$\begin{aligned} P_{\text{set}} &\leq P_{\text{inv}} + \frac{1}{m_p}\left(\omega - \omega_0 - \alpha(\omega - \omega_{\text{max}})^3\right) \\ \implies \omega &\leq \omega_{\text{max}} \quad (\text{upper limit}) \end{aligned} \tag{8}$$

The setpoint changes suggested by Eq. (7) and Eq. (8) result in a frequency that is within the safe region. However, as the scope of this work is only concerned with maintaining the safe region and allowing the secondary control to drive the frequency to the nominal value, it is sufficient to make minimal setpoint changes such that either $\omega = \omega_{\text{min}}$ or $\omega = \omega_{\text{max}}$ is maintained. This results in the following conditions.

$$\begin{aligned} P_{\text{set}}^{\text{low}} &= P_{\text{inv}} + \frac{1}{m_p}\left(\omega - \omega_0 - \alpha(\omega - \omega_{\text{min}})^3\right) \\ &\implies \omega = \omega_{\text{min}} \end{aligned} \tag{9}$$

$$\begin{aligned} P_{\text{set}}^{\text{up}} &= P_{\text{inv}} + \frac{1}{m_p}\left(\omega - \omega_0 - \alpha(\omega - \omega_{\text{max}})^3\right) \\ &\implies \omega = \omega_{\text{max}} \end{aligned} \tag{10}$$

When any event on the system drives the frequencies outside the safe region either from below (under frequency event) or above (over frequency event), the setpoints to the GFM inverters are adjusted accordingly either by Eq. (9) or Eq. (10) to maintain the safe region.

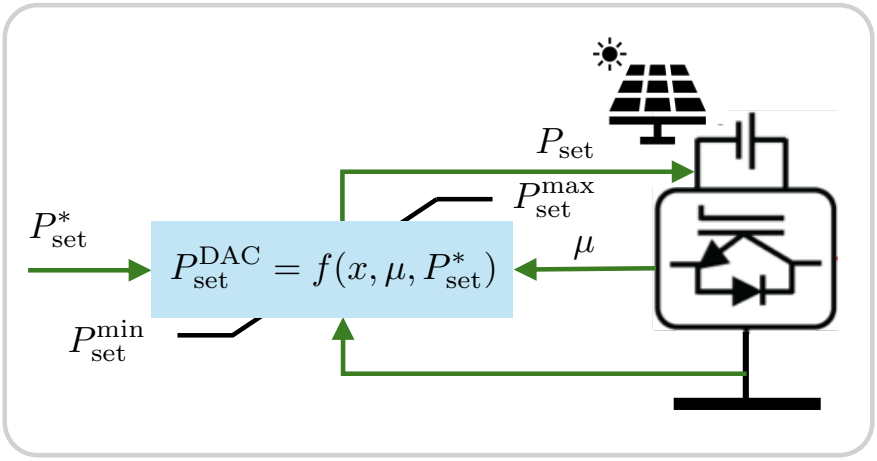


Figure 30: DAC implementation at each inverter. P^*_{set} denotes the setpoint sent by the secondary control, $\mu = (S_{\text{inv}}, m_p)$ denotes the fixed parameters at the inverters and $x = (\omega, P_{\text{inv}}, Q_{\text{inv}})$ denote the measurements from the inverter.

These two cases are combined to deploy these safety controls at the inverters. Let P^*_{set} be the set point given by the secondary controls or central dispatch to the inverter and let P_{set} be the setpoint seen by the inverter. Since the safe controls always act to maintain the safe region, we have,

$$P_{\text{set}} = \begin{cases} P^*_{\text{set}} & \text{if } \omega_{\text{min}} \leq \omega \leq \omega_{\text{max}} \\ P^{\text{DAC}}_{\text{set}} & \text{otherwise} \end{cases} \tag{11}$$

where

$$P^{\text{DAC}}_{\text{set}} = min(P^{\text{up}}_{\text{set}}, max(P^{\text{low}}_{\text{set}}, P^*_{\text{set}})) \tag{12}$$

where $P^{\text{DAC}}_{\text{set}}$ denotes the setpoints that are modified by the safety-promoting DACs. Note that the control performance gain α is chosen in such a way that these safety-promoting DACs won't act if frequency violations will not happen. Therefore, when $\omega_{\text{min}} \leq \omega \leq \omega_{\text{max}}$, $P_{\text{set}} = P^*_{\text{set}}$, which essentially indicates that the real power setpoint provided by secondary controls will be passed to the inverter. Furthermore, the following criterion ensures that the DACs setpoint changes respect the capacity constraints, we have the following criterion.

$$\begin{aligned} P^{\text{max}}_{\text{set}} &= \sqrt{S^2_{\text{inv}} - Q^2_{\text{inv}}} \\ P_{\text{set}} &= min(P^{\text{max}}_{\text{set}}, max(P^{\text{min}}_{\text{set}}, P_{\text{set}})), \end{aligned} \tag{13}$$

where $Q_{\text{inv}} = \sum_{\phi \in \{a,b,c\}} Q_{\text{grid},\phi}$, and S_{inv} is the size of the inverter. The implementation of the safety-promoting DACs is illustrated in Fig. 30. The DACs use only a few measurements from the inverters to adjust the real power setpoints at the inverters and maintain the safe frequency region (in other words, local resilience). The real power control setpoints are modified according to Eq. (12) which is an algebraic equation, and as a consequence, these controls are computationally efficient. Therefore, due to the decentralized autonomous and computationally efficient properties of the DACs, they are more suitable for real-world applications.

Fig. 31 provides an overview of how the DACs function during the onset of a dynamic event. When a dynamic event occurs that results in an over-frequency event and the safe region is

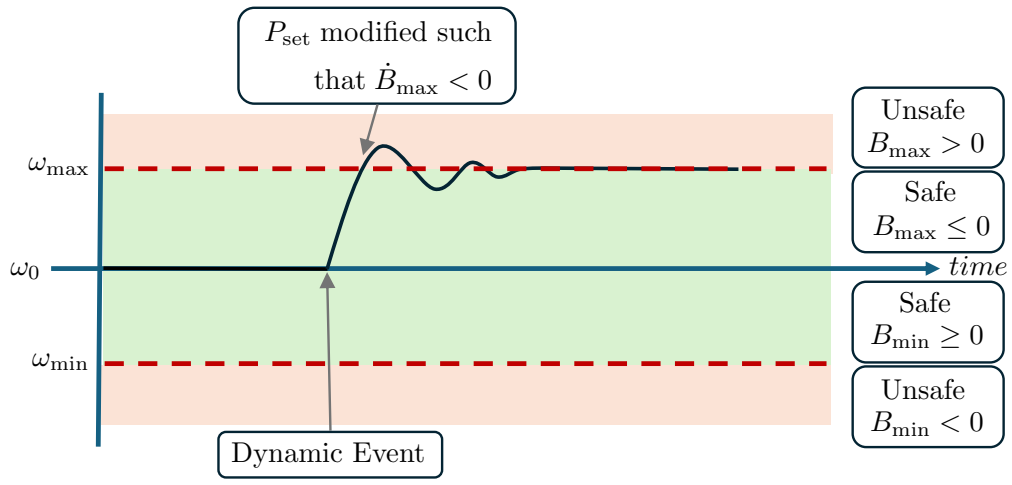


Figure 31: Illustration of the functioning of safety promoting DACs when frequency violations occur.

violated, the control setpoints are modified in such a way that the frequency safe region is maintained (that is, $P_{\text{set}}^{\text{DAC}}$ modified such that $\dot{B}_{\text{max}} \leq 0$).

The following algorithmic procedure encapsulates the entire DAC implementation methodology applicable to any GFM inverter.

Algorithm 1 DAC algorithm to compute the new setpoint at any GFM inverter

Require: Fixed parameters $S_{\text{inv}}, m_p, \alpha$

Inputs: $\omega, P_{\text{inv}}, Q_{\text{inv}}, P^*_{\text{set}}$

Compute $P_{\text{set}}^{\text{low}}, P_{\text{set}}^{\text{up}}$ from Eqs. (9)-(10)

Find P_{set} from Eq. (11).

Use Eq. (13) to ensure P_{set} is within the absolute limits.

Outputs: P_{set}

3.1.2.1 Control Performance Design Guidelines

The control performance parameters, α and q are introduced to ensure certain conditions on the time-derivative of the barrier functions which results in appropriate control setpoint changes to guarantee the safe frequency region. Since we want the control to act minimally inside the safe bounds and act significantly when outside the safe region, we cannot have a constant rate of change for the derivatives of B_{min} and B_{max}, and hence the derivatives of B_{min} and B_{max} need to be nonlinear as a function of the frequency deviation. This is achieved by introducing q and theoretically, q can be any odd natural number.

However, for large q, B_{min}^q or B_{max}^q grows exponentially fast for large deviations in frequency and makes the control extremely aggressive. Thus, for practical consideration, we cannot choose large q. Therefore, for large deviations in ω, $\dot{B}_{min}$ and $\dot{B}_{max}$ becomes extremely large. On the other end, for $q = 1$, the derivatives of the barrier function remain constant both inside and outside the safe region. Hence, we chose $q = 3$, the smallest odd natural number greater than one.

To control the time-derivative of barrier functions, the control performance gain, $\alpha > 0$ is

introduced in Eqs. (5)-(6). For a given $\alpha > 0$, B_{min} or B_{max} reaches steady state exponentially. In practical scenarios, the value α is adjusted to magnify the frequency error, thereby enabling smoother control. Consequently, the selection of α depends on the chosen value of q. For instance, consider $q = 3$, $\omega_{\text{min}} = 59.5Hz$, $\omega_{\text{max}} = 60.5Hz$ and current $\omega = 59.4Hz$. Then α must be at least $|(59.4 - 59.5)^3| = 0.001$. If $\alpha \leq 0.001$, then the control does not see any deviation from the safe region and hence won't act. However, suppose if $\alpha > 0.001$, then the controls would start acting and the control action becomes aggressive if α is larger.

3.1.3 Coordinated Controls

In the onset of adversarial events and other unforeseen events (such as losing an inverter in a microgrid), to ensure system-wide stability guarantees and achieve resilience, we present a cooperative control that appropriately updates the control setpoints to the device or subsystem controls by transitioning the operating point (set by the centralized slower control). The cooperative control operates relatively at a faster timescale when compared to the supervisory centralized coordinator and is located a level below it but lies above the decentralized autonomous controls among the hierarchy of controls.

Following [21], a real-time distributed algorithm based on consensus is considered for all the inverters that updates the setpoints P_{set}, V_{set} to correct the steady-state deviations of system variables. The update rule can be written as follows:

$$P^{+}_{\text{set}_{k,i}} = P_{\text{set}_{k,i}} - \zeta_1(\omega_i - \omega_0) - \zeta_2 \sum_{j \neq i} (m_{p_i} P_{\text{set}_i} - m_{p_j} P_{\text{set}_j}) \tag{14}$$

$$V^{+}_{\text{set}_{k,i}} = V_{\text{set}_{k,i}} - \zeta_3(V_i - V_0) - \zeta_4 \sum_{j \neq i} (m_{q_i} Q_i - m_{q_j} Q_j) \tag{15}$$

where the index k denote the time and i denote the i^{th} inverter. The constants, $\zeta_j,\ j = \{1, 2, 3, 4\}$ are the corresponding weights essentially trading off between achieving consensus against reaching the nominal frequency and voltage. For GFM inverters, as they can regulate their own frequency and voltage independently, $\zeta_1 \neq 0$ and $\zeta_3 \neq 0$ whereas for GFL inverters, $\zeta_1 = 0$ and $\zeta_3 = 0$. Note that Eqs. (14) and (15) denote the distributed implementation of the coordinated controls.

The neighboring inverter measurements (setpoints) are needed the implementation of coordinated controls. These measurements were exchanged via physical aware cyber platform (PACP) introduced later in this section.

3.1.3.1 Implementation Challenges

In the HYPERSIM implementation of these controls, it was noticed that for GFM inverters, they accept the real power and voltage setpoints, however for GFL inverters, they do not accept the voltage setpoint, but accepts the reactive power setpoints. Therefore, the voltage setpoints are appropriately converted to reactive power setpoint as follows:

$$Q_{\text{set}} = (V_0 - V_{\text{set}})/m_q + Q_{\text{inv}} \tag{16}$$

All the setpoint calculations are performed in pu values by considering the measurements in pu values.

Additionally, to avoid extreme control actions during any high impact low probability events, we limit the control actions to reasonable values by putting saturation constraints on the controls.

These saturation limits are included besides the capacity constraints. For real power and voltage, we have the following saturation limits.

$$P_{\text{set}} \in [-1, 1]$$
$$V_{\text{set}} \in [0.8, 1.2]$$

3.2 ALERT Controls

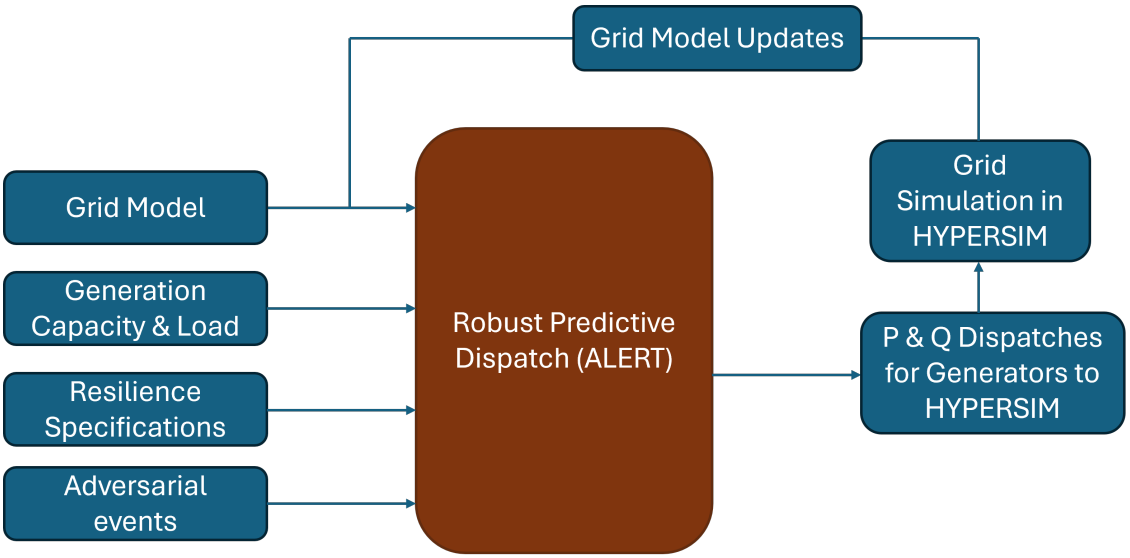


Figure 32: Schematic diagram of ALERT coordination with HYPERSIM.

Figure 32 illustrates the coordination framework of the ALERT controller. The controller receives as inputs the grid model, generation capacity and load, resilience specifications, and adversarial events. Based on this information, it solves a robust multi-period optimal power flow problem to determine the active (P) and reactive (Q) power dispatches of generation resources, while ensuring appropriate reserve margins. Since ALERT operates as a predictive controller, it continuously updates its optimization across multiple periods. Consequently, any grid model changes—such as generator or branch outages, or corrective actions applied within HYPERSIM—must be communicated back to ALERT for consideration in subsequent optimization cycles. The resilience specifications include operational constraints such as reserve margins and solar curtailment limits. Adversarial events represent both cyber-physical and weather-induced disruptions, including load alterations, diesel generator outages, solar inverter failures, and reductions in solar irradiance, as well as combinations thereof. These events are defined by the system operator in accordance with their risk preferences (risk aversion or risk tolerance), and they directly inform the robust dispatch formulation described in the following section.

The detailed formulation of multi-period, robust predictive optimization for ALERT can be found in [4]. Here, we present only the robust formulation of the ALERT controller. The central principle behind making the optimization robust is to allocate sufficient reserves across dispatchable resources, such as diesel generators and solar inverters–so that local controllers have room to adjust their outputs in response to adversarial events.

In its most general form, the robust optimization problem can be written as:

$$\min_{u} \max_{w} f(u, w) \tag{17a}$$

$$\text{s.t.} \quad g(u, w) \leq 0, \quad \forall w \in \Omega_W \tag{17b}$$

where $g(u, w)$ represents the operational and resilience constraints. For details, see [4]. In (17), u denotes the nominal operational decisions, while w represents the adversarial events.

The above formulation (17) can be reformulated using the *explicit maximization method* [22] as:

$$\min_{u} \ \widehat{f}(u), \quad \text{s.t. } \widehat{g}(u) \leq 0 \tag{18}$$

where $\widehat{f}(u) = \max_w f(u, w)$ and $\widehat{g}(u) = \max_w g(u, w)$. This reformulation eliminates the disturbance terms. If both $\widehat{f}(u)$ and $\widehat{g}(u)$ are convex, the resulting optimization problem remains convex and can therefore be solved efficiently.

System resilience is enforced by embedding user-defined weights into reserve margin requirements and by introducing penalty terms into generation costs. The objective of this robust microgrid optimization is to minimize solar and load curtailments, minimize diesel generator usage, and maximize reserves, while simultaneously satisfying network constraints. Furthermore, reserve payments for generators are modeled such that reserve payments decreases with increasing reserve allocation, thereby incentivizing efficient reserve management.

3.2.0.1 Implementation Challenges

The implementation of the ALERT controller in HYPERSIM presented several challenges. First, the ALERT controller incorporates predictive control and online adaptation for adversarial events. However, for this effort, SLAC3R and P2PC were employed as local adaptive controllers. To address this, the predictive and online adaptation components of the ALERT controller were decoupled, and only the predictive component was utilized. Second, while ALERT traditionally coordinates via GridLAB-D, in this implementation, the controller was interfaced with HYPERSIM using virtual machines. Third, the original predictive optimization algorithm did not account for optimized reserve capacity across total resource capacity (kVA); instead, reserves were allocated separately for active power (P) and reactive power (Q). This was modified to incorporate reserve capacity as a proportion of the total resource capacity, which is critical for accurately modeling reserves during generator outages. Furthermore, a linear reserve vs reserve payments is modeled to tradeoff the reserve maximization and dispatches. Fourth, the ALERT code was refactored to accept input and produce output within the execution of a single script. Finally, an automated script was developed to generate ALERT inputs for any power network model, significantly reducing the need for manual input preparation.

3.3 Coordinated Voltage Support with Peer-to-Peer Control

We analyze a system of N microgrids, denoted as $\text{MG}_1, \ldots, \text{MG}_N$, operating in a networked microgrid configuration. Each microgrid is equipped with its own dedicated controller, which gathers data from devices within that specific microgrid and executes various control tasks. Additionally, these controllers have the capability to share information with one another. We consider that the microgrids need to collectively communicate and decide on the additional reactive power support from individual microgrids that are fairly shared with respect to their ratings and available

reactive power headroom. Before the event happened and in nominal operating conditions each MG controller keeps track of their inverter reactive power headroom $Q_1^{h,\text{nom}}, \ldots, Q_N^{h,\text{nom}}$. For the inverter j in the MG i, the reactive power headroom is computed as:

$$Q_{i,j}^{\text{max\_head,nom}} = \sqrt{S_{i,j}^2 - P_{i,j}^2}, \tag{19}$$

$$Q_{i,j}^{h,\text{nom}} = Q_{i,j}^{\text{max\_head,nom}} - Q_{i,j}, \tag{20}$$

where $S_{i,j}$ is the apparent power rating, and $P_{i,j}$ and $Q_{i,j}$ are the active and reactive power measurements, and using peer-to-peer consensus estimates the total nominal reactive power headroom $Q^{h,\text{nom}} = \sum_{i=1}^{N} Q_i^{h,\text{nom}}$. Each MG compute their reactive power headroom and monitor voltages in a continuous manner at fixed periodic intervals. If any of the MG exhibits voltage deviations from the nominal due to some internal disturbances, the P2P controls then assigns a new voltage setpoint to the inverters based on the available reactive power headroom from the other MGs.

At any given time after the event, the MGs compute their available headroom $Q_1^{h,e}, \ldots, Q_N^{h,e}$ using the current active and reactive power measurements similarly as shown above for the nominal condition, and the total availability $Q^{h,e} = \sum_{i=1}^{N} Q_i^{h,e}$ using the peer-to-peer consensus. Suppose if $Q_{i,j}^{h,e} > 0$, then there is non-zero reactive power headroom to support the peer-to-peer control operations and have reactive power sharing among the microgrids. However, if $Q_{i,j}^{h,e} \leq 0$, then, there is no reactive power headroom and will not participate in the peer-to-peer control operations. Subsequently, we estimate the total reactive power needed (i.e., ΔQ) after a voltage event as:

$$\Delta Q = \alpha \frac{(Q^{h,e} - Q^{h,\text{nom}})}{\sum_i S_i} \text{ p.u.}, \tag{21}$$

where the change in reactive power headroom captures the impact of the voltage event in the reactive power deficit. The scaling factor $\alpha > 1$ helps to consider the consumptions of other reactive components such as shunt elements apart from the inverters.

MG controllers then locally determine the amount of additional reactive power injection within their headroom.

$$Q_i^{\text{add\_head}} = \frac{Q_i^{h,e}}{(\sum_{i=1}^{N} Q_i^{h,e})} \Delta Q \text{ p.u.} \tag{22}$$

Subsequently this additional reactive power requirement has been transferred to individual inverters in the microgrids using the following:

$$\Delta Q_{i,j}^{\text{inv\_share}} = \frac{Q_i^{\text{add\_head}} \cdot Q_{i,j}^{h,e}}{Q_i^{h,e}} \text{ p.u.}$$

$$\Delta V_{\text{set\_P2P}_{i,j}} = -m_q^{i,j} \cdot \Delta Q_{i,j}^{\text{inv\_share}} \text{ p.u.}$$

The voltage control setpoints are sent to the inverters via the SLAC3R Coordinated controls as described in the subsequent integration section.

3.3.0.1 Resilient P2P Controls:

To deal with the cyberattack on the MG computations and communication links, we have developed a resilient consensus algorithm in [5, 23] that determines the total nominal reactive power

headroom $Q^{h,0} = \sum_{i=1}^{N} Q_i^{h,0}$, and the total headroom after the voltage event $Q^{h,e} = \sum_{i=1}^{N} Q_i^{h,e}$ in the presence of cyberattacks. For general purpose notational use, let us denote $\tilde{Q}_i$ to be the reactive power headroom variable of interest for the i-th MG during a particular resilient consensus computation interval, i.e., $\tilde{Q}_i$ denotes $Q_i^{h,0}$ for the initial headroom computation, and $Q_i^{h,e}$ for the after-event headroom computation. In this work, we consider that the MG controller can get cyber attacked on the computation when estimating the total reactive power headroom, i.e., the data injection attack to computation of MG controller is considered. In the consensus algorithm, MG controller i updates its estimate of the reactive power headroom at consensus iteration $(k+1)$ based on the previous consensus step k, as follows:

$$\tilde{Q}_i^{k+1} = w_{ii}\tilde{Q}_i^k + \sum_{j \in \mathcal{N}_i} w_{ij}\tilde{Q}_j^k + u_i^k, k = 0, 1, \ldots \tag{23}$$

where w_{ij} represents the weight assigned to controller j, and u_i^k accounts for the perturbations introduced by a data injection attack.

Suppose there are f malicious MG controllers, indexed by the set $\mathcal{F} = \{x_{i1}, \ldots, x_{if}\}$. Let $\mathbf{Q}^k$ denote the vector of reactive power headrooms, and let $\mathbf{W}$ denote the weight matrix. Then, (23) can be expressed compactly as:

$$\mathbf{Q}^{k+1} = \mathbf{W}\mathbf{Q}^k + \mathbf{B}_{\mathcal{F}}\mathbf{u}_{\mathcal{F}}^k, \tag{24}$$

where $\mathbf{B}_{\mathcal{F}} = \begin{bmatrix} \mathbf{e}_{i1,N} & \ldots & \mathbf{e}_{if,N} \end{bmatrix}$, with $\mathbf{e}_{i1,N}$ denoting the N-dimensional unit vector having a 1 in the $i1^{\text{th}}$ position, and $\mathbf{u}_{\mathcal{F}}^k = \begin{bmatrix} (\mathbf{u}_{i1}^k)^\top & \ldots & (\mathbf{u}_{if}^k)^\top \end{bmatrix}^\top$.

During the update step k, the exchanged information among MGs provides controller i with the measurement

$$\mathbf{y}_i^k = \mathbf{C}_i\mathbf{Q}^k, \tag{25}$$

where $\mathbf{C}_i$ is a $(\deg_i + 1) \times N$ matrix containing a single 1 in each row, indicating the positions of the local values available for the linear consensus update law.

The objective is to relate the observable sequence $\mathbf{y}_i^k$ collected over iterations k to the initial vector of reactive power headrooms $\mathbf{Q}^0$. By combining (24) and (25), we obtain the stacked observation model over K iterations:

$$\mathbf{y}_i^{0:K} = \mathcal{O}_{i,K}\mathbf{Q}^0 + \mathcal{M}_{i,K}^{\mathcal{F}}\mathbf{u}_{\mathcal{F}}^{0:K-1}, \tag{26}$$

where $\mathbf{y}_i^{0:K} = \begin{bmatrix} (\mathbf{y}_i^0)^\top & (\mathbf{y}_i^1)^\top & \cdots & (\mathbf{y}_i^K)^\top \end{bmatrix}^\top$, and
$\mathbf{u}_{\mathcal{F}}^{0:K-1} = \begin{bmatrix} (\mathbf{u}_{\mathcal{F}}^0)^\top & (\mathbf{u}_{\mathcal{F}}^1)^\top & \cdots & (\mathbf{u}_{\mathcal{F}}^{K-1})^\top \end{bmatrix}^\top$.

To deal with the cyberattack to MG controllers, each MG controller uses the resilient consensus algorithm (Algorithm 2) following from [23, 24] to determine the total reactive power headroom in the presence of cyberattacks on the MG controllers. As showed in these works, to deal with the risk of cyberattacks on MG controllers, it is necessary that the P2P graph has sufficient connectivity, i.e., if there are f MG controllers got cyberattacks, then then P2P graph needs to have $(2f+1)$ connectivity. This condition also holds in the event of cyber attacks to the communication links, where the resilient consensus can still accurately estimate the total reactive power headroom if the remaining communication graph possesses $2f+1$ vertex connectivity. [5] describes a methodology for risk-informed P2P communication graph generation.

Algorithm 2 Resilient consensus algorithm for estimating the total of reactive power headroom [5]

Data: Generated P2P graph with required connectivity
Result: Estimates of the total reactive power headroom of all the MGs
for i=1:1:N **do**

1.*Receive* the reactive power headroom of the neighbouring controllers $\tilde{Q}_j, j \in \mathcal{N}_i$, and construct $\mathbf{y}_i^k$ based on (25).
2. *Run* linear iterative updates using (24).
3. *Construct* matrices $\mathcal{O}_{i,L}$ and $\mathcal{M}_{i,L}^{\mathcal{F}}$.

4. *Estimate* $\mathbf{Q}^0 = [\tilde{Q}_1, ..., \tilde{Q}_N]$ by reversing (26) with the rank condition given in Theorem 1 in [23]
5. Calculate the total headroom of the NMG $\sum_{i=1}^{N} \tilde{Q}_i$.

3.4 Physically Aware Cyber Platform

The Physically Aware Cyber Platform (PACP) is a middleware solution [6] with the goal to provide increased resilience to high–fidelity cyber-physical systems at the cyber layer. It detects and mitigates various communication latencies, glitches, disconnections, etc., and cyber–attacks such as distributed–denial–of–service (DDoS) and Man–in–The–Middle (MiTM) attacks in support of achieving resilience goals. The platform will proactively handle problems on the cyber layer, inform the control applications with detected problems and aid in applying cyber–informed resilient control strategies. PACP will inspect data packets and monitor network performance to detect malicious behavior such as DDoS attacks and command injection or false data injection attacks through a MiTM attack node. PACP uses ML models to detect the type of attack, confidence level. An RL mitigation agent acts at the cyber layer by using the detection output and other network traffic information such as throughput, network paths as input and try to block the traffic from attack node or reroute the traffic, if possible, through alternate network path such that it reduces the impact of the attack.

PACP's software framework has 4 main components. PACP's main component is its Data Transfer module which handles the data communication between nodes. The other three are the Data Management, Detection, and Mitigation modules. The Data Management module supports storing data sent between publishers and subscribers into a SQL database. The Detection and Mitigation modules include a ML classifier model to detect the presence of DDoS attacks given network data, and a Reinforcement Learning mitigation agent to reroute network traffic.

This Data Transfer module is the main component used in the Capstone 1 project to facilitate communication channels between the 8 VMs in Microgrid–3. A given PACP participant can have multiple publications or subscriptions with other PACP participants and transfer or receive data from them.

Each inverter VM in our Microgrid–3 architecture is both a publisher and subscriber. The 8 nodes all publish and send active and reactive power setpoint values calculated by the SLAC3R coordinated control algorithm to the other 7 nodes while also subscribing to every node to get their updated setpoint values. PACP handles the payload which is defined as a datatype with three fields (Inverter Name, Active Power Setpoint, Reactive Power Setpoint).

A pymodbus [25] script was created and integrated with our cooperative control code which manages data connections between HYPERSIM and the inverter VMs as well as the P2PC VM with the Microgrid–3 inverter VMs using the Modbus protocol.

To facilitate communication for SLAC3R cooperative control, a PACP participant is created for every inverter where we initialize its publications and subscriptions to the other Microgrid–3 nodes. After the initialization phase, a loop runs for the rest of the simulation that continuously receives values from the HYPERSIM simulation via pymodbus that is required by all of the controls.

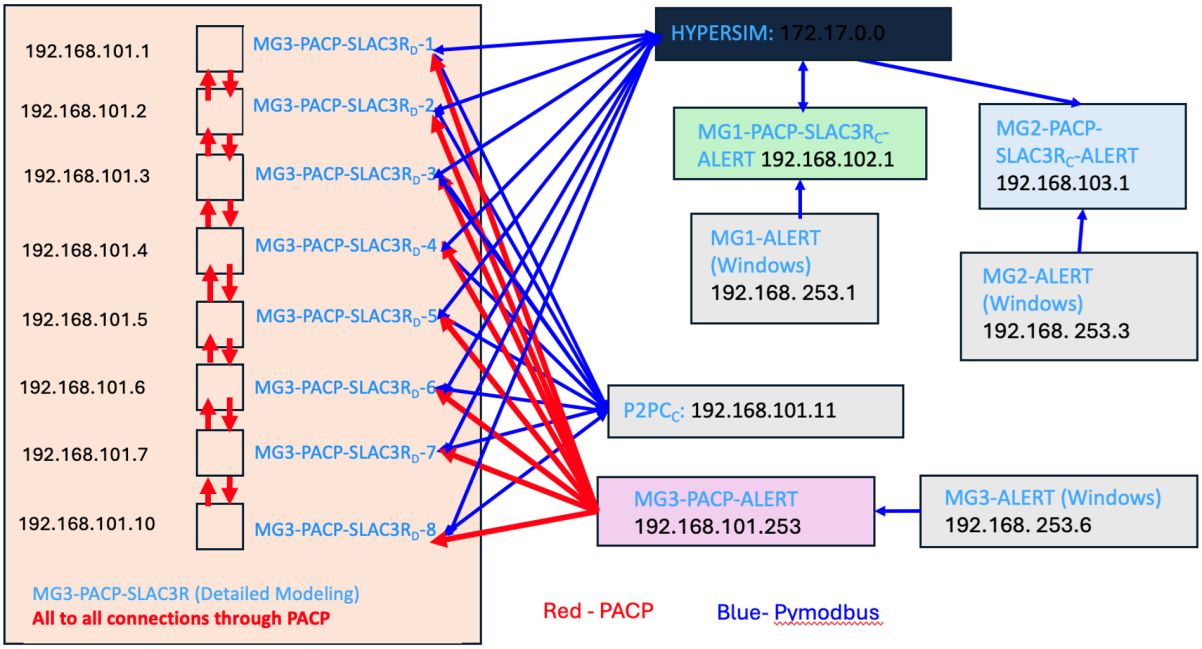


Figure 33: PACP and Modbus-TCP connections diagram.

Each microgrid ALERT algorithm calculates active and reactive power setpoints for the inverters at a slower time scale. These setpoints are transferred to the inverter controller VMs through PACP connections where they are processed and modified (if necessary) by the SLAC3R control algorithm to achieve local safety constraints. In parallel, P2PC voltage setpoints are sent via the pymodbus script running on a separate VM to each Microgrid–3 inverter controller VM. The inverter controller VMs exchange setpoints with their peers in order to accomplish consensus control objectives using PACP connections. Finally, each inverter VM uses a bare Modbus-TCP connection, implemented using pymodbus, to send setpoints to the simulated inverter primary controllers in the HYPERSIM simulation. Figure 33 illustrates the various connections between the networked virtual machines hosting the hierarchical control system and describes which connections are managed by PACP and what control algorithms are hosted on each virtual machine.

3.5 Control Integration

In the above sections, we introduced, the PACP, SLAC3R-L (DACs), SLAC3R-C, P2PC, and ALERT mitigation strategies. The SLAC3R-L controls are implemented directly on HYPERSIM using functional blocks of HYPERSIM and the control gains, frequency limits are embedded into it. The communication between the inverters or MGs is facilitated via PACP. The ALERT and P2PC controls are integrated with the SLAC3R-C controls.

It is important to reiterate that the SLAC3R-L controls are controls at the inverters, SLAC3R-C are distributed controls in the microgrid, ALERT is a centralized predictive optimization with in a microgrid and finally, the P2P controls are between the microgrids in a NMG setting.

First, we discuss the P2P controls integration. The P2P control outputs additional change in the voltage setpoints to the inverters. As this is the correction required besides the existing SLAC3R-C controls, they are added together as follows for the j-th inverter in the i-th MG.

$$V_{\text{set}_{i,j}} = V_{\text{set\_SLAC3R}_{i,j}} + \Delta V_{\text{set\_P2P}_{i,j}} \text{ p.u.} \tag{27}$$

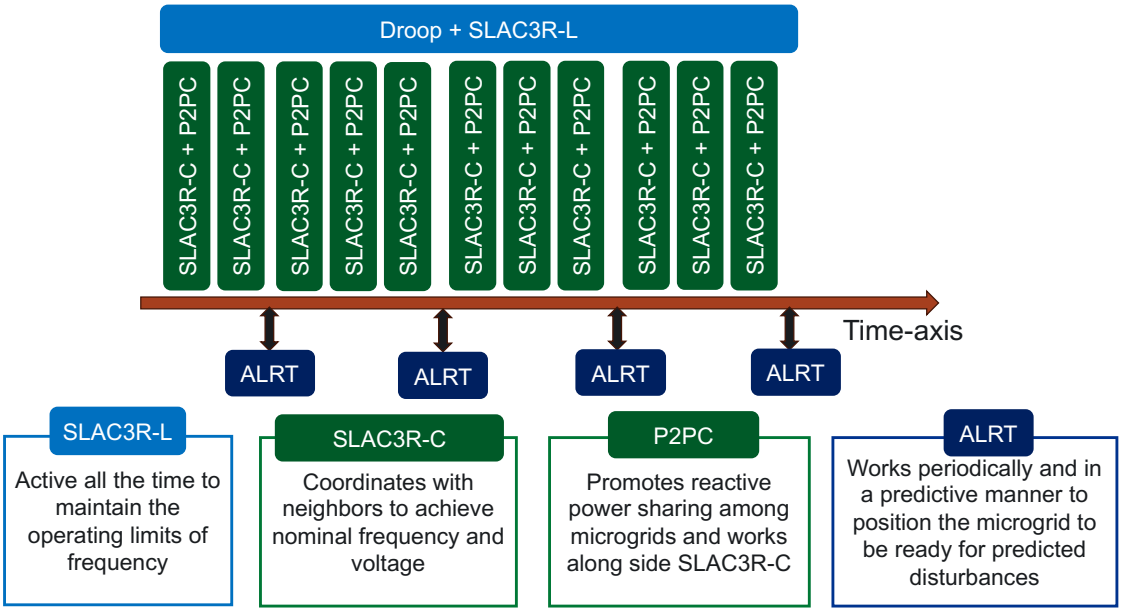


Figure 34: Depiction of timescales of operation of various control strategies

where the SLAC3R-coordinated algorithm's voltage control setpoints are computed based on Eq. (15) for individual inverters. $v_{i,j}^{\text{set}}$ denote the voltage setpoint of the j-th inverter for the MG i.

Next, we discuss the integration of the ALERT controls with the SLAC3R-C controls. First, we recall that ALERT control outputs the real and reactive power setpoints to the inverters. SLAC3R-C outputs the real power and voltage setpoints to the inverters. As ALERT optimization runs at slower timescales compared to SLAC3R-C, we bias the SLAC3R-C setpoints according to ALERT setpoints. Since the control update step involved in SLAC3R-C controls uses the real power setpoints and reactive power measurements, we scale them according to ALERT setpoints. Finally, we now have the following update step of SLAC3R-C controls (Eqs. (14) and (15)) considering the ALERT setpoints.

$$P^{+}_{\text{set}_{k,i}} = P_{\text{set}_{k,i}} - \zeta_1(\omega_i - \omega_0) - \zeta_2 \sum_{j \neq i} \left(m_{p_i} \frac{P_{\text{set}_i}}{P^{\text{ALERT}}_{\text{set}_i}} - m_{p_j} \frac{P_{\text{set}_j}}{P^{\text{ALERT}}_{\text{set}_j}} \right) \tag{28}$$

$$V^{+}_{\text{set}_{k,i}} = V_{\text{set}_{k,i}} - \zeta_3(V_i - V_0) - \zeta_4 \sum_{j \neq i} \left(m_{q_i} \frac{Q_i}{Q^{\text{ALERT}}_i} - m_{q_j} \frac{Q_j}{Q^{\text{ALERT}}_j} \right) \tag{29}$$

3.5.0.1 Summary of Mitigation Strategies and their Timescales

The SLAC3R-L controls are implemented to operate at the timescales of droop and the SLAC3R-C alongside P2PC is run every $0.5s$. Since ALERT is a centralized optimization, it is run every minute. Fig. 34 is an depiction and description of the timescales of operation of all the controls integrated into HYPERSIM.

The Table. 8 captures the inputs and outputs required by each control and the offline data they need for their computations.

	SLAC3R-C (every 0.5s)	P2PC (every 0.5s)	ALERT (every 1 min)
Local Inputs	Inverter voltage, frequency	Generated inverters real and reactive power	Estimated disturbance scenario
Neighbor Inputs (via PACP)	Real and reactive power setpoints	Microgrid reactive power headroom	Independent optimization per microgrid
Outputs (via pymodbus)	GFM: Pset, Vset; GFL: Pref, Qref	ΔQset per for each INV	Pset, Qset for all inverters, DGs
Offline Data	Inverter parameters such as inverter rating, nominal output power Droop coefficients Nominal voltage Inverter type/location	Inverter rating, Inverters nominal real and reactive power	Besides inverter data, they need network parameters, location/magnitude of loads, DG locations, size, nominal output power

Table 8: Summary of control strategies inputs/outputs and the data required for their operation.

4.0 Real-Time Cyber Physical Testbed Overview

4.1 Controller Network Architecture

This section will discuss the specifics of the testbed used to implement the hierarchical control system. The hierarchical control system realized in this project integrates multi-time scale controllers, such as – (a) SLAC3R, (b) ALERT and (c) P2P. Additionally, SLAC3R has 2 different components, (1) SLAC3R-L: local and device-level and (2) SLAC3R-C: distributed, coordinated and system-level.

Figure 35 shows the proposed testbed architecture for an IEEE 9500 node test system. However, due to issues with modeling of the 9500 node system, a smaller distribution system (modified IEEE 123 node test system) is implemented for testing the hierarchical controls. The actual implementation architecture for this system is shown in Fig. 36.

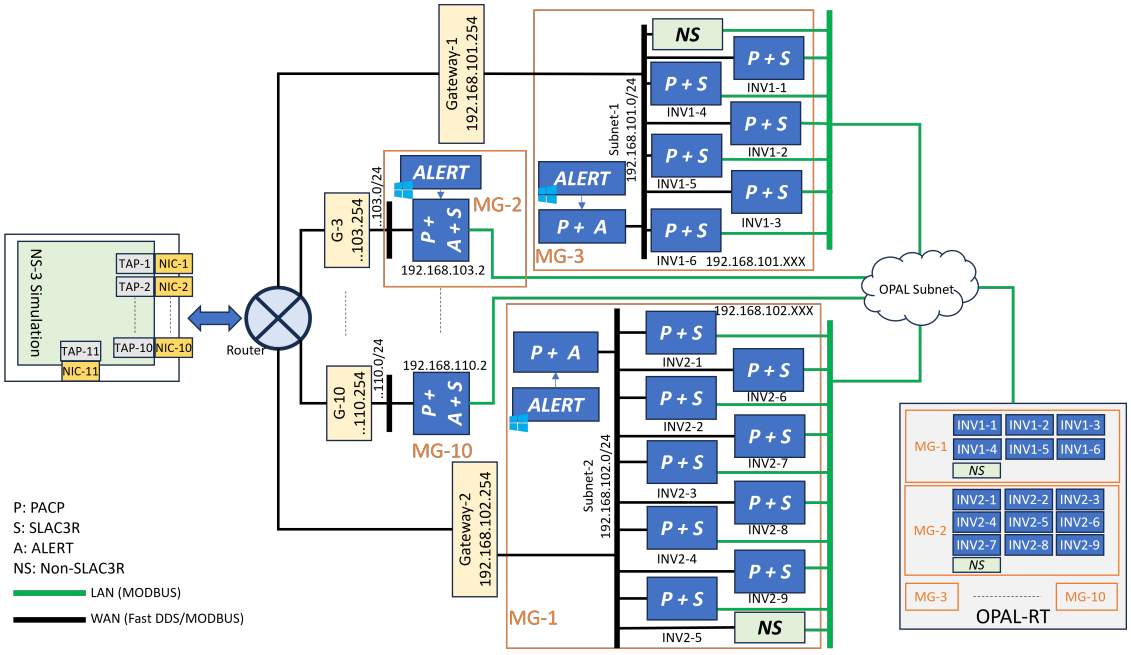


Figure 35: Proposed testbed architecture for large-scale distribution system (such as IEEE 9500 node test system)

The testbed design utilizes a virtual subnetwork for controllers associated with each microgrid. The subnetworks are connected through a virtual router, implemented using a Linux virtual machine (VM). To make the communication network scalable, the subnetworks are also connected through a VM hosting the Network Simulator-3 (NS3), which emulates a virtual communication network. The implementation and usage of NS3 is delineated in Fig. 36.

The first level of control (SLAC3R-C) is connected to the OPAL-RT ethernet interface, acting as the Local-Area Network (LAN), such as a substation communication network, and utilizes MODBUS for communication of measurements and control signals between the simulator and the controllers. The higher-level controls that require peer-to-peer communication, receive local data from SLAC3R-C, and exchange information with peer controllers over the wide-area network (WAN). The WAN allows communication through PACP subscribers and publishers over FAST-DDS protocol as well as through MODBUS protocol. The routing between the different

subnetworks is enabled through IP routes added to each controller VM. For example, for connecting a controller VM from subnet-1 with other subnetworks via the virtual router, the following IP route is added to the VM:

```
sudo ip route add 192.168.0.0/16 via 192.168.101.200 dev ens3
```

where $192.168.101.200$ is the IP address of the virtual router connected to the $ens3$ network interface of the controller VM in subnet-1. The virtual router automatically routes a packet arriving from one subnetwork to the other (connected to one of its interfaces) if the IP forwarding on the router is enabled. The command for enabling IP forwarding on the router is:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

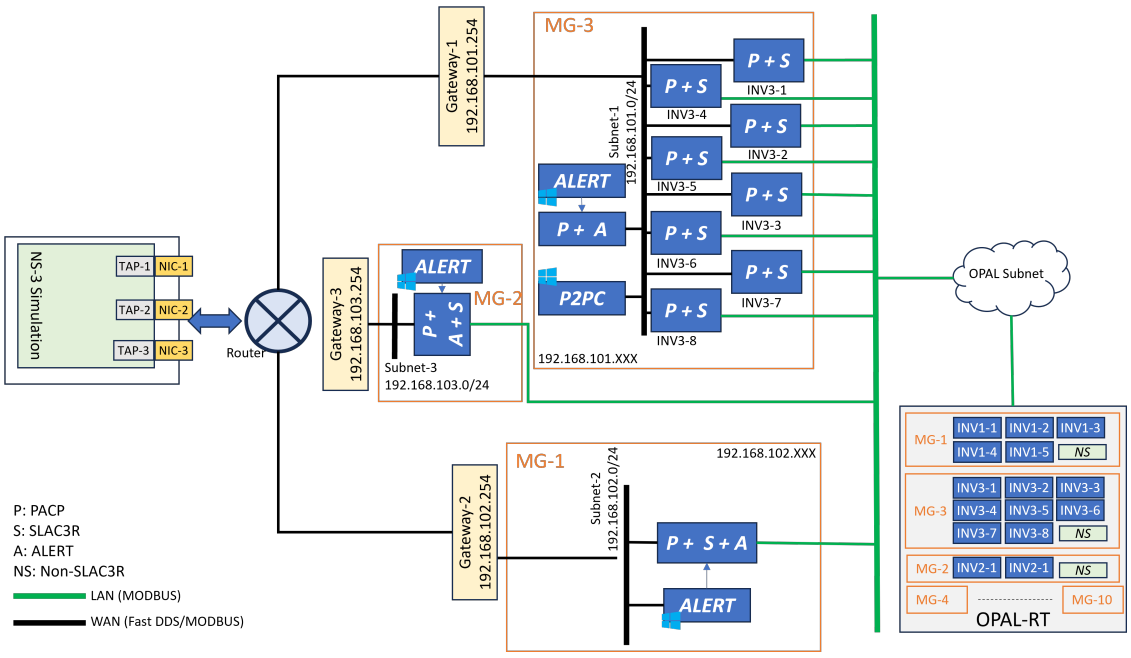


Figure 36: Implemented testbed architecture for the IEEE 123 node test system

The IEEE-123 bus networked microgrid (MG) with 3 self-sustaining MGs is simulated in HYPERSIM OPAL-RT target, where the controllers are communicating with OPAL target utilizing Modbus-based communication. At the system level, there are 3 MGs, namely MG-1, 2 and 3, and respectively, have 1 DG + 1 GFL + 4 GFMs, 1 DG + 1 GFL + 1 GFM, and 1 DG + 1 GFL + 7 GFMs. From control design's perspective, for each GFM and GFL, there are 4 measured inputs from HYPERSIM, which are active power P, reactive power Q, frequency f and voltage V, while for GFM there are 2 control outputs –active power set-point P_{set} and voltage set-point V_{set}, and for GFL, the control outputs are P_{set} and Q_{set} (reactive power set-point). DGs are only designed to receive P_{set} and Q_{set} from ALERT controller.

Without loss of generality, the testbed development includes a detailed distributed implementation architecture of SLAC3R-C for MG-3 only, while for MG-1 and 2, the distributed implementation architecture is not followed. Note that this does not make any modification in the developed SLAC3R-C algorithm. Therefore, we assigned 1 Modbus slave for MG-1 (Slave-1), 1 Modbus slave for MG-2 (Slave-2), 8 Modbus slaves (Slave-3 to 10) for each inverter (GFM+GFL) of MG-3, and 1 Modbus slave (Slave-11) for receiving DG set-points of MG-3. In the controller side, the

Table 9: VM details and its corresponding IP addresses.

VM Index	VM Name	IP Adress	Purpose	Type
VM-1	MG1-PACP-SLAC3RC-ALERT	192.168.102.1	SLAC3R-C + ALERT for MG-1	LINUX
VM-2	MG2-PACP-SLAC3RC-ALERT	192.168.103.1	SLAC3R-C + ALERT for MG-2	LINUX
VM-3	MG3-PACP-SLAC3RD-1	192.168.101.1	SLAC3R-C + ALERT for INV-67 in MG-3	LINUX
VM-4	MG3-PACP-SLAC3RD-2	192.168.101.2	SLAC3R-C + ALERT for INV-72 in MG-3	LINUX
VM-5	MG3-PACP-SLAC3RD-3	192.168.101.3	SLAC3R-C + ALERT for INV-76 in MG-3	LINUX
VM-6	MG3-PACP-SLAC3RD-4	192.168.101.4	SLAC3R-C + ALERT for INV-78 in MG-3	LINUX
VM-7	MG3-PACP-SLAC3RD-5	192.168.101.5	SLAC3R-C + ALERT for INV-80 in MG-3	LINUX
VM-8	MG3-PACP-SLAC3RD-6	192.168.101.6	SLAC3R-C + ALERT for INV-81 in MG-3	LINUX
VM-9	MG3-PACP-SLAC3RD-7	192.168.101.7	SLAC3R-C + ALERT for INV-82 in MG-3	LINUX
VM-10	MG3-PACP-SLAC3RD-8	192.168.101.10	SLAC3R-C + ALERT for INV-97 in MG-3	LINUX
VM-11	MG3-PACP-ALERT	192.168.101.253	ALERT for MG-3	LINUX
VM-12	MG1-ALERT	192.168. 253.3	ALERT for MG-1	Windows
VM-13	MG2-ALERT	192.168. 253.6	ALERT for MG-2	Windows
VM-14	MG3-ALERT	192.168. 253.1	ALERT for MG-3	Windows
VM-15	P2PCC	192.168.101.11	P2P for MG-1, MG-2 and MG-3	Windows

test-bed development involves 15 Virtual Machines (VMs) with specific purposes, as detailed in Table 9.

OPAL-RT target utilizes Modbus communication in sending/receiving data to and from control VMs, as shown in Fig. 37.

1. Modbus Slave-1 Input Registers to send P, Q, f, V of all MG-1 inverters to VM-1 and Modbus Slave-1 Holding registers to receive P_{set}, V_{set} (for GFMs), Q_{set} (for GFL) from VM-1 of all MG-1 inverters and P_{set}, Q_{set} of MG-1 DG.
2. Modbus Slave-2 Input Registers to send P, Q, f, V to VM-2 and Modbus Slave-2 Holding registers to receive P_{set}, V_{set} (for GFMs), Q_{set} (for GFL) from VM-2 of all MG-1 inverters and P_{set}, Q_{set} of MG-2 DG.
3. For MG-3, for each inverter, we designated individual slaves. Therefore, Modbus Slave-3 to 10 Input Registers are utilized to send P, Q, f, V of inverters to VM-3 to 10, respectively. While Modbus Slave-3 to 11 Holding Registers are used to receive P_{set} (for both GFMs and GFLs), V_{set} (for GFMs), Q_{set} (for GFLs and DGs) from VM-3 to 11, respectively.

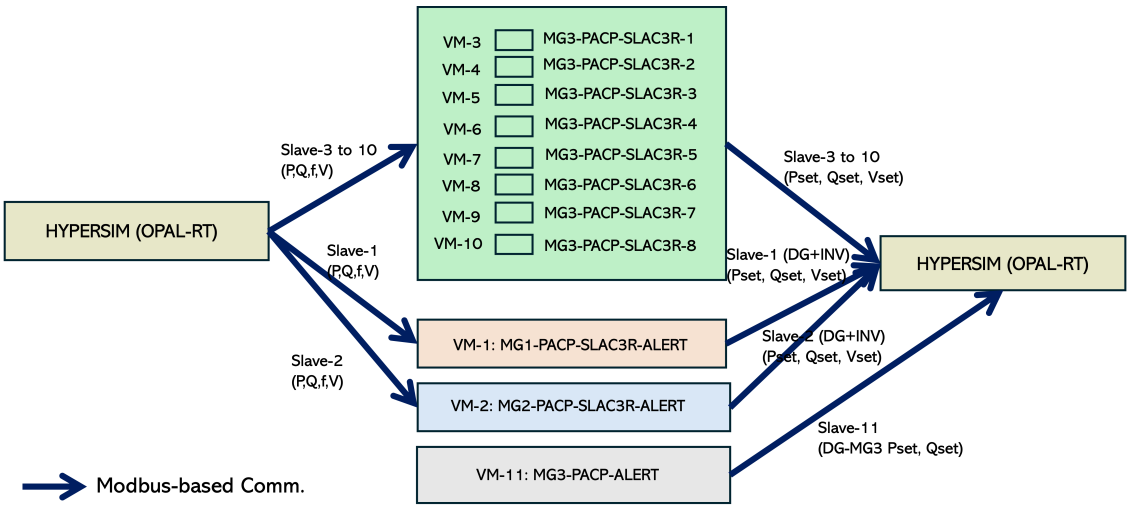


Figure 37: Modbus Communication (Comm.) in between OPAL-RT target and VMs

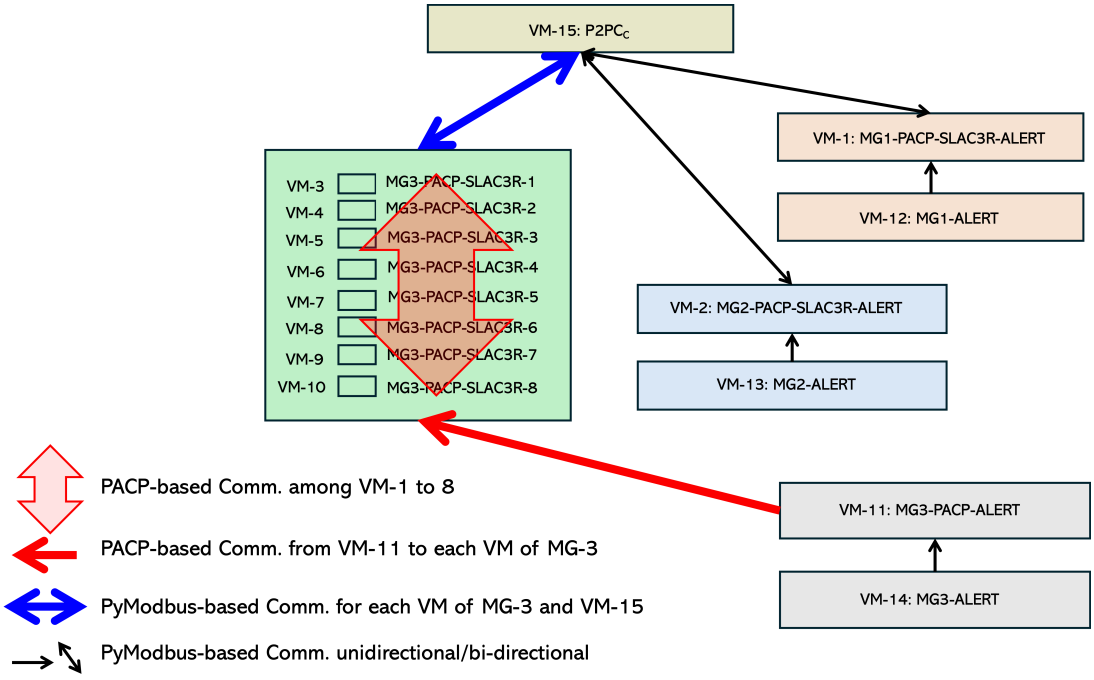


Figure 38: Communication (Comm.) among Control VMs

Next, we describe communication among control VMs.

1. As shown in Fig. 37 and mentioned in Table 9, for each inverter in MG-3, we assigned 1 VM for their SLAC3R-C-based P_{set}, V_{set} (for GFMs) and Q_set (for GFL) computation. In that process, VMs 3 to 10 communicate among themselves for consensus objective through PACP-based communication.
2. For computational simplicity, we assigned separate Windows VM for ALERT computation as detailed in Table 9; now to receive the ALERT set-points from those Windows to Linux counterpart, we utilize PyModbus-based communication for each MG.

3. As detailed distributed modeling is done for MG-3, so ALERT set-points for MG-3 are sent from VM-11 to each MG-3 VM (VM-3 to 10) via PACP-based communication. This kind of communication is not needed for MG-1 and 2.
4. Finally, the P2P control is computed in VM-15 –this necessitates sending P, Q and receiving P2P-based voltage set-points for VM-1 (for MG-1), VM-2 (for MG-2) and each of MG-3 VMs (VM-3 to 10). All these communications are achieved through PyModbus-based communication, as shown in Fig. 38.

4.2 Power System Real-Time Simulation

4.2.1 Simulation Automation Tool

Large-scale simulation models, such as those comprising thousands of interconnected components, require meticulous configuration of numerous parameters to accommodate diverse experimental scenarios. In particular, DER parameters often need to be adjusted to reflect varying operational conditions and research objectives. Manually updating these parameters across such expansive models is not only labor-intensive but also susceptible to human error, which can compromise the accuracy and consistency of simulation results. Furthermore, executing multiple scenarios within these complex environments demands significant time and effort. To overcome these challenges, a Python-based automation framework was developed to streamline both DER parameterization and scenario execution. This solution enables users to efficiently configure model parameters and run simulations with minimal manual intervention, thereby enhancing reproducibility, scalability, and overall workflow efficiency in large-scale power system studies.

This section introduces an automated tool designed to enhance the efficiency and accuracy of real-time simulation workflows. The tool integrates built-in model parameterization capabilities, allowing users to easily configure and customize a wide range of testbed scenarios without manual intervention. By automating the setup process, it significantly reduces the time and effort required to prepare simulations, while ensuring consistency and reproducibility across different testing environments. This functionality is particularly valuable for researchers and engineers who need to rapidly iterate through various configurations to evaluate system performance under diverse conditions. The methodology and execution of the automated tool is described in the following sections.

4.2.1.1 Automated model parameter JSON creation

The automated creation of model parameter JSON files involves dynamically generating structured JSON configurations that define key model parameters from the input CSV files, eliminating the need for manual input and reducing the risk of human error. By leveraging predefined templates and scenario-specific inputs, the system ensures consistency across test runs while enabling rapid customization for different simulation conditions. This automation accelerates the deployment of real-time simulations and enhances reproducibility and scalability. Figure 39 represents the process of updating the DER model parameter in this tool.

CSV file

Figure 40 presents the template containing all relevant model parameters for DERs in the IEEE 123 bus model with added GFM and GFL inverters. Users are required to populate this template

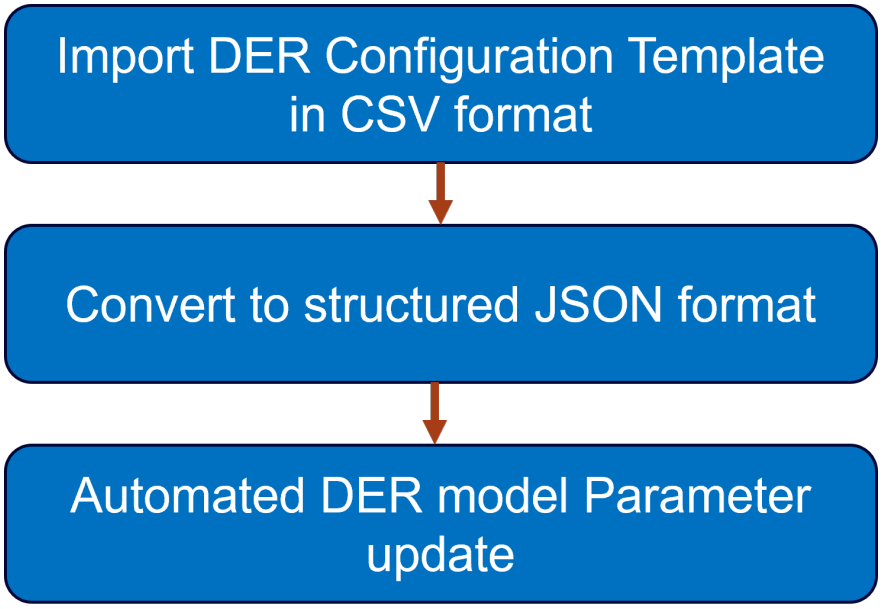


Figure 39: Flowchart representing automated DER model parameter update

with the appropriate DER model parameters in the specified format to enable the creation and updating of DER configurations within the HYPERSIM simulation environment.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	NAME	MODEL	PHASES	SBASE	VBASE	FBASE	XFMR_V1	XFMR_V2	XFMR_W1	XFMR_W2	XFMR_RS	XFMR_LS	XFMR_RM	XFMR_LM	LCL_L1	LCL_L2	LCL_R1	LCL_R2	LCL_C1	LCL_ESR1	STATOR_W	STABILITY_R
2	DG50	diesel_dg	3	600000	4160	60	-1	-1	NA	NA	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	D	20
3	DG300	diesel_dg	3	600000	4160	60	-1	-1	NA	NA	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	D	20
4	DG100	diesel_dg	3	600000	4160	60	-1	-1	NA	NA	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	D	20
5	INV42	inv_3ph_g	3	80000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
6	INV76	inv_3ph_g	3	90000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
7	INV101	inv_3ph_g	3	60000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
8	INV51	inv_3ph_g	3	140000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
9	INV80	inv_3ph_g	3	80000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
10	INV105	inv_3ph_g	3	190000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
11	INV135	inv_3ph_g	3	150000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
12	INV40	inv_3ph_g	3	180000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
13	INV48	inv_3ph_g	3	200000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
14	INV67	inv_3ph_g	3	90000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
15	INV97	inv_3ph_g	3	140000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
16	INV72	inv_3ph_g	3	200000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
17	INV78	inv_3ph_g	3	110000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
18	INV81	inv_3ph_g	3	120000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1
19	INV82	inv_3ph_g	3	170000	480	60	4160	480	D	Y	0.012	0.0562	100	100	0.05	0.05	0.005	0.005	0.05	0.0001	NA	-1

Figure 40: Sample of CSV file with DER model parameters

JSON template files

A Python-based script, der_instantiation.py, was developed to automate the conversion of DER data from a CSV file into JSON format. This transformation is a critical step in preparing the DER model parameters for instantiation within the HYPERSIM simulation environment. By standardizing the input format, the script ensures seamless integration and accurate representation of DER configurations in the IEEE 123-bus test feeder model. By automating the conversion of DER parameters and standardizing the input format, the workflow significantly improves reproducibility and scalability. This allows users to efficiently replicate the modeling process across various test feeders and adapt it to evolving simulation requirements with minimal manual intervention. Figure 41 shows sample contents of a DER parameter JSON file. Each DER type used in the HYPERSIM models has a template JSON file containing default model parameters reflecting the default DER size (such as 4.0 MVA for the diesel generator). For each DER entry in the CSV

file, the appropriate template JSON is copied and overwritten with the DER instance specific parameters according to the row contents in the CSV file.


```
{
  "INV40": {
    "BASE": {
      "SBASE": {
        "A": "180000"
      },
      "VBASE": {
        "A": "480"
      },
      "FBASE": {
        "A": "60"
      }
    },
    "MEASUREMENT": {
      "VD_FIL1": {
        "Tc": "0.01"
      },
      "VQ_FIL1": {
        "Tc": "0.01"
      },
      "ID_FIL1": {
        "Tc": "0.0004"
      },
      "IQ_FIL1": {
        "Tc": "0.0004"
      },
      "P_PU1": {
        "Tc": "0.01"
      },
      "Q_PU1": {
        "Tc": "0.01"
      }
    },
    "IMPEDANCE": {
      "XFMR1": {
        "ConnexPrim": "Delta lead",
        "R1": "1.7306e+00,1.7306e+00,1.7306e+00",
        "L1": "2.1499e-02,2.1499e-02,2.1499e-02",
        "VnomPrim": "4.1600e+00,4.1600e+00,4.1600e+00",
```


Figure 41: Sample of JSON file with DER model parameters

4.2.1.2 Automated scenario execution

To facilitate experimentation across multiple scenarios, an automation script is also employed for scenario execution. This script orchestrates the execution of model updates and simulation runs, enabling efficient and repeatable testing of scenarios in the test feeder model. The key features in the automation in scenario execution are as follows:

- **Scalability**: Handles thousands of model components efficiently.
- **Flexibility**: Easily adapts to new scenarios or parameter sets.
- **Error minimization**: Minimizes manual input, reducing human error.
- **Speed**: Automates repetitive tasks to accelerate simulation process.

Figure 42 shows the flowchart that outlines a structured, step-by-step process designed to automate the configuration and execution of scenarios in HYPERSIM using the simulation automation tool. The process begins with the user sets and defines the input files and information to an input .toml file for the automation tool. The model information, the location of the json files for DER model update, the structured JSON for updating the model parameters for scenario setup

and default initial conditions are provided by the user. Figure 43 presents a structured template containing key headings that define the model parameters subject to modification across various scenarios within the test system. The structured JSON for scenario setup distinctively includes the model parameters for switches, DER parameters, tripping of DERs, enabling/disabling the control parameters of different DERs, datalogging, etc. The user selects one or more scenarios to run, each with distinct DER configurations or operating conditions.

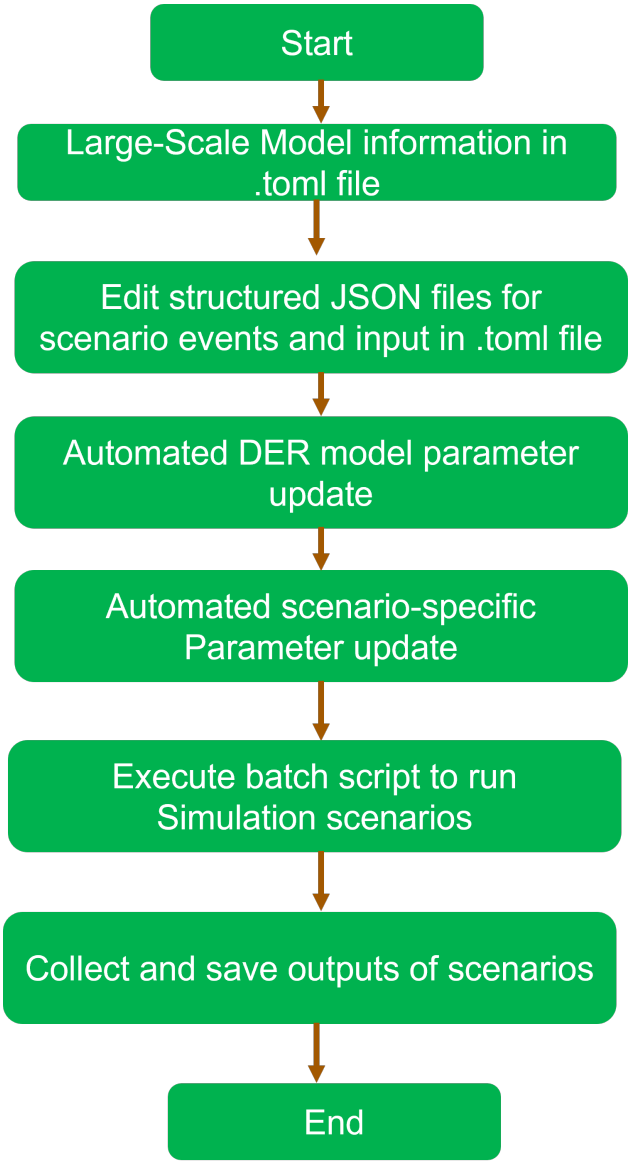


Figure 42: Flowchart representing the automation of running scenarios using the automation tool

A batch script is used to run the scenarios and DER model update. The tool loops through each DER or model component and applies scenario-specific parameters. This eliminates manual editing and ensures consistency. Users define scenarios in a structured format—each scenario might represent a different DER penetration level, fault condition, or control strategy. The tool can run simulations for multiple scenarios and can interface with HYPERSIM's API or command-line utilities to trigger runs. After each run, the tool gathers output data (e.g., voltage profiles,

```
{
  "MG POI Settings": {
    "SW_TR18TO135": {
      "K": "1"
    },
    "T_SW_TR18TO135": {
      "Td": "0"
    },
    "SW_TR151TO300": {
      "K": "1"
    },
    "T_SW_TR151TO300": {
      "Td": "230"
    },
    "SW_TR97TO197": {
      "K": "1"
    },
    "T_SW_TR97TO197": {
      "Td": "230"
    },
    "SW_TR54TO94": {
      "K": "1"
    },
    "T_SW_TR54TO94": {
      "Td": "0"
    },
    "SW_TR60TO160": {
      "K": "1"
    },
    "T_SW_TR60TO160": {
      "Td": "200"
    }
  },
  "IBR PDISP Settings": {
    "INV76.PSET_0": {
      "A": "1"
    },
    "INV76.PSET_CONS": {
      "K": "0"
    },
```


Figure 43: Sample of JSON file with model parameters to set different scenario events

frequency deviations, DER responses) in form of datalogging, which is stored for further data visualization. This tool can edit and run multiple scenarios in much accelerated fashion and reduces the risks of misconfigurations of models in HYPERSIM. The tool is perfect for large-scale models with thousands of DERs and components and enables experimentation with less efforts and time.

4.2.2 DNP3 Interface for External DER Aggregators

4.2.2.1 Motivation

Numerous testbeds have been developed to simulate and evaluate distribution systems with high levels of RE integration, often focusing on areas such as grid stability, inverter behavior, and energy management under varying load and generation conditions [26]. Many of these testbeds also explore the use of communication protocols, such as Modbus, DNP3, IEC 61850, and IEEE 2030.5, to facilitate data exchange between physical devices and virtual components in the system. To ensure interoperability and interconnection between DERs and existing grid infrastructure, several testbeds have been studied and validated using communication protocols and standards [27]. Among these protocols, DNP3 uses a peer-to-peer communication model and publish-subscribe capabilities, enabling e_cient, scalable data transfer and supporting systems with a large number of DERs [28, 29]. Some testbeds demonstrate DER coordination through DNP3 or related protocols, while others focus on real-time automation, cybersecurity impact assessments, or market participation mechanisms. However, the majority of these platforms either center around individual DER devices or examine general communication frameworks, without explicitly modeling the role of DER aggregators as a distinct, functional layer within the system. While DER management systems (DERMS) are commonly used in testbed development to manage grid constraints and prevent violations, DERMS differ from DER aggregators because they cannot participate in the wholesale energy market [30]. DER aggregators are emerging as key players in the evolving energy landscape, enabling groups of DERs to function as a single controllable entity. Aggregators enhance system operations through real-time monitoring, localized exibility, and bidirectional communication with utilities and grid operators [31]. Yet, few existing testbeds incorporate a dedicated aggregator layer or investigate how communication protocols can be adapted to support secure, responsive communication between DERs, aggregators, and control centers. This gap in current research motivates the development of the proposed testbed. The objective is to create a defined aggregator-level within a DNP3-based communication network that enables two-way, real-time communication between the grid, aggregator, and control center. By bridging the divide between current testbed implementations and the operational realities of DER aggregation, this research establishes a foundation for secure, scalable, and ready communication architectures in modern distribution systems.

4.2.2.2 Two-Way Communication for DER Aggregators

To create a DER aggregator within a testbed, an agent will perform the two-way communication, allowing the grid and control center to talk to one another through the agent, similarly to how an aggregator would allow communication between the two. Each microgrid within the IEEE 123 bus system will have its own aggregator that will also connect to a central control center as shown in Fig. 44. The DERs modeled in the IEEE 123 HYPERSIM model sends measurements through the aggregator-level to the control center, while the control center will send controls back to the HYPERSIM model represented in Fig. 45. The aggregator-level provides a secure communication

between the grid and control center that is not otherwise present within middleware because of the increased visibility of data packet transfer.

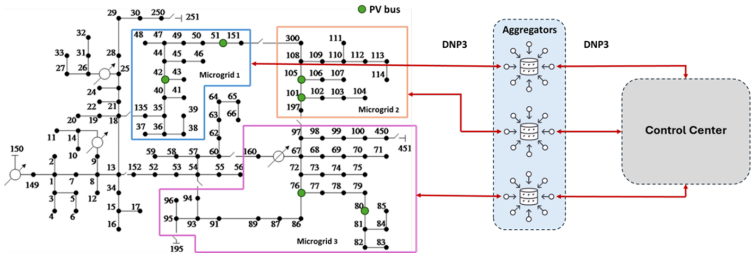


Figure 44: IEEE 123 Bus System with Aggregator-Level One-Line Diagram.

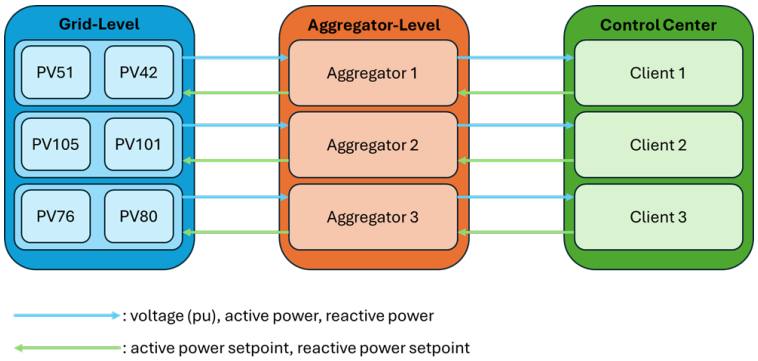


Figure 45: Two-Way Communication Flow.

4.2.2.3 Aggregator Model Test System

The proposed research introduces an aggregator-level architecture for DER-integrated distribution systems, and the cyber physical testbed developed during Capstone 1 provides a platform to implement and validate this concept. The testbed can be used to simulate bidirectional communication between DERs, aggregators, and control centers, mimicking a closed-loop architecture. The testbed's flexible infrastructure supports manual control signals, real-time measurements, and data flow validation. The testbed leverages HYPERSIM to perform real-time cyber-physical experiments. HYPERSIM is a power system simulator created by OPAL-RT [32]. HYPERSIM is compatible with DNP3 through I/O interfaces, which define the client and servers [33]. DNP3 uses TCP protocol, so a Python script can be connected to send and receive data. As shown in Fig. 46, the grid is modeled in HYPERSIM and the aggregator and control center are Python scripts. The HYPERSIM setup communicates through DNP3 with the aggregator master, aggregator outstation, and control center master codes adapted from pydnp3 [34]. The grid communicates with the aggregator-level to send system measurements. The control center also communicates with the aggregator-level, sending back control decisions or adjustments. Each of these components acts as a DNP3-enabled device; however, the two DNP3 connections are between the grid and aggregator master and aggregator outstation and control center. Within the aggregator-level itself, values from the grid and control center are not passed using DNP3 between its internal modules. Instead, data is shared internally through a database manager created through redis.

The aggregator-level can be a redis publisher or subscriber based on the function and where data is being received from and sent to. This is because the aggregator is treated as a single intelligent device that can both receive data from the grid and send data to the control center and vice versa. This design allows the aggregator-level to act as a central processing point, capable of both monitoring and controlling information flow in both directions.

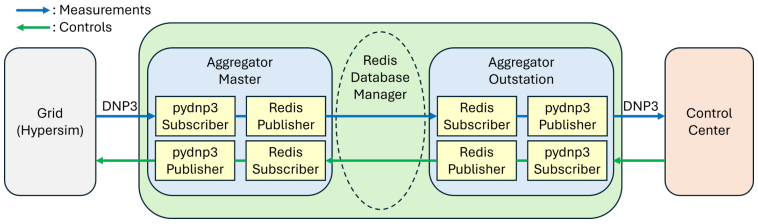


Figure 46: Closed-Loop Testing Structure.

The closed–loop two-way communication works as a forward flow and reverse flow. Once the forward flow is completed, the control center reviews the incoming measurements and sends controls back through the reverse flow to adjust the devices to work optimally. The process for both flows is as follows.

1. Forward Flow:
   a. The grid (Hypersim) sends values to the aggregator-level, which is received by the aggregator master through a DNP3 binding.
   b. The aggregator master publishes the data to a redis database manager that the aggregator outstation subscribes to.
   c. The aggregator outstation sends the data to the control center through their DNP3 binding.
2. Reverse Flow:
   a. The control center publishes values through a DNP3 binding to the aggregator outstation.
   b. The aggregator outstation publishes the controls to the redis database manager, which the aggregator master subscribes to.
   c. The aggregator master sends those controls back to the grid through a DNP3 binding.

In the current implementation, three VMs are utilized. The Hypersim model is in one VM, the aggregator–level is another VM, and the control center is in its own VM. The Hypersim model sends a per unit voltage from the PVs as analog inputs, which is sent through the forward flow. Once the control center receives the voltages, it sends back reactive power analog outputs to ensure the PVs are working within ANSI limits through volt–var control (VVC). Because PVs are an intermittent resource, VVC helps to stabilize grid voltage levels. The DNP3 analog inputs and outputs along with their function are listed in Table 10 and Table 11. The aggregators need to have unique IPs, local addresses, and ports for each microgrid connection. The I/O interface for the aggregator masters and outstations is outlined in Table 12.

4.2.2.4 Aggregator Test Results

The following test is an example of the two–way communication between the grid, aggregators, and control center and focuses on the VVC. Two tests were done for VVC. The first test assumes that the base voltage is 1.05 pu, which is the stable voltage of the PVs in the Hypersim model. ANSI limits are $+/-5\%$ of the nominal voltage, so the upper and lower limits for the first test is 1.1 pu and 1.0 pu respectively. The deadband is between 1.03–1.07 pu. Reactive power is injected when the voltage is less than 1.03 pu, with full reactive power injection starting at 1.0 pu. Full reactive power absorption starts at 1.1 pu, and reactive power begins to absorb at more than

Table 10: HYPERSIM DNP3 Analog Input Sensor Summary

Inverter Bus	Signal	Server	Analog Input Point	Purpose
51	Inverter terminal voltage	Microgrid 1	0	PV voltage feedback
42	Inverter terminal voltage	Microgrid 1	1	PV voltage feedback
105	Inverter terminal voltage	Microgrid 2	0	PV voltage feedback
101	Inverter terminal voltage	Microgrid 2	1	PV voltage feedback
76	Inverter terminal voltage	Microgrid 3	0	PV voltage feedback
80	Inverter terminal voltage	Microgrid 3	1	PV voltage feedback

Table 11: HYPERSIM DNP3 Analog Output Sensor Summary

Inverter Bus	Signal	Analog Output Point	Purpose	
51	Reactive Power Setpoint	Microgrid 1	0	Reactive Power Correction
42	Reactive Power Setpoint	Microgrid 1	1	Reactive Power Correction
105	Reactive Power Setpoint	Microgrid 2	0	Reactive Power Correction
101	Reactive Power Setpoint	Microgrid 2	1	Reactive Power Correction
76	Reactive Power Setpoint	Microgrid 3	0	Reactive Power Correction
80	Reactive Power Setpoint	Microgrid 3	1	Reactive Power Correction

Table 12: PYDNP3 Aggregator Connection Summary

Device	IP Connection	DNP3 Address	Port
Master 1	X.X.X.117	101	20001
Master 2	X.X.X.118	102	20002
Master 3	X.X.X.119	103	20003
Outstation 1	0.0.0.0	104	20004
Outstation 2	0.0.0.0	105	20005
Outstation 3	0.0.0.0	106	20006

1.07 pu. In the second test, the nominal voltage is set at 1.0 pu, which is the nominal voltage of the overall distribution system. The lower limit is 0.95 and the upper limit was 1.05. The deadband is between 0.98 and 1.02 pu. The reactive power injection is between 0.95 and 0.98, with full reactive power injection starting at 0.95 pu. The reactive power absorption is between 1.02 and 1.05 pu, with full reactive power absorption starting at 1.05 pu. VVC aims to keep the voltage within the deadband limits. When the nominal voltage is set to 1.05 pu, reactive power is not injected or absorbed by the inverters because the voltage of the inverters stays within the deadband limits, as shown in Fig. 47, Fig. 48, and Fig. 49. However, when the nominal voltage is set to 1.0 pu, the control center sends corresponding reactive power for injection or absorption in order to get the voltages of the inverters to be within the deadband limits of 0.98–1.02 pu as shown in Fig. 50, Fig. 51, and Fig. 52.

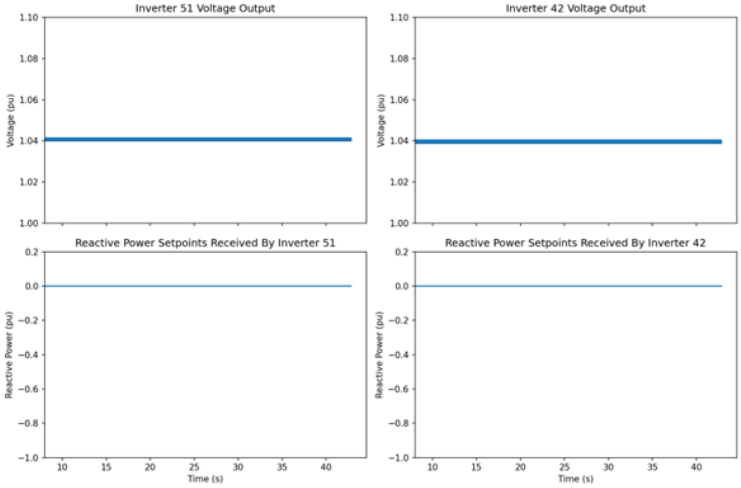


Figure 47: Microgrid 1 VVC 1.05 pu Nominal Voltage.

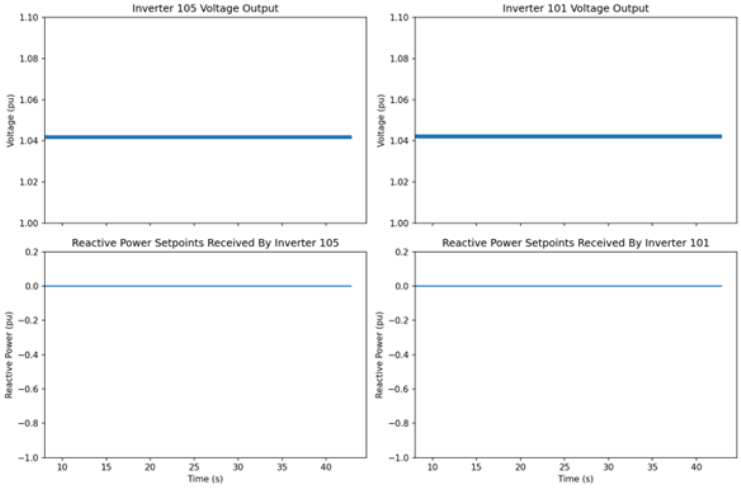


Figure 48: Microgrid 2 VVC 1.05 pu Nominal Voltage.

When the nominal voltage is set to 1.0 pu, the VVC activates. The control center sends reactive power setpoints to be absorbed by the grid to try to lower the inverter output voltage

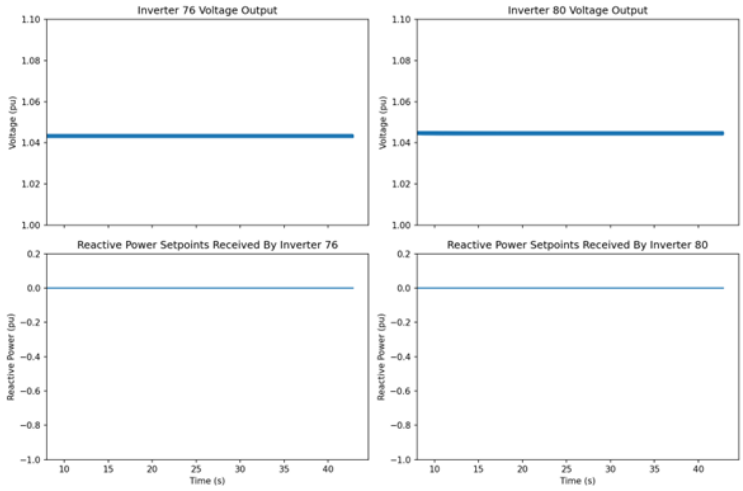


Figure 49: Microgrid 3 VVC 1.05 pu Nominal Voltage.

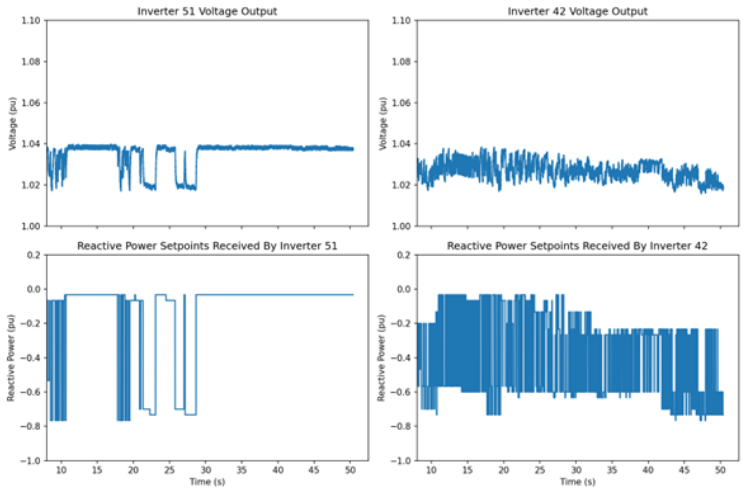


Figure 50: Microgrid 1 VVC 1.00 pu Nominal Voltage.

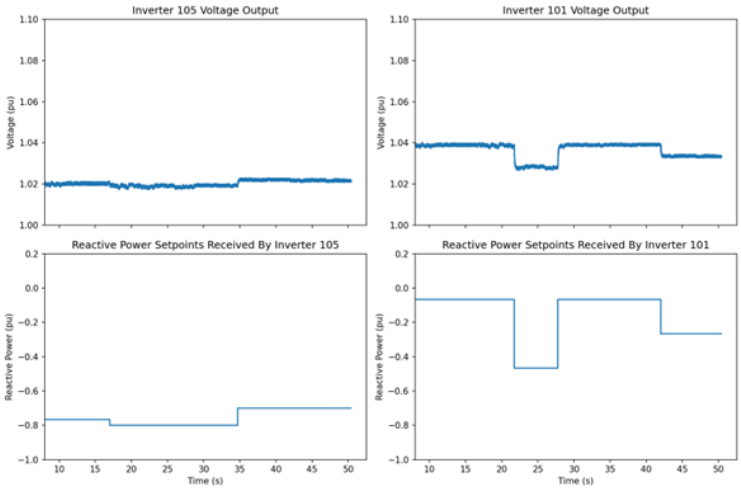


Figure 51: Microgrid 2 VVC 1.00 pu Nominal Voltage.

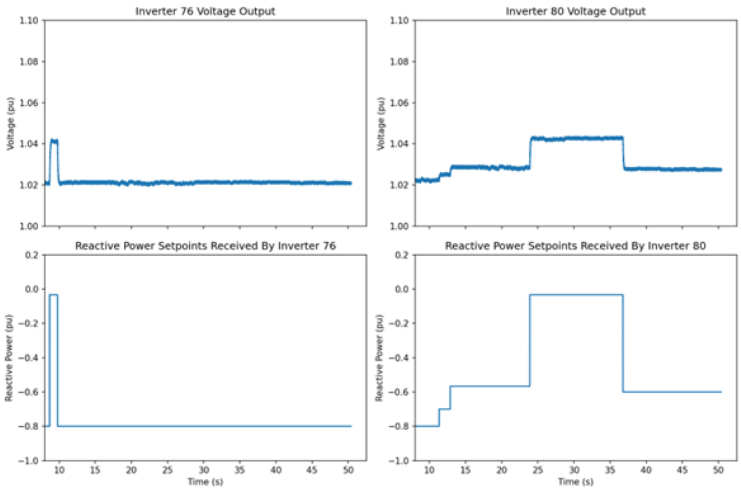


Figure 52: Microgrid 3 VVC 1.00 pu Nominal Voltage.

to be within the deadband of 0.98–1.02 pu. The nominal voltage is set to 1.0 pu because that is the nominal voltage across the whole distribution system and allows for more devices to be added for VVC. It is acknowledged that the control performance is sub-optimal in Microgrid 1 as indicated in Figure 50 and requires additional tuning or stabilizing mechanisms such as set point rate limiting or low pass filtering of local measurements. However, the important outcome was that a remote control center application was able to connect to and control a fleet of DERs based on local information using the aggregator model discussed in this section.

4.2.2.5 DER Aggregator Modeling Closing Remarks

This section shows how to implement an aggregator within a testbed using DNP3 as the communication protocol. The aggregator adds another component in a distribution system within a testbed, and it also allows for more visibility to packets being transmitted. The two–way communication scheme sends measurements, namely inverter voltage, from the grid to the aggregator to the control center, then the control center sends controls, reactive power from VVC, to the aggregator to the grid in order to continue the voltage stability. Future work includes changing the communication protocol to match the needs of other testing. This can also be extended to the use of cyberattacks by having the aggregators screen packets before it reaches the destination. The aggregators add an extra layer of visibility, so detection methods can be used by checking packets and finding anomalies; therefore, the control center can implement mitigation methods. Aggregators also allow for small–scale distributed energy resources to be used in the energy market by combining multiple units to reach a specified threshold for successful operation. Testing can be done to emulate an aggregator in the real–world, as described above, to evaluate if a community's distributed solar generation can be leveraged for distribution system operations.

4.3 Communication Network Simulation

The experiment architecture is composed of virtual LANs for each microgrid and a WAN that provides peer–to–peer communication between the microgrids as described in Section 4.1. In order to simulate communications between the microgrid LANs the team created a simple, multi-route model of a WAN in NS3 [10]. The model is illustrated in Fig. 53. The model provided a way to introduce disturbances in the cyber layer that would be difficult to emulate within the Cybernet cloud computing infrastructure.

We chose NS3 for its precise packet-level control that synchronizes network and physical processes. The NS3 model provides real-time data processing to track attack propagation through network layers, particularly for systems with strict timing requirements like DNP3 in smart grids and IoT standards. The NS3 model was integrated into the cyber–physical testbed by installing a NATIG container on a Cybernet virtual machine and providing the virtual machine with interfaces on each of the microgrid networks. The NATIG container contains an NS3 installation as well as tools that simplify the communication network modeling process [35]. Its customizability simulates Fig. 54 provides a detailed diagram for the NS3 WAN model implementation showing how the data flows between the VMs and the NS3 network. The NS3 WAN model provides the overall cyber–physical testbed with:

- Virtual network interfaces (TAP devices) that bridge external virtual machines with the NS3–simulated network.
- Simulated WAN routes with configurable latency, bandwidth, and packet loss parameters.
- Multiple communication nodes representing control centers, substations, and field devices.

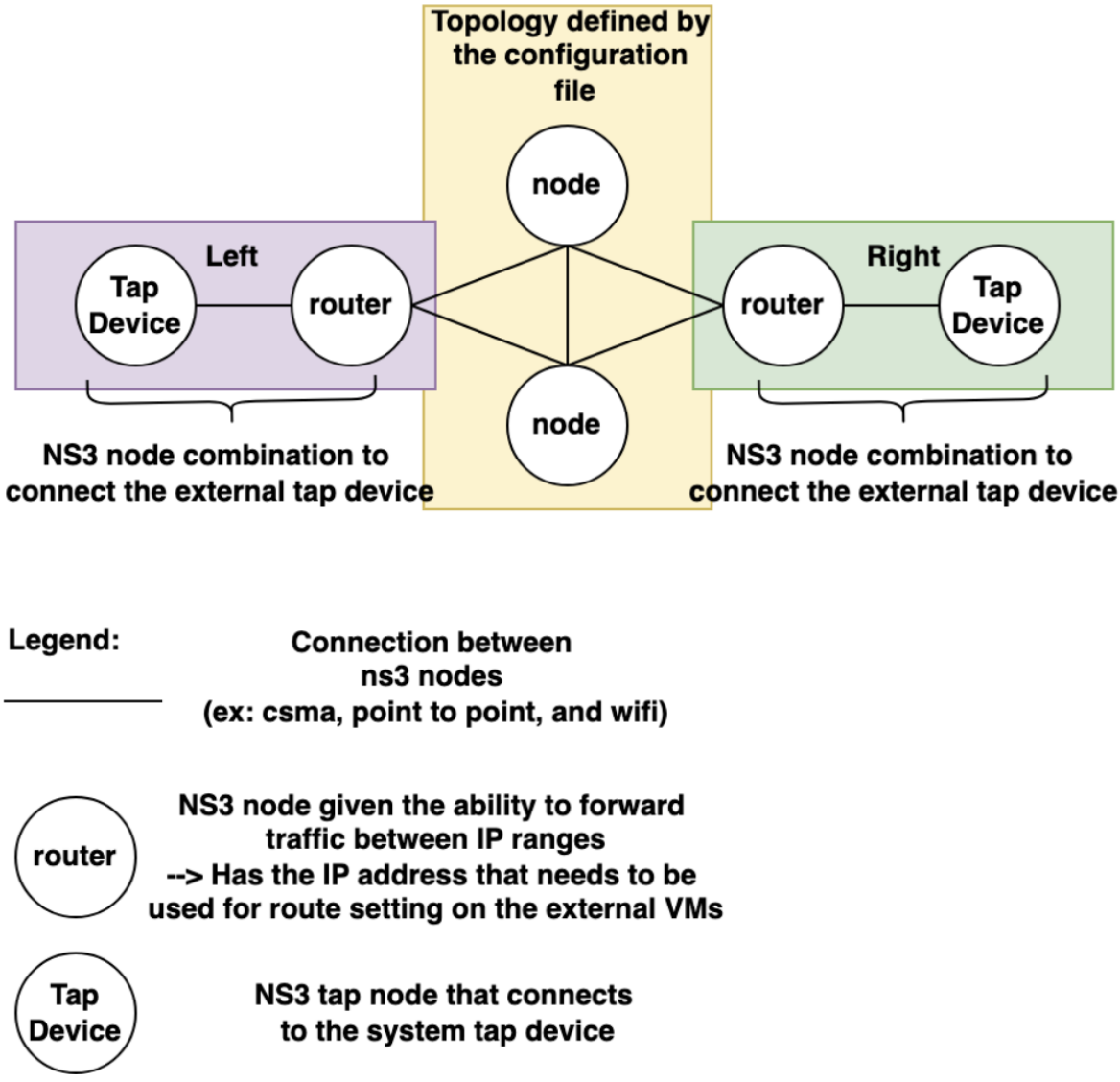


Figure 53: Multi-route WAN model implemented in NS3.

- Bi-directional data flows carrying SCADA protocols between external controller devices hosted on the disparated cyber–physical testbed LANs.
- The ability to simulate resource-constrained IoT environments, including rapid packet generation and protocol vulnerabilities.

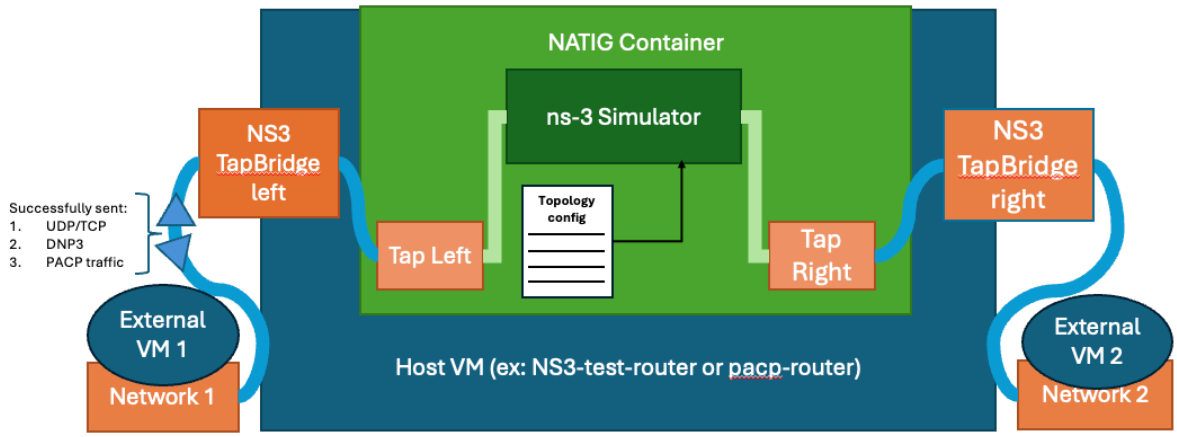


Figure 54: Multi-route WAN model implemented in NS3.

Inside the NS3 setup, a user can specify the attack and the network topology using a config.json file. For example, this allows testbed users to execute a distributed denial of service (DDoS) attack targeting DNP3 traffic between control centers and substations. The model would generate thousands of packets to exhaust server resources while synchronizing with HYPERSIM to observe physical impacts. The NS3 simulation measures protocol-specific metrics like DNP3 command latency. When with the HYPERSIM simulation data collection, experiments with the cyber–physical testbed developed in this work yields valuable data for developing targeted defenses, such as protocol-aware intrusion detection systems that recognize DNP3-specific attack patterns before physical systems are compromised—demonstrating the practical value of high-fidelity cyber-physical attack simulation.

5.0 IEEE 123 Node Distribution System Use Case Studies

To assess how the multi-level controller hierarchy improves the resilience of the IEEE 123 Node Test Feeder distribution system to physical disturbances (such as natural weather events) and cyberattacks, several scenarios were developed to test the dynamic response of the controls. The scenarios feature multiple disturbing events, which reflects how an unexpected coincidence of distribution system stressors can result in greater degradation of system performance. Additionally, as cyberattacker techniques become more advanced, distribution systems may expect to see attackers attempting and potentially succeeding at synchronizing the compromise of resources with natural disasters in order to crippled an already weakened power system. Table 13 gives a brief summary of the scenarios that were developed for the IEEE 123 Node Test Feeder model discussed in Section 2.2. The scenario numbering is not consecutive because the project developed preliminary narratives for a number of different scenarios. Due to time constraints, only the subset of scenarios presented in Table 13 were completed. The team preserved the original numbering scheme to be usable for future work, but did not include the details of the other scenarios in Table 13 to avoid any confusion regarding the actual scenario testing scope that was accomplished.

Table 13: Brief Overview of Scenarios Used for Resilience Experiments

Scenario Number	Brief Scenario Description
1D	An unplanned islanding event separates the three microgrids as a group. A series of cyberattacks lead to separation of the microgrids and Microgrid 3 loses a generation resource. Underfrequency load shedding protection is deployed to avoid excessive frequency dips.
1E	An unplanned islanding event separates the three microgrids as a group. A series of cyberattacks lead to separation of the microgrids and Microgrid 3 loses a generation resource. Underfrequency load shedding protection is deployed to avoid excessive frequency dips. The initial DER dispatch is modified from Scenario 1D.
7A	An unplanned islanding event separates the three microgrids as a group. A capacitor bank located in Microgrid 3 experiences a malfunction and is disconnected. Shortly after, a cyberattack disconnects the diesel generator in Microgrid 3, increasing the deficit of reactive power support.

Each scenario formed the basis for an experiment involving multiple trials. The goal of each experiment was to understand how the system behavior changes as one or more mitigation strategies are integrated into the controller hierarchy. Each trial incorporated a different combination of mitigation strategies including the trials where no mitigation strategies were active,

which provided the baseline system performance. A trial numbering scheme was devised by enumerating all combinations of mitigation strategies including the infeasible combinations for completeness. The feasible enumerations are provided in Table 14.

Table 14: Experiment Trial Numbering Scheme

Trial #	SLAC3R-L	SLAC3R-C	PACP	ALERT	P2P
0	No	No	No	No	No
1	Yes	No	No	No	No
4	No	No	Yes	No	No
5	Yes	No	Yes	No	No
6	No	Yes	Yes	No	No
7	Yes	Yes	Yes	No	No
12	No	No	Yes	Yes	No
13	Yes	No	Yes	Yes	No
14	No	Yes	Yes	Yes	No
15	Yes	Yes	Yes	Yes	No
20	No	No	Yes	No	Yes
21	Yes	No	Yes	No	Yes
22	No	Yes	Yes	No	Yes
23	Yes	Yes	Yes	No	Yes
28	No	No	Yes	Yes	Yes
29	Yes	No	Yes	Yes	Yes
30	No	Yes	Yes	Yes	Yes
31	Yes	Yes	Yes	Yes	Yes

The following subsections will provide a description of each scenario with the specific sequence of events, time domain simulation results, and an assessment of the how the controller hierarchy improved the distribution system response to the disturbing events.

5.1 Scenario 1D

5.1.1 Description and Sequence of Events

In this scenario, a combination of physical and cyber layer events combine to isolate and weaken the microgrids in the IEEE 123 node feeder model. The cyber layer events are emulated by simulating the end effects in the power system model (such as the opening of a circuit breaker due to a malicious remote breaker command). The scenario is representative of a well coordinated cyberattack, where adversaries gained persistence in the microgrid control network and identified the critical communication links that allow the microgrid centralized controller to send remote breaker operate commands to a select number of circuit breakers giving them control over microgrid connectivity and availability of distributed generation. The adversaries use this

control to coordinate their attacks with a hypothetical local weather event that causes a distribution system fault leading to the islanding of the three microgrids. The goal of the adversaries are to temporarily create a generation deficiency in MG3 that is large enough to induce underfrequency loadshedding, thereby causing a disruption to the electricity supply to potentially critical infrastructure that is relying upon MG3 for energy supply. The adversaries first open the POI circuit breakers to isolate the microgrids to prevent pooling of excess generation and storage reserves. Then they execute a remote breaker open command in the Microgrid 3 control network to disconnect Inverter 76, which represents a PV system. The specific sequence and timing of events is summarized in Table 15.

Table 15: Scenario 1D Sequence of Events

Simulation Time (s)	Breaker Event	DER Dispatch Event
0^-	Node 60 to Node 160 breaker is **Open** Node 18 to 135 breaker is **Closed** Node 54 to 94 breaker is **Open** Node 151 to 300 breaker is **Closed** Node 97 to 197 breaker is **Closed**	All GFM INV $V_{SET,0}$ = 1.05 pu All DG $V_{SET,0}$ = 1.015 pu All GFL INV $Q_{SET,0}$ = 0.6 pu MG1 DER $P_{SET,0}$ = 0.5 pu MG2 DER $P_{SET,0}$ = 0.5 pu MG3 Diesel Generator $P_{SET,0}$ = 1.0 pu MG3 GFL Inverter $P_{SET,0}$ = 1.0 pu MG3 GFM Inverter $P_{SET,0}$ = -0.6 pu
200	Node 60 to Node 160 breaker **Opens**	
230	Node 151 to Node 300 breaker **Opens** Node 97 to 197 breaker **Opens**	
280	Inverter 76 PCC breaker **Opens**	

Under frequency load shedding relays have been integrated into the model and will activate if the system frequency should remain below the safety threshold for a pre-defined time out duration. Microgrids will typically have multiple tiers of load priority and the lowest tier loads will be shed in the event of under frequency or under voltage conditions to ensure that the microgrid can remain stable in the event of a sudden, unexpected generation deficit. Table 16 gives the under frequency load shedding parameters.

5.1.2 Simulation Results and Resilience Assessment

As discussed in Section 4.0, a centralized architecture is deployed for the controls in MG-1 and MG-2. However, for MG-3, a detailed implementation is chosen to implement controls, that is, there is a unique VM for each inverter where the SLAC3R-C controls were implemented and a separate VM for ALERT controls is chosen to mimic centralized implementation of ALERT. A pymodbus communication is chosen to transfer the setpoint data between ALERT and SLAC3R-C VMs. PACP is used to facilitate the setpoint exchange between each inverter VM for SLAC3R-C functioning. More details around this implemented architecture were provided in Table 9.

Table 16: UFLS Parameters

UFLS Level	Threshold (Hz)	Time Delay (s)	Action
1	59.5	2.0 s	Open Node 76 to Node 86 breaker, 260 kW
2	59.5	3.0 s	Open Node 67 to Node 68 breaker, 120 kW

The event sequence in Table 15 is used to create multiple system disturbances. These perturbations drive the frequency away from nominal, and with only droop control and no mitigation in place, the frequency does not recover (see Fig. 55(a)). Any subsequent events would therefore potentially destabilize the system. The blue spike in the frequency traces is a numerical artifact that will be corrected in future updates to the measurement model.

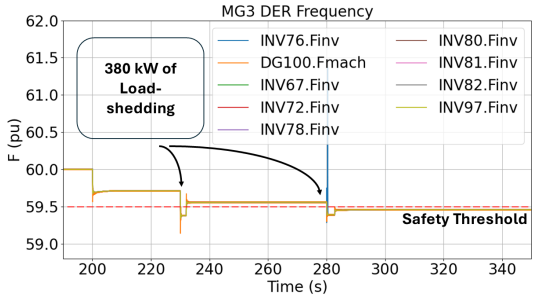
We then enabled mitigation strategies hierarchically, starting with decentralized autonomous controls (DAC, also referred to as SLAC3R-L). These DACs are embedded directly in the HYPERSIM model and use only local measurements, requiring no communications. The SLAC3R-L allowable frequency band is [59.5, 60.5] Hz, which normalizes to [0.991, 1.008]. Because the disturbances slightly exceeded this band, the DAC response engaged minimally to limit excursions, as shown in Fig. 56(c) and the corresponding real power setpoint changes due to the action of SLAC3R-L as shown in Fig. 56(d). This demonstrates the advantage of the decentralized controls that always act only based on local measurements and limits the frequency excursions beyond the allowable pre-defined frequencies. Besides the local measurements, these controls are only a function of the frequency limits chosen which appropriately dictates the control performance gain. This demonstration shows that these controls possess higher TRLs and since we are validating these controls on the HYPERSIM, a high-fidelity simulation platform with hardware in the loop, we conclude that the controls has TRL-6.

Next, we activated the SLAC3R-C coordinated controls. With these controls, the frequencies are driven back to nominal after the first two islanding events at 200 and 230 seconds. Following the third event at 280s, the inverter headroom constraints prevent a full return to nominal, but the frequency remains very close (see Fig. 55(e)). Because the coordinated controls promote consensus while steering frequency to nominal, the setpoints are either in consensus or move toward it after each disturbance (see Fig. 55(f)). For the SLAC3R coordinated controls, neighbor-to-neighbor setpoint exchange is carried out via PACP and all the GFM inverters reach consensus after every disturbance demonstrating the resilience capability of the hierarchy of controls developed under the RD2C initiative.

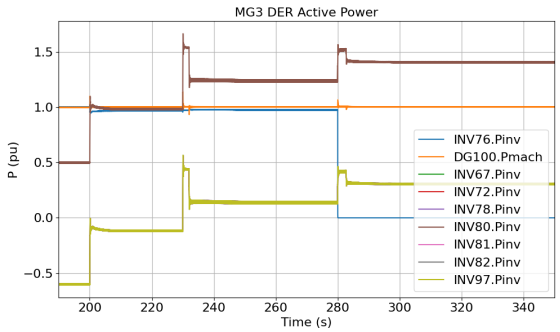
5.2 Scenario 1E

5.2.1 Description and Sequence of Events

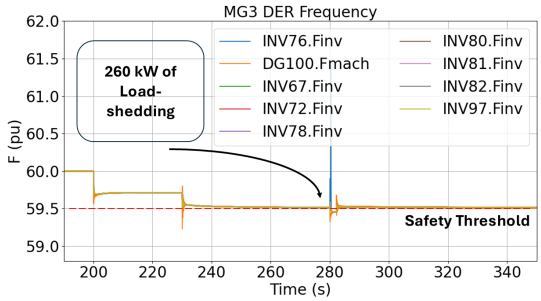
The narrative and sequence of events in Scenario 1E are the same as in described for Scenario 1D in Section 5.1.1 with the exception that the initial active power dispatch of the DERs are zero, which causes the GFM DER to share the load in proportion to their equipment ratings. Scenario 1E also features the load shedding scheme described in Section 5.1.1. Table 17 provides the initial DER setpoints.



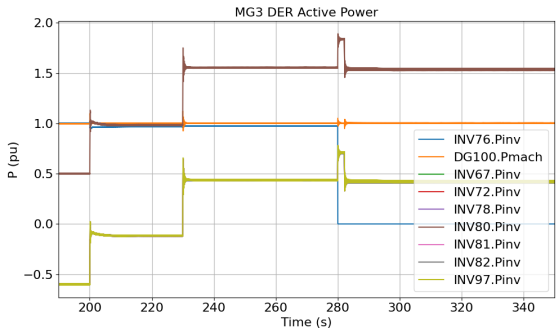
(a) No mitigation strategies.



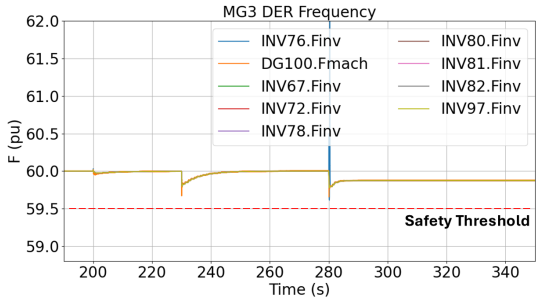
(b) No mitigation strategies.



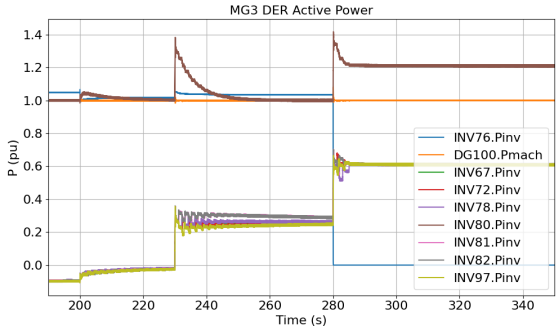
(c) SLAC3R-L control.



(d) SLAC3R-L control.



(e) SLAC3R-L and SLAC3R-C via PACP.



(f) SLAC3R-L and SLAC3R-C via PACP.

Figure 55: Scenario 1D Microgrid 3 dynamic frequency response.

Table 17: Scenario 1E Sequence of Events

Simulation Time (s)	Breaker Event	DER Dispatch Event
0^-	Node 60 to Node 160 breaker is **Open** Node 18 to 135 breaker is **Closed** Node 54 to 94 breaker is **Open** Node 151 to 300 breaker is **Closed** Node 97 to 197 breaker is **Closed**	All GFM INV $V_{SET,0}$ = 1.05 pu All DG $V_{SET,0}$ = 1.015 pu All GFL INV $Q_{SET,0}$ = 0.6 pu All DER $P_{SET,0}$ = 0.0 pu
200	Node 60 to Node 160 breaker **Opens**	
230	Node 151 to Node 300 breaker **Opens** Node 97 to 197 breaker **Opens**	
280	Inverter 76 PCC breaker **Opens**	

5.2.2 Simulation Results and Resilience Assessment

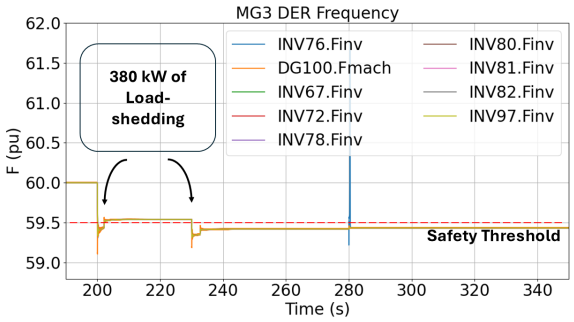
The sequence of events shown in Table 17 are followed to create multiple disturbances on the system. These disturbances result in frequency deviations from the nominal and clearly without any mitigation strategies in place and with just droop controls, the frequency is not recovered (see Fig. 56(a) and (b)) and subsequent disturbances would de-stabilize the system. The blue spike in frequency measurements is a numerical error and will be fixed in the subsequent updates to the measurement model. Next, we started turning on mitigation strategies in a hierarchical manner starting with decentralized autonomous controls (DAC or alternatively referred to as SLAC3R-L). These DACs are directly embedded into the HYPERSIM model and do not need any communication as it is based on local measurements. The allowable frequency limits to the SLAC3R-L controls are $[59.5, 60.5]$ when normalized gives $[0.991, 1.008]$. Since these disturbances led to deviations slightly outside the specified range of allowable frequencies, the DAC controls are engaged minimally to limit the frequencies as shown in Fig. 56(c) and (d).

In the next phase, we turned the SLAC3R-C controls and in the presence of these coordinated controls, the frequencies are steered back into their nominal values during the first two islanding events respectively at 200 and 230 seconds. However, due to the limitations of the inverter headroom, the frequency after the third disturbance at 280s is not nominal but it is very close (see Fig. 56(e)). Since, the coordinated controls are promoting consensus while steering the frequency to nominal, we can observe the setpoints in consensus or moving towards consensus after the disturbance (see Fig. 56(f)). It is clear that all the GFM inverters are in consensus and the GFM inverter is achieving consensus relatively slow. This brings out the challenges while coordinating GFMs and GFLs. It is important to note that, in our previous efforts of demonstrating these coordinated controls using GridLAB-D co-simulation framework, this behavior is not observed that is, the GFMs and GFLs achieve consensus. The setpoint data exchange between the neighbor inverters for SLAC3R coordinated controls implementation is facilitated via PACP.

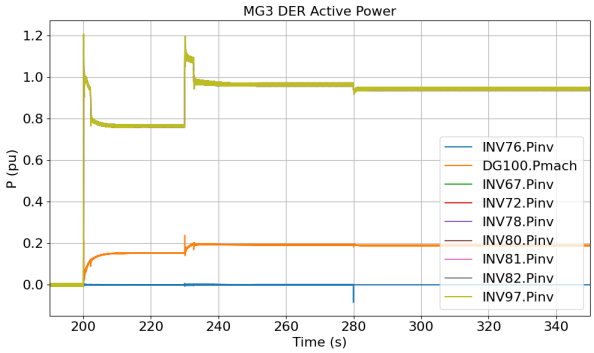
In the last phase, the ALERT supervisory controls that re-adjusts the setpoints in anticipation of an event (possible cloud cover so one of the inverters output power generation is reduced) is integrated alongside the SLAC3R controls. In this test run, ALERT controls are turned ON before the onset of disturbances. It is observed from Fig. 56(h) that in the presence of ALERT

controls, in the anticipation of an event, the controls re-adjusted the DG setpoint in MG3 where it is implemented. As a result, the post disturbance state yielded the nominal frequency or 60 Hz (1 pu). It is important to notice that the ALERT controls complemented the SLAC3R controls and cohesively worked together to achieve nominal frequency as shown in Fig. 56(g). Due to the presence of ALERT and the re-adjustment of setpoints pre-disturbance, the transient evolution of the system is different in the case with SLAC3R and ALERT controls (see Fig. 56(g)) when compared to just with SLAC3R controls (see Fig. 56(e)).

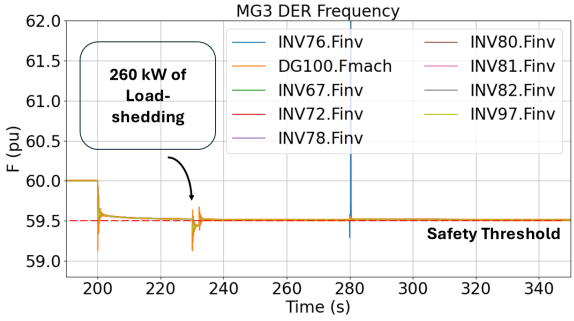
In summary, we observed that the SLAC3R-L acts reactively to correct any frequency disturbances locally and the SLAC3R-C comes in to steer the system frequency to nominal with coordination among various DERs and finally, the ALERT control proactively adjusts the setpoints anticipating a cloud cover event. All these mitigation strategies operating at different time-scales work in a complementary manner to promote resilience.



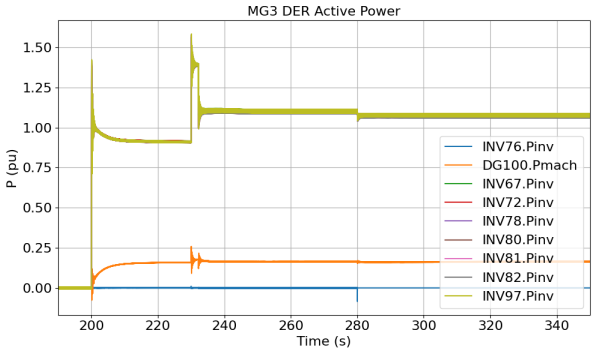
(a) No mitigation strategies.



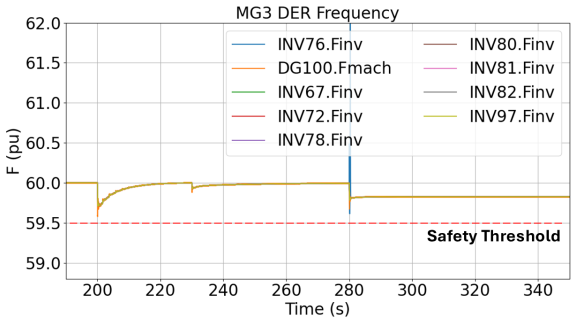
(b) No mitigation strategies.



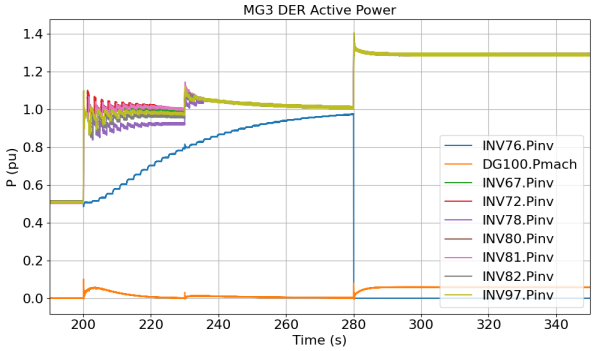
(c) SLAC3R-L control.



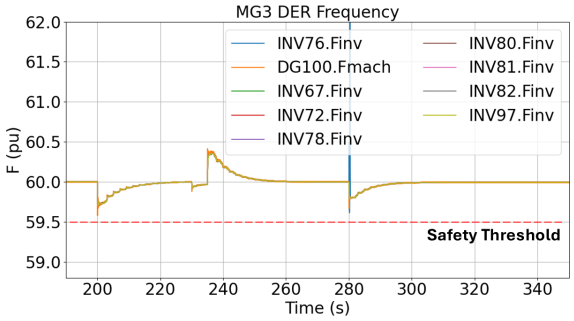
(d) SLAC3R-L control.



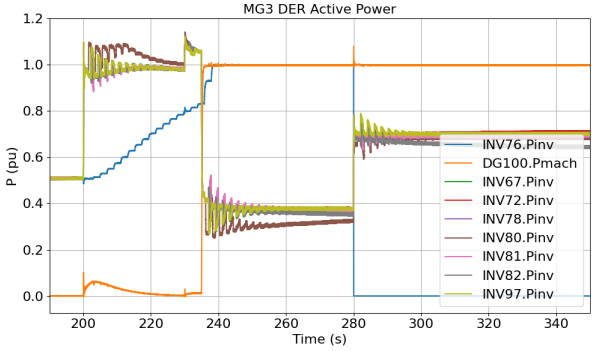
(e) SLAC3R-L and SLAC3R-C via PACP.



(f) SLAC3R-L and SLAC3R-C via PACP.



(g) SLAC3R-L, SLAC3R-C via PACP, and ALERT.



(h) SLAC3R-L, SLAC3R-C via PACP, and ALERT.

Figure 56: Scenario 1E Microgrid 3 dynamic frequency response.

5.3 Scenario 7A

5.3.1 Description and Sequence of Events

This scenario also features a combination of physical and cyber layer events. In this case, a sequence of events results in a significant disturbance to bus voltages in the IEEE 123 node feeder model resulting from a loss of reactive power support. An upstream feeder fault forces the microgrids to separate from the grid by opening the Node 60 to Node 160 circuit breaker. Then an equipment failure causes the fuses to blow on a capacitor bank located at Node 83 in Microgrid 3 causing a loss of voltage support. Finally, an adversary cleverly times a remote breaker command to disconnect the diesel generator in Microgrid 3 causing further loss of voltage support. With a significant deficit of reactive capability, Microgrid 3 experiences a significant voltage sag on its primary feeder. The goal of this experiment is to determine if consensus based reactive power sharing amongst the microgrids can allow Microgrid 1 and Microgrid 2 to automatically provide voltage support to Microgrid 3. Table 18 provides the detailed sequence of events.

Table 18: Scenario 7A Sequence of Events

Simulation Time (s)	Breaker Event	DER Dispatch Event
0^-	Node 60 to Node 160 breaker is **Open** Node 18 to 135 breaker is **Closed** Node 54 to 94 breaker is **Open** Node 151 to 300 breaker is **Closed** Node 97 to 197 breaker is **Closed**	All GFM INV $V_{SET,0}$ = 1.0 pu All DG $V_{SET,0}$ = 1.0 pu All GFL INV $Q_{SET,0}$ = 0.0 pu All DER $P_{SET,0}$ = 0.0 pu
200	Node 60 to Node 160 breaker **Opens**	
230	Capacitor 83 breaker **Opens**	
280	Generator 100 PCC breaker **Opens**	

5.4 Quantitative Resilience Assessment Using Time Domain Simulation Results

This section presents the results of an in-depth literature review and investigation into developing processes for assessing power system resilience based on time series data collection from real-time, EMT simulations. The methodology is discussed in the first subsection and the following subsection will provide an example applied to Scenario 1D, which is presented in Section 5.1. Due to time constraints we were unable to perform a quantitative resilience assessment on the simulation results from Scenario 1E.

At a high-level, the resilience improvement quantification presented in the following sections is performed by:

1. Collecting the time series data from the real-time, scenario simulations.

2. Selecting simulation variables indicative of system health (such as frequency).
3. Computing metrics that measure positive or negative aspects of the system health variable for each dataset.
4. Aggregating the metrics for each dataset into a single, unified metric measuring the system health.
5. Comparing the measure of system health between trials with no mitigation strategies vs trials with one or more mitigations applied.

Where we observe a significant, positive change in the aggregated, system health metric from a trial with no mitigation strategies compared to a trial with one or more mitigation strategies, we can conclude that the mitigations would be successful at reducing the negative effects from the events in the scenario being assessed. The methodology and an example are provided in the following subsections.

5.4.1 Methodology

To assess the overall power system resilience to unplanned adversarial actions when engaging resilience-driven mitigation strategies, performance-based metrics have been identified and applied to quantitatively measure the controller performance. Given the currently tested mitigation strategies, the power grid assets they act upon (the IBRs), and the main scope of their controlling actions, there are two indexes according to which resilience is going to be analyzed:

- Ability of the IBRs to provide frequency support to the microgrids during the disturbances,
- Ability of the IBRs to alleviate synchronous legacy generators effort to accommodate for sudden demand variation due to unforeseen events.

By their definition, these indexes are related to certain system measurements, also named figures of merit (FOM) or measures of performance (MOP). Specifically, the frequency of the synchronous generators is an indicator of how balanced the system is when it comes to serving the demand and avoiding load shedding. At the same time, the power generated by the distributed energy resources, that is the IBRs in this case, gives information about how capable the demand can be sustain by local DERs, rather than spinning new synchronous generator reserves.

As resilience is defined and measured based on the system performance, it is important to select an appropriate measure of performance (MOP) to match the specific service provided by the analyzed system. In our case, the MOP that drives the performance analysis is the system frequency. After conducting the literature review, a promising set of metrics was identified in [36], which could be applied to the frequency variables recorded from the real-time, hardware-in-the-loop testing of the mitigation strategies. Fig. 57 defines the phases of a resilient response as indicated by a FOM. The metrics are applied to the different phases to determine an overall performance.

Robustness (or Resistance) (R) - measure to assess the absorptive capability of the system during the disruptive phase (DR), that is $t_1 \leq t < t_2$. R quantifies the nadir/zenith frequency value between t_1 and t_4

$$R = \min_{t_1 \leq t \leq t_4} \omega(t) \tag{30}$$

where t_1 represents the time the system enters the disruptive phase, and t_4 is the time the system reaches a new steady state.

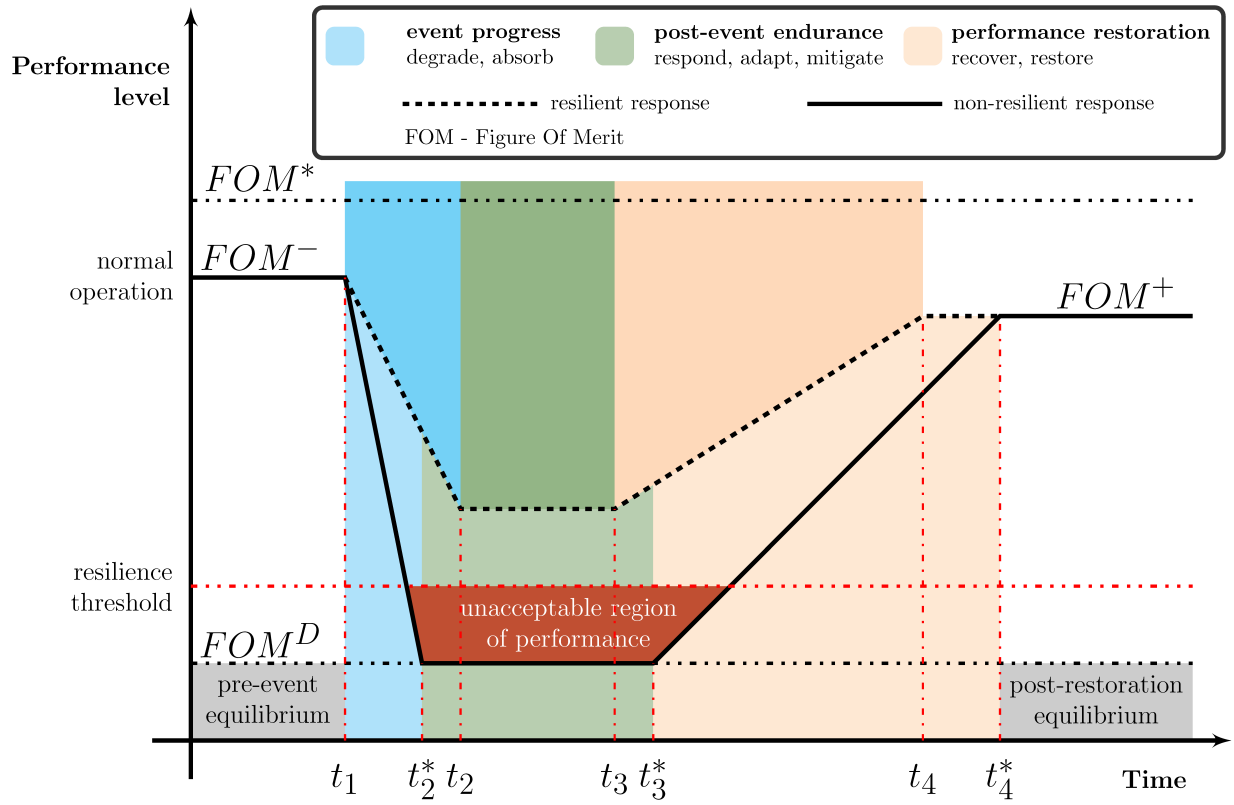


Figure 57: Evolution of a performance indicator (FOM) of a system in the presence of an event (resilience curve).

Rapidity during the disruptive phase ($RAPI_{DP}$) - average slope of the MOP (frequency) function during the disruptive phase, that is $t_1 \leq t < t_2$.

$$RAPI_{DP} = \frac{|\sum_{t=t_1}^{t_2} \frac{\omega(t)-\omega(t-\Delta t)}{\Delta t}|}{N} \tag{31}$$

where N represents the number of Δt time samples between the beginning of the disruptive phase t_1 and the peak time t_2

Performance loss during the disruptive phase (PL_{DP}) - the area of the region bounded by the MOP (frequency) curve before and after the disruptive event causes negative effects.

$$PL_{DP} = \int_{t_1}^{t_2} (\omega(t) - \omega(t_0))dt \tag{32}$$

where t_0 represents the time when the system is in its initial steady state (could also be the nominal value, that is 60 Hz for the frequency).

Time averaged performance loss during disruptive phase ($TAPL_{DP}$) - "considers the time of appearance of negative effects due to disruptive events up to full system recovery and provides a time-independent indication of both adaptive and restorative capabilities as responses to the disruptive events"

$$TAPL_{DP} = \frac{\int_{t_1}^{t_2}(\omega(t) - \omega(t_0))dt}{t_2 - t_1} \tag{33}$$

Rapidity during the recovery phase ($RAPI_{RP}$) - average slope of the MOP (frequency) function during the recovery phase, that is $t_2 \leq t < t_3$.

$$RAPI_{RP} = \frac{|\sum_{t=t_2}^{t_3} \frac{\omega(t)-\omega(t-\Delta t)}{\Delta t}|}{N} \tag{34}$$

where N represents the number of Δt time samples between the beginning of the recovery phase t_2 and the time t_3 when reaching the new steady state.

Performance loss during the recovery phase (PL_{RP}) - the area of the region bounded by the MOP (frequency) curve before and after the extreme negative effect of the disruptive event starts to be mitigated.

$$PL_{RP} = \int_{t_2}^{t_3}(\omega(t) - \omega(t_0))dt \tag{35}$$

Time averaged performance loss during recovery phase ($TAPL_{RP}$)

$$TAPL_{RP} = \frac{\int_{t_2}^{t_3}(\omega(t) - \omega(t_0))dt}{t_3 - t_2} \tag{36}$$

Recovery ability (RA) - measure of how close to the nominal/initial steady state the system manages to recover.

$$RA = \left|\frac{\omega(t_3) - \omega(t_2)}{\omega(t_2) - \omega(t_0)}\right| \tag{37}$$

Finally, the metrics above have been combined to produce the *integrated resilience metric* (GR). GR has been defined to assess system resilience "with an overall perspective and to allow comparisons among different systems and system configurations" by combining the capabilities during both the disruptive and the recovery phases.

$$GR = R * \frac{RAPI_{RP}}{RAPI_{DP}} * (TAPL)^{-1} * RA \tag{38}$$

where the time averaged performance loss TAPL combines the performance loss during both disruptive and recovery phases, as in

$$TAPL = \frac{\int_{t_1}^{t_3}(\omega(t) - \omega(t_0))dt}{t_3 - t_1} \tag{39}$$

We apply these metrics to demonstrate quantitative assessment of resilience improvement in Scenario 1D in the next section.

5.4.2 Quantitative Resilience Assessment of Scenario 1D

We demonstrate application of the quantitative metrics to the disturbances in Scenario 1D. In this case, a hypothetical fault on the distribution system causes the breaker at Node 60 to open to allow the microgrids to operate autonomously while not energizing the fault feeder segment. Following the fault, the microgrids are separated by a hypothetical, malicious command injection

attack on the POI breakers to prevent the microgrids from pooling resources. Finally, the hypothetical adversary performs an additional command injection attack to disconnect one of the inverters in MG3 for a total of three events. We compare the system frequency response with no mitigation strategies to the response with SLAC3R-L and SLAC3R-C in the loop. Note that PACP manages the communication between the SLAC3R-C peers. Fig. 58 through Fig. 60 show the frequency response and highlight the three stages of the resilient response for each of the scenario events.

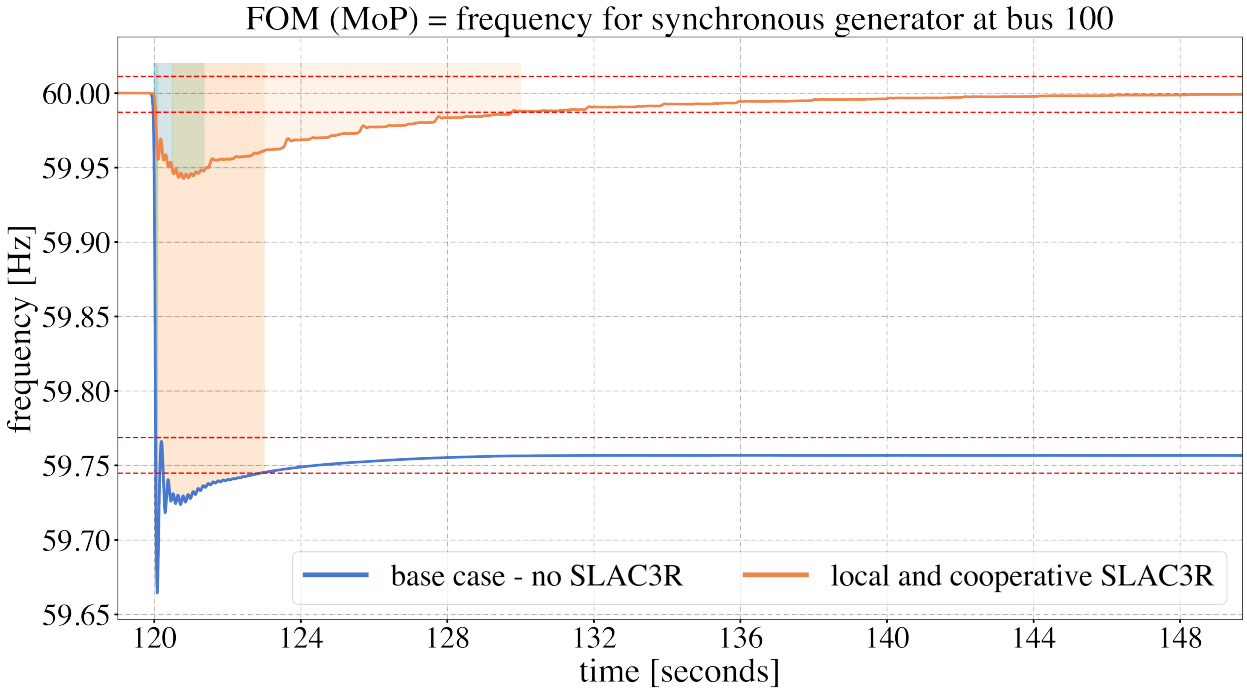


Figure 58: Analysis of the performance regions from the trapezoidal model of resilient response for the first event in Scenario 1D : system degradation (blue); system response (green); and system recovery (orange).

Visually, it is clear that when SLAC3R and PACP are in the loop the frequency response is substantially improved. We applied the quantitative performance metrics and have summarized them in Table 19.

The integrated resilience metric GR indicates the substantial frequency response improvement that is apparent from Fig. 58. A relatively smaller improvement in frequency response is observed in Fig. 59 and Fig. 60 due to activation of underfrequency loadshedding protections for the case where no mitigations are present. A well established method for visualizing multiple metrics is to present them as axes on a radar (or spider) chart. Generally, the convention for radar charts is for larger values along an axis to indicate better performance. Thus, the larger the area of a polygon representing the overall performance of a system, the better the overall performance. Furthermore, the sizes of the polygons can readily be compared visually. The integrated resilience metric calculation in Eq. (38) indicates that several metrics are negative metrics – the larger the value, the worse the system has performed – and hence they appear in the denominator. In this case, these metrics do not conform to the radar chart convention. Therefore, we use the inverse of the negative metrics to conform to the convention of radar charts. After inversion is applied to the negative metrics, all metric scores are normalized using the larger of the two resulting scores. An example calculation for PL_{RP} is given as follows.

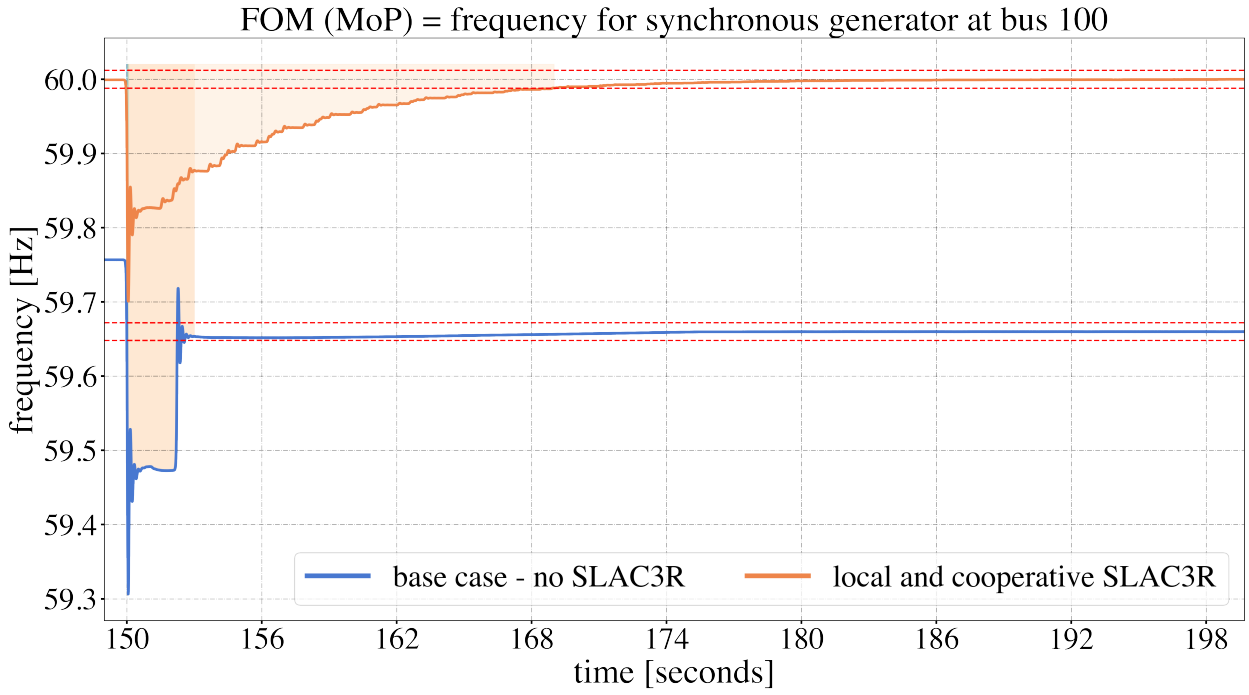


Figure 59: Analysis of the performance regions from the trapezoidal model of resilient response for the second event in Scenario 1D : system degradation (blue); system response (green); and system recovery (orange).

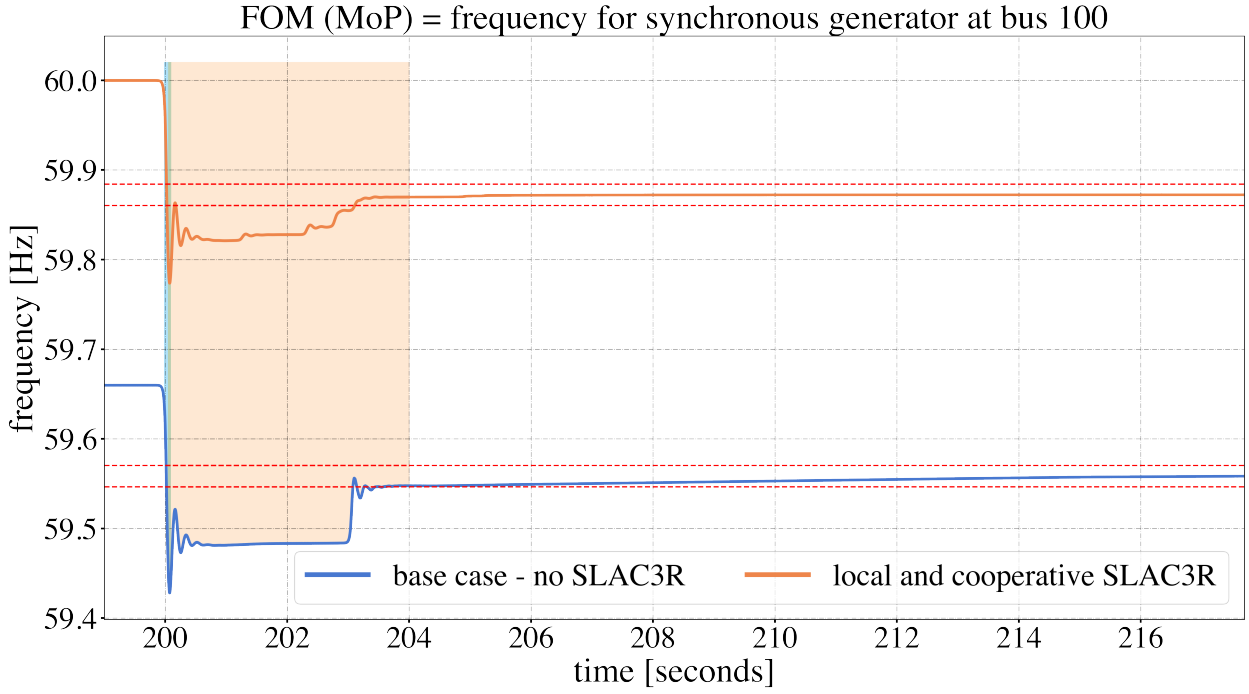


Figure 60: Analysis of the performance regions from the trapezoidal model of resilient response for the third event in Scenario 1D : system degradation (blue); system response (green); and system recovery (orange).

Table 19: Scenario 1D Quantitative Resilience Metrics

Metric	Event 1 No Mitigations	Event 1 SLAC3R + PACP	Event 2 No Mitigations	Event 2 SLAC3R + PACP	Event 3 No Mitigations	Event 3 SLAC3R + PACP
R	59.6642	59.9427	59.3063	59.7009	59.4280	59.7737
ΔR	0.3357	0.0573	0.6937	0.2991	0.5720	0.2283
$RAPI_{DP}$	3.2261	0.1624	5.0585	3.3675	2.4804	2.4208
PL_{DP}	0.0197	0.0357	0.0421	0.0145	0.0417	0.0118
$RAPI_{RP}$	0.1023	0.0142	0.4035	0.0428	0.1177	0.0986
PL_{RP}	0.7694	0.2633	1.4071	1.2543	1.9701	0.4987
$TAPL_{DP}$	0.2063	0.0457	0.5048	0.1749	0.4760	0.1347
$TAPL_{RP}$	0.2636	0.0286	0.4796	0.0662	0.5016	0.1704
$TAPL$	0.2618	0.0300	0.4804	0.0667	0.5011	0.1694
RA	0.2675	0.8742	0.8537	1.0664	0.5730	0.3975
GR	1.9332	152.7313	8.4067	12.1314	3.2246	5.7128

1. Calculate the inverse of the PL_{RP} metrics: without any controls the score is then 1.2997; with SLAC3R–in–the–loop the score is then 3.7980.
2. Normalize both scores using the PL_{RP} score from the case with SLAC3R–in–the–loop: the normalized score for the case without any controls is then 0.3422

The resulting radar charts for all three events are shown in Fig. 61. The areas covered by the polygons representing the system performance with SLAC3R–in–the–loop are significantly larger indicating much improved resilience of the MG3's frequency to the events in Scenario 1D. For the $RAPI_{RP}$ metric, the system without any resilient controls applied appears to consistently perform better. This is an artifact of the calculation method for these metrics, which can be difficult to apply accurately for high-order systems that have much more complex dynamics than 2nd order systems, which are often used to develop such metrics. For example, the higher score for $RAPI_{RP}$ is driven by the very fast first swing of the frequency for the case with no controls (see the blue trace in Fig. 58. Similarly, the inverted PL_{DP} score in Fig. 61(a) suggests that the case with no controls experiences less performance degradation during the disturbance phase. This is because the algorithm to identify the disturbance phase interprets the disturbance phase as concluded during the first swing, which happens very quickly. Therefore, the width of the area under the curve during the disturbance phase of the case without any controls is very narrow, leading to a very small performance loss calculation. Despite this phenomena, the integrated resilience metric GR successfully combines the constituent metrics in a way that accurately reflects the significantly improved MG3 frequency response due to adding the SLAC3R and PACP mitigation strategies for all three events. The impact is that this methodology can be applied to a large number of scenario results that could span many different types and combinations of natural or adversarial events. The GR metric can be calculated for each trial to understand the incremental improvement (or lack of improvement) in resilience confered by each mitigation strategy to understand the coverage for potential events and where weaknesses may persist.

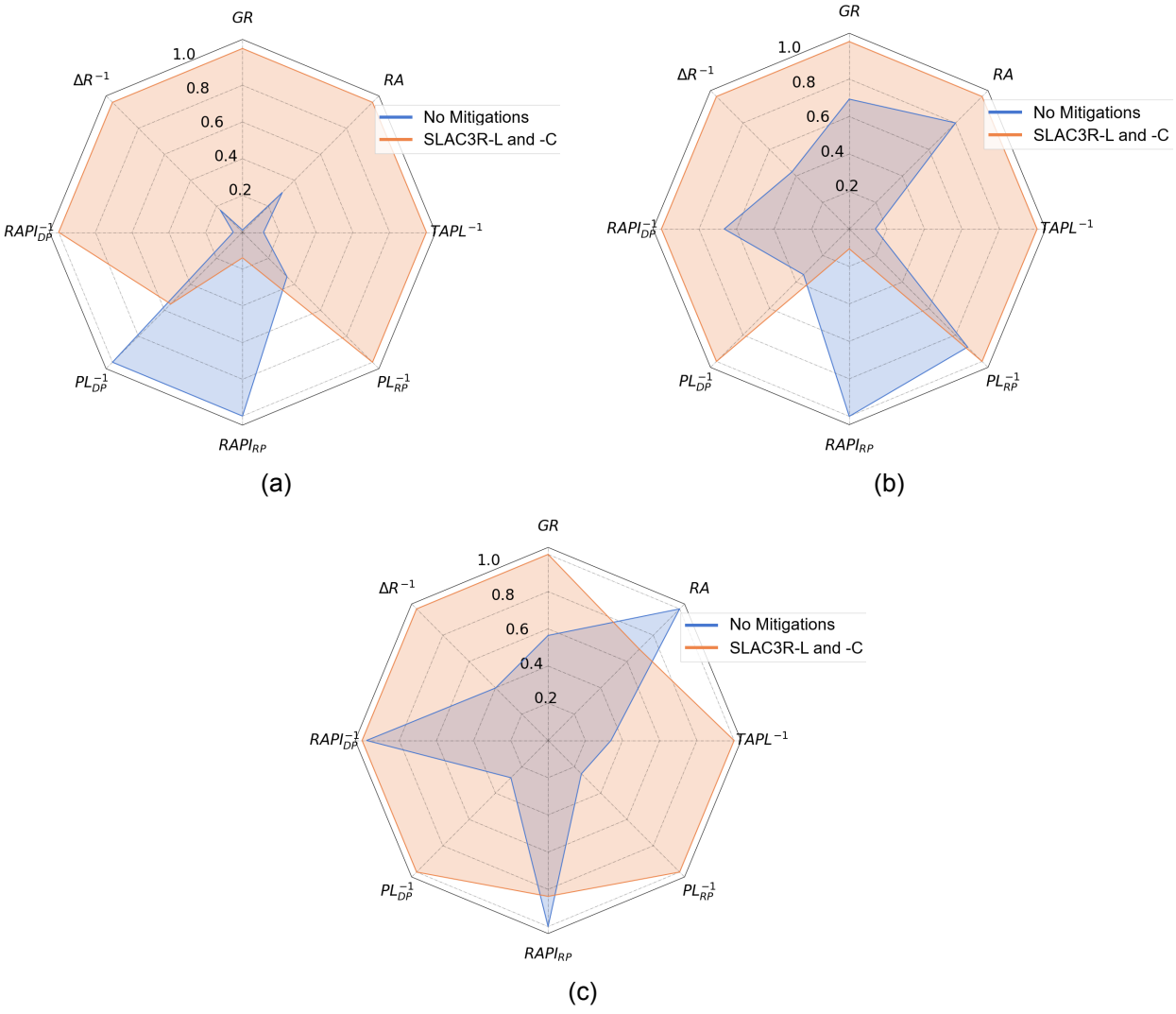


Figure 61: Resilience visualization for Scenario 1D using the Microgrid 3 diesel generator frequency as the figure of merit.

6.0 Conclusions

The capstone effort aimed to exercise an integrated framework that included SLAC3R, P2P, PACP, and ALERT in a controlled, modeled environment with sufficient complexity to span the range of resilience mitigations provided. This aim required: 1) increasing the scale and complexity of our modeling and simulation capabilities; 2) developing an integration framework to enable the separate controls to operate cohesively; and 3) running the integrated framework through a suite of test cases on the HYPERSIM testbed and performing analyses to quantify and characterize the resilient response. Consequently, the specific outcomes of this capstone effort are:

1. **NMG Model Development:** This effort improved the IEEE 123 Node Test Feeder model. The new model version is more stable due to replacement of older DER models with documented stability issues. A modbus interface was added to integrate external control strategies deployed via Cybernet virtual machines. Also, developed a EMT model of the 3000 Node Test Feeder - the third feeder of the IEEE 9500 Node Test Feeder benchmark model. This effort overcame several large–scale, real–time, EMT modeling challenges.
2. **Communication Network:** This effort developed a real–time NS3–based WAN model suitable for real–time, hardware–in–the–loop experimentation. The communication network model in the NS3 simulator emulated the communication infrastructure connecting device– and application–layer controls. NS3 also supports a bridge adapter to connect the simulated network to a physical network, thereby attaching the physical system to the simulated environment. This enabled us to set up complex communication topologies and create scenarios such as generating an attack in a simulated NS3 node. The integrated models have the capability to induce physical disturbances, mimic real-time communication challenges, and serve as a platform for adversarial actions. We defined a variety of scenarios representing both physical and cyber events to validate system performance, assess control sensitivities, and provide a basis for developing mitigation strategies.
3. **Integration of Mitigation Strategies:** We integrated the RD2C control strategies across all CPS layers in the emulated test environment into a unified, hierarchical distribution-system control scheme. SLAC3R-L continuously monitored local measurements and intervened on fast timescales to enhance local resilience, implemented in HYPERSIM as an add-on to the inverter controls. ALERT and P2PC were coupled with SLAC3R-C. ALERT computes setpoints anticipating possible disruptions (such as cloud cover) and used them to scale the SLAC3R-C setpoints, so the centralized predictive controller biased the secondary SLAC3R-C actions. This achieved power sharing, stabilized frequency and voltage, and improved preparedness for near-term disturbances. P2PC was also integrated with SLAC3R-C and activated in networked microgrids when disturbances caused voltage deviations in any MG. It calculates the incremental voltage support required from each MG and its inverters. Since the P2PC outputs are change in voltage support, they were superimposed on the SLAC3R-C setpoints. Overall, these controls complement one another across different operational timescales to promote resilience. The PACP middleware provided the communications layer among DER controls.
4. **Demonstration:** We demonstrated the unified controls on the high–fidelity HYPERSIM testbed. A range of scenarios were developed to analyze the system performance at different timescales, DER responses at device level and system level and the challenges associated with data flows across different DERs and controls. From the range of experiments, it was clear that the proposed unified control strategies effectively mitigate various disturbances and achieve resilience. Overall, these experiments supported the design and validation of proactive and adaptive tuning measures to improve resilience in cyber-physical systems.

6.1 Future Work Guidance

The Capstone 1 Team faced many challenges with developing the cyber–physical testbed. Due to time constraints, numerous opportunities for future work remain. From the HYPERSIM modeling effort, the single phase constant power load models presented a road–block to stable, islanded operation of the microgrids in the 3000 Node Test Feeder model. These loads are ubiquitous in the model, representing every spot load in the feeder. Further model development efforts need to address the unstable behavior of these loads when not connected to a stiff voltage source. Additionally, there are several potential features for the GridLAB–D to HYPERSIM conversion tool that could further reduce the manual modeling effort including:

- Automatic placement of off–page connectors at every node to facilitate population of shunt connected, dynamic models on separate pages.
- A decoupling component location identification tool.
- Further improvement of the automatic DER population feature.
- Configurable selection of load model type by node.

Furthermore, integration of the modeling and simulation automation python based tool with the conversion tool presents an opportunity to fundamentally change the way HYPERSIM users can interact with large–scale models. By combining the automatic network drawing capability of the conversion tool with the automatic model parameterization capability of the tool developed in this capstone, the user can shift from a graphical interface to a scripting based interface. In terms of scenario orchestration, the team developed a manual process for coordinating the start of simulations with the start of control software on the virtual machines. This was a manageable approach at the scale of the IEEE 123 Node Test Feeder model containing three microgrids and 18 DERs. However, a larger–scales (such as 10 microgrids and hundreds of DERs) this approach is unlikely to succeed. As a result, future work conducting large–scale, real-time tests of hierarchical control systems will require development of experiment automation. One potential solution is the use of Ansible scripting, which can automatically execute processes such as starting the power system and communication system simulations while simultaneously activating control system software on the host virtual machines.

References

[1] N. C. of State Legislatures, G. Anderson, M. Cleveland, D. Shea, Modernizing the electric grid: State role and policy options, National Conference of State Legislatures, 2019.

[2] S. Patel, Eight critical reliability challenges nerc is confronting for grid stability, https://www.powermag.com/eight-critical-reliability-challenges-nerc-is-confronting-for-grid-stability/, accessed: 10-15-2024.

[3] NERC, 2022 long-term reliability assessment, https://www.nerc.com/pa/RAPA/ra/Reliability%20Assessments%20DL/NERC_LTRA_2022.pdf, accessed: 10-30-2024.

[4] N. Nazir, T. Ramachandran, S. Bhattacharya, A. Singhal, S. Kundu, V. Adetola, Optimization-based resiliency verification in microgrids via maximal adversarial set characterization, in: 2022 American Control Conference (ACC), IEEE, 2022, pp. 2214–2220.

[5] T. L. Vu, S. Mukherjee, K.-B. Kwon, V. A. Adetola, Peer-to-peer communication control for resilient operations of networked cyberphysical systems, Tech. rep., Pacific Northwest National Laboratory (PNNL), Richland, WA (United States) (2024).

[6] B. Hyder, A. Ahmed, P. Mana, T. Edgar, S. Niddodi, Leveraging high-fidelity datasets for machine learning-based anomaly detection in smart grids, in: 2023 11th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES), IEEE, 2023, pp. 1–6.

[7] S. P. Nandanoori, A. K. Bharati, S. Sinha, S. Kundu, V. Adetola, K. Schneider, Empowering the grid: Decentralized autonomous control for effective utilization and resilience, arXiv preprint arXiv:2410.17143 (2024).

[8] R. A. Jinsiwale, M. Maharjan, P. T. Mana, A. Ashok, T. Becejac, S. E. Harpool, W. J. Hofer, A. R. Vielma, G. Gloria, T. W. Edgar, Design and development of a high fidelity cyber-physical testbed, Tech. Rep. PNNL-34860, Pacific Northwest National Laboratory (PNNL), Richland, WA (United States) (2023).

[9] A. A. Anderson, S. V. Vadari, J. L. Barr, S. Poudel, A. Dubey, T. E. McDermott, R. Podmore, Introducing the 9500 Node Distribution Test System to Support Advanced Power Applications: An Operations-Focused Approach, Tech. Rep. PNNL-33471, Pacific Northwest National Laboratory (PNNL), Richland, WA (United States) (Sep. 2022). doi:10.2172/1922914.
URL https://www.osti.gov/biblio/1922914

[10] G. F. Riley, T. R. Henderson, The ns-3 network simulator, in: Modeling and tools for network simulation, Springer, 2010, pp. 15–34.

[11] R. A. Jinsiwale, M. J. Rice, M. Maharjan, P. T. Mana, B. Hyder, Scalability of real-time distribution models, Tech. Rep. PNNL-37021, Pacific Northwest National Laboratory (PNNL), Richland, WA (United States) (2024).

[12] R. Jinsiwale, M. Maharjan, T. Becejac, A. Ashok, Evaluating a real-time model decoupling compensation approach for developing scalable, high-fidelity microgrid models, in: 2023 IEEE Texas Power and Energy Conference (TPEC), IEEE, 2023, pp. 1–6.

[13] S. Chiniforoosh, J. Jatskevich, A. Yazdani, V. Sood, V. Dinavahi, J. A. Martinez, A. Ramirez, Definitions and applications of dynamic average models for analysis of power systems, IEEE Transactions on Power Delivery 25 (4) (2010) 2655–2669. doi:10.1109/TPWRD.2010.2043859.

[14] W. H. Kersting, Distribution System Modeling and Analysis, Taylor & Francis, 2017, 4th Ed.

[15] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, Analysis of Electric Machinery and Drive Systems, IEEE, 2002, 2nd Ed.

[16] A. Yazdani, A. R. Di Fazio, H. Ghoddami, M. Russo, M. Kazerani, J. Jatskevich, K. Strunz, S. Leva, J. A. Martinez, Modeling guidelines and a benchmark for power system simulation studies of three-phase single-stage photovoltaic systems, IEEE Transactions on Power Delivery 26 (2) (2011) 1247–1264. doi:10.1109/TPWRD.2010.2084599.

[17] Model specification of droop-controlled, grid-forming inverters (regfm_a1) (2023). doi:10.2172/2229442.
URL https://www.osti.gov/biblio/2229442

[18] R. H. Lasseter, J. H. Eto, B. Schenkman, J. Stevens, H. Vollkommer, D. Klapp, E. Linton, H. Hurtado, J. Roy, Certs microgrid laboratory test bed, IEEE Transactions on Power Delivery 26 (1) (2010) 325–332.

[19] W. Du, F. K. Tuffner, K. P. Schneider, R. H. Lasseter, J. Xie, Z. Chen, B. Bhattarai, Modeling of grid-forming and grid-following inverters for dynamic simulation of large-scale distribution systems, IEEE Transactions on Power Delivery 36 (4) (2020) 2035–2045.

[20] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. De Vicuña, M. Castilla, Hierarchical control of droop-controlled ac and dc microgrids—a general approach toward standardization, IEEE Transactions on industrial electronics 58 (1) (2010) 158–172.

[21] A. Singhal, T. L. Vu, W. Du, Consensus control for coordinating grid-forming and grid-following inverters in microgrids, IEEE Transactions on Smart Grid 13 (5) (2022) 4123–4133.

[22] X. Bai, L. Qu, W. Qiao, Robust ac optimal power flow for power networks with wind power generation, IEEE Transactions on Power Systems 31 (5) (2015) 4163–4164.

[23] T. L. Vu, S. Mukherjee, V. Adetola, Resilient communication scheme for distributed decision of interconnecting networks of microgrids, in: 2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE, 2023, pp. 1–5.

[24] S. Sundaram, C. N. Hadjicostis, Distributed function calculation via linear iterative strategies in the presence of malicious agents, IEEE Transactions on Automatic Control 56 (7) (2010) 1495–1508.

[25] G. Collins, J. Iversen, pymodbus, https://pypi.org/project/pymodbus/ (2025).

[26] G. Krishnan S., C. Konstantinou, Design and evaluation of a dc microgrid testbed for der integration and power management, in: 12th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2024, pp. 1–6.

[27] H. Bilil, A. Hathah, A. Battou, Design and experimentation guidelines for der's emulation testbed, IEEE Transactions on Power Systems 37 (3) (2010) 1730–1738. doi:10.1109/TPWRS.2021.3114117.

[28] B. Pham, C. Huff, P. N. Vendittis, A. Smit, A. Stinskiy, S. Chanda, Implementing distributed intelligence by utilizing dnp3 protocol for distribution automation application, in: IEEE/PES Transmission and Distribution Conference and Exposition (T&D), IEEE, 2018, pp. 1–7. doi:10.1109/TDC.2018.8440305.

[29] D. Moldovan, R. Ayyanar, Dnp3 implementation in a high der penetration distribution system, in: IEEE Kansas Power and Energy Conference (KPEC), 2024, pp. 1–5. doi:10.1109/KPEC61529.2024.10676137.

[30] A. Garg, M. Hernandez, M. Carney, B. Currie, A. Akhtar, D. Weisser, Flexible interconnection analysis with epri der integration testbed, in: CIRED 2020 Berlin Workshop (CIRED 2020), Vol. 2020, 2020, pp. 353–356. doi:10.1049/oap-cired.2021.0056.

[31] L. Strezoski, Utility derms and der aggregators: An ideal case for tomorrow's dso, in: IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2022, pp. 1–5. doi:10.1109/ISGT-Europe54678.2022.9960384.

[32] OPAL-RT, The power system simulator of tomorrow, https://www.opal-rt.com/systems-hypersim/ (2024).

[33] OPAL-RT, Dnp3 master, https://www.opal-rt.com/software-communication-protocols/dnp3-master/ (2024).

[34] ChargePoint, pydnp3, https://github.com/ChargePoint/pydnp3 (2022).

[35] O. Bel, J. Kim, W. J. Hofer, M. Maharjan, B. Hyder, S. Purohit, S. Niddodi, Co-simulation framework for network attack generation and monitoring, IEEE Access (2024).

[36] J. Palma-Oliveira, Resilience and Risk: Methods and Application in Environment, Cyber and Social Domains, Springer Dordrecht, 2017. doi:10.1007/978-94-024-1123-2.

Appendix A – HYPERSIM DER Models and Default Parameters

A.1 Diesel Synchronous Generator Block Diagrams and Default Parameters

Table A.1: Diesel Synchronous Generator Default Parameters.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Grid Connection							
Volt.	12.47 kV	SCC	15.0 MVA	X/R	10		
Step Up Transformer							
Pri. Voltage	12.47 kV	Sec. Voltage	4.16 kV	Pri. Winding	WYE−G	Sec. Winding	DELTA
Rating	4.0 MVA	Z	4.75%	X/R	11.4		
Machine							
Rating	4.0 MVA	Stator Voltage	4.16 kV	Nom. Freq.	60 Hz	Pole Pairs	2
R_S	0.0016 pu	L_s	0.03 pu	X_0	0.00052 pu	X_C	0.0 pu
X_d	0.7 pu	X_q	0.8 pu	X'_d	0.13 pu	X'_q	0.1 pu
X''_d	0.08 pu	X''_q	0.08 pu	T'_{d0}	3.0 s	T'_{q0}	3.0 s
T''_{d0}	0.2 s	T''_{q0}	0.2 s				
Governor and Prime Mover							
m_p	0.05 pu	K_p	5.0 pu	K_i	9.0 pu	T_{act}	0.1 s
$+Lim$	1.1 pu	$-Lim$	0.0 pu	T_P	0.1 s		
AVR and Exciter							
m_q	0.05 pu	T_r	0.02 s	K_a	90 pu	T_a	0.02 s
K_f	0.03 pu	T_f	1.0 s	T_e	0.8 s	K_e	1.0 pu
K_d	0.38 pu						
Protection System							
TOC1		1.5 pu, 1.0 s		TOC2		2.5 pu, 0.4 s	
OV1		1.12 pu, 0.33 s		UV1		0.88 pu, 0.33 s	
OF1		69.0 Hz, 0.33 s		UF1		50.4 Hz, 0.33 s	

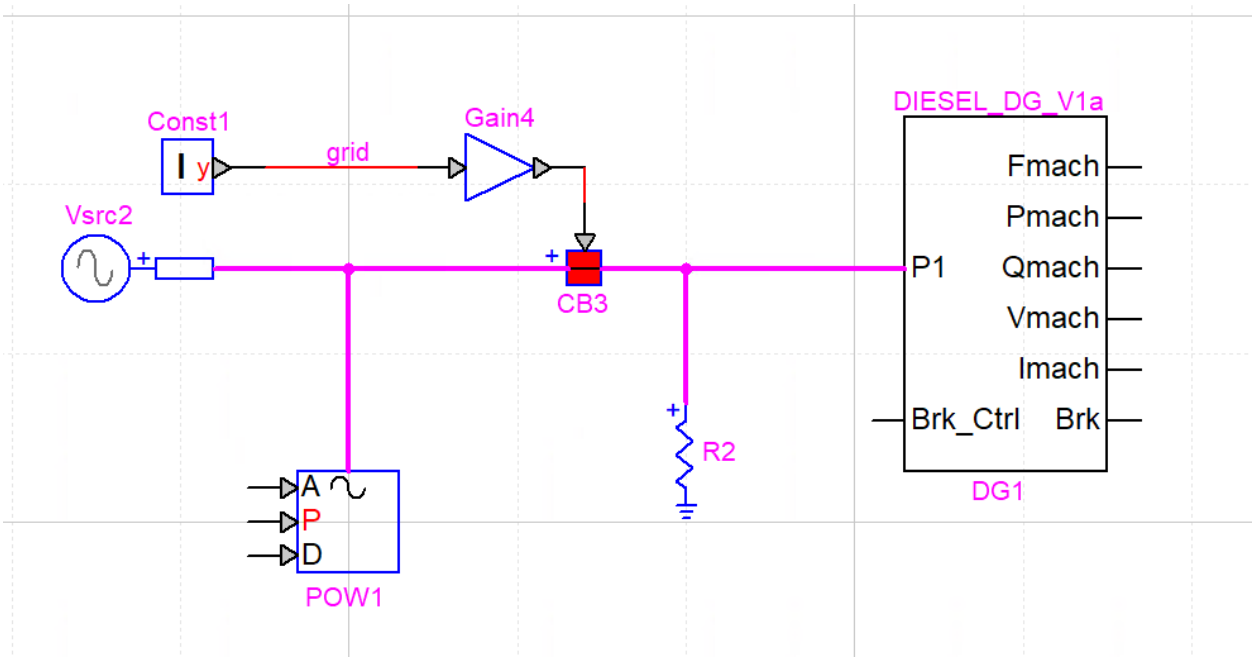


Figure A.1: Diesel synchronous generator model subcircuit and test circuit model.

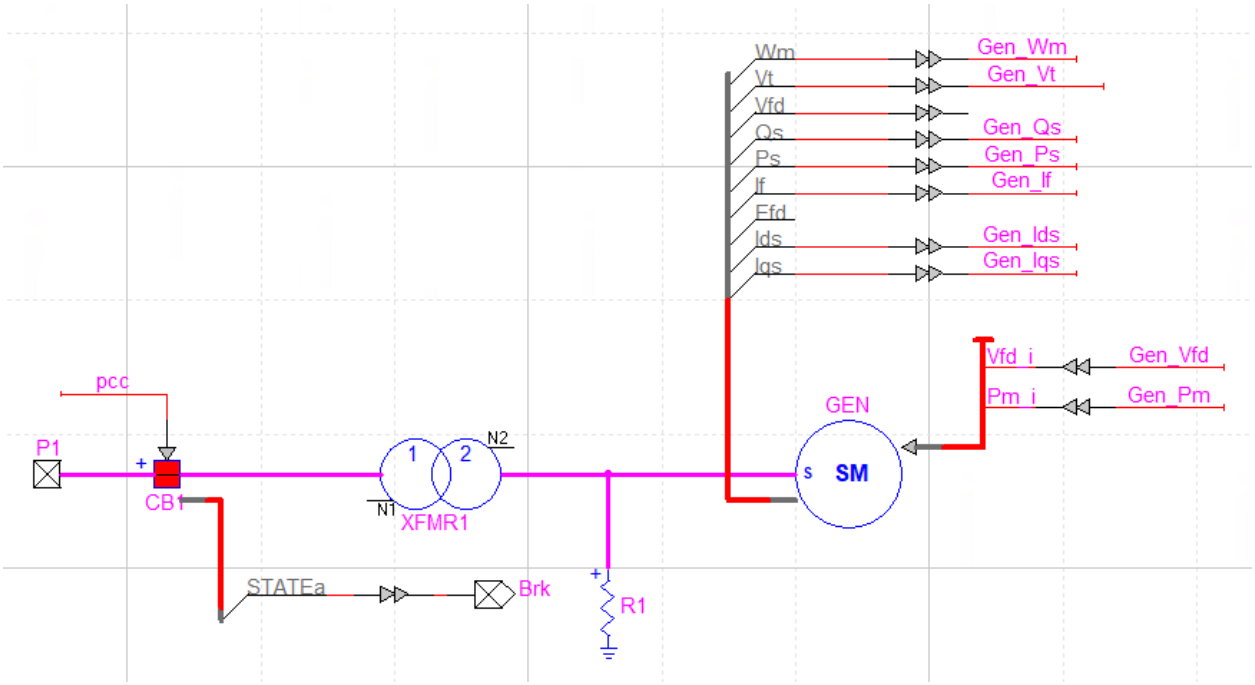


Figure A.2: Diesel synchronous generator physical model.

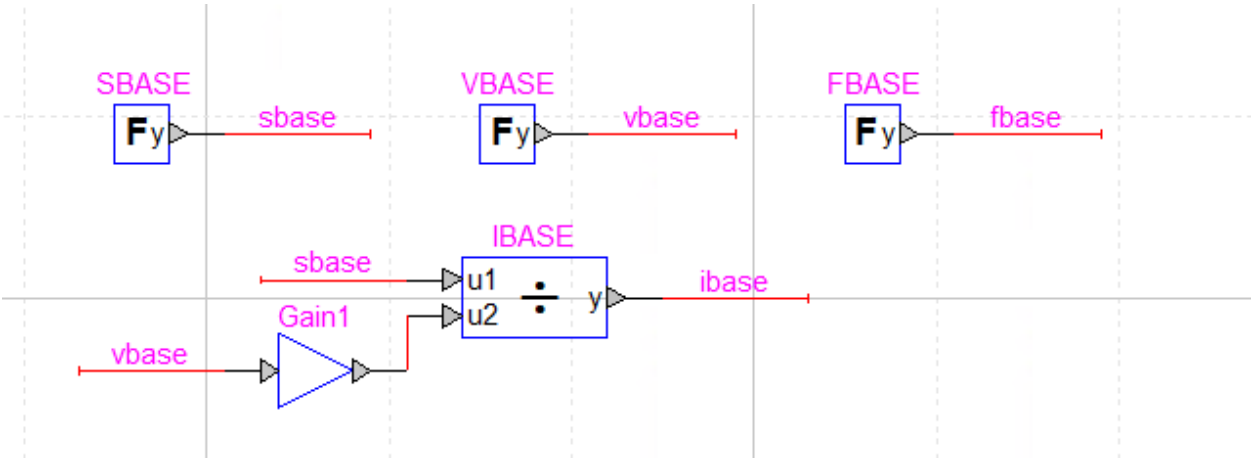


Figure A.3: Diesel synchronous generator system base variables for per-unitizing signals.

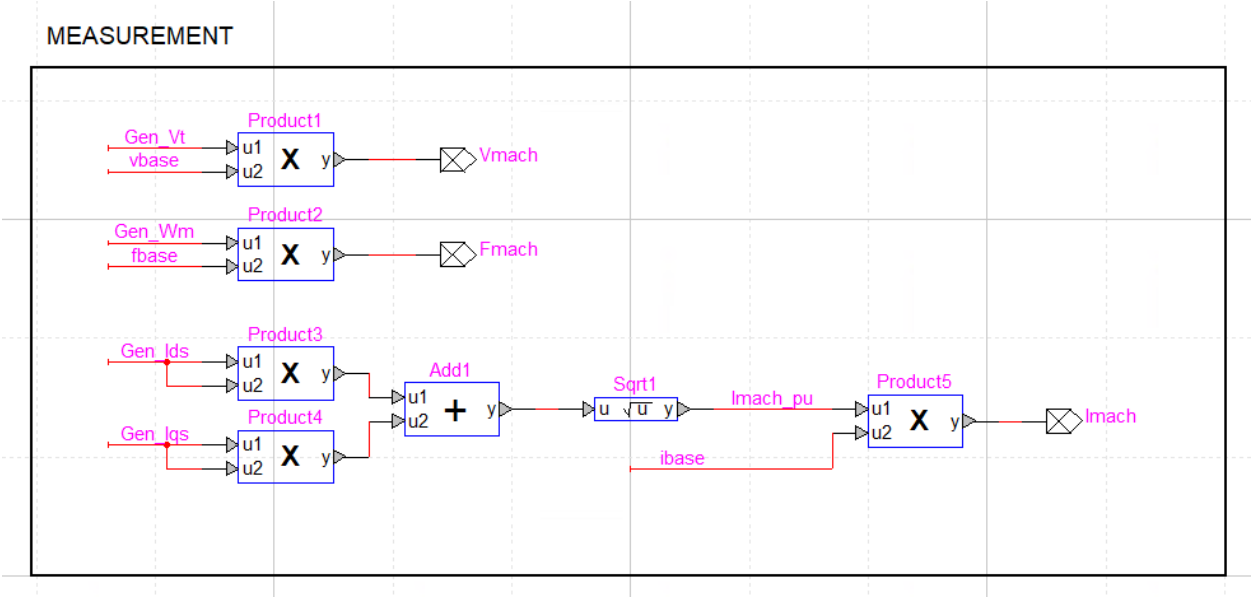


Figure A.4: Diesel synchronous generator measurement of frequency, RMS voltage, and RMS current.

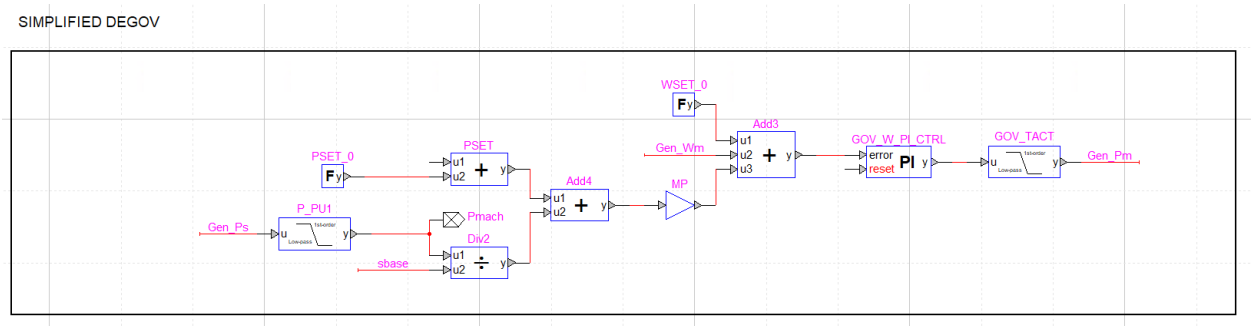


Figure A.5: Diesel synchronous generator governor and prime mover model.

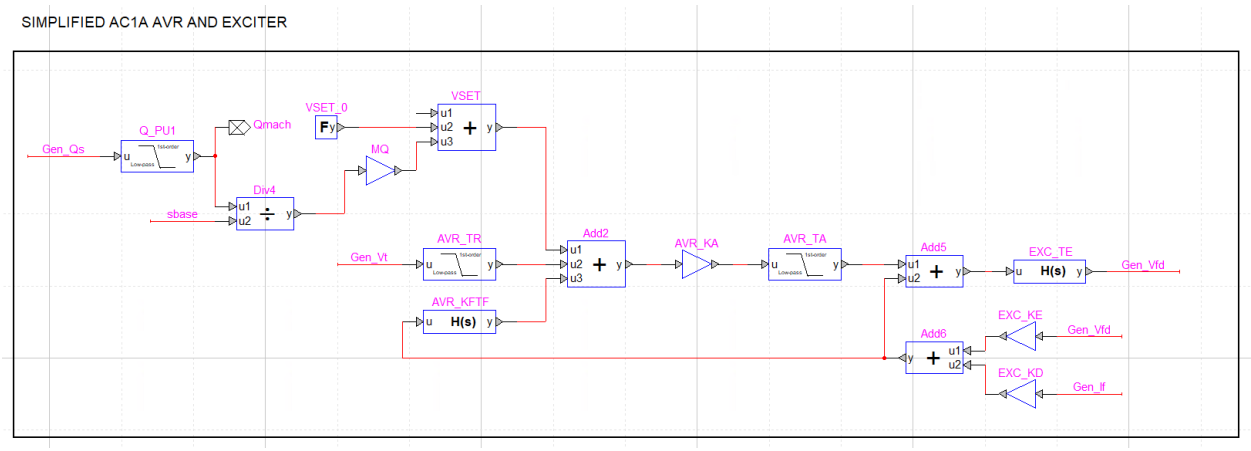


Figure A.6: Diesel synchronous generator AVR and exciter model.

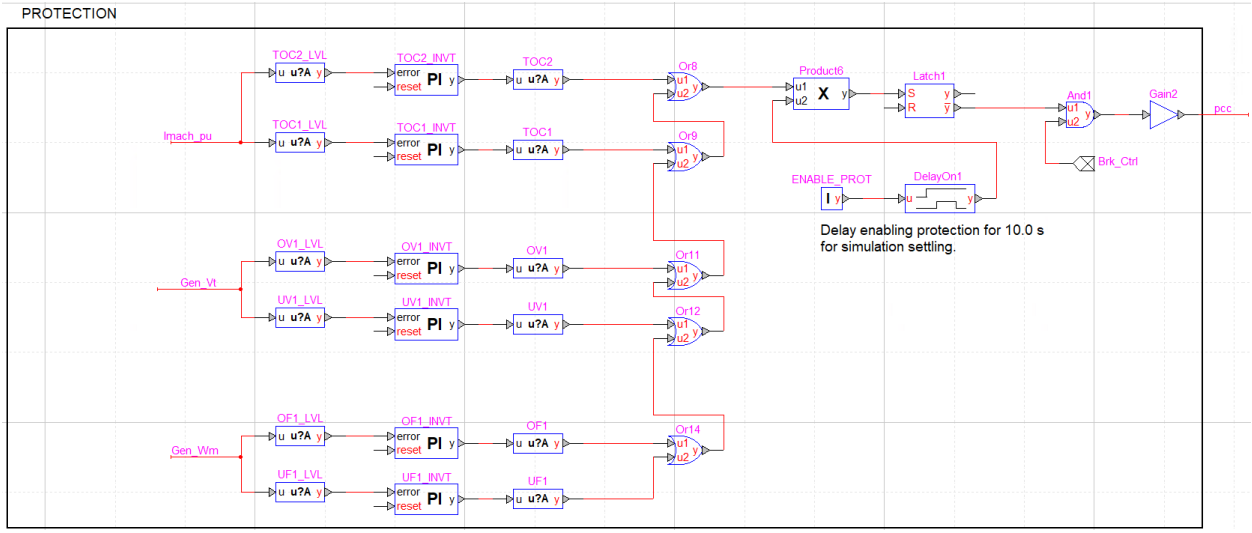


Figure A.7: Diesel synchronous generator protection logic circuit.

A.2 Three Phase GFM Inverter Block Diagrams and Default Parameters

Table A.2: Three Phase GFM Inverter Default Parameters.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Grid Connection							
Volt.	12.47 kV	SCC	15.0 MVA	X/R	10		
Step Up Transformer							
Pri. Voltage	12.47 kV	Sec. Voltage	0.48 kV	Pri. Winding	WYE−G	Sec. Winding	DELTA
Rating	1.5 MVA	Z	5.75%	X/R	4.7		
LCL Filter							
X_1	0.05 pu	R_1	0.005 pu	X_2	0.05 pu	R_2	0.005 pu
C	0.0521 pu	R_C	0.0001 pu				
Droop Controllers							
m_p	0.05 pu	m_q	0.05 pu	ω_0	1.0 pu	V_{set}	1.0 pu
Voltage Controller							
K_p	0.0 pu	K_i	20.0 pu/s	$+Lim$	1.5 pu	$-Lim$	0.0 pu
Low Pass Filters							
T_v	0.01 s	T_i	0.0004 s	T_P	0.01 s	T_Q	0.01 s
Protection System							
TOC1		1.2 pu, 1.0 s		IOC		2.0 pu, 0.0 s	
OV1		1.1 pu, 12.0 s		OV2		1.2 pu, 0.0 s	
UV1		0.88 pu, 20.0 s		UV2		0.7 pu, 10.0 s	
UV3		0.5 pu, 1.0 s		OF1		62.0 Hz, 0.0 s	
UF1		57.0 Hz, 0.0 s					

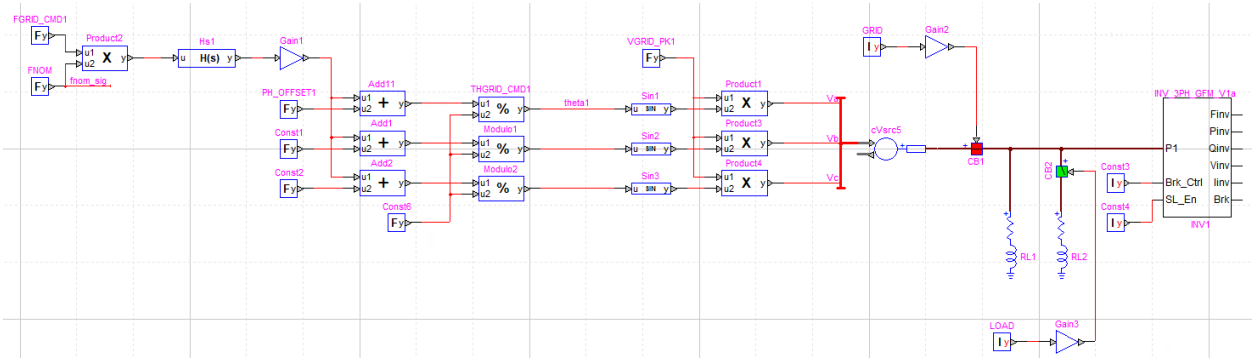


Figure A.8: Three phase GFM inverter model subcircuit and test circuit model.

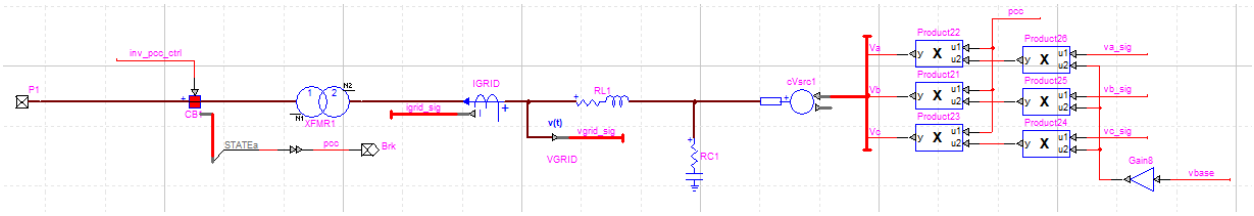


Figure A.9: Three phase GFM inverter physical model.

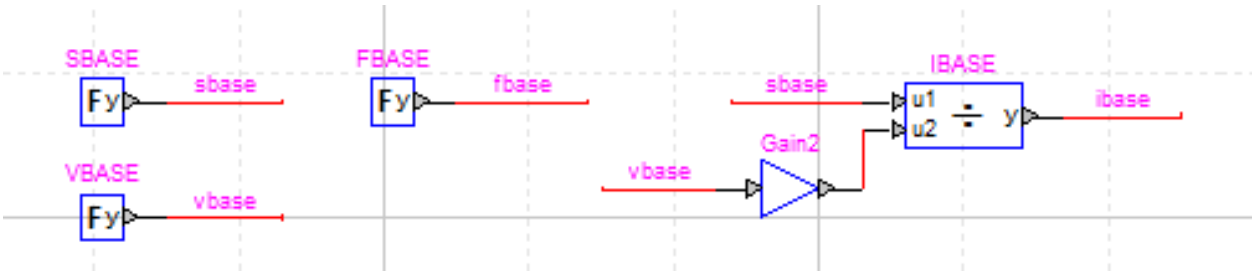


Figure A.10: Three phase GFM inverter system base variables for per-unitizing signals.

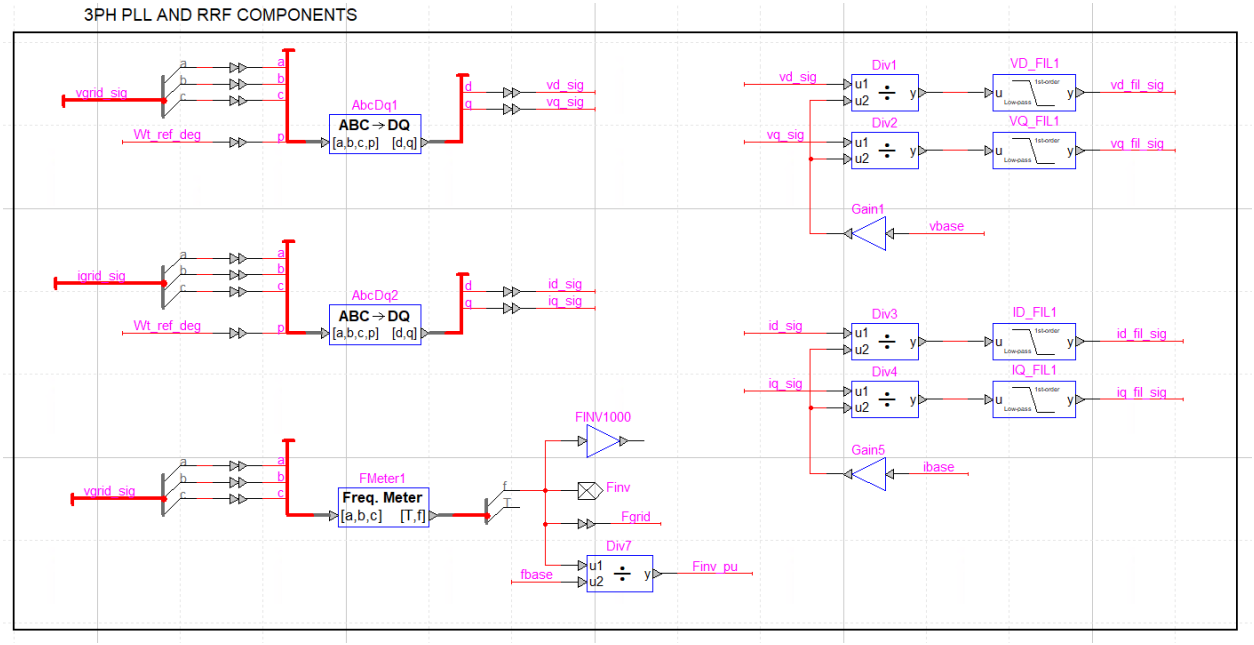


Figure A.11: Three phase GFM inverter calculation of rotating reference frame voltage and current components.

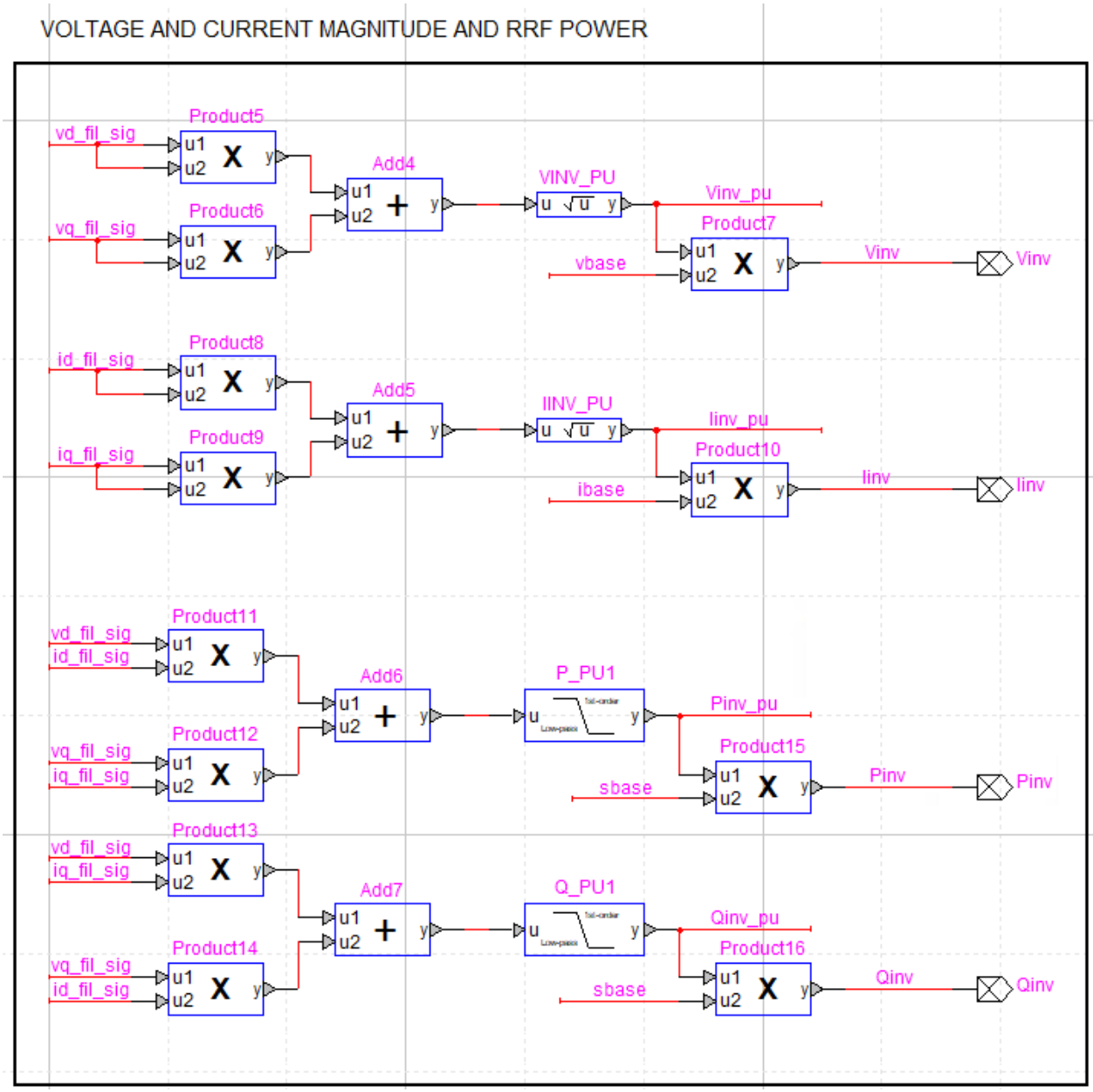


Figure A.12: Three phase GFM inverter measurement of RMS voltage, RMS current, instantaneous real power, and instantaneous reactive power.

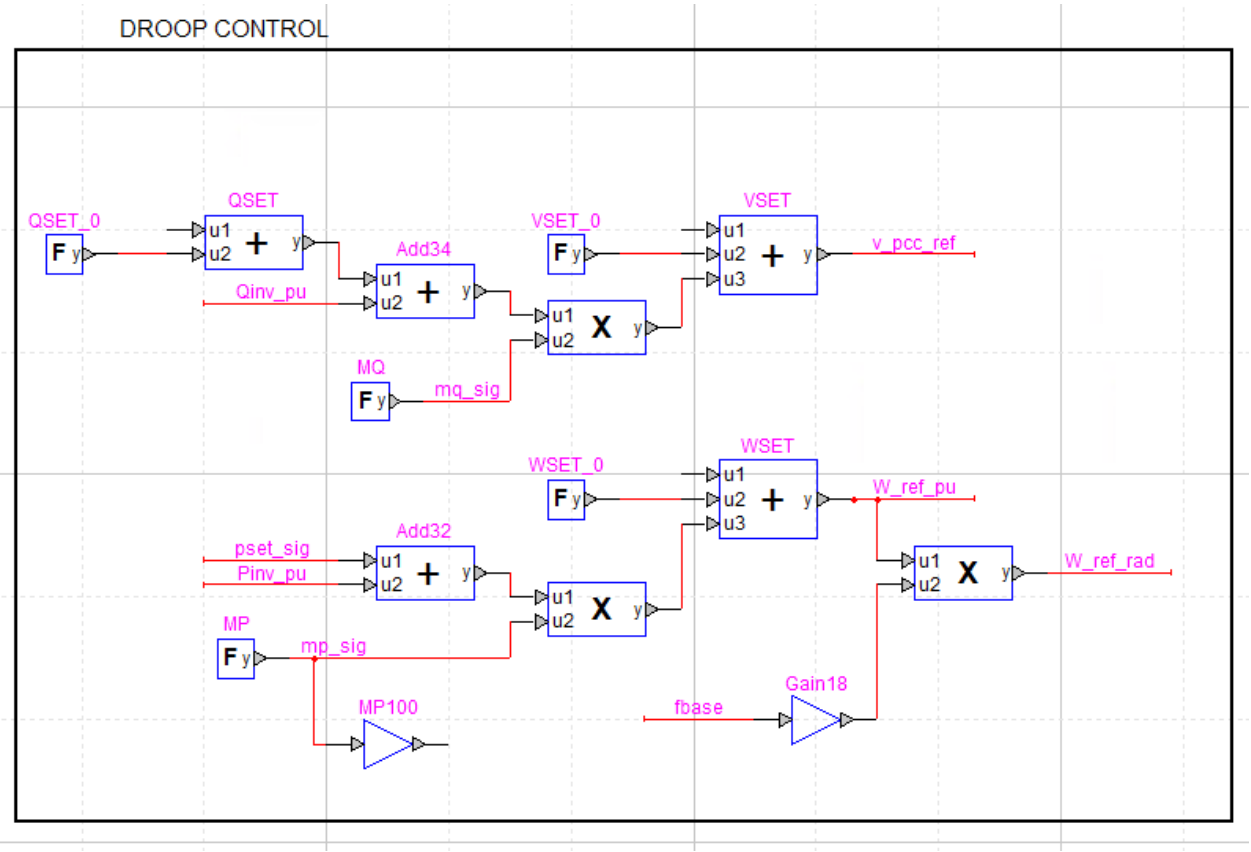


Figure A.13: Three phase GFM inverter droop control.

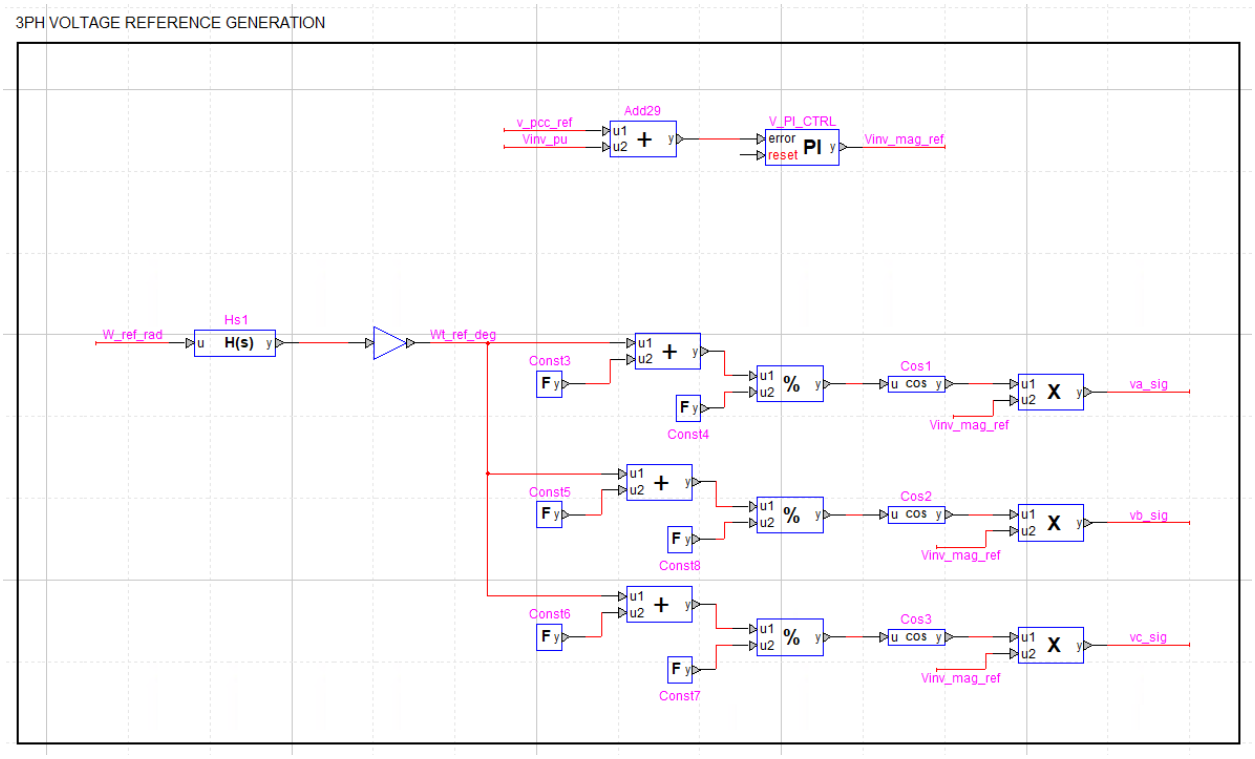


Figure A.14: Three phase GFM inverter voltage control and modulation signals.

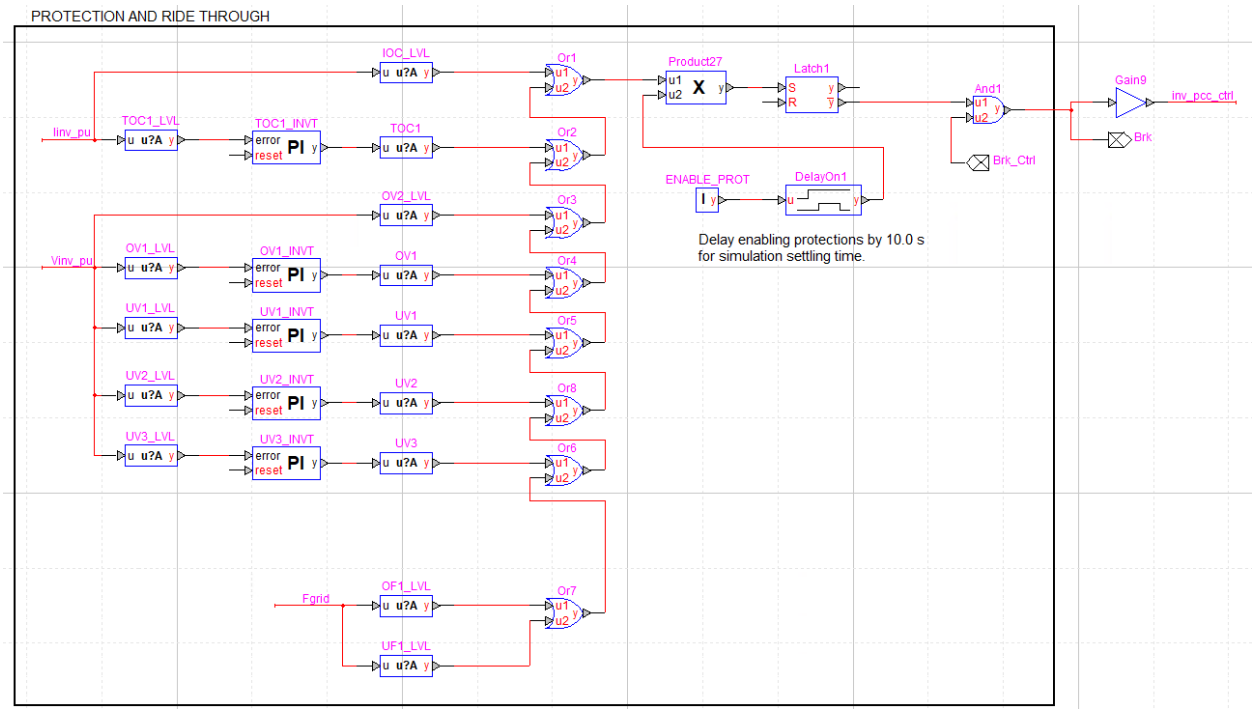


Figure A.15: Three phase GFM inverter protection logic circuits.

A.3 Three Phase GFL Inverter Block Diagrams and Default Parameters

Table A.3: Three Phase GFL Inverter Default Parameters.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Grid Connection							
Volt.	12.47 kV	SCC	15.0 MVA	X/R	10		
Step Up Transformer							
Pri. Voltage	12.47 kV	Sec. Voltage	0.48 kV	Pri. Winding	WYE−G	Sec. Winding	DELTA
Rating	1.5 MVA	Z	5.75%	X/R	4.7		
LCL Filter							
X_1	0.05 pu	R_1	0.005 pu	X_2	0.05 pu	R_2	0.005 pu
C	0.0521 pu	R_C	0.0001 pu				
Active and Reactive Power Controllers							
K_p	0.0 pu	K_i	5.0 pu/s				
Current Controllers							
K_p	1.0 pu	K_i	10.0 pu/s	$I_q + Lim$	1.1 pu	$I_q - Lim$	-1.1 pu
$I_d + Lim$	$\sqrt{1.1^2 - I_q^2}$	$I_d - Lim$	0.0 pu				
Low Pass Filters and PLL							
T_v	0.01 s	T_i	0.0004 s	T_P	0.01 s	T_Q	0.01 s
$K_{p,pll}$	400 rad/pu	$K_{i,pll}$	40e3 rad/pu*s				
Protection System							
TOC1		1.2 pu, 1.0 s		IOC		2.0 pu, 0.0 s	
OV1		1.1 pu, 12.0 s		OV2		1.2 pu, 0.0 s	
UV1		0.88 pu, 20.0 s		UV2		0.7 pu, 10.0 s	
UV3		0.5 pu, 1.0 s		OF1		62.0 Hz, 0.0 s	
UF1		57.0 Hz, 0.0 s					

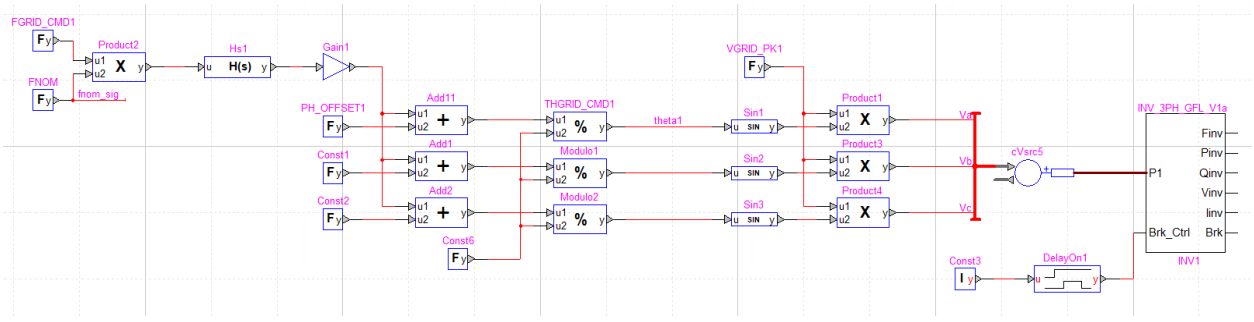


Figure A.16: Three phase GFL inverter model subcircuit and test circuit model.

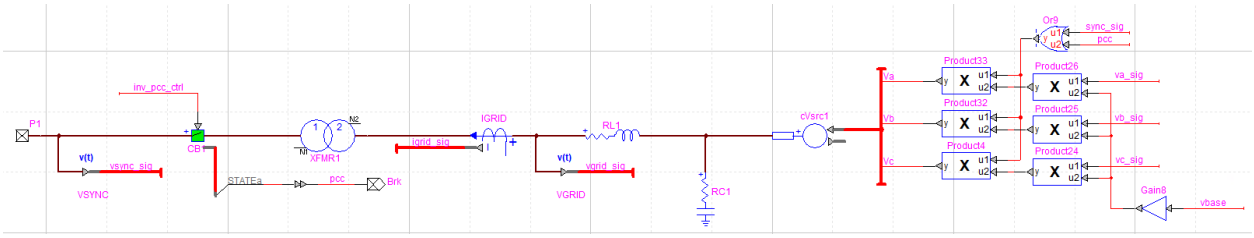


Figure A.17: Three phase GFL inverter physical model.

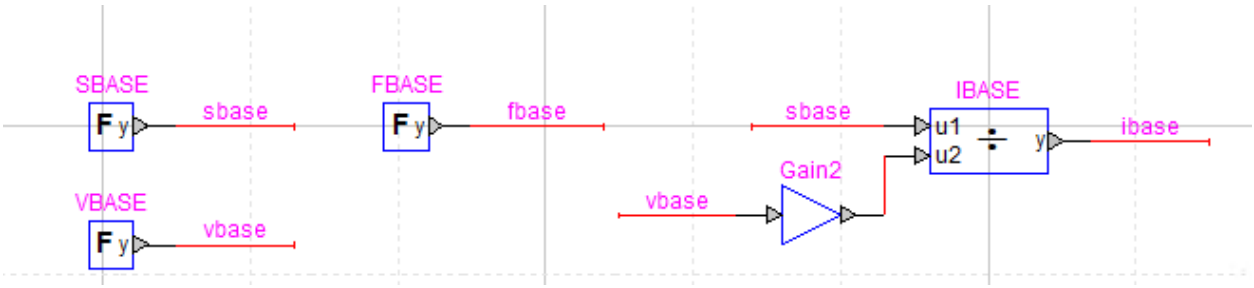


Figure A.18: Three phase GFL inverter system base variables for per-unitizing signals.

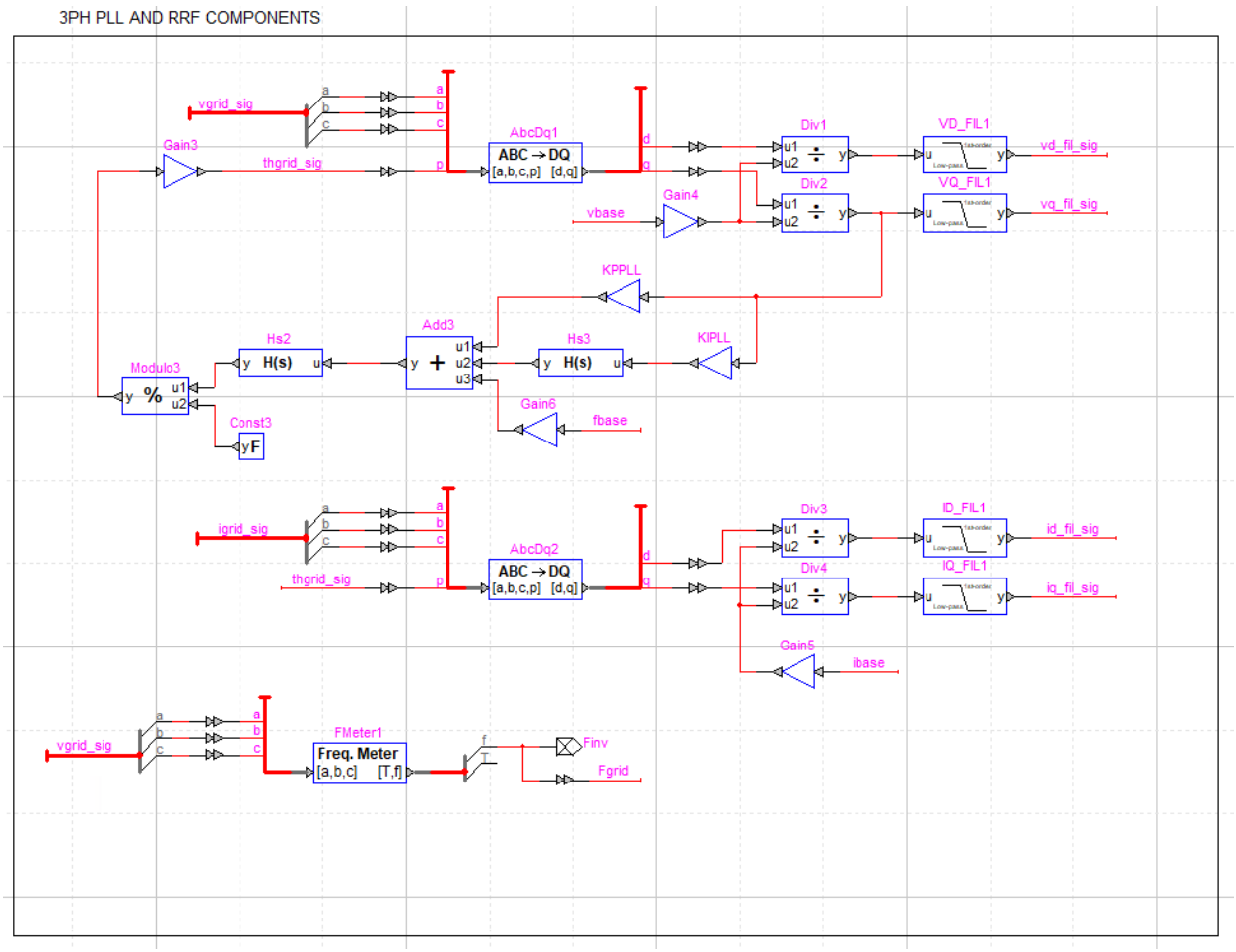


Figure A.19: Three phase GFL inverter calculation of rotating reference frame voltage and current components using three phase, synchronous reference frame phase locked loop (SRFPLL).

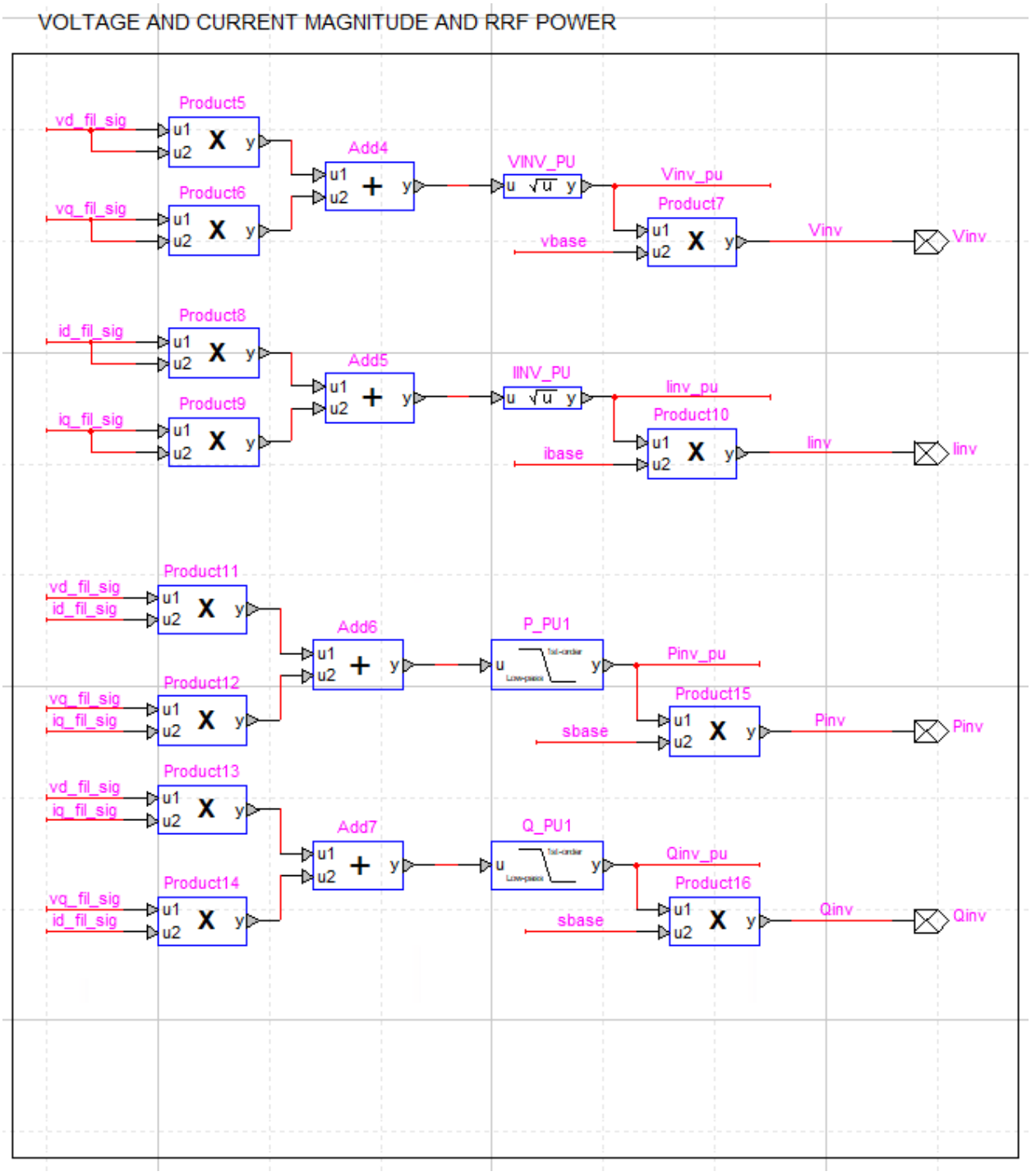


Figure A.20: Three phase GFL inverter measurement of RMS voltage, RMS current, instantaneous real power, and instantaneous reactive power.

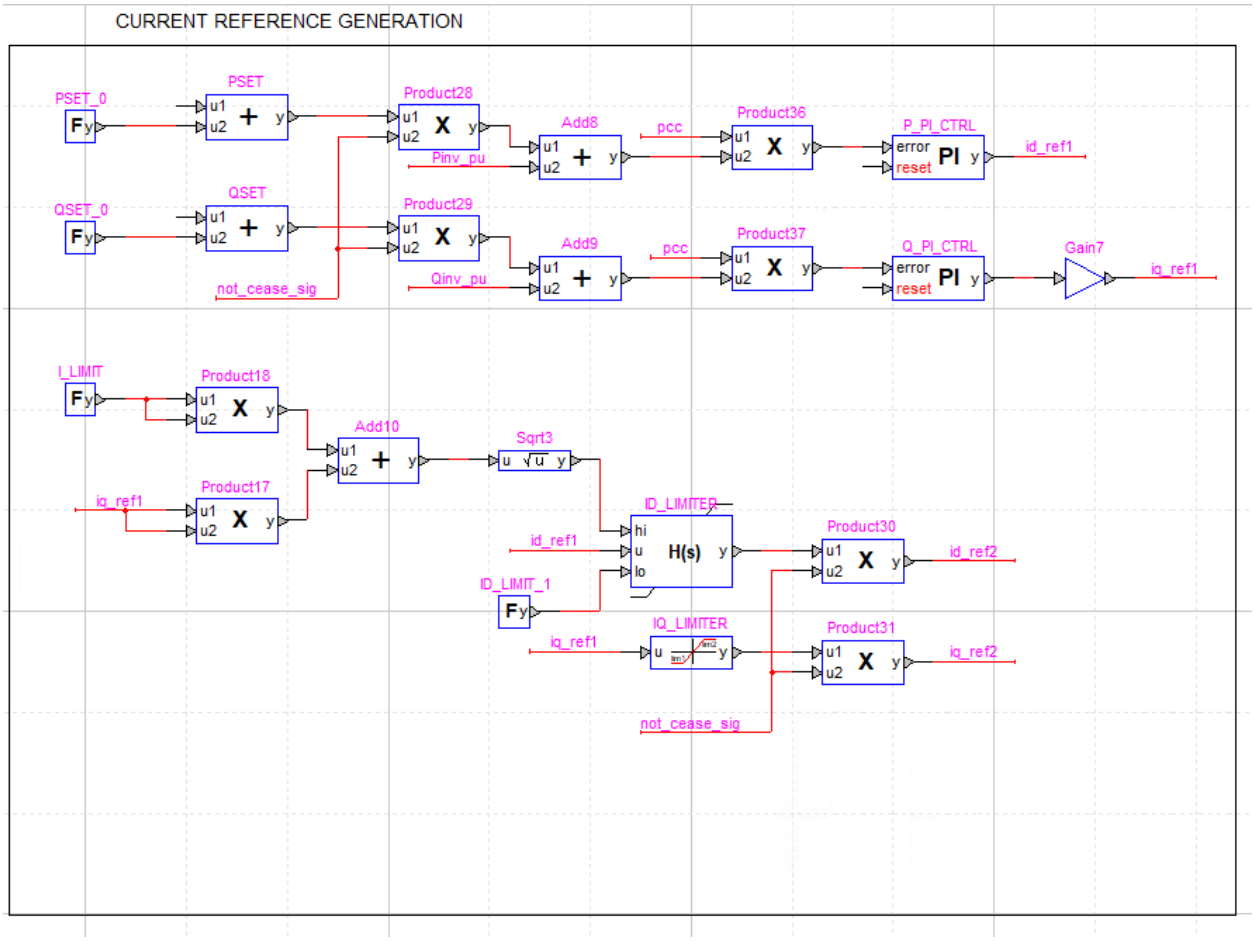


Figure A.21: Three phase GFL inverter active and reactive power controllers.

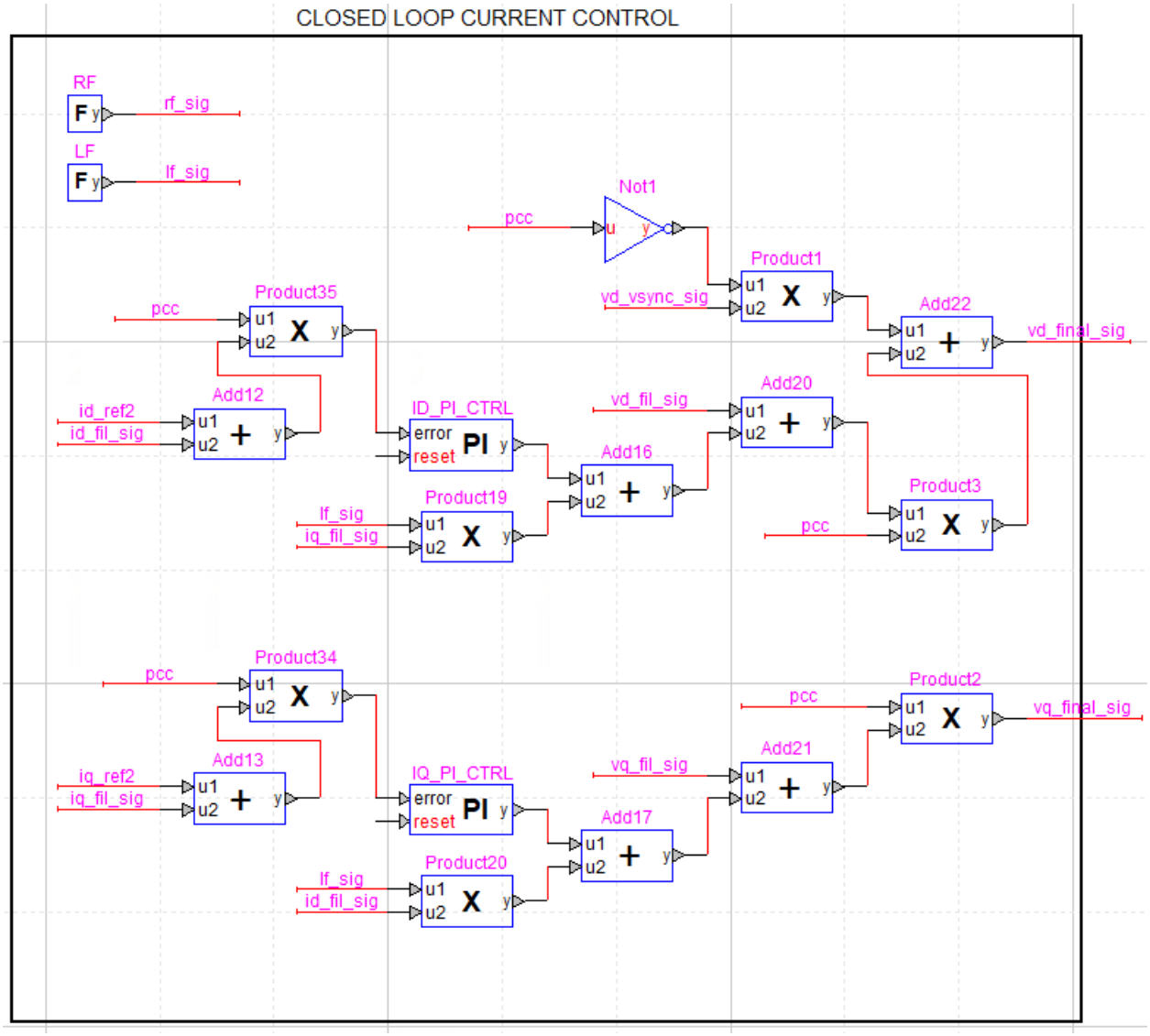


Figure A.22: Three phase GFL inverter d- and q-axis current controllers.

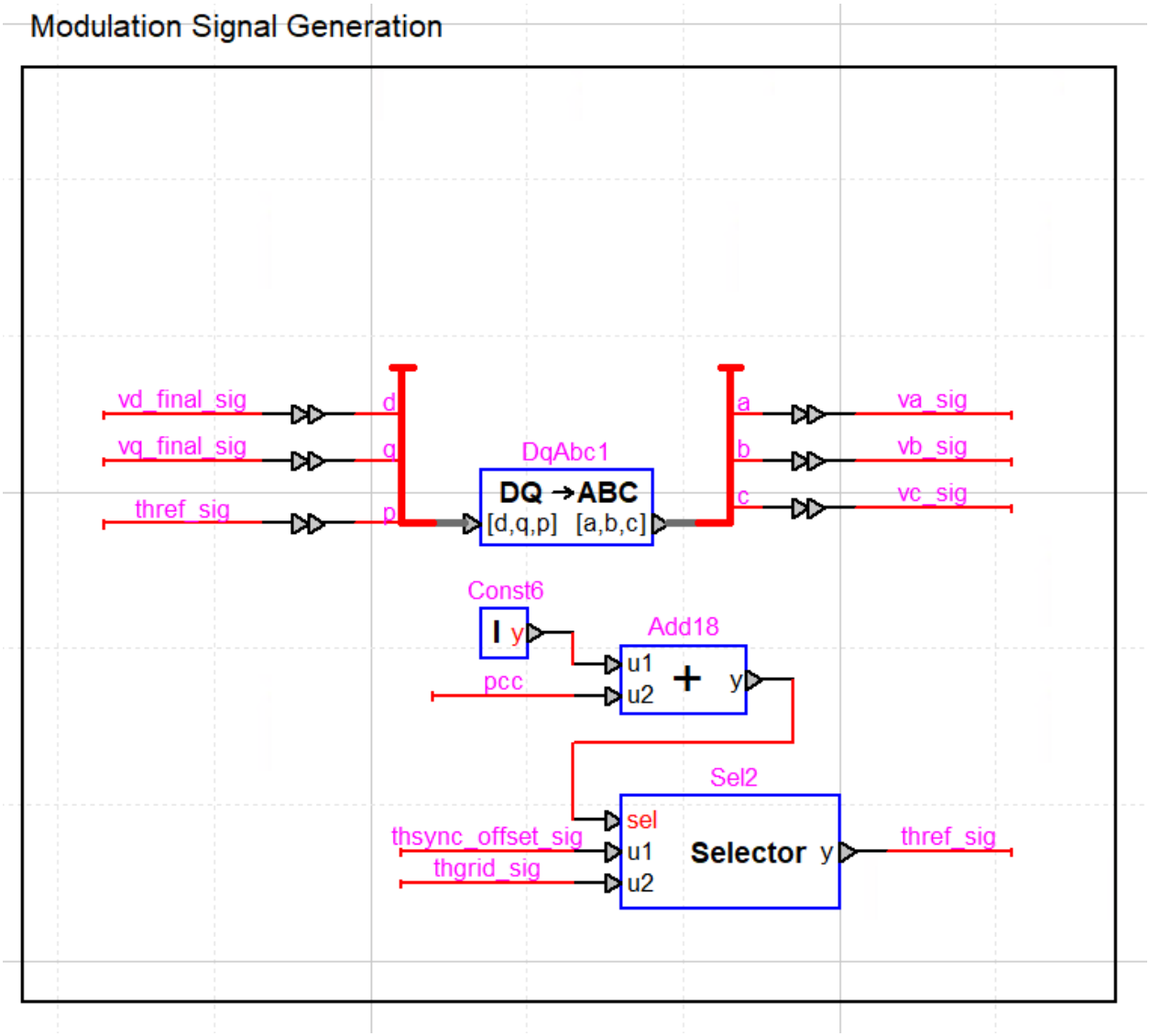


Figure A.23: Three phase GFL inverter voltage modulation signals.

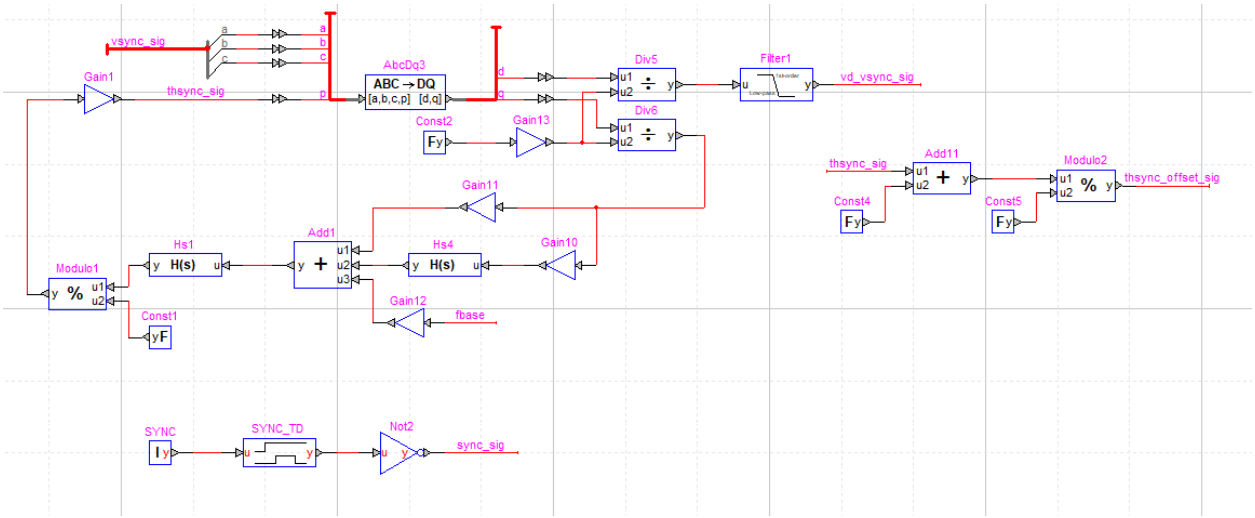


Figure A.24: Three phase GFL inverter scheme for synchronizing to grid voltage and closing PCC circuit breaker.

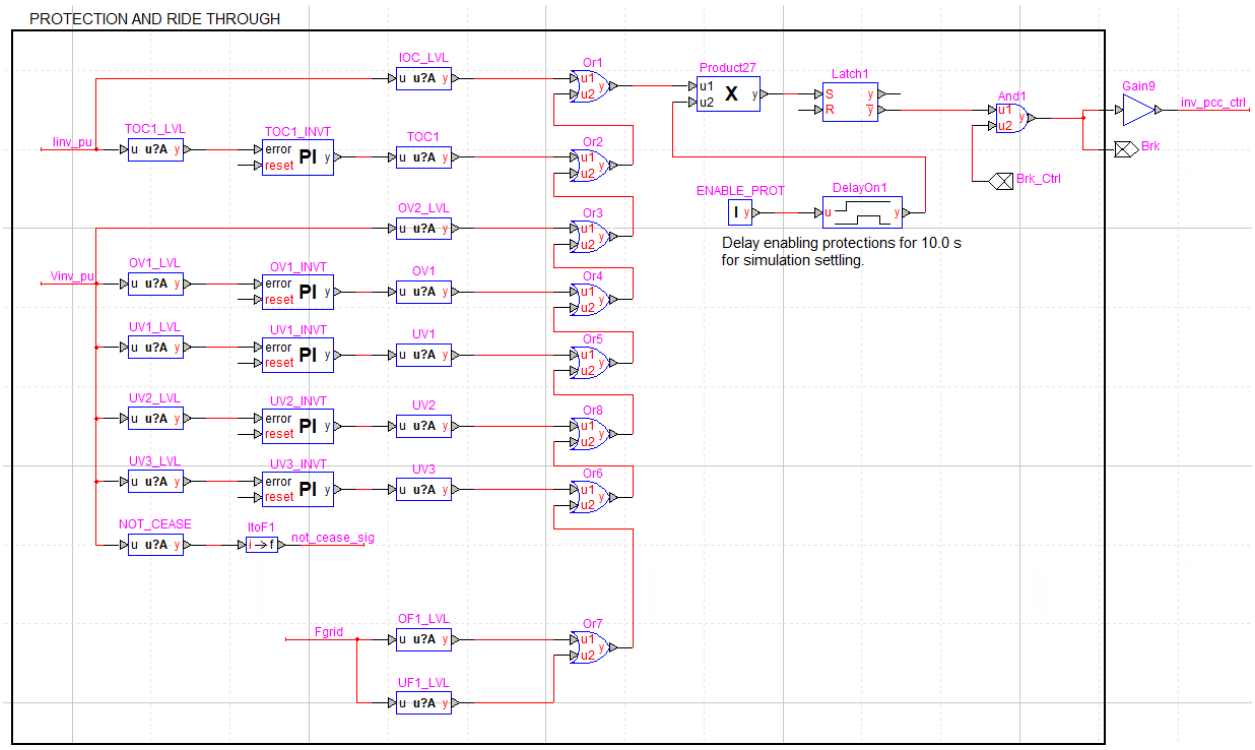


Figure A.25: Three phase GFL inverter protection logic circuits.

A.4 Single Phase GFM Inverter Block Diagrams and Default Parameters

Table A.4: Single Phase GFM Inverter Default Parameters.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Grid Connection							
Volt.	12.47 kV	SCC	0.1 MVA	X/R	5		
Step Up Transformer							
Pri. Voltage	7.2 kV	Sec. Voltage	0.24 kV	Rating	0.01 MVA	Z	2.326%
X/R	2.1						
LCL Filter							
X_1	0.05 pu	R_1	0.005 pu	X_2	0.01 pu	R_2	0.001 pu
C	0.0543 pu	R_C	0.0002 pu				
Droop Controllers							
m_p	0.01 pu	m_q	0.05 pu	ω_0	1.0 pu	V_{set}	1.0 pu
Voltage Controller							
K_p	0.0 pu	K_i	20.0 pu/s	$+Lim$	1.2 pu	$-Lim$	0.0 pu
Low Pass Filters and SOGI-FLL							
T_v	0.01 s	T_i	0.0004 s	T_P	0.01 s	T_Q	0.01 s
γ	-50 pu	ω_0	377 rad/s				
Protection System							
TOC1		1.2 pu, 1.0 s		IOC		2.0 pu, 0.0 s	
OV1		1.15 pu, 0.5 s		OV2		1.2 pu, 0.0 s	
UV1		0.7 pu, 0.16 s		UV2		0.5 pu, 0.0 s	
OF1		62.0 Hz, 0.0 s		UF1		57.0 Hz, 0.0 s	

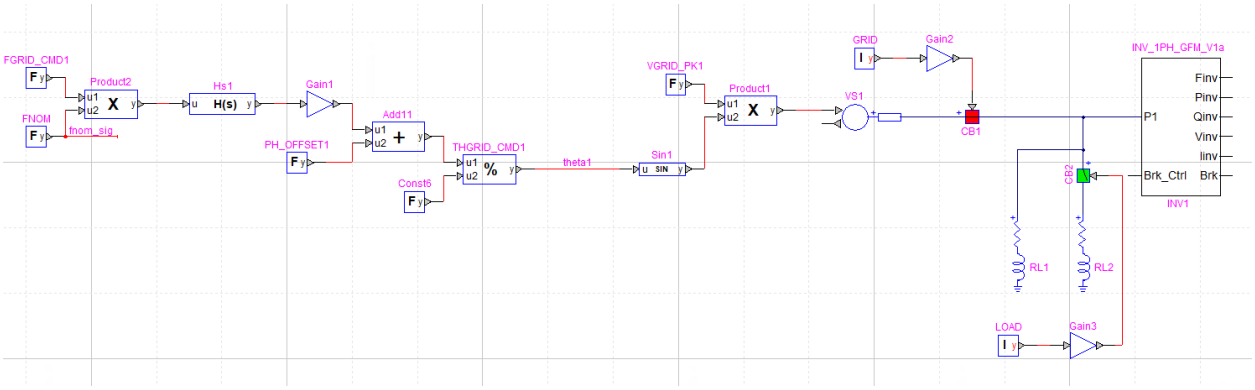


Figure A.26: Single phase GFM inverter model subcircuit and test circuit model.

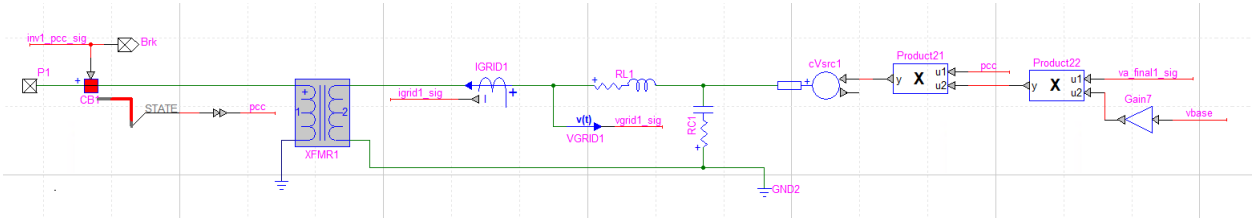
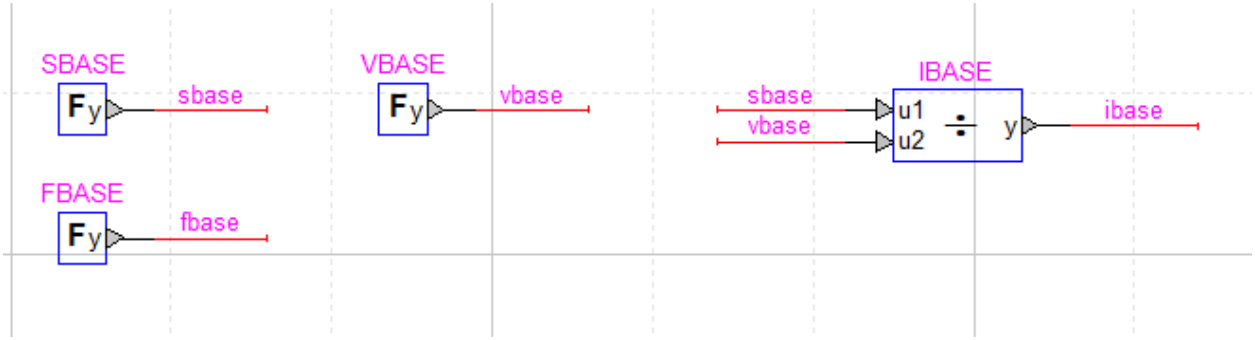


Figure A.28: Single phase GFM inverter system base variables for per-unitizing signals.



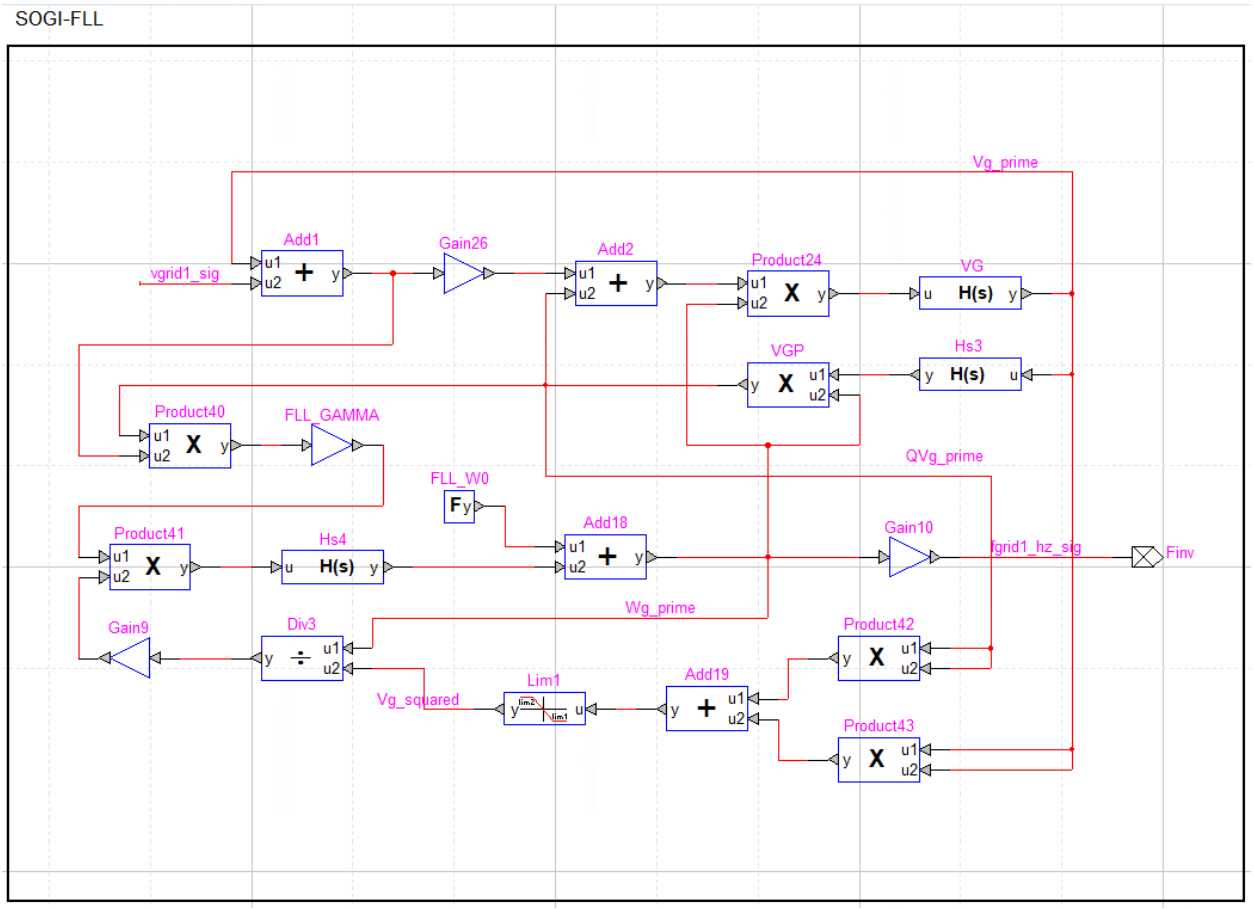


Figure A.29: Single phase GFM inverter measurement of system frequency using Second Order Generalized Integrator Frequency Locked Loop (SOGI-FLL).

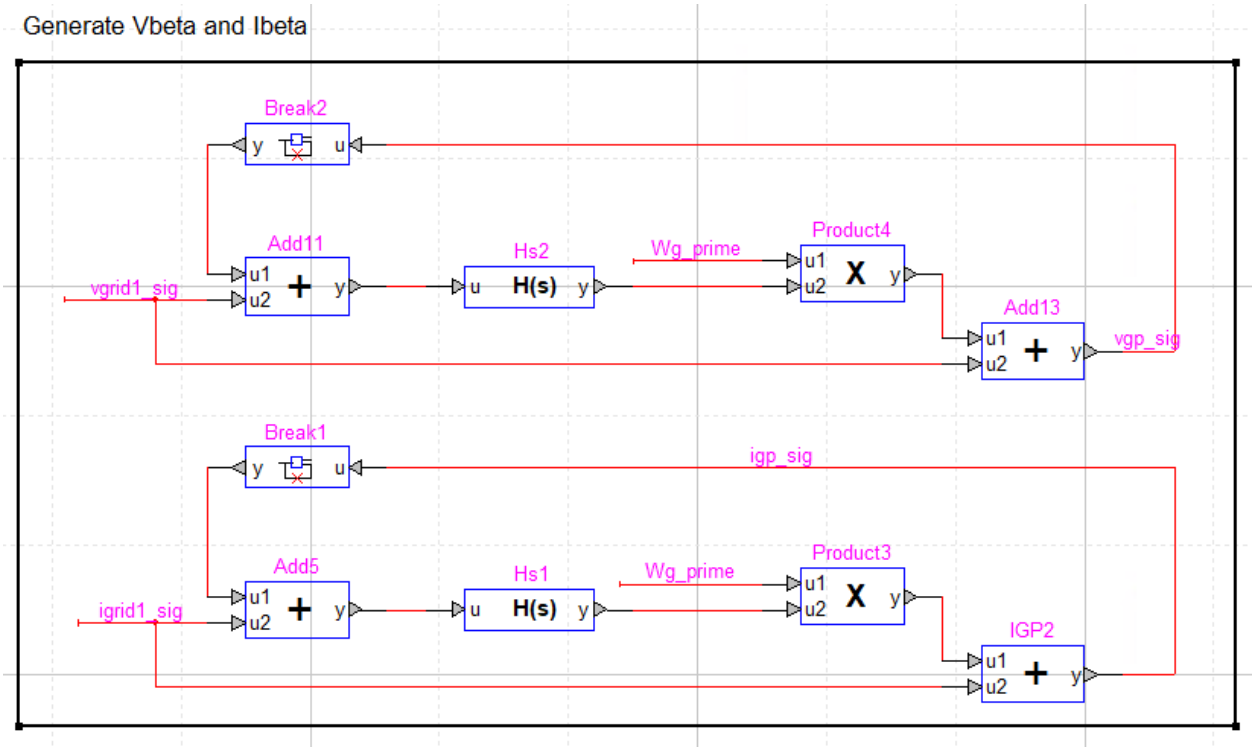


Figure A.30: Single phase GFM inverter calculation of orthogonal components of single phase voltage and current.

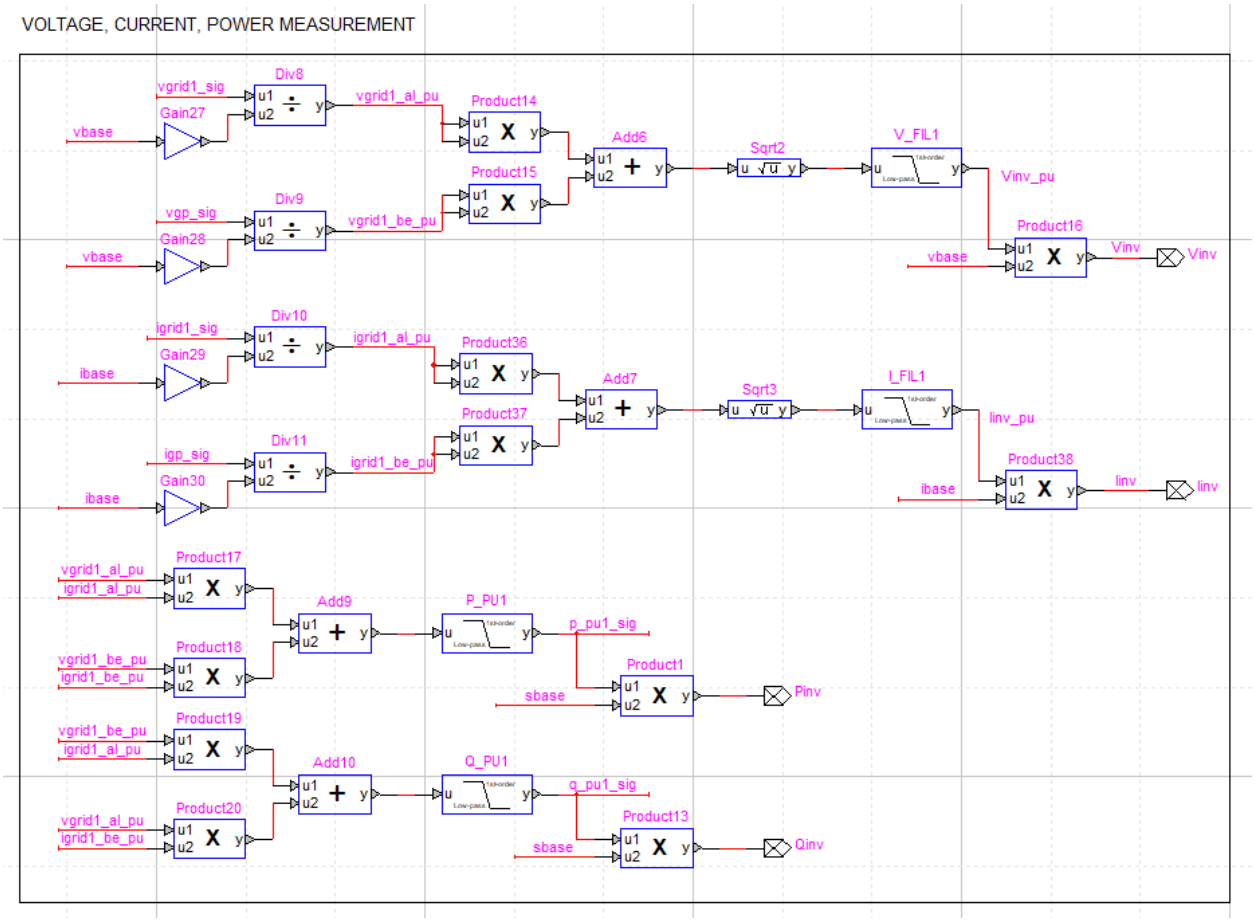


Figure A.31: Single phase GFM inverter measurement of RMS voltage, RMS current, instantaneous real power, and instantaneous reactive power.

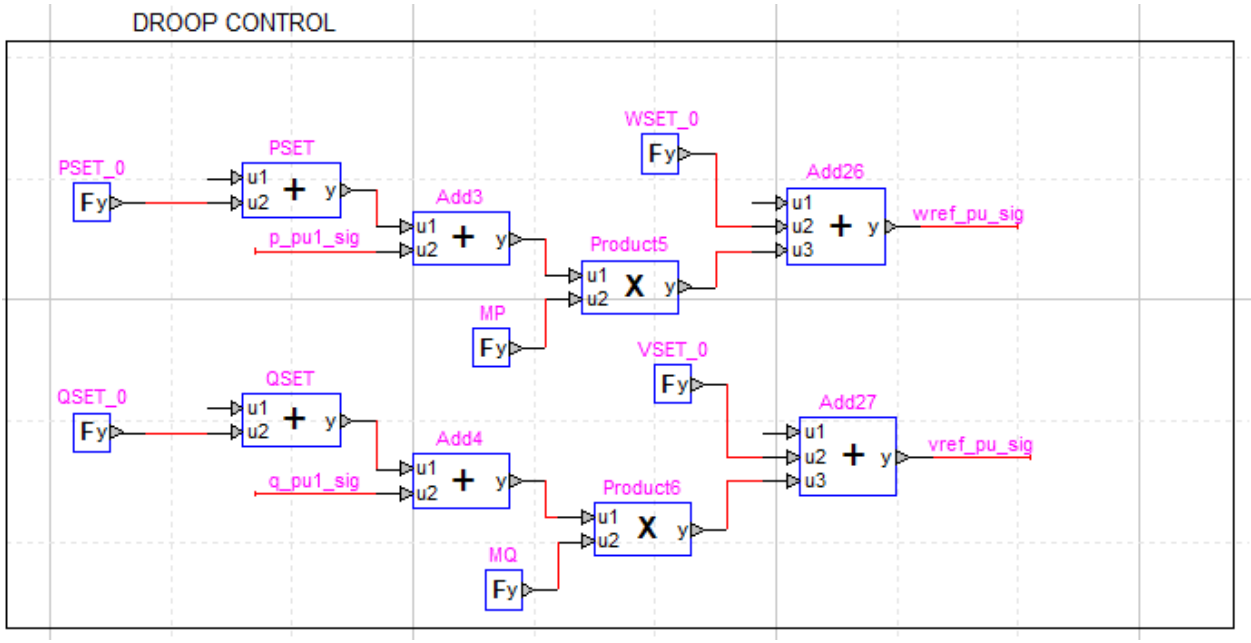


Figure A.32: Single phase GFM inverter droop control.

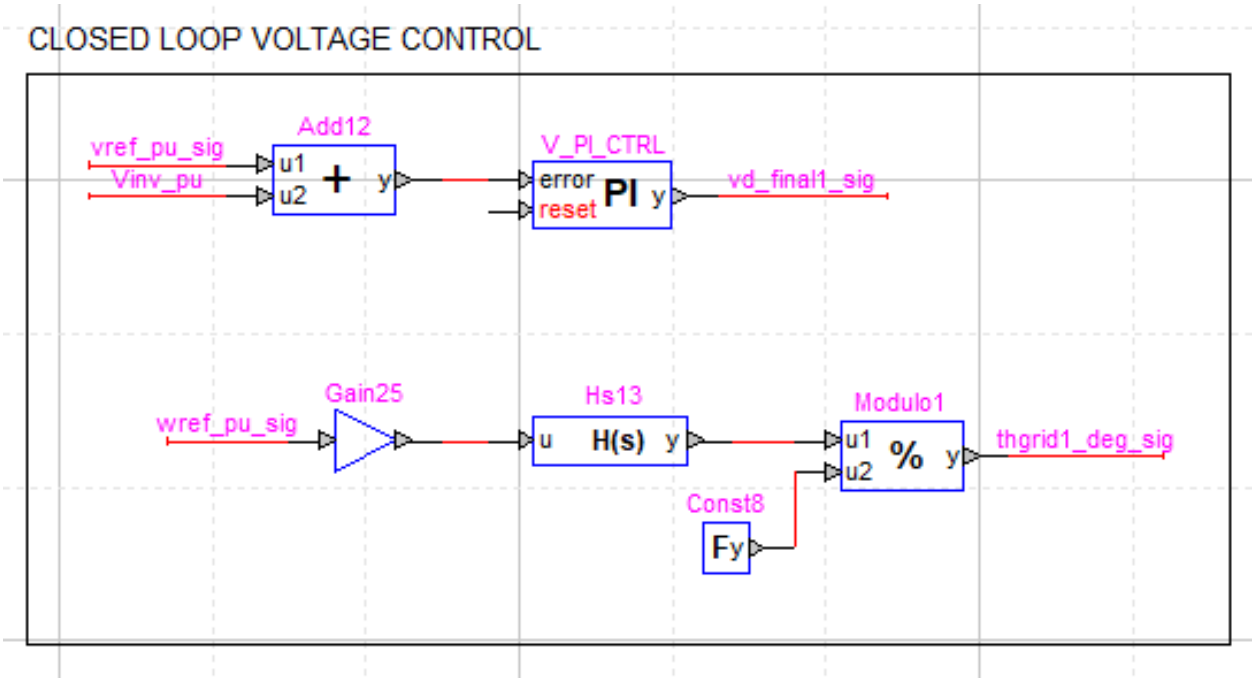


Figure A.33: Single phase GFM inverter voltage control.

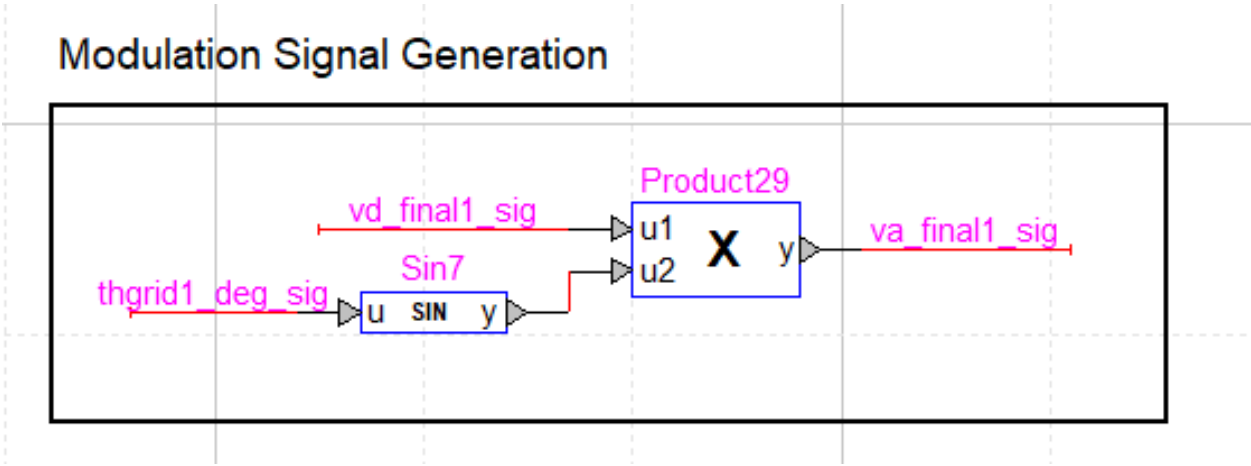


Figure A.34: Single phase GFM inverter generation of modulation signals.

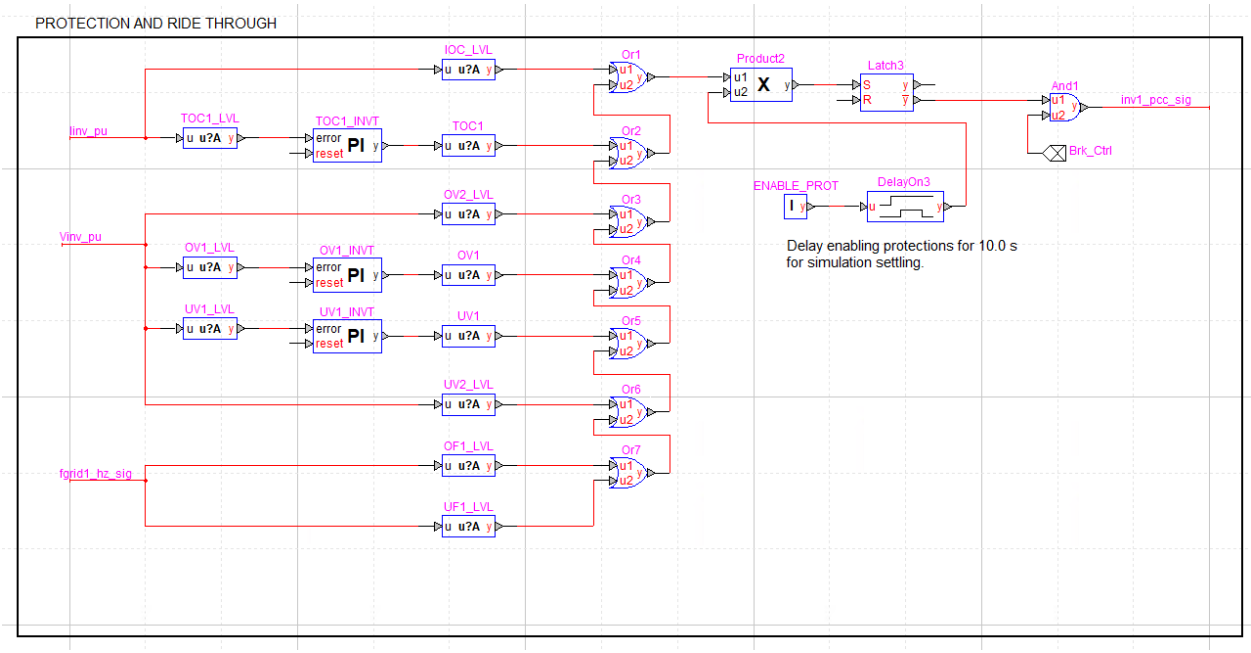


Figure A.35: Single phase GFM inverter protection logic circuits.

A.5 Single Phase GFL Inverter Block Diagrams and Default Parameters

Table A.5: Single Phase GFL Inverter Default Parameters.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Grid Connection							
Volt.	12.47 kV	SCC	0.1 MVA	X/R	5		
Step Up Transformer							
Pri. Voltage	7.2 kV	Sec. Voltage	0.24 kV	Rating	0.01 MVA	Z	2.326%
X/R	2.1						
LCL Filter							
X_1	0.05 pu	R_1	0.005 pu	X_2	0.05 pu	R_2	0.005 pu
C	0.0543 pu	R_C	0.0002 pu				
Active and Reactive Power Controllers							
K_p	0.0 pu	K_i	5.0 pu/s				
Current Controllers							
K_p	1.0 pu	K_i	1.0 pu/s	$I_q + Lim$	1.1 pu	$I_q - Lim$	-1.1 pu
$I_d + Lim$	$\sqrt{1.1^2 - I_q^2}$	$I_d - Lim$	0.0 pu				
Low Pass Filters and Enhanced PLL							
T_v	0.01 s	T_i	0.0004 s	T_P	0.01 s	T_Q	0.01 s
H_0	400 rad/pu	H_1	20e3 rad/pu*s	H_3	200 pu		
Protection System							
TOC1		1.2 pu, 1.0 s		IOC		2.0 pu, 0.0 s	
OV1		1.15 pu, 0.5 s		OV2		1.2 pu, 0.0 s	
UV1		0.7 pu, 0.16 s		UV2		0.5 pu, 0.0 s	
OF1		62.0 Hz, 0.0 s		UF1		57.0 Hz, 0.0 s	

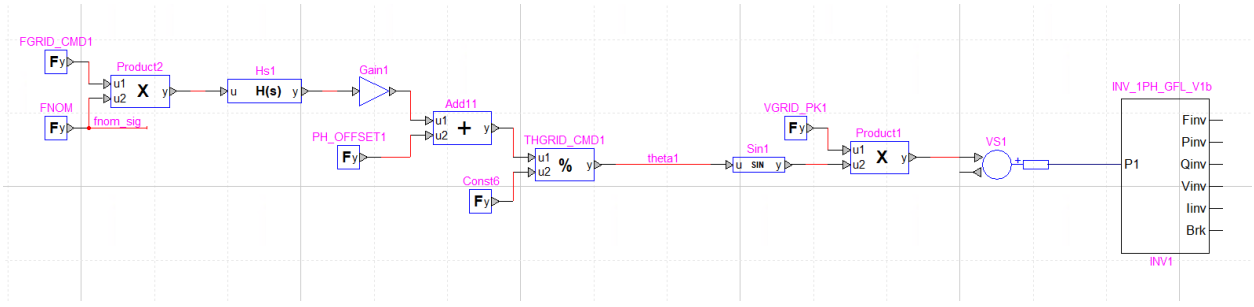


Figure A.36: Single phase GFL inverter model subcircuit and test circuit model.

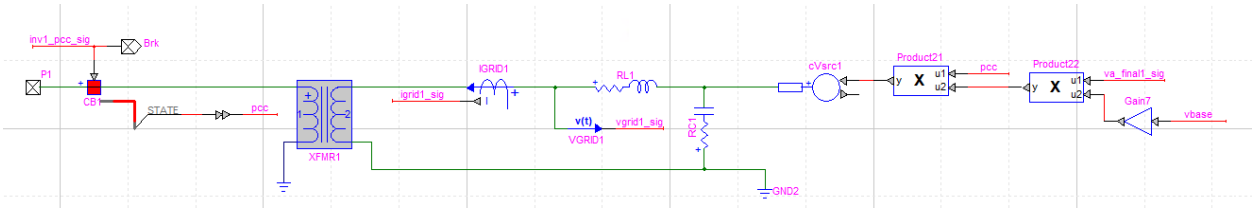


Figure A.37: Single phase GFL inverter physical model.

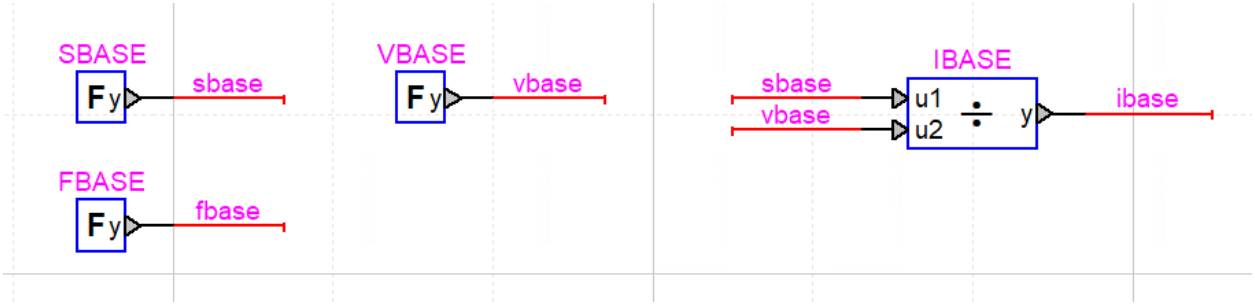


Figure A.38: Single phase GFL inverter system base variables for per-unitizing signals.

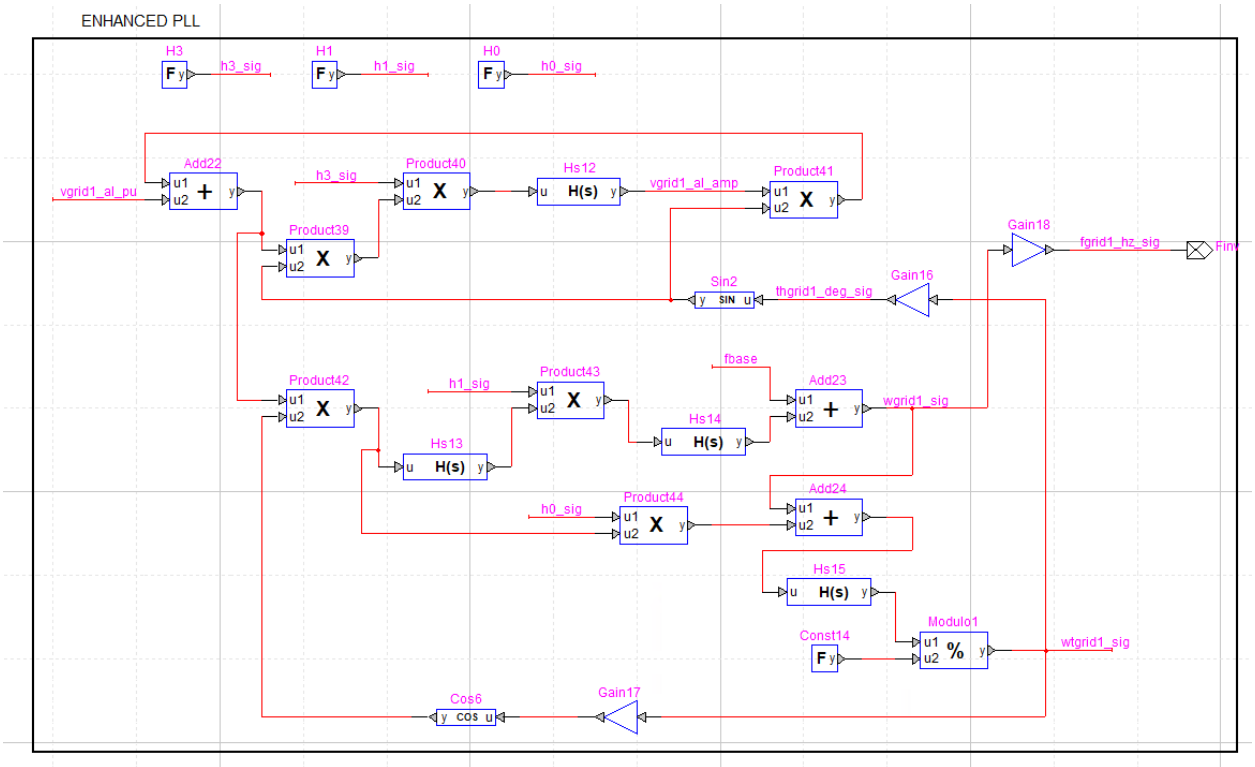


Figure A.39: Single phase GFL inverter calculation of voltage magnitude and phase angle using single phase enhanced phase locked loop (EPLL).

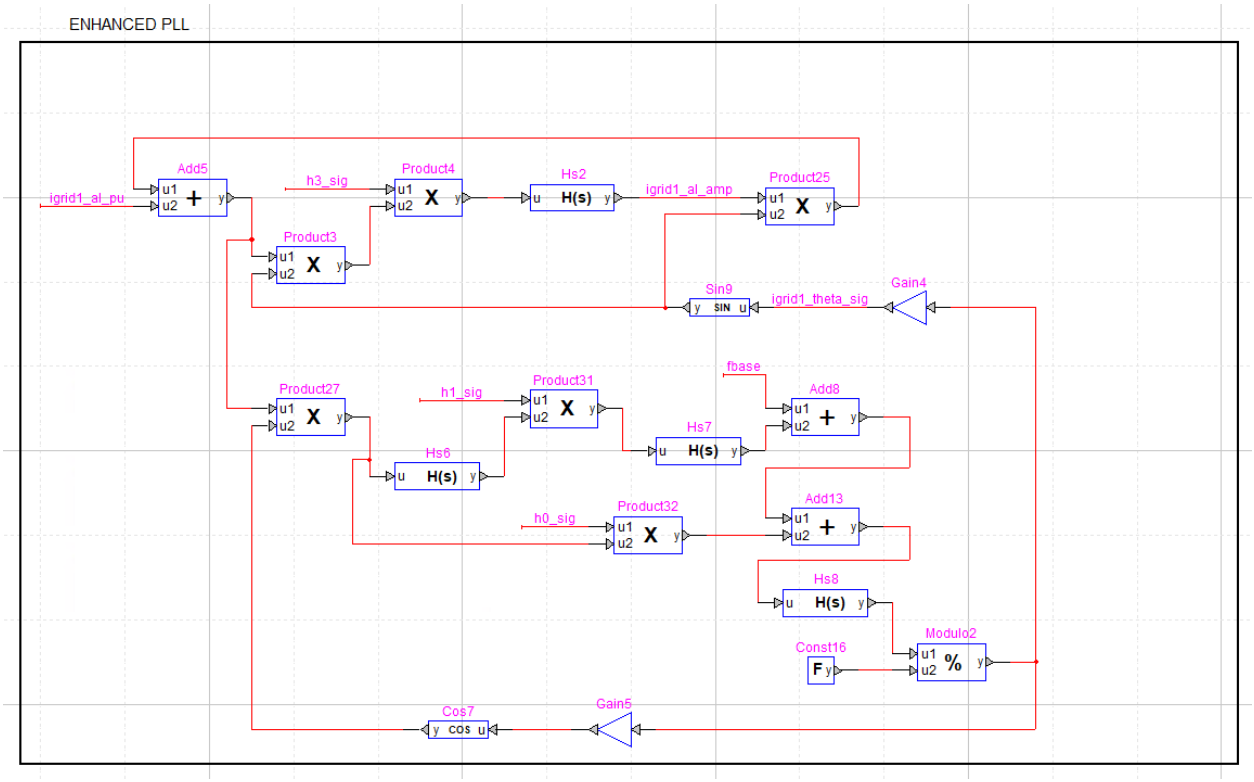


Figure A.40: Single phase GFL inverter calculation of current magnitude and phase angle using single phase enhanced phase locked loop (EPLL).

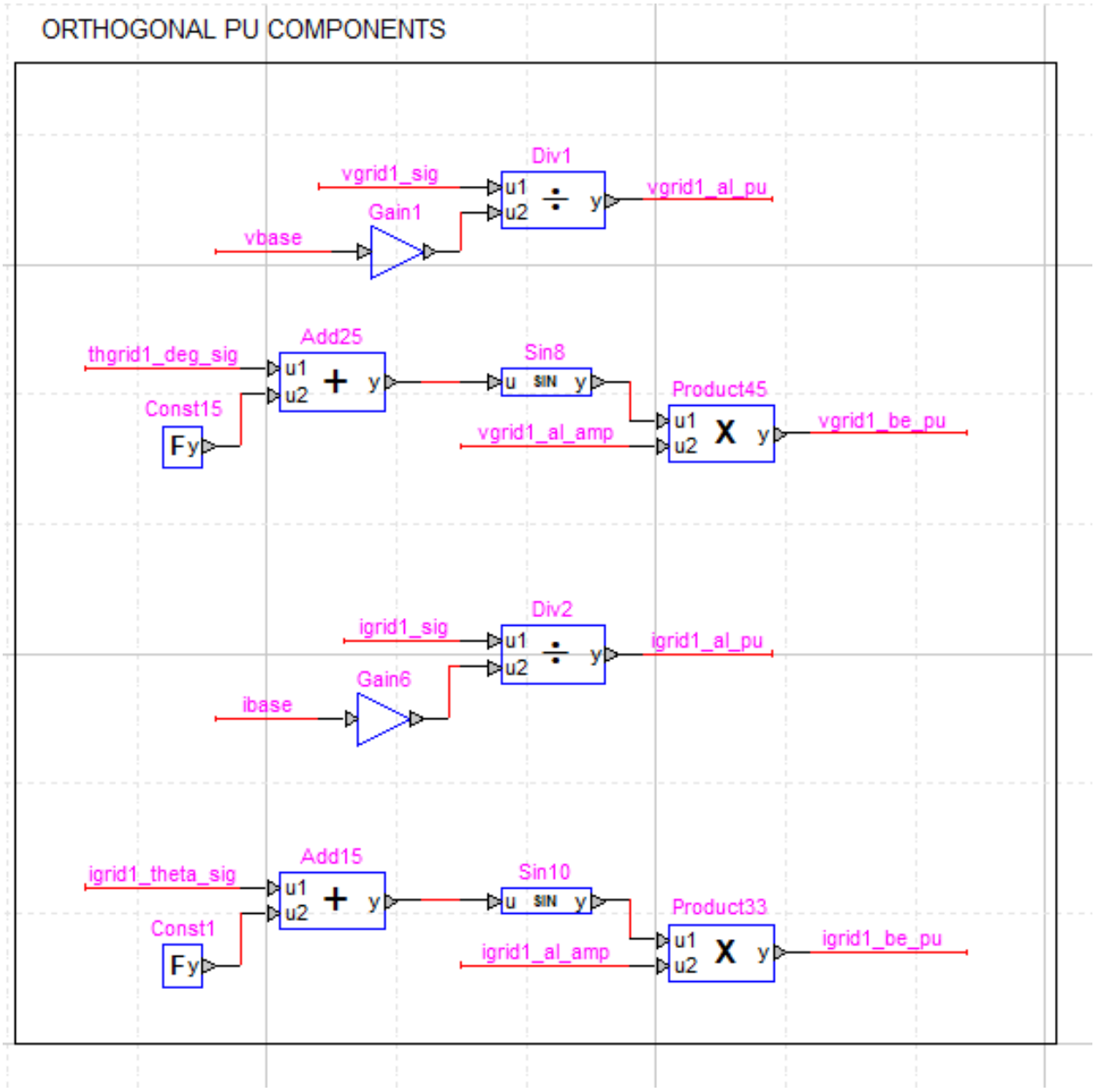


Figure A.41: Single phase GFL inverter generation of orthogonal components of single phase voltage and current.

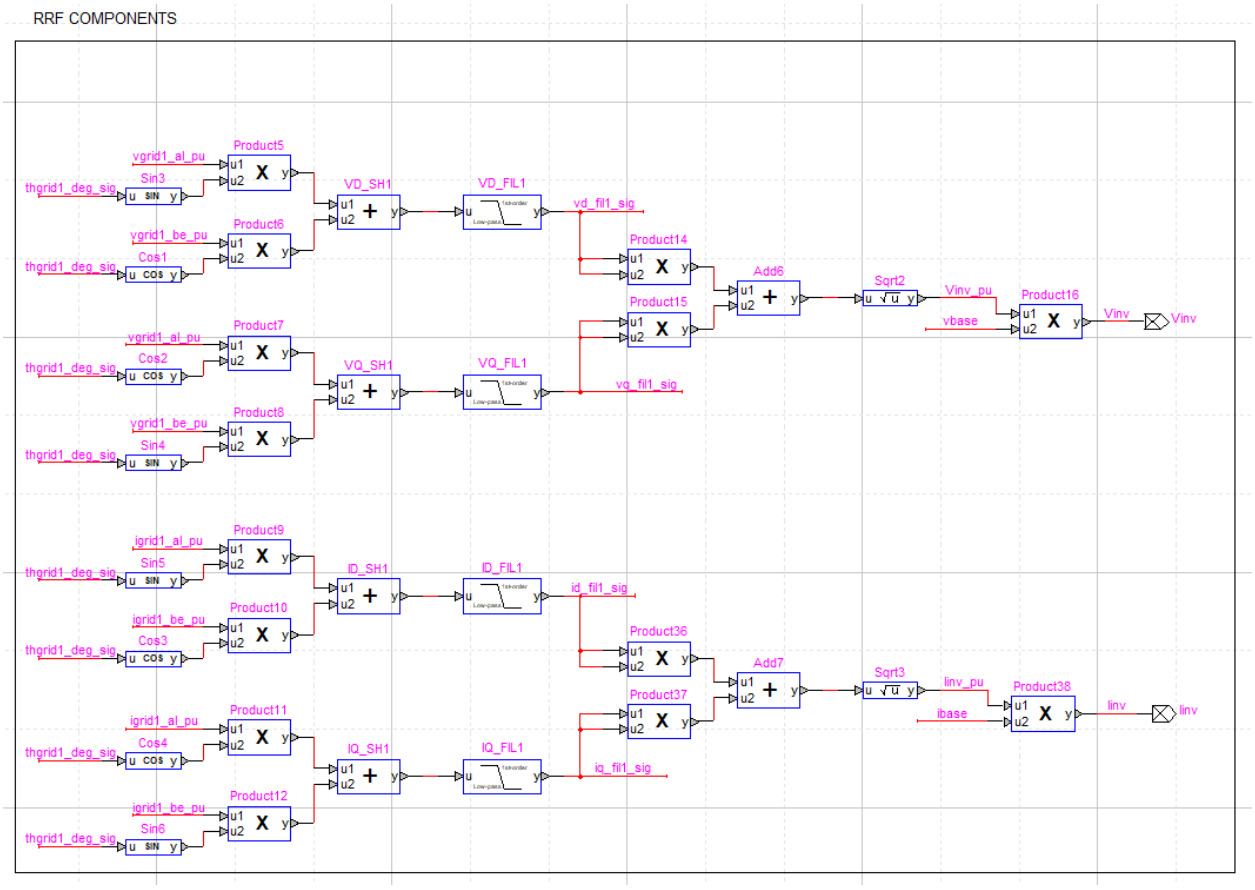


Figure A.42: Single phase GFL inverter measurement of RMS voltage and RMS current.

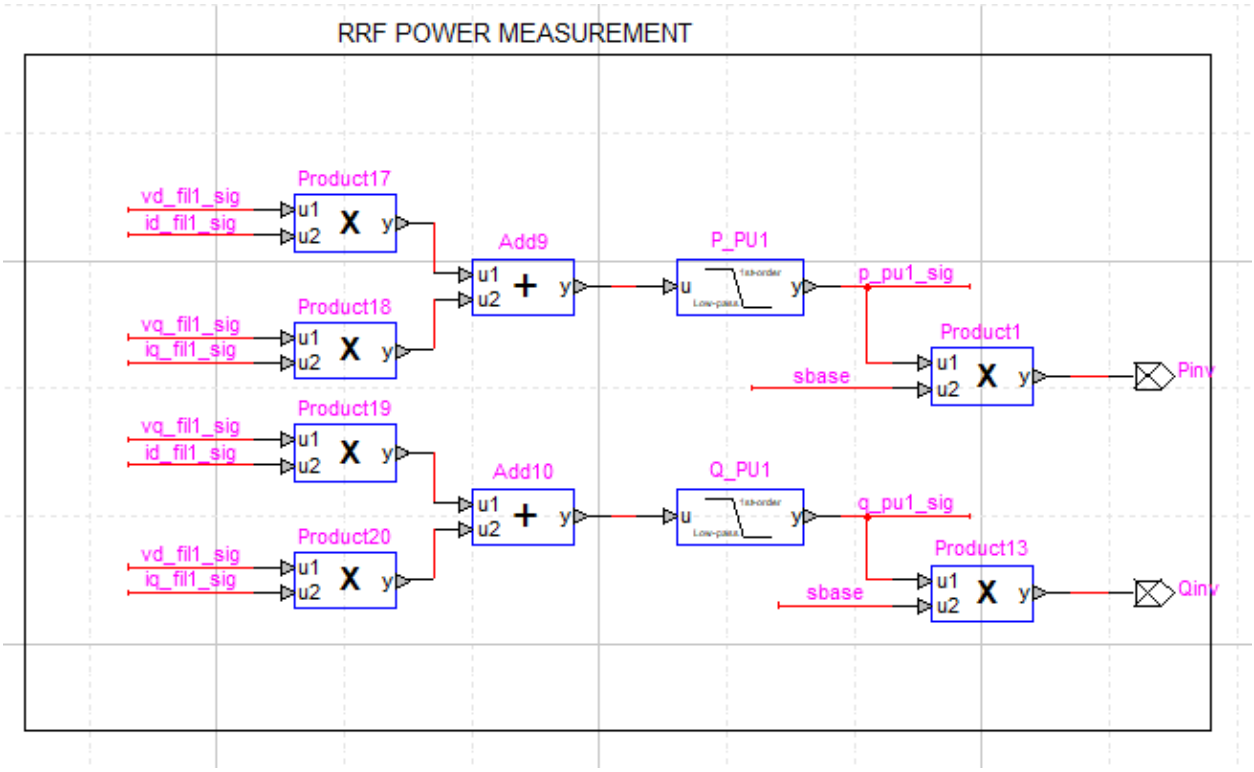


Figure A.43: Single phase GFL inverter measurement of instantaneous real and reactive power.

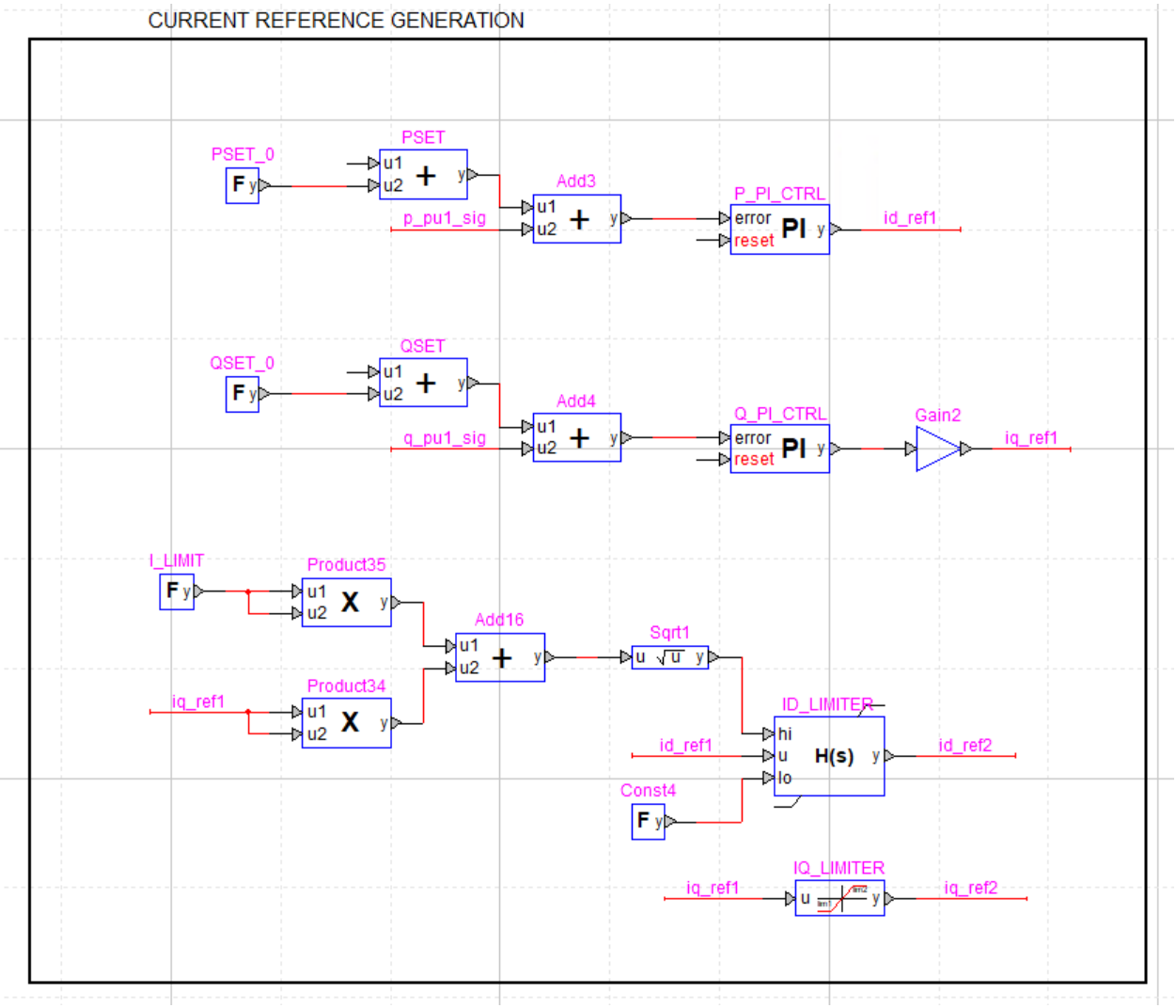


Figure A.44: Singlephase GFL inverter active and reactive power controllers.

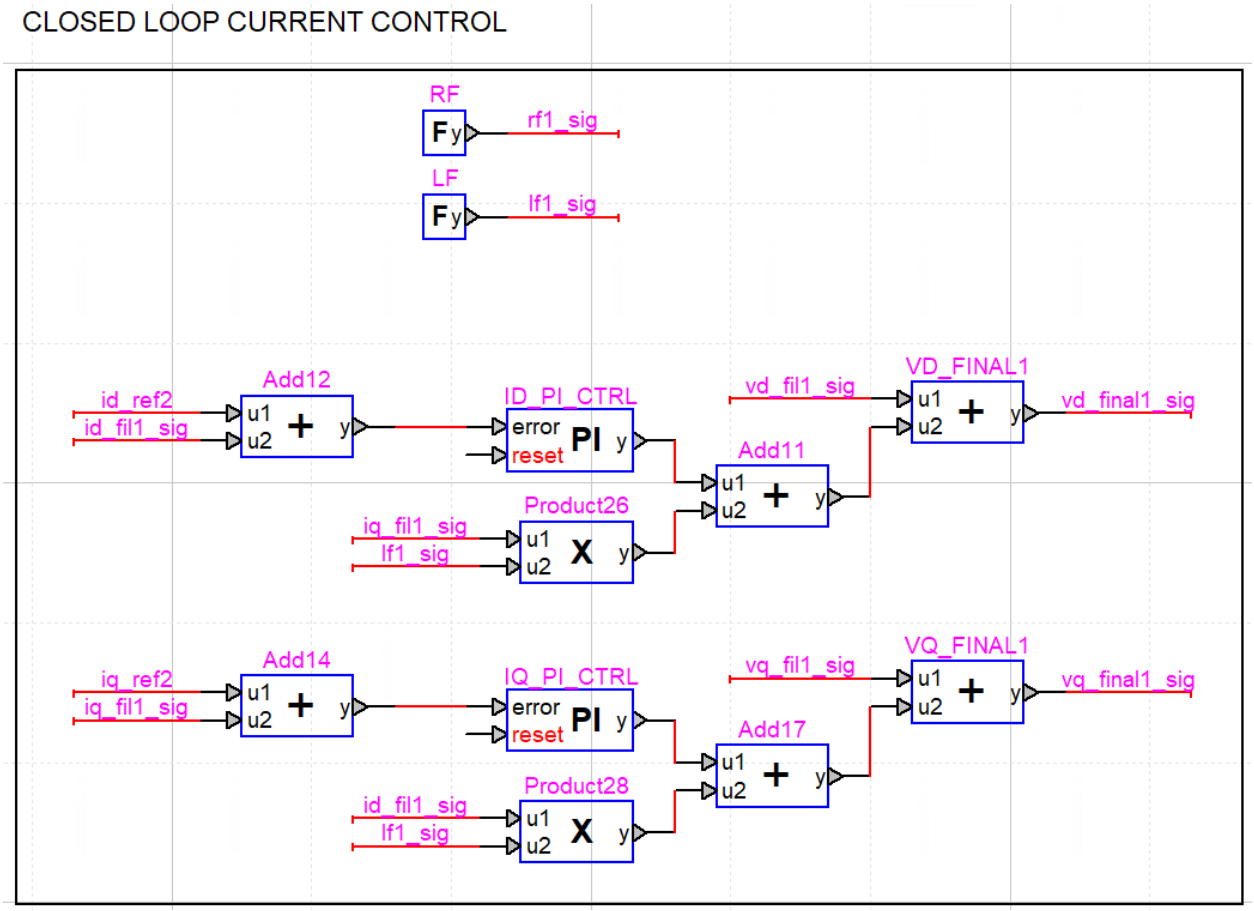


Figure A.45: Single phase GFL inverter d- and q-axis current controllers.

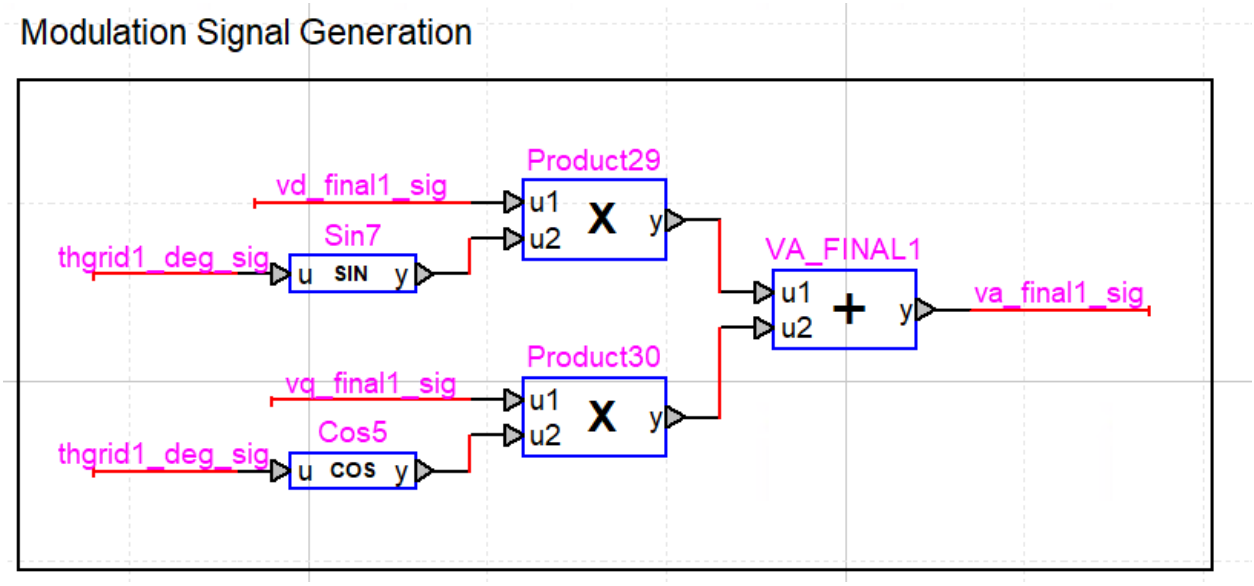


Figure A.46: Single phase GFL inverter voltage modulation signals.

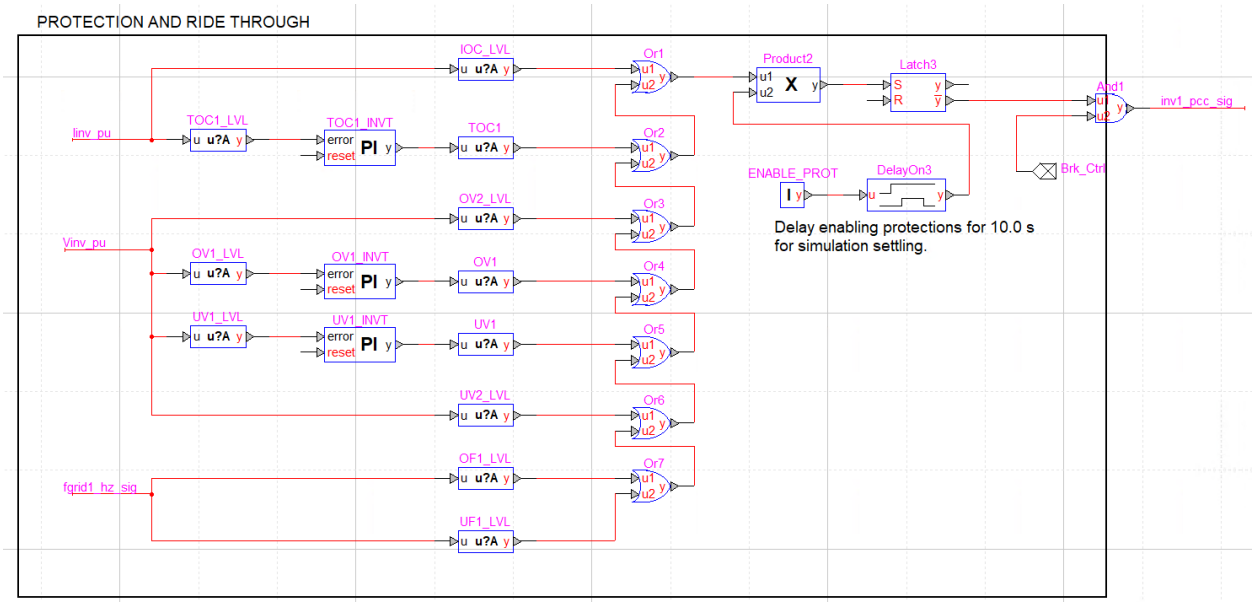


Figure A.47: Single phase GFL inverter protection logic circuits.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov