European Research Infrastructure supporting Smart Grid and Smart Energy Systems Research, Technology Development, Validation and Roll Out – Second Edition

Project Acronym: **ERIGrid 2.0**

Project Number: **870620**

Technical Report Lab Access User Project

Cyber-Physical Resilient Distribution System (CYPRESS)

Access Duration:

03/08/2022 - 26/08/2022 (remote access: 17 days)
03/10/2022 - 27/10/2022 (physical access: 25 stay days, 17 access days)

Funding Instrument:	Research and Innovation Action
Call:	H2020-INFRAIA-2019-1
Call Topic:	INFRAIA-01-2018-2019 Integrating Activities for Advanced Communities
Project Start:	1 April 2020
Project Duration:	54 months
User Group Leader:	Chaudhry Talha Hassan (LUMS)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 870620.

Report Information

Document Administrative Information	
Project Acronym:	ERIGrid 2.0
Project Number:	870620
Access Project Number:	142
Access Project Acronym:	CYPRESS
Access Project Name:	Cyber-Physical Resilient Distribution System
User Group Leader:	Chaudhry Talha Hassan (Lahore University of Management Sciences-LUMS)
Document Identifier:	ERIGrid2-CYPRESS LabAccess_Report_Template v1
Report Version:	vn.01
Contractual Date:	07/09/2022
Report Submission Date:	12/12/2022
Lead Author(s):	Chaudhry Talha Hassan (LUMS)
Co-author(s):	Muhammad Shamaas
Keywords:	[Mixed-integer second order cone programming, inverter-based distributed generator, resilience, cyber-physical distribution system, Synchronous generator], European Union (EU), H2020, Project, ERIGrid 2.0, GA 870620
Status:	Draft

Change Log

Date	Version	Author/Editor	Summary of Changes Made
03/10/2022	v1.0	E. Mrakotsky (AIT)	Draft report template adapted
14/11/2022	v1.1	Chaudhry Talha Hassan (LUMS)	Draft report preparation
05/12/2022	v1.2	Dr. Catalin Gavriluta (AIT)	Hosting institute review
12/12/2022	v1.3	Chaudhry Talha Hassan (LUMS)	Revised draft

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AGC	Automatic Gain Control
AIT	Austrian Institute of Technology
BUG	Back-up Generator
CUF	Current Unbalance Factor
CIL	Controller in the Loop
CPS	Cyber-Physical Systems
DER	Distributed Energy Resource
ESS	Energy Storage System
GFL	Grid-Following
GFM	Grid-Forming
HILP	High Impact Low Probability
MAS	Multi-Agent System
MG	Microgrid
MISOCP	Mixed Integer Second Order Conic Program
RoCoF	Rate of Change of Frequency
SG	Synchronous Generator
SOC	State of the Charge
TA	Transnational Access
TCL	Thermostatically Coupled Load
VOC	Virtual Oscillator Control

Executive Summary

Advances in smart grid technology have resulted in a higher reliability of service to consumers. Although faults and outages are inevitable, service must be restored to normal in the event of a blackout. Events such as hurricane, earth-quakes and floods are high impact low probability (HILP) events that can cause wide spread blackouts. However, deployments of Microgrids (MGs) and distributed energy resources (DERs) at the distribution level provide opportunities for service restoration in islanded mode. Bulk power distribution networks have limited number of switchable lines and loads. Hence, the network can often be sectioned into large bus blocks. Switching multiple loads simultaneously can result in severe frequency and voltage oscillations especially in inverter dominated weak inertial systems. Therefore, limits on frequency and transient voltage of DERs shall be incorporated as constraints of the restoration problem.

This work is an effort to validate a multi-layer sequential service restoration framework on an experimental setup at AIT Energy Systems Lab. An IEEE-123 node test feeder was emulated in OPAL-RT 5600/5700 with a controller in the loop (CIL) setup under the ERIGrid Transnational Access (TA) program. eMegaSim (real-time electromechanical solver for power systems) was used to emulate the test feeder at Ts = 100uSec. Moreover, multi-agent system (MAS) based on Rpi controller was used to implement distributed secondary control for frequency and voltage regulation during the restoration process.

During the online (17 days) and physical (17 days) access period several tests were conducted to validate the framework for different scenarios.

- Grid-following inverters (GFL) with and without grid-support functions as defined in IEEE 1547:2018 for Freq/Watt and Volt/Var regulation.
- Centralized automatic gain control (AGC) for frequency and voltage regulation.
- Role of distributed secondary control in frequency and voltage regulation during the black-start process.
- Consensus control law in the distributed controllers.
- Random location of faults and MG formation.

The overall schematics for the experimental setup is shown below

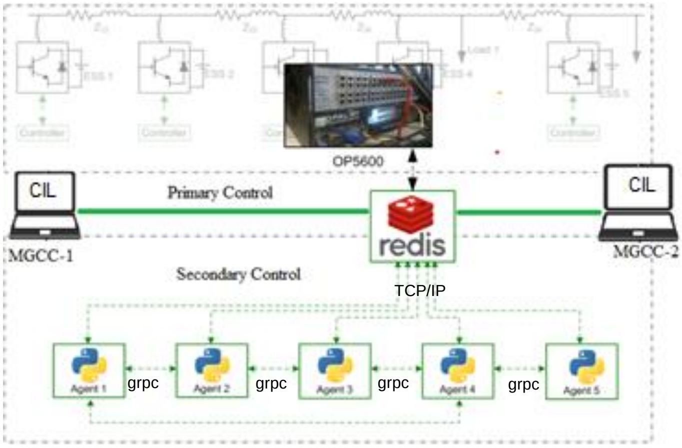


Figure 1: Experimental setup

1 Lab-Access User Project Information

1.1 Overview

USER PROJECT	
User Project Acronym	CYPRESS
USER Project Title	**Cyber-Physical Resilient Distribution System**
ERIGrid Reference	142
ERIGRID2.0	4th TA Call
USER GROUP	
Name (Lead)	Chaudhry Talha Hassan
Organization, Country	Lahore University of Management Sciences (LUMS), Pakistan
Names	Chaudhry Talha Hassan; Muhammad Shamaas
Organizations, Country	Lahore University of Management Sciences (LUMS), Pakistan
HOST RESEARCH INFRASTRUCTURE	
Name	AIT Austrian Institute of Technology
Countries	Austria
Start and End Dates	**03/08/2022 - 26/08/2022 (17 remote access days),**
(‘In-person’)	**03/10/2022 - 27/10/2022 (physical access, 25 stay days, 17 access days), AIT Vienna**
Stay and Access days	**In total: Stay Days 25 | Access Days 17 / AIT SmartEST Vienna**
	Chaudhry Talha Hassan | Stay Days 25 | Access Days 34 (remote and physical)
	Muhammad Shamaas | Stay Days 0 | Access Days 17 (remote)

1.2 Research Motivation, Objectives, and Scope

Extreme events such as hurricanes, earth-quakes, flooding and cyber-attacks can cause cascaded damages to the transmission and distribution system. This may result in a widespread blackout and restoration may take several days or even weeks. However, with the integration of renewable energy resources at the middle-voltage and low-voltage level the distribution systems are now active power networks with bi-directional flow of powers. Availability of generation resources at the distribution level provides a unique opportunity for black-start in the islanded system. This enhances the resiliency of the power system. DERs such as Photo-voltaic and wind are intermitted and stochastic in nature. Emergency synchronous generators (SG) and energy storage systems (ESS) can provide a stable and reliable power to the load. Therefore, Grid forming (GFM) inverters are usually designed with a stable ESS as the back-up DC source. An active distribution system may have several PV-based

GFL inverters, storage based GFM inverters and emergency synchronous generators. Similarly the distribution system may have several smart switches for network reconfiguration.

In the absence of a strong grid voltage, frequency and voltage references are provided either by a GFM inverter or a synchronous generator. These DERs act as a controlled voltage source and provide the black-start capability for the islanded MG. Despite having a significant difference in their construction and operations they both possess similarities which mimic the GFM inverters as SGs. Control techniques such as virtual synchronous machines (VSM), droop controls and virtual oscillator control (VOC) are used in GFM inverters. GFL inverters are constant PQ-source and can be considered as a controlled current source which requires external voltage signal at the terminal. Therefore, active distribution systems with high penetration of DERs provide an opportunity for fault isolation, MG formation and service restoration.

The objective of this investigation is to develop a cyber-physical test-bed for validation of sequential service restoration framework. The dynamic stability of the restored power system is ensured at every restoration stage. A distributed secondary control over a multi-agent system (comprising Rpi processors) is developed for voltage and frequency regulation during the restoration process.

The scope of the work includes modelling a modified IEEE-123 node test feeder into OPAL-RT target for electro-mechanical (EM) transient analysis with Ts = 100uS. The optimizer/controller in the loop is based on MATLAB R2019b with Gurobi solver. The framework generates switching commands for power system reconfiguration and ramps up the power generation of DERs with an objective to maximize restored load. However, the dynamic constraints of frequency and voltage and considered at every step to ensure transient stability. Additionally, a consensus based secondary control law in the agents (associated with DERs) regulate frequency and voltage during the restoration process. This accounts for inverter dynamics, TCL type loads, low x/r ratio of distribution lines and optimization processing delays between two consecutive stages.

1.3 Structure of the Document

This document is organised as follows: Section 2 briefly outlines the state-of-the-art that provides the basis of the realised Lab Access (LA) User Project (UP). Section 3 describes the executed experiments including methodology, set-ups, data management and processing. Section 4 summarises the results and conclusions. Potential open issues and suggestions for improvements are discussed in Section 5. Finally, additional information is provided in the Appendix A. Optimization Problem, Appendix B. Distributed MAS server code and Appendix C. Distributed MAS client code.

Introductory Note:

This Lab Access project was performed in two phases: Due to the delays in the attribution of VISA for travelling from Pakistan to Austria, first remote access of staff members of LUMS was organised and foreseen at AIT Austrian Institute of Technology at the SmartEST Laboratory in Vienna, Austria.

A second physical stay was performed several weeks later by LUMS staff member (lead) at the SmartEST Laboratory in Vienna, Austria to implement the developed and prepare methods on site.

2 State-of-the-Art/State-of-Technology

After a major blackout, electric power is usually restored in a top-down approach where large synchronous generators connected to the transmission network resumes operation and restore power to the associated distribution systems. However, fault location and repair work on the transmission/distribution network may take a significant amount of time hence delaying the restoration process. Similarly, faults in transmission network cause wide-spread blackouts and affect a larger geographical area. With the adoption of renewable resources such as wind and PV in the middle and low voltage distribution networks, power can be restored in a bottom-up approach where local DERs can provide power to critical loads in islanded mode. This approach significantly reduces the blackout time, its scale, financial and social impacts. Therefore, utilities are encouraging the adoption of DERs, MGs and reconfigurable power networks at the distribution level. An effective method for emergency service restoration is sectionizing the power network into isolated MGs [1] [2]. Bulk power distribution networks have limited number of switchable lines and loads [5]. Hence, the network can often be sectioned into bus blocks [6]. Switching several loads simultaneously can result in severe frequency and voltage oscillations especially in inverter dominated weak inertial systems. Therefore, limits on frequency and transient voltage of DGs shall be incorporated as constraints of the restoration problem [7].

In the literature, two main methodologies have been adopted to solve the load restoration problem. One approach considers it as a classical deterministic optimization problem with an objective to maximize (critical) load pickup incorporating several constraints. This is usually based on a central optimizer/control. The other considers it as a multi-agent system (MAS) assigning agents for each network element (such as DERs, switches and ESS) and restores the loads based on heuristics and agent behavioural modelling. Due to inherent advantages of distributed approaches, MAS restoration techniques are gaining grounds especially for larger distribution networks where centralized techniques cannot scale-up and present a single point of failure. Therefore distributed MAS techniques are scalable, reliable and robust.

Smart grids can have an extensive range of loads, switches and auxiliary equipment. Practical and operational limitations must be considered for the sequential switching of manual and remote-controlled switches [6]. The operating time of a switch can be impacted by its type, its location [8] and the constraints of the communication channel [9]. Coordination of switches is also important to prevent overloading of power generators. Possible changes in network topology must be considered to make feasible microgrids with radial structure and sufficient generation capacity [9]. Microgrid restoration can be challenging for inverter dominated smart grids because these networks suffer from poor dynamic stability and have lower inertia and damping. The author in [10] proposed a two-level simulation assisted framework for the service restoration of weak grids. However, the role of cyber-network is missing in his work. The most serious problem with restoring three-phase unbalanced power distribution networks is the overheating of generators due to current unbalance, excessive ramping, frequency drops, and voltage oscillations [13], [14]. The microgrid frequency response rate can be approximated based on the total generation capacity of generators [16], [17]. The RoCoF can be reduced by limiting the ramp rate of synchronous generators [12],[15], [18].

Energy storage systems (ESS) have a significant role in mitigating the effects of lower inertia in inverter dominated power systems. A strategic charging and discharging scheme must be formulated to maintain SOC within a safe limit [15]. Such approximation methods can effectively constrain the maximum frequency drop by limiting the maximum recovered load in each stage [14]. However, these techniques cannot be applied to inverter-based generators with unbalanced loading [10], [11]. The frequency response of low-inertia inverters is highly unpredictable. Static as well as dynamic constraints for voltage, current, and frequency must be imposed to prevent damage to sensitive electronic equipment [5], [7], [10],[19],[20].

Distribution system restoration is traditionally formulated as a combinatorial problem. Conventional restoration is implemented through reconfiguration of a faulted feeder section through tie-lines with a healthy neighbouring feeder. Modern distribution feeders with high penetration of DERs provide an opportunity for service restoration by forming MGs around black-start DGs [1]. However, most existing algorithms formulate this problem as a single-step optimization problem that cannot handle inter-temporal constraints like ramp rates for DGs, charging/discharging rates for ESS and stability of the power systems. The optimal power flow problem can be simplified through several approximations and relaxations. Constant current and constant impedance loads can be represented using linearized power flow models [13]. The problem of power flow in an unbalanced multi-phase radial network has been solved using semi-definite programming. Mixed-integer models with quadratic, conic, and semi-definite optimization have been successful for power loss analysis [17]. Load flow problems in radially distributed networks have been solved by convex or conic optimization programs [17]. Although complex optimization problems can accurately model the steady-state behaviour of these power system elements, their transient behaviour cannot be predicted.

The role of a synchronous generator as a black-start capable DG has not been studied in detail for service restoration problems. Besides improving electrical stability, high-inertia diesel generators can act as reliable power sources. There are new opportunities to consider synchronous generators or backup generators (BUGs) as a source of black-start capability [12]. Ramp constraints, inertial capacities of prime-movers, current unbalance factor (CUF) and generator excitation characteristics are important for modelling their operation. By incorporating BUGs as dispatchable DGs in a microgrid, the utility of BUGs can be increased to harness power from non-dispatchable DGs and improve microgrid stability and coverage. Standardization of smart grid communication technologies provides new opportunities for improving protection and control of smart grids. With the increasing installation of remote-controlled switches, tie-lines, isolators and distributed energy resources, there is a growing need for incorporating communication layer into the power systems making them cyber-physical systems (CPS). Power quality monitoring modules integrated with high frequency transceivers allow real-time monitoring of transient fault records with millisecond precision. Active management of smart grids is now possible with improved processing power and reliable communication technologies.

Cyber-physical power systems control frameworks require sophisticated cyber-physical test-beds for real-time emulation and validation. Emulating the three components (power layer, communication layer and control layer) in real-time is a challenging task [21]. This work implements a cyber-physical test bed for evaluating the service restoration framework on IEEE-123 node test feeder. The target power system is emulated in OPAL-RT. The central controller runs the optimization and it forms a controller in the loop (CIL) setup.

3 Executed Tests and Experiments

3.1 Test Plan, Standards, Procedures, and Methodology

The AIT Austrian Institute of Technology is the largest research and technology organization (RTO) in Austria. The energy systems department hosts two main laboratories including AIT smartEST and power electronics laboratory. Research infrastructure in the power electronics laboratory is detailed as follows.

- Multicore Opal-RT (5600/5700) real-time simulator (eMegaSim/ePhasorSim).
- Typhoon HIL real-time simulator
- MATLAB TPC Target HIL simulation environment
- PLECS RT-Box real-time simulator
- Power-HIL and Controller-HIL experiments at full power in a closed control loop
- General simulation tools: Matlab/Simulink, SimPowerSystems, PSpice/Cadence
- Network simulation tools: DigSILENT PowerFactory, PSAT
- Communication network simulator: OMNeT++
- Batch of Raspberry Pi microprocessors for real-time distributed control
- TMS320/C2000 DSP processor boards
- Powerful simulation cluster for complex and large-scale system simulations
- Coupling middleware AIT LabLink

3.1.1 Validation of the framework was planned on a real-time OPAL-RT target running RT-LAB software. The IEEE-123 node test feeder was developed in MATLAB/Simulink using Simscape and other library models. A detailed model of the feeder was developed considering ZIP type loads, GFL and GFM types of DERs, line impedances and capacitances, transformers, voltage regulators and breakers. RT-LAB transformers Simulink models into compatible code that runs on OPAL-RT target. The feeder was split into 4 distinct isolated MGs and they were considered as autonomous power systems. Each MG was controlled by its own optimizer. Data reads and writes to the real-time target were performed by REDIS cache. A REDIS serve in the network allowed reads and writes between the optimizer and OPAL-RT target. Figure 2 below illustrates the test plan.

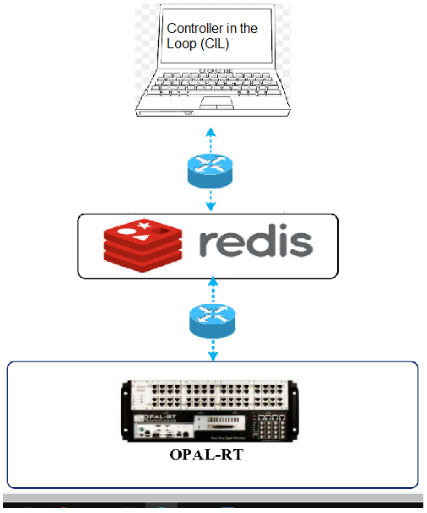


Figure 2: Test plan

Following test cases were designed to validate the restoration framework.

- **Cy_v1.1**: Porting MG-1 (power systems and REDIS interface) on OPAL-RT target with REDIS server accessible from cloud over a static IP address. The optimizer (with a REDIS client) accessed the cache server remotely for signal reads and command writes to OPAL-RT target.
- **Cy_v1.2**: Porting other MGs (power systems and REDIS interface) on OPAL-RT target with REDIS server accessible from cloud over a static IP address. The optimizer (with a REDIS client) accessed the cache server remotely for signal reads and command writes.
- **Cy_v1.3**: UI based “Chronos” dash board for control and monitoring of OPAL-RT simulations. Data visualization and file records for user defined parameters such as frequency, voltage and power.
- **Cy_p1.1**: Integrated testing of all components (such as OPAL-RT, Optimizer and REDIS server) in a single data LAN environment for all microgrids.
- **Cy_p1.2**: Validating the framework with centralized secondary control for frequency and voltage regulation.
- **Cy_p1.3**: Adding grid-support functions (of freq-watt and volt-var) to GFL inverters and their effect on frequency/voltage regulation according to IEEE 1547:2018.
- **Cy_p1.4**: Tuning optimization parameters such as DER capacities, frequency and voltage limits and their effect on restoration dynamics.
- **Cy_p1.5**: Validating the framework with distributed secondary control for frequency and voltage regulation on raspberry Pi (rpi) processors.
- **Cy_p1.6**: Implementing gRPC for inter-agent communication. Communication between Rpi Agent and DER (in OPAL-RT) is managed through REDIS over TCP/IP
- **Cy_p1.7**: Implementing Leader-Follower consensus (LFC) control law in DERs (agents) for proportional power sharing, frequency and voltage regulation.

Table 1: Tests schedule (Remote access)

Week 1	03/08/2022	Kick-off meeting. Introduction and login details of “Chronos”.
	04/08/2022	**Cy_v1.1:** Simulink model for MG-1 was shared with AIT technical staff. Interfacing signals (measurements and commands) were discussed.
	05/08/2022	**Cy_v1.1:** The model was appended with IEEE-123 node init file, Ts (sampling time) appropriate for eMegaSim and REDIS interface.
Week 2	08/08/2022	**Cy_v1.1:** After a successful build and download into the OPAL-RT target, the model was simulated from Chronos UI. The restoration was performed remotely over the cloud. Results were monitored from Chronos.
	09/08/2022	**Cy_v1.1:** Restoration was performed with centralized secondary control for frequency and voltage regulation.
	10/08/2022 11/08/2022	**Cy_v1.2:** Restoration of MG-2
	12/08/2022	**Cy_v1.2:** Restoration of MG-2 (with secondary control)
Week 3	15/08/2022	**Cy_v1.2:** Restoration of MG-3
	16/08/2022	**Cy_v1.2:** Restoration of MG-3 (with secondary control)
	17/08/2022 18/08/2022 19/08/2022	**Cy_v1.2:** Restoration of MG-4
Week 4	22/08/2022 23/08/2022	**Cy_v1.2:** Restoration of MG-4 (revised Simulink model)
	24/08/2022 25/08/2022 26/08/2022	**Cy_v1.1/Cy_v1.2/Cy_v1.3:** Validating the framework for all MGs from Chronos. Maintaining file records and graphs.

Table 2: Tests schedule (Physical access)

Week 1	03/10/2022	Kick-off meeting. Visit, orientation and safety instructions
	04/10/2022	**Cy_p1.1:** Integrating all components in the local data LAN
	05/10/2022 06/10/2022 07/10/2022	**Cy_p1.1:** Testing MGs in the local LAN setup. Comparing results with offline and remote simulation.
Week 2	10/10/2022 11/10/2022	**Cy_p1.2:** Validating the framework with centralized sec-

		ondary control for all MGs.
	12/10/2022 13/10/2022	**Cy_p1.3:** Validating the framework with grid-support functions added to GFL inverters for all MGs.
	14/10/2022	**Cy_p1.4:** Testing the restoration framework by tuning optimization parameters and operating limits for all MGs.
Week 3	17/10/2022	**Cy_p1.5:** Setting up Raspberry Pi processors for gRPC tools and REDIS client
	18/10/2022 19/10/2022	**Cy_p1.5/ Cy_p1.6:** Implementing gRPC client stub and server in each agent.
	20/10/2022	**Cy_p1.7:** Implementing LFC control law in each agent.
	21/10/2022	**Cy_p1.7:** Validating restoration framework with distributed secondary control for frequency and voltage regulation.
Week 4	24/10/2022	**Cy_p1.7:** Validating the framework with different configurations and parameter tuning for all MGs.
	25/10/2022	Compiling results and conclusions of the work
	26/10/2022	National Holiday in Austria

Service restoration is a sequential process where load is gradually picked up by closing switches and harnessing power from nonblack-start DERs such as PV and wind inverters. However, dynamic stability of the power system must be ensured at every stage. Incorporating the dynamic feedback as explicit constraints to the optimization problem is the key to a stable black-start process. The Simulink model keeps a track of the frequency nadir and max voltage deviation and reports it to the optimizer. The REDIS cache reads data at a very high speed from the OPAL-RT target. The same data is available for profiling in a graph or writing to record files. Node-RED is a flow-based development tool for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser based flow editor. Data is read from REDIS cache at 500ms and profiled in a graph for analysis during the restoration process.

The controller solves a mixed integer second order conic problem (MISOCP) to determine switch status and generation build-up of DERs. These commands are routed to OPAL-RT through REDIS cache. Power system is re-configured in real-time on the fly and transients are reported as feedback to the controller. Dynamic constraints are used to calculate the total distortion factor of the resulting power system. This modulates the load step for the next stage ensuring a stable restoration with acceptable variations of frequency and voltage.

3.2 Test Set-up(s)

All tests were performed on OPAL-RT 5600/5700 real-time simulator. The target is accessed by the host PC (running RT-LAB software) placed in the same data LAN. IP address map of the LAN is as follows

- 192.168.20.1~20 for 20 Nos. raspberry Pis running Linux/Debian with Python-3.
- 192.168.20.24 for host PC running RT-LAB
- 192.168.20.101 for OPAL-RT 5600 real-time simulator.
- 192.168.20.102 for OPAL-RT 5700 real-time simulator.
- 192.168.20.111 for Optimizer (CIL) running MATLAB/YALMIP with Gurobi solver.
- 192.168.20.254 for REDIS server.

Phase-1 of the project was performed remotely through “Chronos”. Chronos is a web service provided by AIT which enables users to create cyber physical test beds. Chronsos integrates well with OpalRT’s ePhasorSim and offers several automation and convenience tools for creating these testing environments. However, since for this project eMegaSim was used, Chronos was mainly used as a web UI based dashboard which allowed remote users to interact with the real-time simulation. Figure 3 illustrates a detailed structure of the Chronos framework for remote access to a cyber-physical test bed.

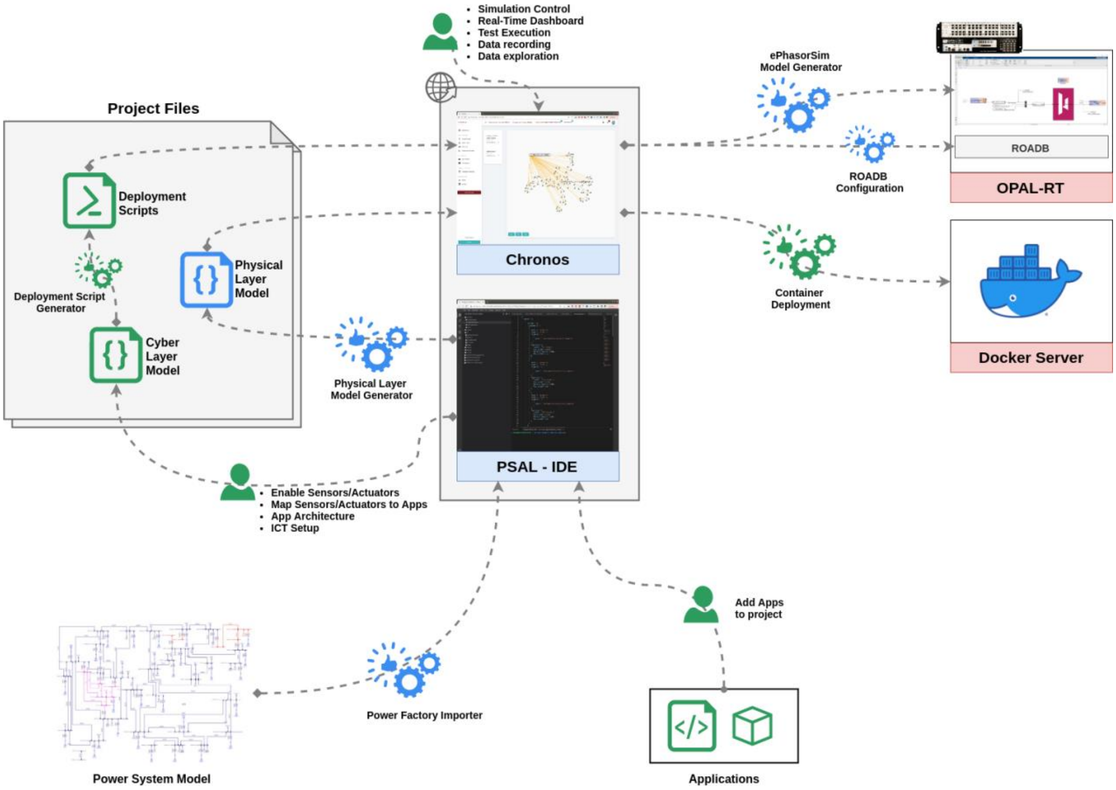


Figure 3: Chronos – AIT framework for automatic setting up cyber-physical test beds

During the remote access phase, the Simulink model for power system was loaded to the OPAL-RT target with the help of AIT technical staff. However, after incorporating the power system into the Chronos platform, the PSAL editor provided an added flexibility to reconfigure the I/O data streams and data access rate without the need to reload power system into the target. Several tests were performed to test the efficacy of the user framework using the Chronos platform. This web application proved to be an effective tool to setup cyber-physical test-beds for smart energy applications. Figure 4,5 and 6 highlight key features and interfaces of the Chronos platform.

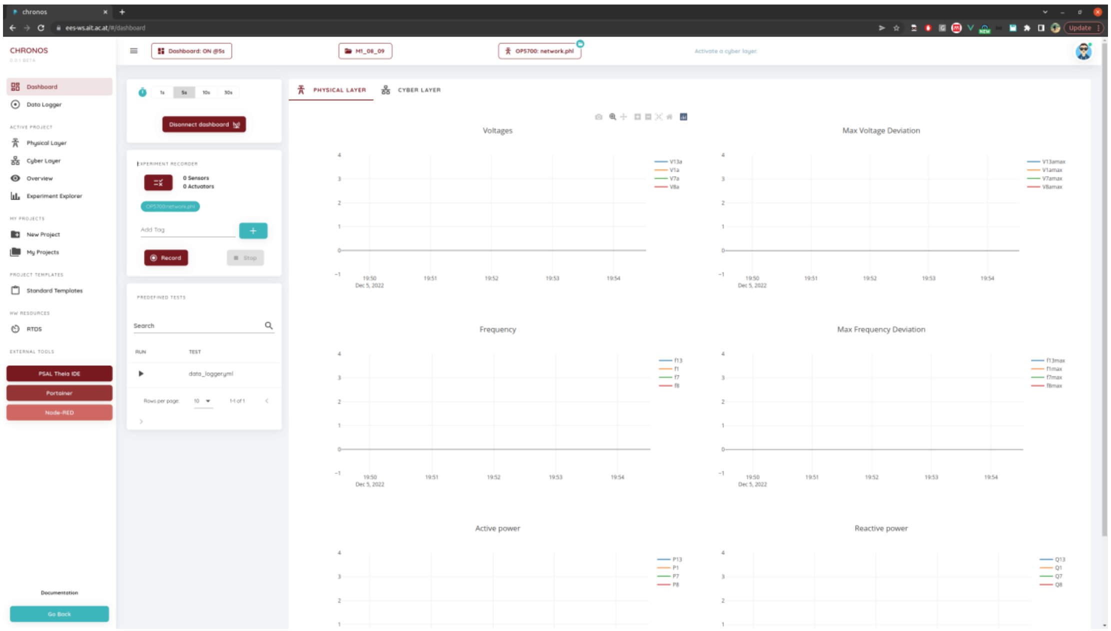


Figure 4: Chronos real-time dashboard panel

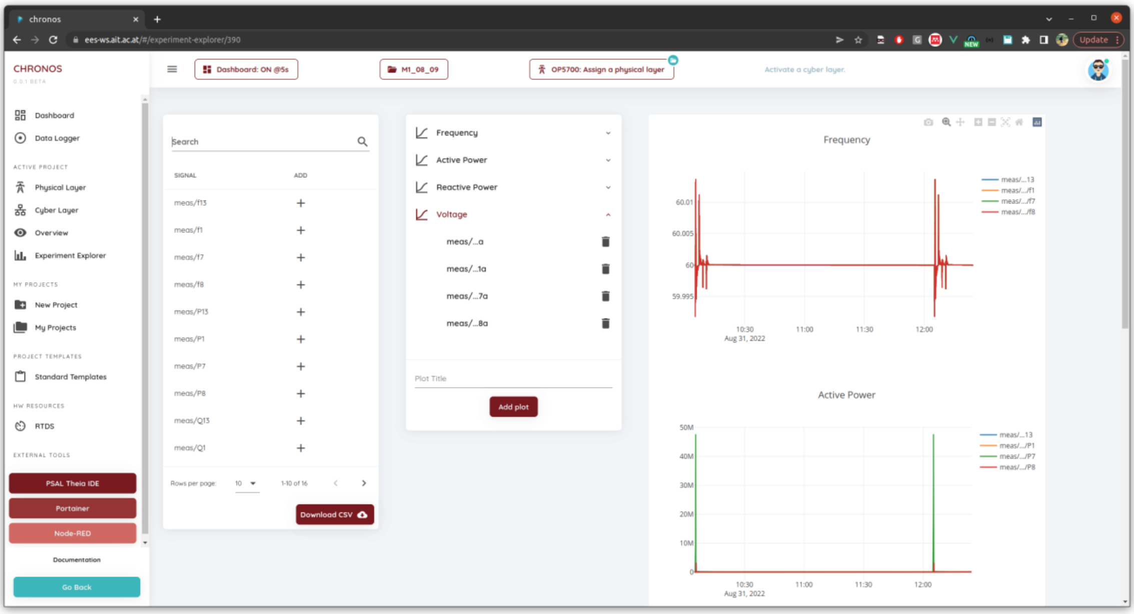
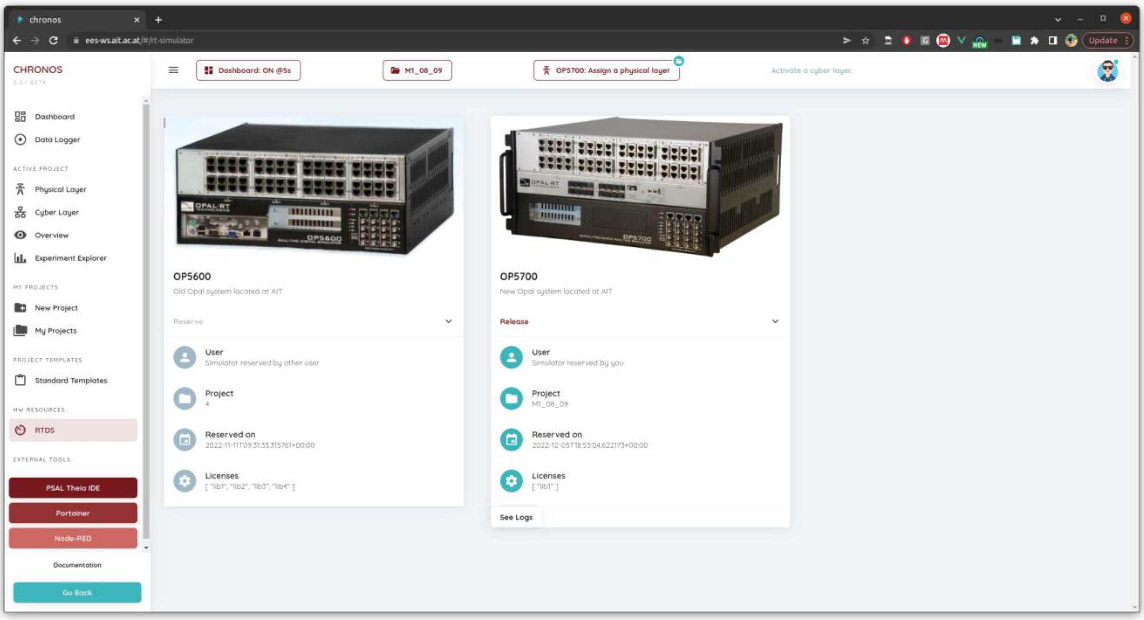


Figure 5: Chronos data and experiment exploration panel



OP5600
Old Opal system located at AIT

User
Simulator reserved by other user

Project
4

Reserved on
2022-11-11T09:31:33.315761+00:00

Licenses
["lib1", "lib2", "lib3", "lib4"]

OP5700
New Opal system located at AIT

User
Simulator reserved by you

Project
M1_08_09

Reserved on
2022-12-05T18:53:04.622173+00:00

Licenses
["lib1"]

Figure 6: Chronos real-time simulator HW reservation panel

The scope of user access project included distributed secondary control on embedded hardware such as Rpi. Therefore, it was necessary to have a physical access to the host lab infrastructure. This helped in the setup of a holistic cyber-physical test bed without experiencing delays/latencies of inter-domain routing and network latencies. Phase-2 of the project was performed physically in AIT power electronics laboratory by placing optimizer (CIL), REDIS server, OPAL-RT, host PC and Rpis in the same LAN. Figure 7 below shows the schematics of the LAN environment. Physical access also helped in maintaining power systems emulation in OPAL-RT and modifying the power system for several tests and performance improvements (which was not possible in remote access without the involvement of an AIT staff).

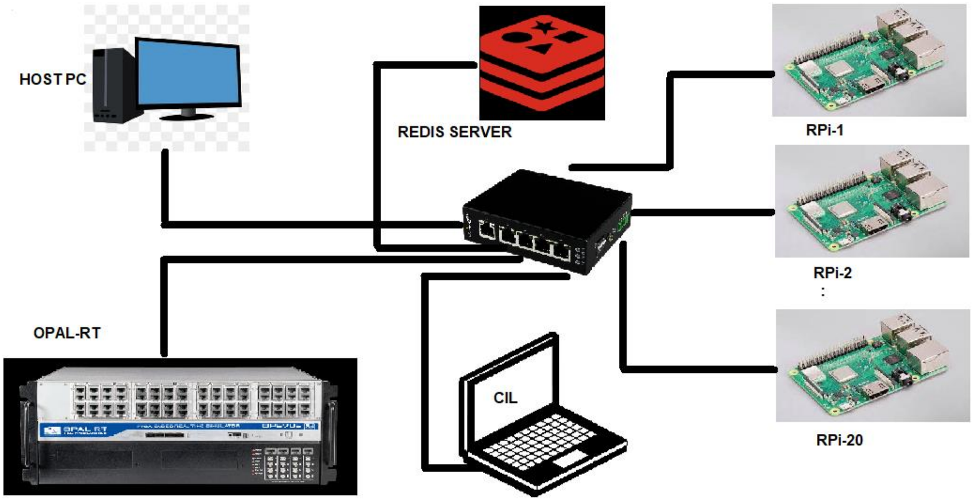


Figure 7: DATA LAN Environment

The proposed framework was validated on a modified IEEE-123 node test feeder. The feeder operates at a nominal voltage of 4.16 kV. It is characterized by overhead and underground lines, unbalanced loading with constant impedance, current and power (ZIP) loads, voltage regulators, shunt capacitor banks, and multiple switches. The test feeder is famous among the power and energy research community for validating distribution and network reconfiguration algorithms. Figure 8 illustrates the modified test feeder considered in our work. The power system was modelled in Simulink and later built and transferred to real-time target using RT-LAB software. Figure 9 shows all MGs and their schematics along with interface commands and measurements.

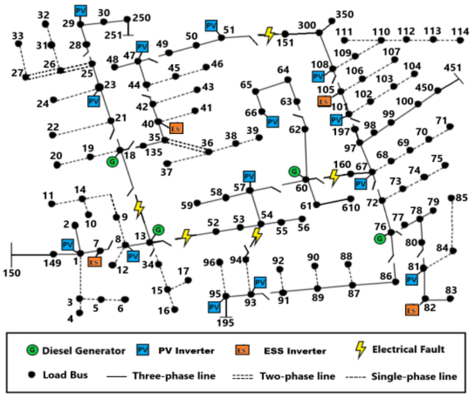


Figure 8: Modified IEEE-123 node feeder

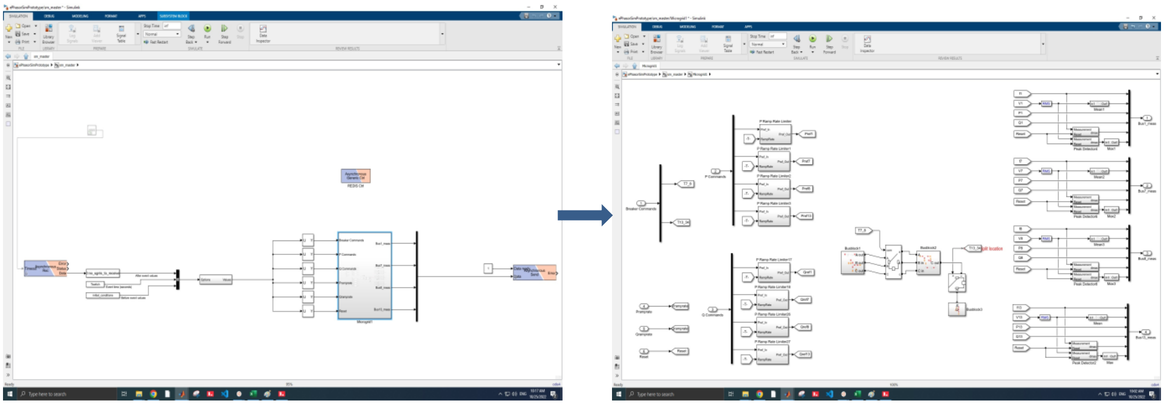


Figure 9 (a): MG1

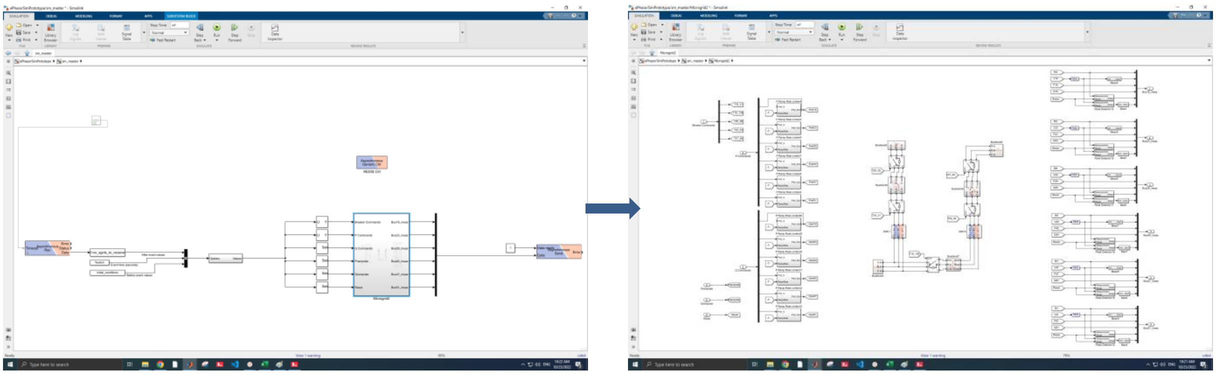


Figure 9 (b): MG2

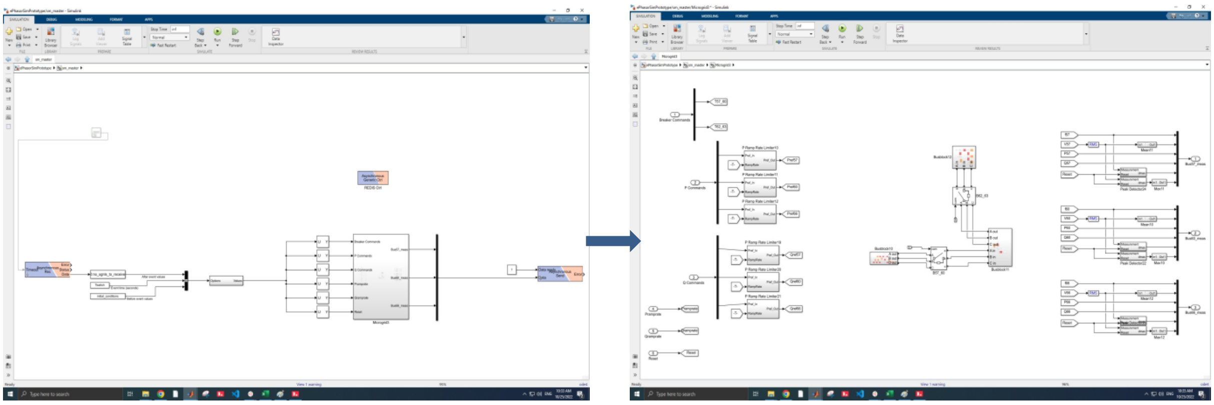


Figure 9 (c): MG3

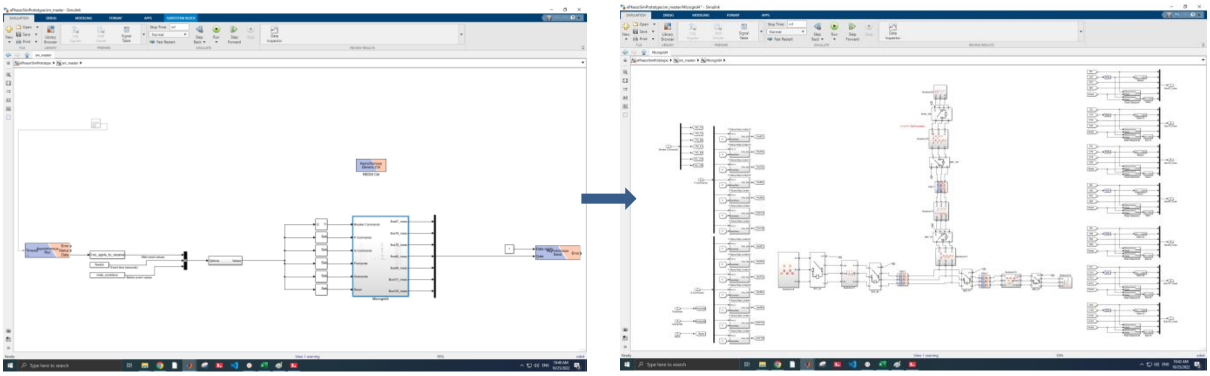


Figure 9 (d): MG4

Modern distribution systems with high penetration of renewable energy resources have inverter based DERs. These DERs are classified as GFM and GFL inverters. Emergency or back-up synchronous generators are also a part of distribution systems. They are evenly distributed within a feeder for providing backup power in case of emergency to localized loads. However, with the installation of smart switches, the power network can be reconfigured. This provides a unique opportunity of service restoration in blackout areas having DERs. Although, GFM inverters and synchronous generators differ in construction but they exhibit similar electrical characteristics. GFM inverter control strategies include droop control, virtual synchronous machine (VSM) and virtual oscillator control (VOC). In our work, we considered synchronous generators as black-start voltage sources that provide frequency and voltage reference to GFL inverters. GFL inverters are constant PQ or current sources that require external voltage signal at their terminals to operate. Figure 10 shows the model of a synchronous generator as a controlled voltage source driven by P-f and Q-v droop controls, governor/inertial control and excitation control. Figure 11 depicts the model of a GFL inverter with grid-support functions.

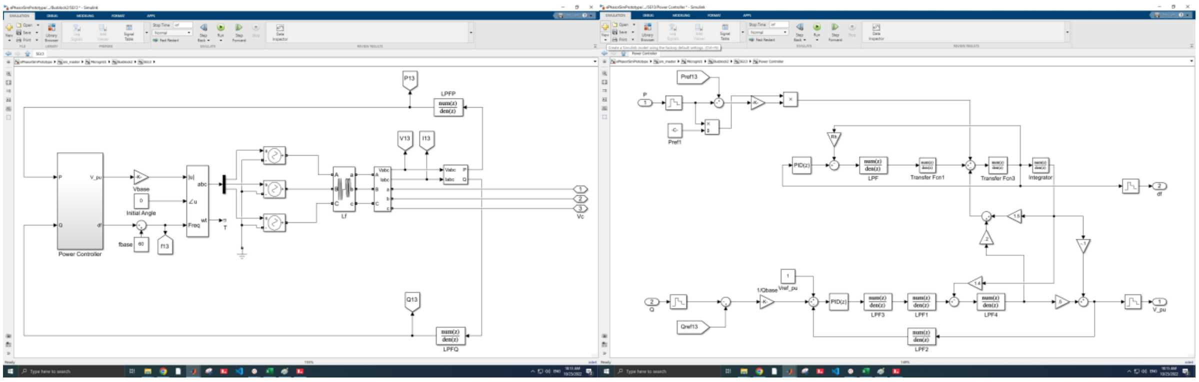


Figure 10: Synchronous Generator model

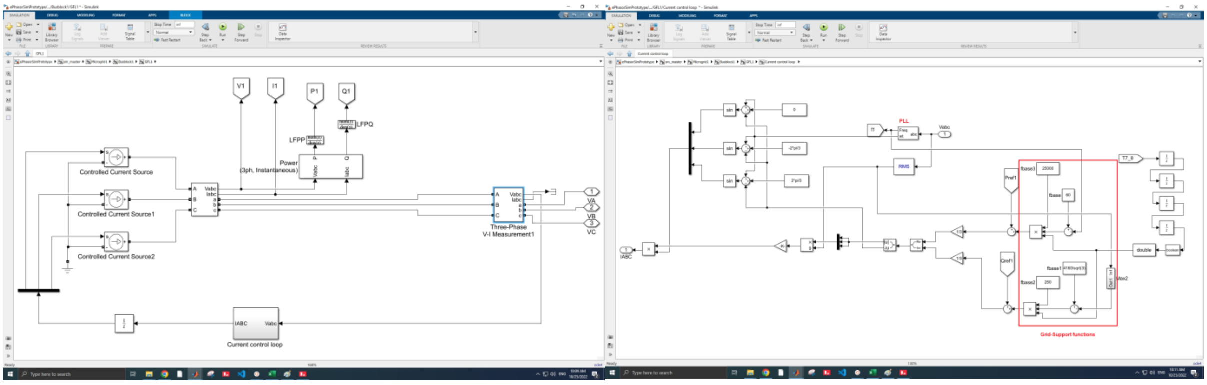


Figure 11: GFL inverter model

3.3 Data Management and processing

REDIS cache was used for data flow among the setup components including OPAL-RT, Optimizer and distributed agents. REDIS is an open source, in-memory data buffer used by developers as a database, streaming engine and a message broker. REDIS performed the data exchange at a very fast rate and did not affect the fidelity of the framework. Node-RED was used for data management and profiling the transient behaviour of the framework.

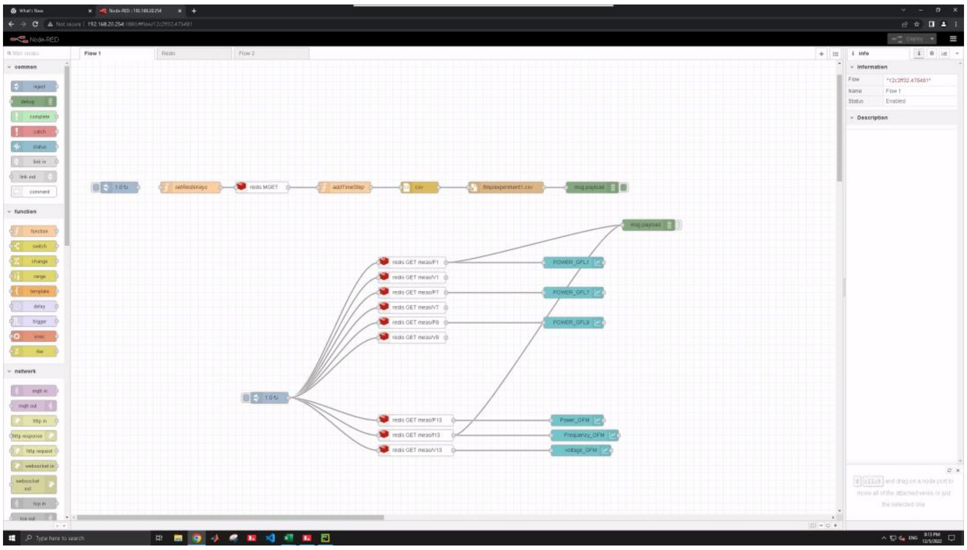


Figure 12: Node-red flow for extracting simulation data

4 Results and Conclusions

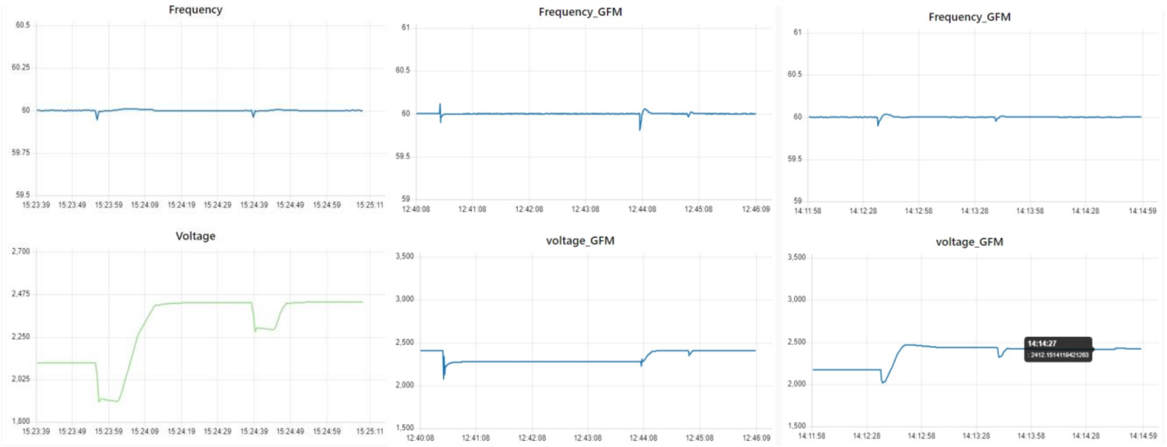
4.1 Discussion of Results

The restoration framework was validated for the following conditions.

- Restoration without secondary regulation (case-1).
- Restoration with grid-support functions and no secondary regulation (case-2).
- Restoration with secondary regulation (case-3).

During the restoration process, load is picked up by closing neighbouring switches and power is harnessed from GFL inverters. This results in frequency oscillations and voltage deviations due to electro-mechanical nature of power electronics based inverters, distribution line impedance/reactance and different ramp rates of GFL/GFM inverters. Therefore secondary regulation of frequency and voltage is required between two restoration stages. Results for different cases are illustrated below.

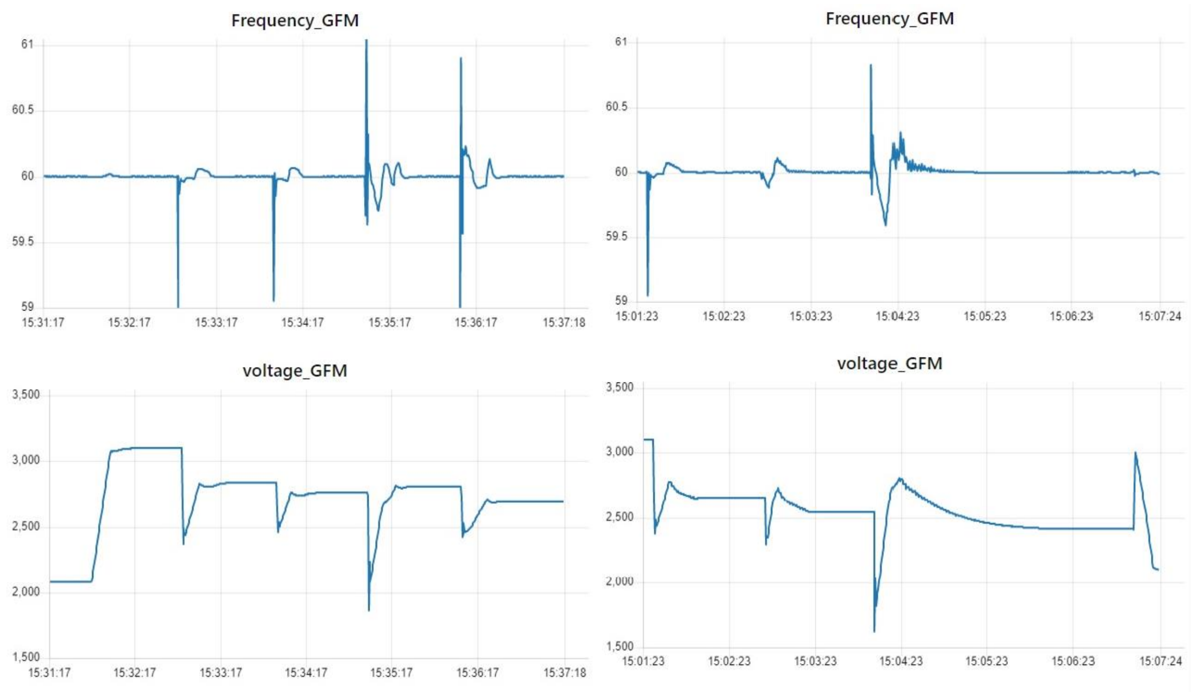
Microgrid-1 consists of 3 bus blocks, 2 switches, 2 GFL inverters and 1 SG. Total load present in the MG is 400kW. However, we have considered that available generation capacity is larger than the load requirement. Results are recorded at the SG bus of MG-1 and presented below in Figure 13.



(a) Case-1 (b) Case-2 (c) Case-3

Figure 13: Microgrid-1 response

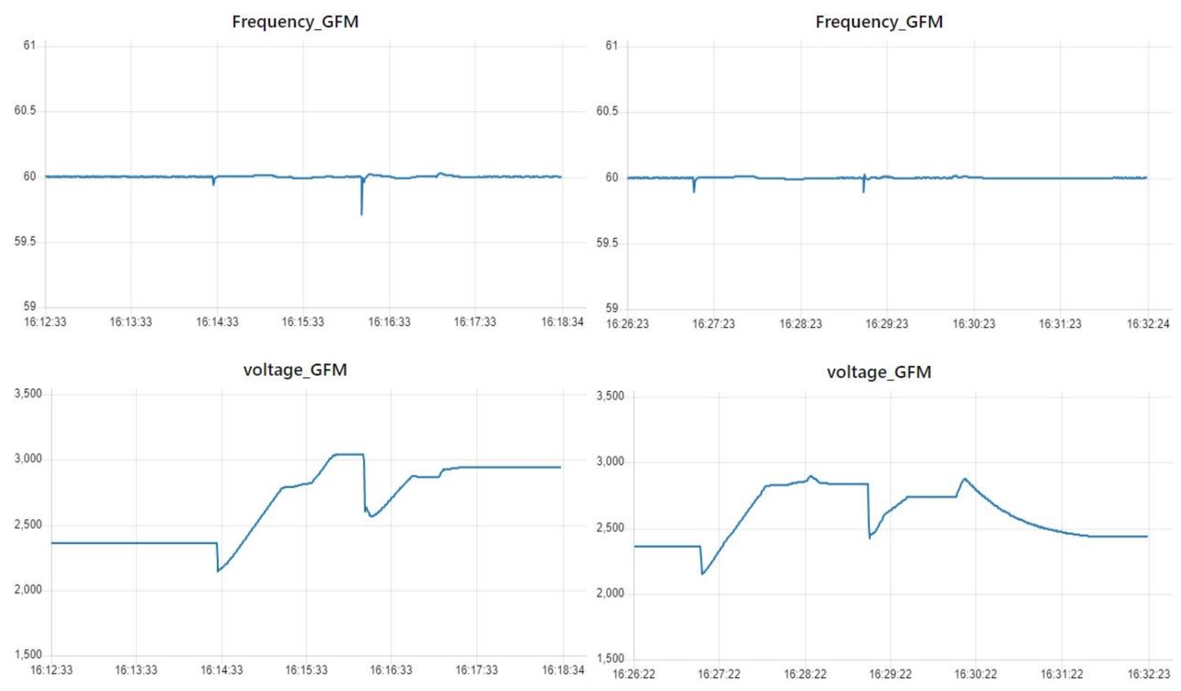
Microgrid-2 consists of 6 bus blocks, 5 switches, 5 GFL inverters and 1 SG. Total load present in the MG is 1145kW. However, we have considered that available generation capacity is larger than the load requirement. Results are recorded at the SG bus of MG-2 and presented below in Figure 14.



(a) Case-1 *(b) Case-3*

Figure 14: Microgrid-2 response

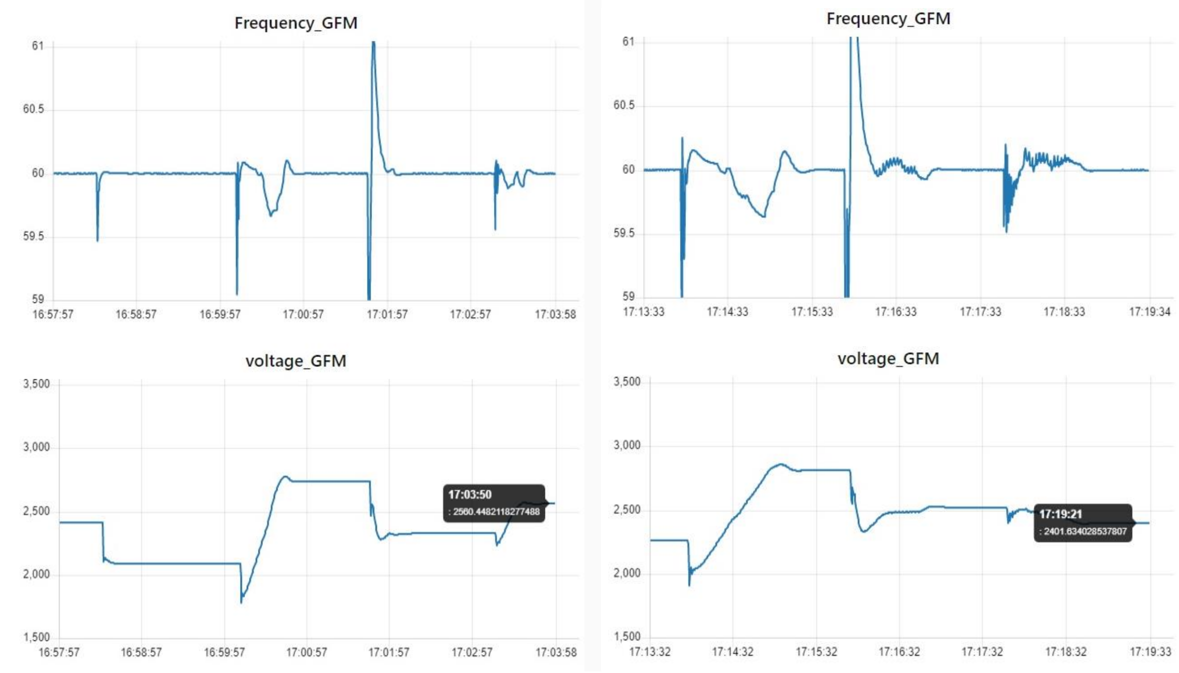
Microgrid-3 consists of 3 bus blocks, 2 switches, 2 GFL inverters and 1 SG. Total load present in the MG is 550kW. However, the available generation capacity is larger than the load. Results are recorded at the SG bus of MG-3 and presented below in Figure 15.



(a) Case-1 *(b) Case-3*

Figure 15: Microgrid-3 response

Microgrid-4 is the largest islanded power system in our case study. It consists of 8 bus blocks, 7 switches, 6 GFL inverters and 1 SG. Total load present in the MG is 1425kW. However, we have considered that available generation capacity is larger than the load requirement. Results are recorded at the SG bus of MG-4 and presented below in Figure 16.



(a) *Case-1*

(b) Case-3

Figure 16: Microgrid-4 response

Restoration results reveal the importance of incorporating grid-support functions in GFL inverters. We can see the performance of restoration framework for case-2 in MG-1. However, for distribution systems where grid-support functions are not available in the GFL inverters than distributed secondary regulation for frequency and voltage shall be implemented to ensure supply security to the customers. Another interesting aspect to note here is that larger MGs (2 and 4) take more stages (up to 4 or 5) to restore. Frequency oscillations/dips are the result of bus block connection by closure of corresponding switches. Larger bus blocks afflict severe oscillations to microgrid frequency and voltage.

4.2 Conclusions

Validation of cyber-physical energy systems requires sophisticated test-beds that can emulate the framework in real-time across all domains of power systems, controller and communication network. This work was tested with a controller in the loop (CIL) approach where the controller interacted with the power system simulating in real-time. Different components of the test-bed including OPAL-RT and the optimizer may have different processing times and speed. However, in this setup the aim was to validate the restoration framework by analysing the transient stability of reconfigured power system.

This framework was only tested on offline event based simulators like GridLAB-D, ns-3 and MATLAB. However, transnational access under the ERIGrid 2.0 program provided a unique opportunity to validate our approach on real-time power systems simulator. Cyber-physical energy systems with holistic test-bed designs ensure fidelity of the algorithm.

Future work includes role of communication layer among distributed MAS interacting to perform secondary regulation of frequency/voltage and networking of islanded MGs for energy management. Similarly impact of communication layer on stability of the power system during switching commands and power reference generation shall also be studied.

5 Open Issues and Suggestions for Improvements

Physical access to smart grid research infrastructure fosters system-level support and education for industrial and academic researchers in power and energy systems research. However, due to administrative problems of physical access such as travel restrictions and VISA processing times, Virtual access is gaining importance and becoming an efficient medium of access.

The development and testing of “Chronos” platform for virtual access in this TA user project is very encouraging. It has all the required interfaces for testing, validating and simulation of a cyber-physical test-bed. However, during the detailed testing and project loading we encountered a few minor issues that were later resolved by the AIT digitalization team (responsible for maintaining this platform). It is observed that ROADB interface dll/batch file gets corrupted after some time and the user has to manually replace the corrupted file with the fresh copy. Similarly, the OPAL-RT target needs a reset after a continuous operation for 24~30 hours.

References

[1] Z. Wang and J. Wang, “*Self-healing resilient distribution systems based on sectionalization into microgrids*”, IEEE Transactions on Power Systems, vol. 30, no. 6, pp. 3139-3149, Nov. 2015.

[2] A. Arif and Z. Wang, “*Networked microgrids for service restoration in resilient distribution systems*”, IET Generation, Transmission and Distribution, vol. 11, no. 14, pp. 3612-3619, September 2017.

[3] J. A. Taylor and F. S. Hover, “*Convex models of distribution system reconfiguration*”, IEEE Trans. Power Syst., vol. 27, no. 3, pp. 1407–1413, Aug. 2012.

[4] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, “*Applications of second-order cone programming, Linear Algebra and Its Applications*”, vol. 284, pp. 193–228, Nov. 1998

[5] L. Che and M. Shahidehpour, “*Adaptive formation of microgrids with mobile emergency resources for critical service restoration in extreme conditions*”, IEEE Trans. Power Syst., vol. 34, no. 1, pp. 742-753, 2019.

[6] A. Arif, B. Cui and Z. Wang, “*Switching device-cognizant sequential distribution system restoration*”, IEEE Transactions on Power Systems, vol. 37, no. 1, pp. 317-329, Jan. 2022.

[7] Y. Xu, C. -C. Liu, K. P. Schneider, F. K. Tuffner and D. T. Ton, “*Microgrids for service restoration to critical load in a resilient distribution system*”, IEEE Transactions on Smart Grid, vol. 9, no. 1, pp. 426-437, Jan. 2018.

[8] B. Chen, Z. Ye, C. Chen and J. Wang, “*Toward a MILP modeling framework for distribution system restoration*”, IEEE Transactions on Power Systems, vol. 34, no. 3, pp. 1749-1760, May 2019.

[9] C. Chen, J. Wang, F. Qiu and D. Zhao, “*Resilient distribution system by microgrids formation after natural disasters*”, IEEE Transactions on Smart Grid, vol. 7, no. 2, pp. 958-966, March 2016.

[10] Q. Zhang, Z. Ma, Y. Zhu and Z. Wang, “*A two-level simulation-assisted sequential distribution system restoration model with frequency dynamics constraints*”, IEEE Trans. Smart Grid, vol. 12, no. 5, pp. 3835-3846, 2021.

[11] Y. Du, X. Lu, J. Wang and S. Lukic, “*Distributed secondary control strategy for microgrid operation with dynamic boundaries*”, IEEE Transactions on Smart Grid, vol. 10, no. 5, pp. 5269-5282, Sept. 2019.

[12] Y. Wen, W. Li, G. Huang and X. Liu, “*Frequency dynamics constrained unit commitment with battery energy storage*”, IEEE Transactions on Power Systems, vol. 31, no. 6, pp. 5115-5125, Nov. 2016.

[13] B. Chen, C. Chen, J. Wang and K. L. Butler-Purry, “*Sequential service restoration for unbalanced distribution systems and microgrids*”, IEEE Trans. Power Syst., vol. 33, no. 2, pp. 1507-1520, March 2018.

[14] O. Bassey, K. L. Butler-Purry and B. Chen, “*Dynamic modeling of sequential service restoration in islanded single master microgrids*”, IEEE Transactions on Power Systems, vol. 35, no. 1, pp. 202-214, Jan. 2020.

[15] B. Chen, C. Chen, J. Wang and K. L. Butler-Purry, “*Multi-time step service restoration for advanced distribution systems and microgrids*”, IEEE Trans. Smart Grid, vol. 9, no. 6,

pp. 6793-6805, Nov. 2018.

[16] J. Zhao, H. Wang, Y. Liu, Q. Wu, Z. Wang and Y. Liu, "*Coordinated restoration of transmission and distribution system using decentralized scheme*", IEEE Trans. Power Syst., vol. 34, no. 5, pp. 3428-3442, 2019.

[17] J. Zhao, H. Wang, Y. Hou, Q. Wu, N. Hatziargyriou, W. Zhang and Y. Liu, "*Robust distributed coordination of parallel restored subsystems in wind power penetrated transmission system*", IEEE Transactions on Power Systems, vol. 35, no. 4, pp. 3213-3223, July 2021.

[18] H. Gu, R. Yan, T. K. Saha, E. Muljadi, J. Tan and Y. Zhang, "*Zonal inertia constrained generator dispatch considering load frequency relief*", IEEE Trans. Power Syst., vol. 35, no. 4, pp. 3065-3077, Jul. 2020.

[19] J. Li, X. Ma, C. Liu and K. P. Schneider, "*Distribution system restoration with microgrids using spanning tree search*", IEEE Transactions on Power Systems, vol. 29, no. 6, pp. 3021-3029, Nov. 2014.

[20] J. Zhao, Q. Wu, N. D. Hatziargyriou, F. Li and F. Teng, "*Decentralized data-driven load restoration in coupled transmission and distribution system with wind power*", IEEE Transactions on Power Systems, vol. 36, no. 5, pp. 4435-4444, Sept. 2021.

[21] Catalin Gavriluta, Cedric Boudinet, Friederich Kupzog, Antonio Gomez-Exposito, and Raphael Caire. "*Cyber-physical framework for emulating distributed control systems in smart grids*". International journal of electrical power & energy systems, 114:105375, 2020.

Appendix A. Optimization Problem

The optimizer solves a mixed integer conic program for calculating the restoration sequence and power generation for DERs. The objective is to maximize the restored load while accounting for several constraints.

The objective function (1) aims to maximize the total recovered loads over a rolling horizon [t , t+τ]. The priorities of the bus blocks were assigned randomly. All loads within a bus block had the same weight.

$$max \sum_{t\epsilon[t,t+\tau]} \sum_{i\epsilon\Omega_Z} \sum_{\phi\epsilon\Omega_\phi} w_i^Z x_{i,t}^Z P_{i,\phi,t}^Z \qquad (1)$$

DistFlow equations: The DistFlow equations [3] were used to determine the electrical power flow in the network. According to DistFlow, the generated power must be equal to the sum of load demand, net transmitted power and line losses observed at the bus. Similarly, they also define the voltage drop across buses as a function of line admittance. Constraints (2)-(3) define the nodal balance equations of active and reactive power for each phase. Equation (4), (5), (8) and (9) are disjunctive constraints that are active only when the corresponding line is energized i.e $x^L_{l,t}$ = 1 and enforces no constraint if $x^L_{l,t}$ = 0 considering M a considerably large number [3]. Equation (6) and (7) are hyperbolic constraints which is a special case of SOCP constraints [4]. Taking advantage from LinDistFlow, here it is reasonable to drop the squared term for resistance and reactance of distribution line to form constraint (8) and (9). Constraint (10) guarantees that the voltage is restricted within a specified limit [U^{min}_i , U^{max}_i], and will be 0 if the bus is in an outage area $x^B_{i,t}$=0

$$\sum_{l\epsilon\Omega_L(i,.)} P_{l,\phi,t}^L - \sum_{l\epsilon\Omega_L(.,i)} P_{l,\phi,t}^L = P_{i,\phi,t}^G + P_{i,\phi,t}^{ES} - x_{i,t}^Z P_{i,\phi,t}^Z$$
$$-P_{i,\phi,t}^{loss}, \forall i,\phi,t \qquad (2)$$

$$\sum_{l\epsilon\Omega_L(i,.)} Q_{l,\phi,t}^L - \sum_{l\epsilon\Omega_L(.,i)} Q_{l,\phi,t}^L = Q_{i,\phi,t}^G + Q_{i,\phi,t}^{ES} - x_{i,t}^Z Q_{i,\phi,t}^Z$$
$$-Q_{i,\phi,t}^{loss}, \forall i,\phi,t \qquad (3)$$

$$\tilde{U}_{i,\phi,t} \le U_{j,\phi,t} + M(2 - x_{l,t}^L - p_{l,\phi}), \forall l, ij\epsilon\Omega_L,\phi,t \qquad (4)$$

$$\tilde{U}_{i,\phi,t} \ge U_{j,\phi,t} - M(2 - x_{l,t}^L - p_{l,\phi}), \forall l, ij\epsilon\Omega_L,\phi,t \qquad (5)$$

$$R_l(P_{l,\phi,t}^{L2} + Q_{l,\phi,t}^{L2}) \le \tilde{U}_{i,\phi,t} P_{i,\phi,t}^{loss}, \forall l\epsilon\Omega_L, i,\phi,t \qquad (6)$$

$$X_l(P_{l,\phi,t}^{L2} + Q_{l,\phi,t}^{L2}) \le \tilde{U}_{i,\phi,t} Q_{i,\phi,t}^{loss}, \forall l\epsilon\Omega_L, i,\phi,t \qquad (7)$$

$$\begin{aligned} U_{i,\phi,t} - U_{j,\phi,t} &\ge 2(R_l P_{l,\phi,t}^L + X_l Q_{l,\phi,t}^L) \\ &+M(x_{l,t}^L + p_{l,\phi} - 2), \forall l, ij\epsilon\Omega_L,\phi,t \end{aligned} \qquad (8)$$

$$\begin{aligned} U_{i,\phi,t} - U_{j,\phi,t} &\le 2(R_l P_{l,\phi,t}^L + X_l Q_{l,\phi,t}^L) \\ &+M(2 - x_{l,t}^L - p_{l,\phi}), \forall l, ij\epsilon\Omega_L,\phi,t \end{aligned} \qquad (9)$$

$$x_{i,t}^B U_i^{min} \le U_{i,\phi,t} \le x_{i,t}^B U_i^{max}, \forall i,\phi,t \qquad (10)$$

Capacity limit constraints: Constraints (11) and (12) represent the active and reactive power limits of the distribution lines. Similarly, constraints (13)-(16) define the active and reactive power generation limits for DGs including both black-start capable synchronous generators and inverter based grid-following (GFL) DGs. It is worth-mentioning that grid-forming DGs are connected at the beginning to provide reference voltage and frequency for the microgrid. GFL DGs connect at a later stage to provide active and reactive powers to the loads. Constraint (15) and (16) of the GFL inverters are multiplied by the binary variable which makes them relaxed when the corresponding GFL inverter is not energized. Constraint (17) and (18) imposes a limit on the ramping of output power of the generators.

$$-x^L_{l,t}P^{Lmax}_l \le P^L_{l,\phi,t} \le x^L_{l,t}P^{Lmax}_l, \forall l \epsilon \Omega_L, \phi, t \qquad (11)$$

$$-x^L_{l,t}Q^{Lmax}_l \le Q^L_{l,\phi,t} \le x^L_{l,t}Q^{Lmax}_l, \forall l \epsilon \Omega_L, \phi, t \qquad (12)$$

$$0 \le P^G_{i,\phi,t} \le P^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{SG}}, \phi, t \qquad (13)$$

$$0 \le Q^G_{i,\phi,t} \le Q^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{SG}}, \phi, t \qquad (14)$$

$$0 \le P^G_{i,\phi,t} \le x^G_{i,t}P^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{IDG}}, \phi, t \qquad (15)$$

$$0 \le Q^G_{i,\phi,t} \le x^G_{i,t}Q^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{IDG}}, \phi, t \qquad (16)$$

$$-\Delta P^{Gmax}_i \le P^G_{i,\phi,t} - P^G_{i,\phi,t-1} \le \Delta P^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{G}}, \phi, t \quad (17)$$

$$-\Delta Q^{Gmax}_i \le Q^G_{i,\phi,t} - Q^G_{i,\phi,t-1} \le \Delta Q^{Gmax}_i, \forall i \epsilon \Omega_{\mathrm{G}}, \phi, t \quad (18)$$

ESS constraints: Constraint (19) defines the active power discharge limits for ESS. Constraint (20) illustrates that ESS and generator can not co-exist at a particular bus i. Constraint (21) sets a limit on the SOC of ESS. Similarly constraint (22) defines the operational limits for discharging of energy storage systems for a duration Δt with efficiency η^{ES}

$$0 \le P^{ES}_{i,\phi,t} \le x^{ES}_{i,t}P^{Emax}_i, \forall i \epsilon \Omega_{\mathrm{ES}}, \phi, t \qquad (19)$$

$$x^{ES}_{i,t} + x^G_{i,t} \le 1, \forall i, t \qquad (20)$$

$$SOC^{Em}_i \le SOC^{ES}_{i,\phi,t} \le SOC^{EM}_i, \forall i \epsilon \Omega_{\mathrm{ES}}, \phi, t \qquad (21)$$

$$SOC^{ES}_{i,\phi,t} = SOC^{ES}_{i,\phi,t-1} - \eta^{ES}_i \Delta t P^{ES}_{i,\phi,t}, \forall i \epsilon \Omega_{\mathrm{ES}}, \phi, t \quad (22)$$

Inter-connection constraints: Constraints (23)-(30) defines physical connections among buses, lines, IBDGs and loads during the restoration process. Constraint (23) ensures that grid-following inverter will be switched on, if the corresponding bus is energized. Constraint (24) implies that a switchable line can only be energized when both end buses are energized. Whereas, constraint (25) implies that a non-switchable line can be energized when either of its end buses are energized. Constraint (26) ensures that a switchable load can only be energized if the corresponding bus is energized, otherwise it will be off. On the other hand, constraint (27) defines that a non-switchable load adopts the same energizing status as that of its corresponding bus. Constraints (28)-(30) enforces that grid-following inverters, switchable line and switchable loads once energized can not be tripped during the restoration process.

$$x^G_{i,t} \leq x^B_{i,t}, \forall i \epsilon \Omega_{\mathrm{IDG}} | \Omega_{\mathrm{ES}}, t \qquad (23)$$

$$x^L_{l,t} \leq x^B_{i,t} \wedge x^L_{l,t} \leq x^B_{j,t}, \forall l, ij \epsilon \Omega_{\mathrm{SL}}, t \qquad (24)$$

$$x^L_{l,t} = x^B_{i,t} \vee x^L_{l,t} = x^B_{j,t}, \forall l, ij \epsilon \Omega_{\overline{SL}}, t \qquad (25)$$

$$x^Z_{i,t} \leq x^B_{i,t}, \forall i \epsilon \Omega_{\mathrm{SZ}}, t \qquad (26)$$

$$x^Z_{i,t} = x^B_{i,t}, \forall i \epsilon \Omega_{\overline{SZ}}, t \qquad (27)$$

$$x^G_{i,t} - x^G_{i,t-1} \geq 0, \forall i \epsilon \Omega_{\mathrm{IDG}} | \Omega_{\mathrm{ES}}, t \qquad (28)$$

$$x^L_{l,t} - x^L_{l,t-1} \geq 0, \forall l \epsilon \Omega_{\mathrm{SL}}, t \qquad (29)$$

$$x^Z_{i,t} - x^Z_{i,t-1} \geq 0, \forall i \epsilon \Omega_{\mathrm{SZ}}, t \qquad (30)$$

Structural constraints: DistFlow model is valid only for a radial distribution network. This group of constraints ensures that the formed MGs remain isolated from each other and there exist a radial topology within the MGs. Constraint (31) enforces that if a bus is located in a bus block, then the bus would have the same energizing status as that of the corresponding bus block. In order to avoid loops, constraint (32) ensures that if both ends of a switchable line are already energized at a previous interval t − 1, then the line cannot be closed. Similarly if a bus block is not energized previously, then constraint (33) implies that if can only be energized by at most one of the connected switchable lines. Constraint (34) ensures that an energized switchable line can energize the connected bus block. In support of constraint (33), the constraint (35) requires that a switchable line can only be energized at time t if at least one of the connected bus block is already energized at time t − 1.

$$x^B_{i,t} = x^{BB}_{i,t}, \forall i \epsilon \Omega_{\mathrm{BB}}, t \qquad (31)$$

$$\begin{aligned}(x^{BB}_{i,t} - x^{BB}_{i,t-1}) + (x^{BB}_{j,t} - x^{BB}_{j,t-1}) \geq x^L_{l,t} - x^L_{l,t-1} \\ , \forall l, ij \epsilon \Omega_{\mathrm{SL}}, t \geq 2\end{aligned} \qquad (32)$$

$$\begin{aligned}\sum_{ki, k\epsilon\Omega_i} (x^L_{ki,t} - x^L_{ki,t-1}) + \sum_{ij, j\epsilon\Omega_i} (x^L_{ij,t} - x^L_{ij,t-1}) \\ \leq 1 + M x^{BB}_{i,t-1}, \forall k, ij \epsilon \Omega_{\mathrm{SL}}, t \geq 2\end{aligned} \qquad (33)$$

$$x_{i,t-1}^{BB} \leq \sum_{ki,k\epsilon\Omega_i} x_{ki,t}^{L} + \sum_{ij,j\epsilon\Omega_i} x_{ij,t}^{L}, \forall k, ij\epsilon\Omega_{\mathrm{SL}}, t \geq 2 \quad (34)$$

$$x_{ij,t}^{L} \leq x_{i,t}^{BB} + x_{j,t}^{BB}, \forall ij\epsilon\Omega_{\mathrm{SL}}, t \geq 2 \quad (35)$$

Dynamic constraints: After a restoration stage is executed, the maximum values for nodal frequency and voltage deviation is recorded. These values are translated into a numerical parameter for expressing the state of the microgrid. Equation (36) presents the formula for calculating the microgrid disturbance factor. This factor is used to restrain or modulate the amount of restored load in a microgrid for the next stage, as expressed in (38)-(39). The maximum nodal voltage deviation of each phase was incorporated to discourage unbalanced loading of the lines and generators. Constraint (37) determines the total number of active generators in a microgrid. This limits the disturbance factor within [0, 1].

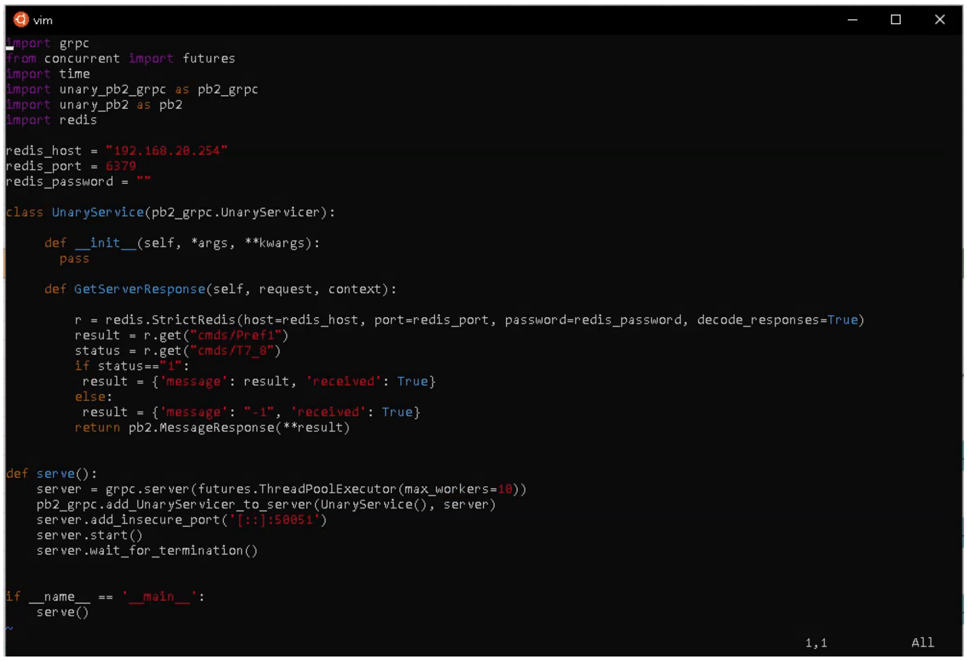
$$\begin{aligned} D_{m,t} = \frac{1}{N_t}[\sum_{i\epsilon\Omega_G} \frac{1}{4} x_{i,t}^{G}(\frac{\Delta f_{m,t}^{meas}}{f_0} + \\ \frac{\Delta V_{i,\phi_A,t}^{meas} + \Delta V_{i,\phi_B,t}^{meas} + \Delta V_{i,\phi_C,t}^{meas}}{V_0})], \forall i\epsilon\Omega_{\mathrm{G}}, m, t \end{aligned} \quad (36)$$

$$N_t = \sum_{i\epsilon\Omega_G} x_{i,t}^{G}, \forall i\epsilon\Omega_{\mathrm{G}}, t \quad (37)$$

$$\Delta P_{m,t}^{res} = \Delta P_{m,t-1}^{res} + \alpha(1 - D_{m,t-1}), \forall m, t \quad (38)$$

$$0 \leq P_{m,t}^{res} \leq (P_{m,t-1}^{res} + \Delta P_{m,t}^{res}), \forall m, t \quad (39)$$

Appendix B. Distributed MAS server code

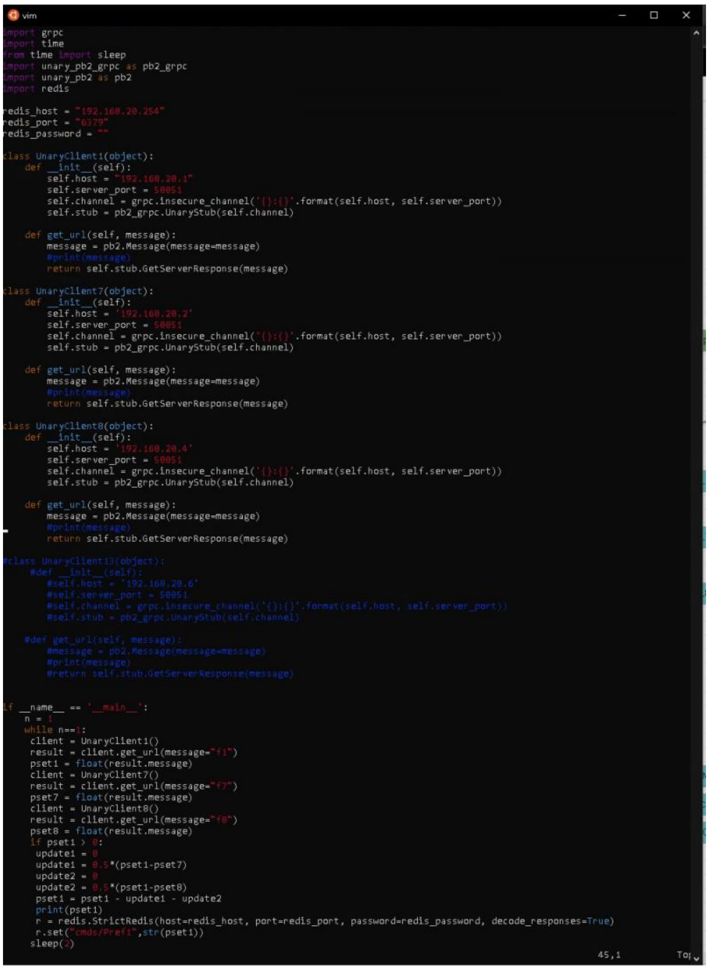


```
import grpc
from concurrent import futures
import time
import unary_pb2_grpc as pb2_grpc
import unary_pb2 as pb2
import redis

redis_host = "192.168.20.254"
redis_port = 6379
redis_password = ""

class UnaryService(pb2_grpc.UnaryServicer):

     def __init__(self, *args, **kwargs):
       pass

     def GetServerResponse(self, request, context):

         r = redis.StrictRedis(host=redis_host, port=redis_port, password=redis_password, decode_responses=True)
         result = r.get("cmds/Pref1")
         status = r.get("cmds/T7_8")
         if status=="1":
          result = {'message': result, 'received': True}
         else:
          result = {'message': "-1", 'received': True}
         return pb2.MessageResponse(**result)


def serve():
    server = grpc.server(futures.ThreadPoolExecutor(max_workers=10))
    pb2_grpc.add_UnaryServicer_to_server(UnaryService(), server)
    server.add_insecure_port('[::]:50051')
    server.start()
    server.wait_for_termination()


if __name__ == '__main__':
    serve()
~
                                                                         1,1           All
```

Figure 17: gRPC Server code

```
import grpc
from concurrent import futures
import time
import unary_pb2_grpc as pb2_grpc
import unary_pb2 as pb2
import redis

redis_host = "192.168.20.254"
redis_port = 6379
redis_password = ""

class UnaryService(pb2_grpc.UnaryServicer):

    def __init__(self, *args, **kwargs):
      pass
    def GetServerResponse(self, request, context):

        r = redis.StrictRedis(host=redis_host, port=redis_port, password=redis_password, decode_responses=True)
        result = r.get("cmds/Pref1")
        status = r.get("cmds/T7_8")
        if status=="1":
         result = {'message': result, 'received': True}
        else:
         result = {'message': "-1", 'received': True}
        return pb2.MessageResponse(**result)


def serve():
    server = grpc.server(futures.ThreadPoolExecutor(max_workers=10))
    pb2_grpc.add_UnaryServicer_to_server(UnaryService(), server)
    server.add_insecure_port('[::]:50051')
    server.start()
    server.wait_for_termination()


if __name__ == '__main__':
    serve()
```

Appendix C. Distributed MAS client code



```
import grpc
import time
from time import sleep
import unary_pb2_grpc as pb2_grpc
import unary_pb2 as pb2
import redis

redis_host = "192.168.20.254"
redis_port = "6379"
redis_password = ""

class UnaryClient1(object):
    def __init__(self):
        self.host = "192.168.20.1"
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

class UnaryClient7(object):
    def __init__(self):
        self.host = '192.168.20.2'
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

class UnaryClient8(object):
    def __init__(self):
        self.host = '192.168.20.4'
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

#class UnaryClient13(object):
     #def __init__(self):
        #self.host = '192.168.20.6'
        #self.server_port = 50051
        #self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        #self.stub = pb2_grpc.UnaryStub(self.channel)

    #def get_url(self, message):
        #message = pb2.Message(message=message)
        #print(message)
        #return self.stub.GetServerResponse(message)


if __name__ == '__main__':
    n = 1
    while n==1:
     client = UnaryClient1()
     result = client.get_url(message="f1")
     pset1 = float(result.message)
     client = UnaryClient7()
     result = client.get_url(message="f7")
     pset7 = float(result.message)
     client = UnaryClient8()
     result = client.get_url(message="f8")
     pset8 = float(result.message)
     if pset1 > 0:
      update1 = 0
      update1 = 0.5*(pset1-pset7)
      update2 = 0
      update2 = 0.5*(pset1-pset8)
      pset1 = pset1 - update1 - update2
      print(pset1)
      r = redis.StrictRedis(host=redis_host, port=redis_port, password=redis_password, decode_responses=True)
      r.set("cmds/Pref1",str(pset1))
    sleep(2)
                                                                                      45,1          Top
```

Figure 18: gRPC Client Stub

```
import unary_pb2_grpc as pb2_grpc
import unary_pb2 as pb2
import redis

redis_host = "192.168.20.254"
redis_port = "6379"
redis_password = ""

class UnaryClient1(object):
    def __init__(self):
        self.host = "192.168.20.1"
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

class UnaryClient7(object):
    def __init__(self):
        self.host = '192.168.20.2'
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

class UnaryClient8(object):
    def __init__(self):
        self.host = '192.168.20.4'
        self.server_port = 50051
        self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        self.stub = pb2_grpc.UnaryStub(self.channel)

    def get_url(self, message):
        message = pb2.Message(message=message)
        #print(message)
        return self.stub.GetServerResponse(message)

#class UnaryClient13(object):
    #def __init__(self):
        #self.host = '192.168.20.6'
        #self.server_port = 50051
        #self.channel = grpc.insecure_channel('{}:{}'.format(self.host, self.server_port))
        #self.stub = pb2_grpc.UnaryStub(self.channel)

    #def get_url(self, message):
        #message = pb2.Message(message=message)
        #print(message)
        #return self.stub.GetServerResponse(message)



if __name__ == '__main__':
    n = 1
    while n==1:
     client = UnaryClient1()
     result = client.get_url(message="f1")
     pset1 = float(result.message)
     client = UnaryClient7()
     result = client.get_url(message="f7")
     pset7 = float(result.message)
     client = UnaryClient8()
     result = client.get_url(message="f8")
     pset8 = float(result.message)
     if pset1 > 0:
      update1 = 0
      update1 = 0.5*(pset1-pset7)
      update2 = 0
      update2 = 0.5*(pset1-pset8)
      pset1 = pset1 - update1 - update2
      print(pset1)
      r = redis.StrictRedis(host=redis_host, port=redis_port, password=redis_password, decode_responses=True)
      r.set("cmds/Pref1",str(pset1))
     sleep(2)
```

Disclaimer

This document contains material, which is copyrighted by the authors and may not be reproduced or copied without permission.

The commercial use of any information in this document may require a licence from the proprietor of that information.

Neither the Lab Access User Group as a whole, nor any single person warrant that the information contained in this document is capable of use, nor that the use of such information is free from risk. Neither the Lab Access User Group as a whole, nor any single person accepts any liability for loss or damage suffered by any person using the information.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Copyright Notice

This project has received funding from the European Union's
Horizon 2020 research and innovation programme under
Grant Agreement No. 870620.