Microgrid Management using a Distributed Multi-agent Control System

Mohamad Fares Al Jajeh

Department of Electrical and Computer Engineering
McGill University, Montreal, Canada

April 2020

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering

Abstract

This thesis proposes a distributed microgrid control system (DCS) framework for controlling microgrid assets, including distributed energy resources, loads and the point of interconnection. The design employs a multi-agent system, where each asset is assigned an agent. The agents use neighbor-to-neighbor communications to coordinate. The objective of the work is to demonstrate the DCS capability of meeting microgrid dispatch and transition function requirements. As part of the agent design, distributed optimization for each microgrid function is fulfilled by the consensus algorithm. The dispatch of controllable distributed energy resources is optimized in terms of cost while renewable energy resources operate at maximum power point tracking. Distributed load curtailment is carried out considering different load types, sizes, and costs. Furthermore, a distributed priority queuing algorithm is proposed for emergency load shedding that can select a specific load using predetermined criteria. The control system features centralized sourcing of critical commands during emergency events but keeps the logic and execution of those commands distributed.

A microgrid benchmark is developed and validated in Simulink to test the DCS against challenging microgrid settings that include high penetration of renewable energy. The performance of the DCS is quantified with control performance metrics and is compared against standard metrics. Moreover, DCS assets optimization and reliability indices are compared to that of centralized control. Practical testing scenarios were developed for each microgrid control objective. To validate the DCS operation, the tests were conducted on a hardware-in-the-loop testbed created using hardware controllers and a real-time simulator. It is shown that the main advantages of a microgrid DCS are in its enhanced flexibility and reliability.

Résumé

Cette thèse propose un cadre de système de contrôle de microréseaux distribués (DCS) pour contrôler les atouts de microréseaux, y compris les ressources énergétiques distribuées, les charges et le point d'interconnexion. La conception utilise un système multi-agents, dans lequel chaque atout est affecté à un agent. Les agents utilisent des communications de voisin à voisin pour se coordonner. L'objectif des travaux est de démontrer la capacité du DCS à répondre aux exigences de répartition et de transition des microréseaux. Dans le cadre de la conception de l'agent, l'optimisation distribuée pour chaque fonction de micro-réseau est réalisée par l'algorithme de consensus. La répartition des ressources énergétiques distribuées contrôlables est optimisée en termes de coût tandis que les ressources énergétiques renouvelables fonctionnent au suivi du point de puissance maximale. La réduction de la charge répartie est effectuée en tenant compte des différents types, tailles et coûts de charge. De plus, un algorithme de mise en file d'attente à priorité distribuée est proposé pour le délestage d'urgence qui peut sélectionner une charge spécifique en utilisant des critères prédéterminés. Le système de contrôle propose un approvisionnement centralisé des commandes critiques lors des événements d'urgence, mais conserve la logique et l'exécution de ces commandes distribuées.

Un référentiel de microréseau standard est développé et validé dans Simulink pour tester le DCS par rapport aux paramètres de microréseaux difficiles qui incluent une forte pénétration des énergies renouvelables. Les performances du DCS sont quantifiées avec des métriques de performances de contrôle et sont comparées aux métriques standard. De plus, les indices d'optimisation et de fiabilité des atouts DCS sont comparés à ceux du contrôle centralisé. Des scénarios d'essais pratiques ont été élaborés pour chaque objectif de contrôle du microréseau. Pour valider le fonctionnement du DCS, les tests ont été effectués sur un hardware-in-the-loop créé à l'aide de contrôleurs matériels et d'un simulateur en temps réel. Il est démontré que les avantages d'un microréseau DCS résident dans sa flexibilité et sa fiabilité accrues.

Acknowledgements

I would like to sincerely thank my supervisor, Prof. Géza Joós, for his academic support and the valuable opportunities he provided me throughout my master's program. I highly appreciate the chance he gave me to work with an industrial partner and to become an intern there. I am also very grateful for his technical and financial support that allowed me to present my work at a prime power systems conference. I am humbled by his technical expertise and outlook. Being his student has been a privilege and an honor.

I would like to further acknowledge the financial support provided by the NSERC/Hydro-Quebec Industrial Research Chair on the Integration of Renewable Energies and Distributed Generation, and the InnovÉÉ project.

I would like to thank my friend and mentor Dr. Syed Qaseem Ali for his guidance on life and research in general. Syed believed in me, in my work and helped me transition from the lab to the industry. On this note, I would also like to thank the rest of my colleagues at OPAL-RT for their feedback on my work when I presented my demo on distributed control at the company and for the great times that we shared during my internship. Specifically, I would also like to thank Jean-Nicolas Paquin, Shijia Li, Carlos Rangel and the rest of my colleagues at OPAL-RT for their feedback on my work during the various stages of my master's.

I would like to express my gratitude toward Sanja Dzeletovic for being a great friend inside and outside the lab. Sanja made a mark on my personality, towards the better. I also thank her for her valuable time in providing me with editorial feedback on my thesis. I want to also give thanks to my friend and colleague Ilja Novickij, for the good memories we made in the lab and the technical discussions and arguments about research and life. I would like to thank Chu Sun for being a good travel and work buddy, and for his time and comments while attending the meetings at OPAL-RT. I would also like to thank my lab buddies, Marzuq Ajani and Mohammad Ata, and the rest for all the good lab memories. I also extend my appreciation to Quishi Cui and Dmitry Rimorov for their valuable guidance during my early lab days.

Finally, I thank my dad, mom, my brother and his family, and my sister and her family, for supporting my decision to go to Montreal to pursue my studies. My gratitude goes to the one God, who has blessed me throughout my life with your love and everything that I am today.

Preface

One conference paper resulted from the work presented in this thesis [1]. The candidate is the primary contributor in defining the problem, proposing the solution, designing and developing the solution, and designing the test setup, testing the solution, analyzing the results, and writing the resulting conference paper and this thesis.

> M. F. A. Jajeh, S. Q. Ali, G. Joos, and I. Novickij, "Islanding of a Microgrid Using a Distributed Multi-agent Control System," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 29 Sept.-3 Oct. 2019, pp. 6286-6293, doi: 10.1109/ECCE.2019.8912499.

Dr. Syed Qaseem Ali from OPAL-RT Technologies Inc. contributed by providing editorial comments on the paper writing and technical comments on the test cases and results. Prof. Geza Joos contributed by supervising the primary author, by helping him shape and refine the contributions of the paper and in providing feedback on the paper's presentation, technical contents and results. Ilja Novickij contributed in helping to establish the test setup IEC 61850 communications.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

EMS	Energy Management System
ESS	Energy Storage System
CCS	Centralized Control System
DCS	Distributed Control System
DER	Distributed Energy Resource
DPQ	Distributed Priority Queuing
DSO	Distribution System Operator
EICC	Equal Incremental Cost Criteria
EENS	Expected Energy Not Served
MAS	Multi-agent System
MCB	Microgrid Circuit Breaker
MGCC	Microgrid Central Controller
MGCS	Microgrid Control System
MPPT	Maximum Power Point Tracking
MTTF	Mean Time to Fail
MTTR	Mean Time to Repair
PLL	Phase-locked Loop
POI	Point of Interconnection
RT	Response Time
SAIFI	System Average Interruption Frequency Index
SS	Steady-state
ST	Settling time

List of Symbols

$C_i(P_{Gi})$	The cost function of asset *i*
μ^{up}	Upper inequality constraints
μ^{down}	Lower inequality constraints
r_i	Incremental cost at agent *i*
L	Laplacian Connectivity Matrix
ε_i	Learning Rate
$\mathrm{d}[t]$	Objective function
$r_j[t]$	Consensus variable at agent j
P_{poi}	Active power measurement at the POI
Q_{poi}	Reactive power measurement at the POI
$\mathrm{P_{DSO}}$	Active power DSO setpoint
$f_i(\mathrm{t})$	Frequency measurement at agent *i*
μ_i	Volt/VAR Consensus variable
γ_i	Load *i* price
λ_i	Load curtailment consensus variable
ND	DPQ Network discovery

Chapter 1

Introduction

1.1 Thesis Context and Scope

A microgrid is defined as "A group of interconnected loads and distributed energy resources with clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid and can connect and disconnect from the grid to enable it to operate in both grid-connected or island modes." [2]. The definition distinguishes the microgrid from other electrical entities in being able to survive on its own when the macrogrid fails. Therefore, the need for a resilient power grid and quick electricity recovery have put microgrids under the spotlight. In 2017, Hurricane Maria wiped out most of Puerto Rico's power grid, putting the country on an economic halt and millions of lives in danger. Less than a year later, the Energy Commission in Puerto Rico published an official regulatory document for microgrid development, acknowledging the need for microgrids as a resilient technology [3, 4]. Nonetheless, the innovation of microgrids is not necessarily reserved for disasters. As part of the 2050 Vision by the Canadian Electricity Association, the future power grid will need to take into account electric vehicles, distributed generation, and consumer participation by either demand response or generation [5]. Microgrids are pathways to achieve these goals [6]. Also, there have been many microgrid case studies around the world that have provided lessons learned [7-9]. One key element in these case studies is the microgrid control system, which is responsible for controlling its assets according to specified objectives [10].

1.2 Problem Definition

The common method of using a microgrid central controller (MGCC) to establish the microgrid control system (MGCS) is subject to several limitations. First, the MGCC is directly connected to all microgrid assets, receiving measurements from each asset and sending commands in return. Therefore, the MGCC puts the microgrid under jeopardy because it is a single point of failure. The failure of the MGCC leads to a blackout which entails economic consequences. Second, as the number of controllable nodes increases in a microgrid, the complexity of the MGCS becomes significantly larger. This may impose limitations since fast microgrid functions such as

islanding and reconnection from and to the macrogrid need to be solved in a short period [11]. Third, data privacy is a concern when detailed generation and consumption patterns are exposed to a single centralized entity [12].

In this thesis, the application of a distributed control system (DCS) as an alternative to centralized control is investigated. There is a need for a complete DCS framework that considers grid-connected and islanded modes, and transitions between the two modes [13], but such framework is missing from the literature. Furthermore, using a DCS for microgrid control is not straightforward. On transition, as will be covered in the literature review, very few papers exist on the DCS transition function, which includes planned and unplanned islanding, and many of its challenges remain to be addressed. For planned islanding, the DCS must coordinate the output power of DERs and, if needed, the curtailment of loads. For unplanned islanding, proper timing of load shedding and fast system response are required to save the microgrid from a blackout. On the other hand, on dispatch, the consensus algorithm is a popular choice for distributed coordination. However, tuning the consensus algorithm parameters for achieving fast and optimal dispatch for every microgrid mode of operation is difficult and requires extensive testing.

1.3 Literature Review

1.3.1 Microgrid Control System Frameworks and Topologies

Effective Microgrid control frameworks are critical for the reliable and efficient operation of the microgrid [6]. An important distinction noticed in the literature is between AC microgrids and DC microgrids as their control strategies are different. Hybrid AC/DC microgrids also exist in the literature whereas they are treated as two separate microgrids that work together [14] [15]. The focus of this thesis is on AC microgrids. Furthermore, the control structure primarily depends on the tailored objectives of the microgrid [16]. MGCS topologies include centralized, distributed and decentralized control systems [17]. Decision factors are related to scalability, plug and play, and optimality [18].

On decentralized control topology, authors in [19] proposed the use of a fully decentralized microgrid control system. The motivation is to be able to plug and play any DER without the requirement of setting up communications, hence slashing infrastructure costs. In addition, instead of having a single grid-forming DER, all dispatchable DERs behave as grid-forming nodes. The control would then depend primarily on frequency and voltage droop curves. The problem of

having circulating reactive current due to multiple master nodes is highlighted and argued to be solved by having local voltage control using Q-v droop [20]. Despite the decentralized control scheme's advantages in avoiding a single point of failure and achieving seamless plug and play, the overall control operation seems to fall behind in terms of coordination for economic dispatch, synchronization, and coordinated response during islanding. On the contrary, distributed control is an attempt to overcome the aforementioned decentralized control limitations at the cost of having an appropriate communication infrastructure.

On centralized control topology, centralized control systems (CCS) use a central controller for the dispatching of DERs. As emphasized by [21], centralized control allows the usage of real-time optimization routines, since all assets are connected to a central controller. Centralized approaches are more desirable in remote communities with a fixed infrastructure or campuses with critical demand-supply balances [16].

As opposed to centralized control, a distributed control strategy aims to solve the secondary control problem of microgrids while allowing the decisions on the control variables to be made locally [21]. In addition, it offers a robust control scheme if the communication network is varying or limited [22]. Therefore, a distributed paradigm follows a divide and conquer scheme, where the distributed controllers handle only local processes and perform localized decision making. This feature also paves the way toward assets autonomy, energy trading, and plug and play [13].

1.3.2 Microgrid Control Challenges

The microgrid control system (MGCS) face several control challenges during steady-state and transition operations [2][23]. Conventionally, synchronous machines are widely used to generate electricity. The energy stored in the rotating rotor of a machine can be instantly used to damp any electrical disturbances on the grid, such as sudden load changes or loss of other power sources [24]. On the other hand, solar and wind sources, which are prevalent in microgrids, have power electronic interfaces with the grid which lack the inertial response to disturbances that is normally available in synchronous machines. Furthermore, renewable DERs are non-dispatchable and stochastic by nature, which limits their controllability. Therefore, only dispatchable distributed energy resources (DERs) such as synchronous machines and energy storage can be utilized to establish coordinated control. Furthermore, timely control require that the MCGS frequently process data coming from all microgrid assets in real-time. On the other hand, the MGCS must

optimally dispatch assets, which can have different control options and need to be treated differently. Microgrid control challenges are magnified during microgrid islanded mode of operation as frequency and voltage are subject to a higher degree of variation during a disturbance. Since the microgrid DERs and loads are small when compared to the main grid, any change in the microgrid topology brings about a relatively large supply-demand imbalance that needs to be quickly and smoothly recovered. Finally, unplanned islanding is among the most challenging aspect of microgrid control. Unplanned islanding occurs when the microgrid circuit breaker (MCB) abruptly trips due to a variety of reasons such as an upstream grid fault. Consequently, the microgrid control system is forced to switch to the islanded mode of operation without preparation. Recovering the microgrid from unplanned islanding into steady-state islanded mode requires a great deal of coordination between assets to regulate the voltage and the frequency in their continuous operating regions.

1.3.3 Centralized Control Systems

Numerous methods can be used for centralized dispatching starting from classical literature [17]. The authors in [25] presented an Energy Manager that is composed of agents for DERs and loads and a master controller that imposes dispatches and restrictions. The agents submit demand and supply bids to the master agency, and in turn, the master controller sends dispatch orders. In [26], the authors implemented a double layer energy management system that handles scheduling in the top layer and dispatching in the bottom layer. In [27], the author implemented a formulated multi-objective optimization approach to meet cost, reliability, power fluctuation, peal loading, and GHG emissions. In [28], intelligent optimization techniques using ant-colony and particle swarm optimization were investigated for their ability to converge faster than traditional methods in multi-objective optimization, making them more suitable for real-time implementation. In [29], a rule-based dispatch technique is considered for every microgrid mode operation. In [30], authors consider a centralized model predictive control formulation that solves for an extended horizon. In [31], transition management is performed by means of proportional-integral controllers that take input from locally measured voltage and frequency and a supervisory control unit.

Despite the existence of CCS implementations as in [29], [30], and [31], they suffer several limitations. Their control systems lack flexibility and would need an upgrade if assets are added or removed. They require a large communication bandwidth to achieve the global reach that they

need [32]. Besides, their complete reliance on a central controller makes them more susceptible to user data leaks and complete system failures [33].

1.3.4 Distributed Control Systems

A distributed control system (DCS) uses local controllers to coordinate the dispatching of DERs. There is literature on numerous methods concerning distributed optimization [34]. The most popular way to do distributed optimization is to use the consensus algorithm [35]. The consensus algorithm is about reaching a consensus on a variable between several decision-making agents that have local communications with their neighbors. The consensus control scheme is categorized into either a regulator or a tracking problem. A regulator approach has all agents synchronize to a nominal value such as the frequency, while a tracking approach has agents subscribe to a certain leader agent [36].

The application of the consensus algorithm in economic dispatch has been investigated in the literature. The earliest work on the incremental cost consensus was in [22], where the authors provided important convergence analysis that set the foundation for later research. In [37], the consensus was developed as an economic dispatch technique by estimating the global supply-demand mismatch in a strongly connected network. Authors in [38] continued the work in [37] by taking transmission losses and unit modeling of DERs into consideration. In [39], the authors utilized the sub-gradient algorithm to achieve minimum operational costs using equal incremental rate criteria by doing frequency regulation. Also, adaptive convergence rates were adopted for the consensus algorithm by canceling the effect of the innovation term and switching to the average consensus algorithm once the frequency reached a certain settling threshold. On the other hand, [40] implemented a switching learning rate approach, where different learning rates are designed for different frequency deviation thresholds. It is important to note that utilizing a frequency regulation-based approach for economic dispatch, as opposed to supply and demand matching, solved the problem of estimating transmission losses since all losses are embedded in frequency deviation. However, for frequency deviation to occur, a DER with inertia must be present in the microgrid, which is a limitation to the applicability of the approach.

Although [39] and [40] proposed strategies for implementing an adaptive consensus learning rate, they did not solve the problem of finding the optimal learning rate in the first place, nor provided a detailed analysis on how the choice of the learning rate impacted the performance

of their algorithms. Also, if not properly tuned, the use of the consensus algorithm for real-time dispatch of distributed energy resources (DERs) can cause over-coordination if the learning rate is too large, and vice versa. Therefore, this thesis carries out detailed studies of the impact of consensus algorithm learning rate on the microgrid operation.

On the other hand, although [22, 36-40] contributed several advancements with regard to using the consensus algorithm for distributed microgrid control, none proposed a unified DCS topology for the real-time control of all microgrid assets, which would be required for a complete implementation and application of a distributed microgrid control system in a real microgrid context in all its modes of operation. Therefore, this thesis adds to the body of knowledge by proposing a detailed DCS topology based on a multi-agent system for the control of microgrid assets and describing detailed implementations of its agents.

Load management can be utilized by the MGCS to fix a relatively large supply-demand mismatch. Authors in [40] have taken a simple load shedding approach depending on critical frequency thresholds and integrated it as a part of the overall distributed algorithm. However, a problem observed is that the decentralized responsiveness of loads can cause frequency oscillations. Authors in [41] used a global information discovery technique to achieve load shedding [41]. An important limitation, however, is in the size of data exchanged during the discovery process as each agent gathers data about every other agent. Authors in [42] introduced a distributed randomized demand response to solve the problem of choosing the responsive load, however, its practicability is limited.

Furthermore, literature is scarce on microgrid transition management to islanded mode and vice versa in a distributed control context. In [43], the authors attempted to solve the reconnection problem by implementing a synchronization controller that sends voltage, frequency, and phase tracking setpoints to nearby leading DER nodes. Consequently, other DERs follow the leading DERs in the tracking process and coordinate by sharing active/reactive power outputs, and frequency/voltage measurements. For planned islanding, authors in [44] zeroed the power at the point of interconnection (POI) before disconnection but did not consider the case when generation capacity is insufficient for doing so. This limitation can result in load-supply unbalance in islanded mode and system instability. Authors in [45] only considered post fault islanding, which typically results in heavy transients that can only be sustained by a conventional generation resource with enough inertia. On unplanned islanding, the work in [45] does not showcase a fast load shedding

scheme that can quickly restore supply/demand unbalance to prevent the tripping of protection relays.

1.4 Proposed Solution

The purpose of this thesis is to develop a distributed microgrid control system (DCS) and to investigate its capability to deliver microgrid control core functions. Furthermore, the goal is to analyze the DCS performance in a complex microgrid with high penetration of renewable energy under different scenarios to evaluate its applicability. The proposed DCS is a network of interconnected controllers installed separately at designated assets. These DCS controllers use local measurements and neighbor-to-neighbor communications for calculating the proper dispatch setpoints depending on the microgrid mode of operation.

1.5 Methodology

The approach for designing a complete microgrid DCS was done in two main stages, whereas each stage of research involved testing and validating the multi-agent system (MAS). Specifically, at the first stage of research, the dispatch function was designed and tested. The dispatch function involves steady-state operation in grid-connected and islanded mode. The MAS framework, the role of each agent in fulfilling the dispatch function, and the algorithms used by each agent were designed in this stage. Also, in the first stage, a simple microgrid benchmark was employed, consisting of one asset of each type to test the DCS ability to meet active and reactive power setpoints in grid-connected mode, and frequency and voltage regulation in islanded mode. In the second stage of research, the transition function was designed and tested. The transition function involves the transition of the microgrid between grid-connected and islanded modes. Previously designed agents needed to be updated to accommodate the transition process. Furthermore, a new agent needed to be introduced for load management to meet the need for curtailment and fast load shedding. On the other hand, a more advanced microgrid benchmark was adopted, validated and modified to test the DCS against challenging operating conditions.

In each research stage, the microgrid and its control system were first tested in an offline simulation, then in an RTS environment, and finally on a C-HIL testbed. Finally, relevant quantification of results was done during the final testing stages, which gave insights into the tuning requirements needed by the control system to conform to standards. In this thesis, the

presented results are of the final MAS as tested on the microgrid benchmark using the C-HIL approach.

1.6 Thesis Contributions

1. A multi-agent system topology for the implementation of a distributed microgrid control system towards the real-time management of microgrid assets including controllable DERs, controllable loads, and the point of interconnection in grid-connected mode, islanded mode, and during transition between the two modes.
2. Distributed algorithms for assisting microgrid transition during planned and unplanned islanding. Need-based and cost-optimized load curtailment is utilized during planned islanding using the consensus algorithm considering the discontinuous nature of loads. Additionally, an algorithm is proposed for agent selection for shedding the largest and least critical load post unplanned islanding. The timing and execution of shedding or curtailing a load are centrally decided.
3. Identification of impacts on the microgrid of using the consensus algorithm to implement the dispatch function. A parametric analysis of the consensus algorithm learning rate was carried out to determine which learning rate values would achieve fast convergence and at what cost of operation.

1.7 Thesis Outline

The thesis is composed of 6 chapters that guide the reader from the context of microgrid control to the proposed DCS demonstration. The chapters are outlined as follows:

Chapter 1 introduced the research topic of microgrid control, proposed the DCS as a potential solution, presented the methodology and the overall work contributions.

Chapter 2 outlines the microgrid control requirements and details formulations for distributed control. Formulations include the mathematical formulation for economic dispatch and the consensus algorithm. Also, centralized control theory and its base controller are overviewed. Finally, the C-HIL tested and the MGCS testing requirements are described.

Chapter 3 proposes the control algorithms to be used by the DCS. First, the consensus algorithm applications for coordination of DERs power output and distributed load curtailment. Second, the distributed priority queuing algorithm (DPQ) is outlined in detail and performance analysis is carried out to assess its convergence.

Chapter 4 presents the implementation of the multi-agent system that makes up the DCS. Agents are designed to control the MCB, controllable DERs and loads. The inputs and outputs of each agent, the cross coordination between agents and their control logic are detailed.

Chapter 5 demonstrates the testing results. The chapter presents case studies and measured performance metrics. Furthermore, comparisons with centralized control on the cost of operation and reliability are carried out.

Chapter 6 summarizes the thesis, draws conclusions from the work conducted, and lays out the next steps for future research.

Chapter 2

Context, Theory, Tools, and Approaches

2.1 Introduction

In this chapter, the microgrid control requirements, mathematical formulations for dispatch and distributed control, and validation tools are presented. Setting the stage, the microgrid control system requirements and testing methodology are overviewed. Next, the control and mathematical formulations used for the design of the DCS are detailed. For comparison, the centralized control theory and its designed base case controller are outlined. Finally, the C-HIL testbed, which includes the microgrid benchmark model used and the experimental setup, is described.

2.2 Microgrid Overview and Formulation

2.2.1 Microgrid Structure, Assets, and Control System Requirements

Microgrids are distribution level electrical grids that have a single point of interconnection (POI) with a substation and have enough resources to be self-sufficient when islanded from the grid. A Microgrid contains a variety of assets that allows it to operate in grid-connected and islanded modes. Figure 2.1 shows a generalized microgrid structure. A microgrid control system (MGCS) is needed to control the operation of the assets. Assets may include synchronous machines, energy storage systems (ESS), PV farms, wind turbines, loads, protection relays and the POI circuit breaker. Each type of asset has a unique set of control inputs. For example, as shown in Table 2.1, PV and wind do not receive active power setpoints, as they operate in maximum power point tracking (MPPT) mode, however, if programmed, they can curtail their output power. On the other hand, an ESS can receive both active and reactive power setpoints. The MGCS is expected to send the commands that each asset needs according to the microgrid operational objectives and current mode of operation.

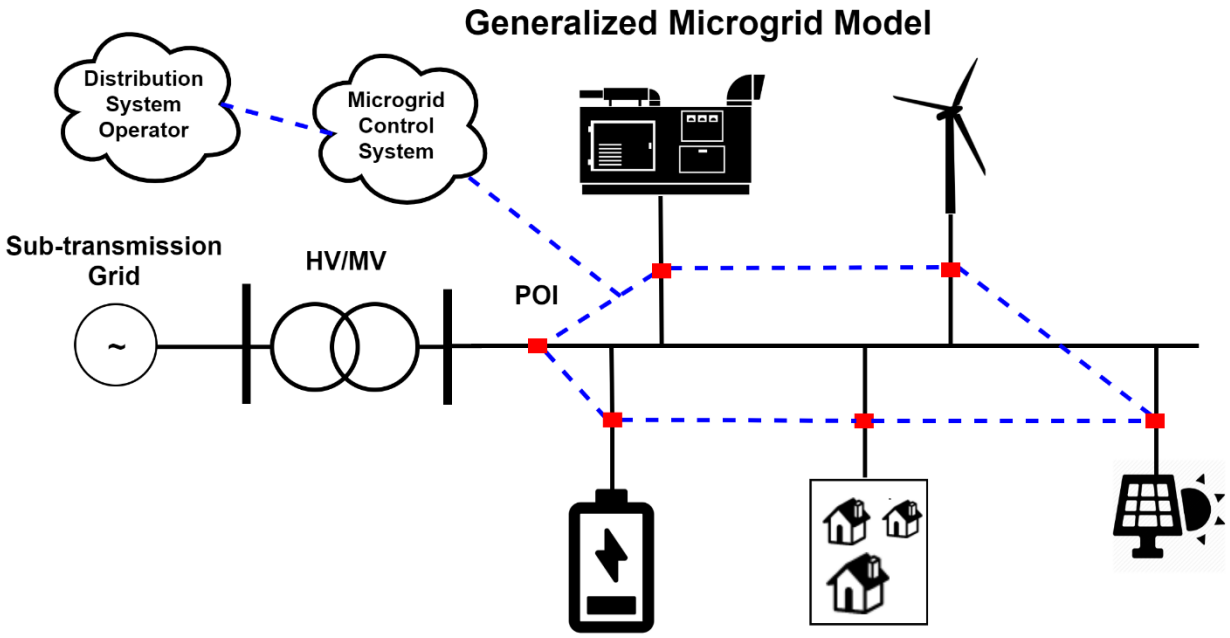

Figure 2.1 General microgrid schematic

Table 2.1 Asset control requirements

Asset	P Setpoint	Q Setpoint	Voltage	Frequency	Mode of Operation	Trip	Curtail
Synchronous machine	Y	Y	Y	Y	Y	Y	N
ESS	Y	Y	Y	Y	Y	Y	N
PV/Wind	N	Y	Y	N	N	Y	Y
POI circuit breaker	N	N	N	N	N	Y	N
Load	N	N	N	N	N	Y	Y
Protection Relay	N	N	N	N	Y	Y	N

2.2.2 Mathematical Formulation for Active Power Dispatch

In this section, the mathematical formulation for economic dispatching is specified. At any point during the microgrid operation, the supply-demand balance must be met, whether that demand is specified as a lumped sum exported power at the POI during grid-connected mode, or supply-demand balance in islanded mode. In both cases, minimizing the cost of operation is an objective that needs to be met. The economic dispatch problem is formulated as in [24]:

$$Min\ \sum C_i(P_{Gi}) - \mathrm{r}^T(\sum P_{Gi} - \sum P_{Di}) - \sum \mu_i{}^{max}(-P_{Gi} + P_{Gi}^{max}) - \sum \mu_i{}^{min}(P_{Gi} - P_{Gi}^{min}) \quad (1)$$

Where P_{Gi} is the total generated power from generation asset G_i, P_{Di} is the power demand from load D_i, C_i is the cost curve of asset i, r_i the incremental cost, μ^{max} and μ^{min} are multipliers associated maximum and minimum power output of unit i, respectively, and P_{Gi}^{max} and P_{Gi}^{min} are the maximum and minimum active power generation capacity, respectively. The corresponding Lagrange function is:

$$L = \sum C_i(P_{Gi}) - \mathrm{r}^T(\sum P_{Gi} - \sum P_{Di}) - \sum \mu^{max}(-P_{Gi} + P_{Gi}^{max}) - \sum \mu^{min}(P_{Gi} - P_{Gi}^{min}) \quad (2)$$

Furthermore, $P_{Gi} > 0$ if it is producing power and $P_{Gi} < 0$ if it is absorbing power. The generation allocation for minimum cost requirements must also satisfy the power balance equation:

$$\sum P_{Gi} - \sum P_{Di} - P_{loss} = 0 \quad (3)$$

Where P_{loss} is the total power loss in the network. The generated power by asset G_i is subject to limits on generation capacity as follows:

$$P_{Gi}{}^{min} < P_{Gi} < P_{Gi}{}^{max} \quad (4)$$

The cost curve C_i is quadratic [24] and is defined for synchronous machines assets as:

$$C_i(P_{Gi}) = \frac{1}{2}a_i P_{Gi}^2 + b_i P_{Gi} + c_i \quad (5)$$

Where the cost coefficient $a_i > 0$, while b_i and $c_i \geq 0$. As for the battery energy storage system (ESS), the cost curve is defined as in [33, 38, 39]:

$$C_i(P_{Gi}) = \frac{1}{2}a_i P_{Gi}^2 \quad (6)$$

This is termed pseudo-cost curve, where the coefficient $a_i > 0$. The implication of using (6) is that there is always a cost to storing and retrieving energy from the battery, hence its dispatch objective would be to minimize its usage. The same cost curve formulation in (6) is defined for controllable loads [33], reflecting the cost to curtail, where $P_{Gi} \geq 0$. A positive P_{Gi} for a load translates to curtailment. The load cost curve is chosen to be quadratic for two main reasons. First, as per common industrial contracts, the curtailment cost increases with the increase in the size of the curtailed load. Second, to simplify the modeling of the load cost curve, on the assumption that the final amount would be discretized afterward depending on the load itself. Finally, there is no cost curve associated with the solar nor wind and they are not involved in economic dispatching.

First-order optimality condition [46] implies:

$$\frac{\partial L}{\partial P_{Gi}} = 0 \quad (7)$$

Hence from (7):

$$a_i P_{G_i} + b_i - r_i - \mu^{max} - \mu^{min} = 0 \quad (8)$$

Therefore, at the optimal solution r_i can be calculated as:

$$r_i = a_i P_{G_i} + b_i - \mu^{max} - \mu^{min} \quad (9)$$

Where the incremental cost r is the derivative of the cost curve $C(P_{Gi})$ with respect to the output power P_{Gi}. It is the slope of the cost curve at a power setpoint P_{Gi}. Furthermore, according to [24], complementary slackness conditions imply that the terms $\sum \mu^{max}(-P_{Gi} + P_{Gi}^{max})$ and $\sum \mu^{min}\left(P_{Gi} - P_{Gi}^{min}\right)$ in (2) are always zero. Specifically, if P_{Gi} is within its generation limits, then $\mu^{max} = 0$ and $\mu^{min} = 0$. For all generation assets that have not reached their limits, the incremental cost r must be equal. Using the conditions above , the active power setpoints for controllable DERs can be calculated as:

$$\begin{cases} if\ b + a * P_{Gmin} \geq r_i\ then\ P_{Gi} = P_{Gmin} \\ if\ b + a * P_{Gmax} \leq r_i\ then\ P_{Gi} = P_{Gmax} \\ else:\ \ P_{Gi} = \frac{r_i - b_i}{a_i} \end{cases} \quad (10)$$

While the mathematical formulation in (1) is centralized, the localized usage of the cost functions and the equal incremental cost criteria equally applies to the distributed problem formulation.

Finally, other dispatch objectives are taken indirectly into consideration that fit in the context of a microgrid. Specifically, the objectives of minimizing fuel consumption and emissions. These are always met by allowing wind and solar to work in maximum power point tracking (MPPT) mode. These objectives imply operational assumptions that would influence the dispatching of controllable DERs during run-time.

2.3 Distributed Control Formulation and Theory

2.3.1 Distributed Control Theory

Figure 2.2 shows a typical DCS topology. As can be noticed, each node in the cyber environment is assigned to a node in the physical environment. Cyber nodes control physical nodes. Nodes that share common objectives and characteristics are grouped together and are expected to communicate. A group of nodes can coordinate with another group or type of nodes for the fulfillment of a higher-level objective that is common to both groups. Furthermore, each cyber node is expected to have a certain degree of autonomy over the control of its assigned

physical node. Finally, the system is capable of scaling both vertically, the computational capability of each controller, and horizontally, the number of controllers.

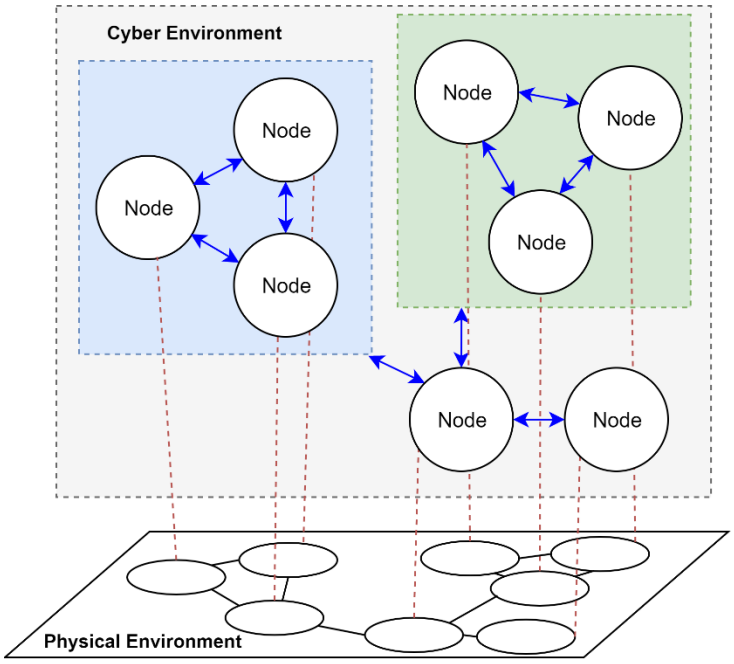

Figure 2.2 Distributed control system topology

2.3.2 Multi-Agent System Formulation

One good approach toward the design of a DCS is to model it as a multi-agent system (MAS). A MAS is a cluster of agents with specialized functions that use local measurements and inter-communications to achieve local or global objectives. The authors in [47] define an intelligent agent as a software program that have reactivity, pro-activeness and social ability. Reactivity is the ability of an agent to react to environmental changes. Pro-activeness is about the capability of an agent to adjust to environmental changes. Social ability is the use of agent communication language by agents to be able to converse beyond simply passing data between them. While reactivity and pro-activeness are retained in the DCS implementation in this thesis, the social ability as defined in [47] is not. The reason is that the functions and actions available for power system assets are well-defined and limited, and therefore it does not justify the use of ontology to converse as that would be a better fit for more general-purpose agents. However, the agents do communicate with data that is necessary to run distributed control algorithms. Finally, it is worth noting that agents are distributed in nature, which can help set modular control schemes for assets and microgrids at large, hence the use of MAS to model the design of a DCS.

2.3.3 Graph Theory

The microgrid communication topology adopted in the thesis can be modeled by a weighted undirected graph $G = (V, E, L)$ [48]. $V = (v_1, v_1, v_3 \ldots)$ is a finite non-empty set representing the vertices. Vertices typically represent the agents in the microgrid that have local controllers and are active in the distributed control paradigm. $E \subseteq vxv$ is a set of pairs of vertices that represent edges. An edge in E is a bi-directional communication link between two agents. A graph G is "strongly connected" if there is a path for every pair of distinct vertices. $L = [L_{ij}]$ is a weighted adjacency matrix [35]. An element L_{ij} is positive if there exists an edge between vertices v_i and v_j, otherwise, it is zero. Matrix $D = I - L$ is called the Laplacian of the graph G. Matrix D has to be a Perron matrix satisfying the following properties: 1. D is a row stochastic matrix, whereas, each row elements sum to one. 2. All eigenvalues are nonnegative and less than one $|\lambda_i| < 1$. Finally, $NEI_i = (v_j \in V : (v_i, v_j) \in E)$ is the set of neighbors of node v_i, and N_i shall denote the number of neighbors of v_i. Since self-loops are permitted, v_i also belongs to NEI_i.

2.3.4 Consensus Theory and Algorithms

The consensus algorithm is about reaching an agreement about the state of a variable x among several entities. The consensus is represented by equation (11) in its discrete form [35]. The data update process at an agent i at time $t+1$ is:

$$x_i[t+1] = \sum_{j=1}^{N_i} L_{ij} x_j[t] \tag{11}$$

Where x_i is the local data being processed such as incremental cost and x_j is the data obtained from a neighboring agent. N_i is the total number of agents connected to agent i. L is the connectivity matrix which essentially defines the weights on the data received from neighboring agents. To introduce objective constraints [49], equation (11) can be updated to:

$$x_i[t+1] = \sum_{j=1}^{N_i} L_{ij} x_j[t] + \varepsilon_i d_i[t] \tag{12}$$

Where $d_i[t]$ is the sub-gradient of agent i objective function. ε_i is the learning rate of agent i. The term $\varepsilon_i d_i[t]$ is called the innovation term. Alternatively, the consensus algorithm can be written in the following form:

$$x_i[t+1] = x_i[t] + \sum_{j=1}^{N_i} L_{ij}\big(x_j[t] - x_i[t]\big) + \varepsilon_i d_i[t] \tag{13}$$

Equations (12) and (13) are mathematically equivalent. It should be noted that the matrix L and the learning rate ε are the primary factors that determine the convergence rate of the

consensus algorithm. The learning rate ε has a direct effect on the convergence rate [37]. A small learning gain would result in a smooth yet slow convergence, whereas a large learning rate would result in a fast yet oscillating convergence [22]. Regarding the design of the connectivity matrix L, it can be shown that the second-largest eigenvalue λ_2 of the Laplacian $D = I - L$ plays an important role in regards to the convergence speed [50]. According to [50], designing a D with minimum λ_2 results in optimal convergence speed. However, solving for such a D in real-time is nearly impossible because many variables need to be solved. Besides, such a design would require knowledge of all generation and line parameters in the microgrid, information that is unavailable to all agents. Therefore, an alternative method is used for designing the coefficients of L that are adaptive to the varying topology. The method adopted is based on the Mean Metropolis method [50]:

$$\boldsymbol{L_{ij}} = \begin{cases} \frac{2}{N_i+N_j+\alpha} & i \neq j, j \in NEI_i \\ 1 - \sum_{j \in NEI_i} \frac{2}{N_i+N_j+\alpha} & i = j \\ 0 & No\ Connection \end{cases} \quad (14)$$

Where α is a small constant of a typical value of 0.001. The L_{ij} elements change values whenever the number of connected agents to itself or its neighbors' changes. Although the method suggested by [50] is sub-optimal, it's implementation is feasible and hence adopted in this thesis. Finally, it is possible for the learning rate ε and connectivity matrix L to be dynamically updated during run-time in order to optimally respond to the system requirements and needs. This is further illustrated in the case studies and analysis results in Chapter 5.

As an application, controllable DER assets can cooperatively regulate the microgrid frequency to its nominal value in islanded mode. These assets would need to operate in frequency-power droop control mode. The algorithm, which is derived in [39], shows that supply-demand discrepancies can be represented by the frequency deviation. Accordingly, the algorithm is stated as follows:

$$r_i[t+1] = \sum_{j=1}^{N_i} L_{ij} r_j[t] + \varepsilon_i(f^* - f_i(\mathrm{t})) \quad (15)$$

Where f^* is the rated frequency, f_i is the measured frequency at asset i, and ε_i is the learning rate used by agent i. Once the local incremental cost r_i [t+1] is calculated from (13), it can be used to calculate the power setpoint using (10).

Another application is voltage regulation by controllable DERs in islanded mode. This is needed to fix voltage perturbations when the load varies in steady-state islanded operation. The algorithm derived in [51], is restated as follows:

$$u_i[t+1] = \sum_{j=1}^{N_i} L_{ij} u_j[t] + \mu_i(\mathrm{v}^* - v_i(\mathrm{t})) \quad (16)$$

Where u_i is the consensus variable calculated by agent *i*. $v_i(t)$ is the measured voltage at asset *i*. v^* is the nominal voltage. Consequently, the calculated consensus u_i is used to set the reactive power setpoint of asset *i* using (28).

2.4 Centralized Control Theory and Comparison

Figure 2.3 shows a typical CCS topology. As can be noticed, all nodes in the physical environment are assigned to one node in the cyber environment. The central cyber node is responsible for controlling all the nodes in the physical environment. Only vertical scaling is possible in a CCS, which is limited by the technological advancement of processors available at the central node.

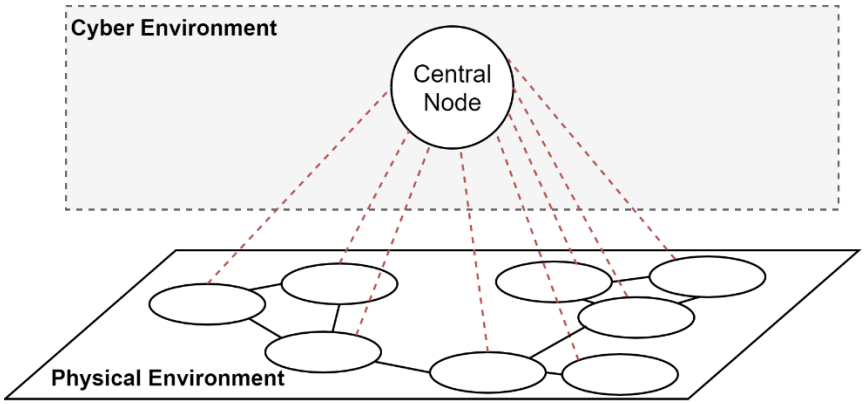

Figure 2.3 Centralized control system topology

The comparison between centralized and distributed control in this thesis will be based on the following points:

- Reliability

The question explored is how does the failure of the controller hardware affect reliability indices. The reliability metrics of interest are the System Average Interruption Frequency Index (SAIFI) and Expected Energy Not Served (EENS) [52]:

$$SAIFI = \frac{\sum_{i=1}^{N} INTRP_i}{N.n_{customer}} \quad (17)$$

$$EENS = \frac{\sum_{i=1}^{N} Ec_i}{N} \quad (18)$$

Where $INTRP_i$ is the total number of customer interruptions per year *i*, N is the total number of years, $n_{customer}$ is the total number of customers and Ec_i is the unserved energy in a year. A previous reliability study by [53] argued that the DCS presents advantages with respect to SAIFI and EENS. However, a similar study is explored in this thesis in section 5.4.1 taking into account the used microgrid benchmark presented in section 2.8 and improved assumptions.

- Cost of operation and resource optimization

The cost of microgrid operation depends on the results of the dispatch algorithm. Therefore, the explored question is how the dollar costs of microgrid operation compare between CCS and DCS dispatch algorithms.

- Hardware computational requirements

The question explored is how do the CCS and DCS differ in terms of processor loading. This is evaluated in terms of the required processor time in seconds to run the dispatch algorithm. The advantage of DCS over CCS was previously asserted in [54] and [55].

- Dispatch algorithm total convergence time

The question explored is how do the CCS and DCS compare in terms of total convergence time of their dispatch algorithms. This question is important to consider since the DCS use communications between controllers for every iteration of the dispatch algorithm which impacts the convergence time.

- Control system complexity and scalability

The question explored is how do the CCS and DCS compare in terms of the convergence rate of their algorithms. In [22], authors argued that the DCS is more scalable than a CCS, where they did a convergence analysis of the consensus algorithm, and found that the convergence is linear when a random communication network is used. In this thesis, this analysis is performed for the proposed distributed priority queuing algorithm in section 3.5.

- Control system flexibility

Control system flexibility is about the adaptability of a control system to the addition of new units or otherwise the removal of units from its control pool. The DCS advantage over CCS regarding control system flexibility was asserted in [17, 55]. In this thesis, the adaptability of the proposed DCS topology to the addition and removal of an agent will be presented.

2.5 Microgrid Control System Requirements

The topic of microgrid control systems has attracted a lot of interest and a functional standard was developed for it. The IEEE standard 2030.7 is dedicated to the specification of microgrid controllers [2]. Details of the standard specifications can be found in Appendix A. It is important to note that the overall microgrid control system hierarchy consists of three levels, namely: primary, secondary and tertiary control, as depicted in Figure 2.4. However, the standard specifies the MGCS to be exclusively about the secondary control level.

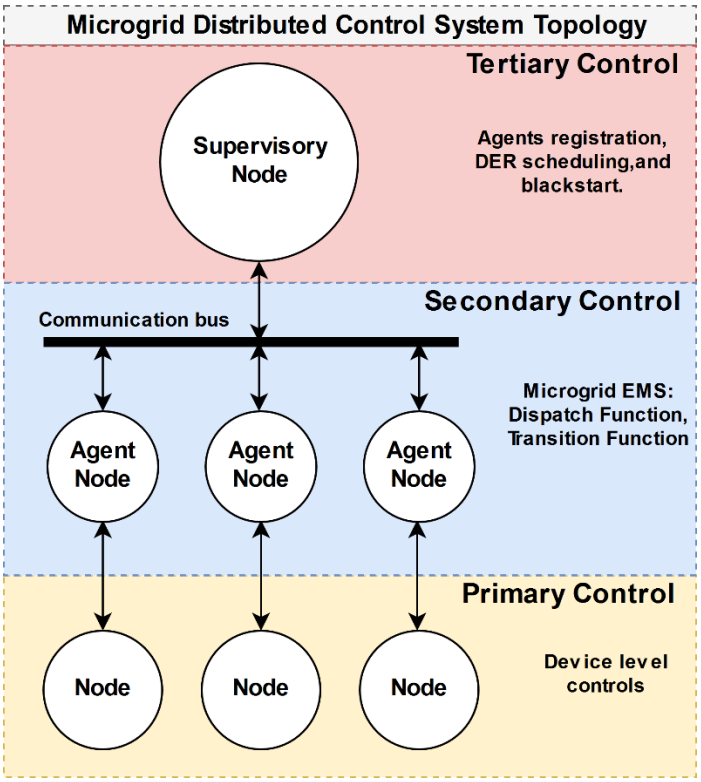

Figure 2.4 Distributed microgrid control system control hierarchy.

The MGCS functions treated in this thesis are specified as follows:

- Grid-connected steady-state power dispatch:

  - Coordination of controllable DER assets to meet the net power exchange requirement at the POI while maintaining economic operation and maximum renewable generation.
- Islanded steady-state power dispatch:
  - Involves the distributed coordination of controllable DER assets to regulate the microgrid frequency and voltage while maintaining economic operation and maximum renewable generation.
- Planned Islanding:
  - Involves the distributed coordination of controllable DER assets to zero the net-power at the POI followed by microgrid disconnection. Hybrid loads curtailment can be utilized if the DER assets' total power is not enough.
- Unplanned Islanding:
  - Involves the immediate response of assets to sudden microgrid disconnection to maintain microgrid stability. The response includes the fast response of DERs and the fast shedding of less critical loads if necessary.

Regarding the timeframe of events, the MGCS is expected to have fast-responding controls for the transient functions of planned and unplanned islanding, as well as slower and perpetual controls for steady-state functions in grid-connected and islanded modes [11]. Fast-responding controls include load shedding, load curtailment, and changing the DER mode of operation and power dispatch when the microgrid changes its mode of operation. On the other hand, slow controls involve solving for the power setpoints by the dispatch function in steady-state.

2.6 Microgrid Control System Testing

Regarding MGCS testing, the IEEE standard 2080.8 is dedicated to the testing of microgrid controllers and specifies testing metrics [11]. It entails testing procedures regarding initial conditions, events, and relevant measurements that pertain to the core functions in the IEEE standard 2030.7 [2]. The testing in this thesis as part of a case study for each MGCS function, which follows some of the suggested initial conditions and initiating events. Through the conducted case studies, the features of the proposed DCS are highlighted.

Furthermore, the results of the case studies are analyzed using the metrics suggested by the IEEE standard 2030.8. Specifically, the POI frequency and voltage should maintain in specific

operational regions. Therefore, measurements of voltage and frequency are recorded as per the data collection specified in the IEEE standard 2030.8 and compared against the allowable regions specified in the IEEE standard 1547 [56]. Furthermore, the system control response is analyzed where applicable in terms of response time, overshoot, settling time and SS error. Such analysis gives a quantified indication of the control system performance in meeting MGCS objectives.

Different research groups have established several benchmarks and testbeds for microgrid controllers. In [57], a comprehensive summary of microgrid testing techniques using real-time (RT), controller hardware-in-the-loop (C-HIL), power hardware-in-the-loop (P-HIL), emulation, and co-simulation by researchers is presented. Each technique, respectively, is one step closer to how the system will operate in real-time. Furthermore, the test fidelity is traded off with its coverage [11]. Specifically, since RT and C-HIL testbeds are based on generic simulated models, they have a larger testing coverage than a P-HIL testbed, which is based on specific physical apparatus. However, the P-HIL testbed has more fidelity than RT and C-HIL, as it reflects more accurately a deployed system. In [58], a PHIL-based microgrid testbed is where DERs are emulated to allow for quasi-realistic testing. The installation allows for microgrid control and optimal dispatching verification and testing. In [59], the authors investigated the prototyping of digital controllers when the microgrid has inverter-based DGs. In [60], a P-HIL testbed for testing multi-agent systems was developed. In [61], the experimental validation was done on a 3-phase 208 V AC microgrid where the setup has four inverters-based sources with similar topologies, two loads, and all connected in a radial connection. A processor board was used to model the communication channels and execute control routines. In [62], the authors implemented MAS HIL testing using a real-time simulator. Their implementation used a central coordinator of the microgrid, so their test-bench has both distributed and centralized components, and the communication between the customer portal and the agents relied on the Contract Net Protocol. In [29], the authors developed a C-HIL testbed using a real-time digital simulator and a hardware controller. The authors argued that the C-HIL approach is most appropriate for testing of MGCS. C-HIL offers enough fidelity for testing microgrid dispatch and transition functions while allowing the testbed to be flexible enough to test the MGCS on other microgrid topologies. This thesis agrees with the authors in [29] and uses for validation a C-HIL testbed presented in section 2.8.

When carried out properly, a real-time simulation can be an effective tool for validating control systems. A simulation on a real-time simulator runs against the wall clock at a discrete

time step. This allows the effective interfacing of external hardware with the model simulation, which helps in validating the designed control system [63]. A real-time simulator can simulate a microgrid with time steps ranging from nanoseconds to milliseconds [64]. Using the nanoseconds timestep to run models is most appropriate for simulating fast transients' events and detailed power electronics switching. However, nanosecond simulations are too detailed, and therefore too slow, to simulate in real-time larger-scale systems such as microgrids. On the other hand, models ran in the milliseconds timestep are appropriate for simulating power systems dynamics involving thousands of busses during faults but are not accurate enough for testing microgrid transition and protection functions. In between, the microseconds timestep is accurate enough for simulating transient microgrid functions, where average models are used for power electronics switching, giving computational room on the simulator processors to scale up the model size for the real-time simulation of the microgrid scale. Therefore, the ideal model timestep for microgrid testing is in the microseconds range, which is adopted in this thesis.

2.7 Simulation Tools

2.7.1 Microgrid Benchmark Model

The microgrid model used is a medium voltage distribution benchmark [65] modified to work as a microgrid. Using the benchmark can aid in the design and validation of MGCS functions under a variety of microgrid conditions. The benchmark development was validated before usage, whereas the validation results are presented in Appendix B. Figure 2.5 shows the modified distribution benchmark model. The model is modified with a POI at bus 2 and DERs at various points of the microgrid beyond bus 1. The locations and ratings of the DERs are a custom configuration and are presented in Table 2.2. The ESS is located at bus 2 in order to provide fast voltage for the POI. On the other hand, the diesel generators are located at busses 4 and 9 to support the bus voltages downstream of the feeder. The design was made considering it is a community microgrid [7]. Specifically, the utility has setup the assets that are essential for the microgrid to come into being, such as the POI, the wind farm, the ESS and the diesel generators. Furthermore, individual investors have PV farms installed at their respective properties.

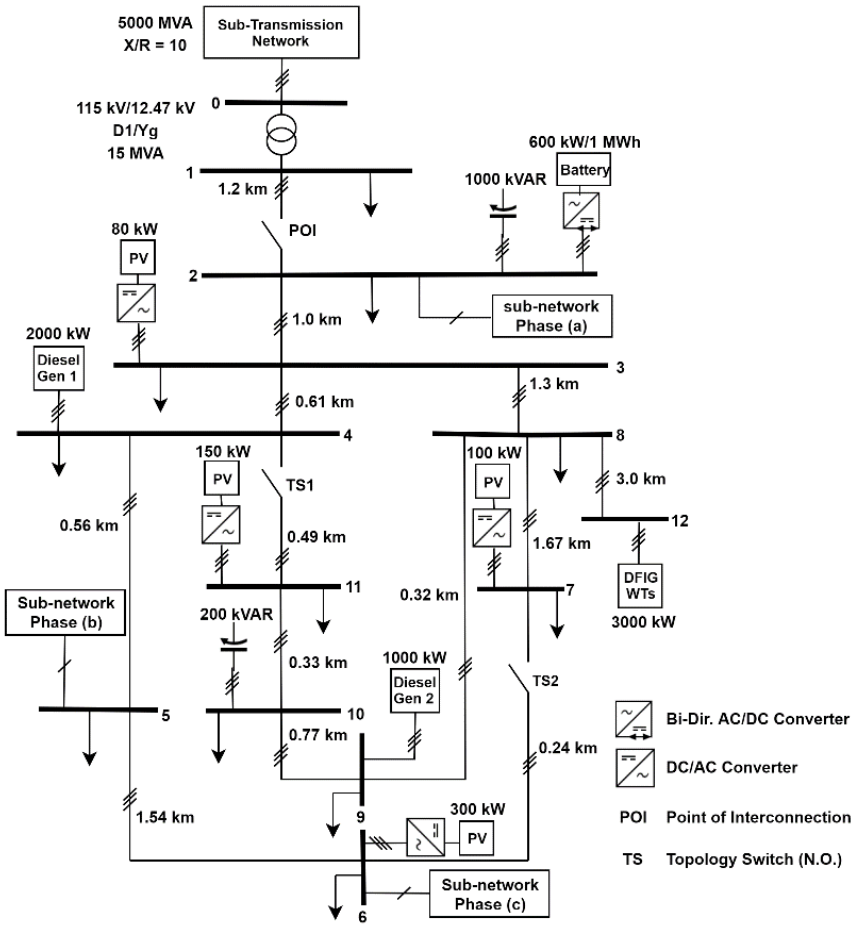

Figure 2.5 Modified benchmark model

The DERs, excluding the ESS, are designed to have a combined rating of 6630 kVA, which includes 3630 kVA of renewable capacity. DER assets include controllable and uncontrollable types. By controllable, it refers to the ability to set active and reactive power setpoints and the mode of operation. The controllable DER assets are the ESS and two Synchronous Machine assets numbered 1, 3 and 6, respectively. The ESS is rated at 1 MWh with 600 kW converter power, and it is needed in this setting to store excess renewable power and prevent energy losses. Furthermore, the synchronous machines exist to provide system inertia and can assist the ESS in meeting the microgrid load in islanded mode of operation. However, the 2000 kVA machine can be used in grid-connected mode to provide auxiliary service to the utility grid during peak hours, while the second 1000 kVA machine is a backup generator that is only used in islanded mode. On the other hand, all renewable assets are assumed to be uncontrollable assets and they always work in MPPT mode. Asset models and their primary controls are explained in Appendix C. The cost parameters

of the two diesel generators are based on industrial charts [66], while the ESS cost parameter is chosen to be cheaper than diesel as in [33, 39]. DERs cost parameters are listed in Table 2.3.

Table 2.2 3-Phase assets ratings

No.	Bus	DER Type	Max Rating (kW)
1	2	Energy Storage	600 – 1 MWh
2	3	PV Farm	80
3	4	Diesel Generator	2000
4	6	PV Farm	300
5	7	PV Farm	100
6	9	Diesel Generator (backup)	1000
7	11	PV Farm	150
8	12	Wind Farm	3000

Table 2.3 DERs ratings and cost parameters

DER Type	Rating (KVA)	Cost Parameter a $/h.kW^2	Cost Parameter b $/h.kW	Cost Parameter c $/h
Diesel generator 4	2000	0.00013782	0.02276286	62.67314286
Diesel generator 9	1000	0.00006971	0.02110657	124.3188571
ESS	600	0.0001	0	0

Moreover, the microgrid consists of ten MV busses, each has a 3-phase aggregated load. The microgrid full load is rated at 4940 kVA. Three types of loads are identified in this work: critical loads, non-critical loads and hybrid loads. A critical load shall not be shed or curtailed at any moment in the microgrid runtime. For example, critical loads consist of hospitals, governmental offices and other sensitive loads. A non-critical load is a load that is treated as a lumped sum sheddable load. Examples of non-critical loads consist of residential feeders. A hybrid load consists of a controllable load component as well as a critical load component, whereas the controllable component can be curtailed. Examples of hybrid loads are charging stations, factories,

and campuses that include heating and cooling elements. For this work, the two largest loads are identified as non-critical. The two smallest loads are identified as critical, and the rest are hybrid loads. The microgrid load ratings and types are listed in Table 2.4.

Table 2.4 Microgrid loads types and ratings

Bus #	Nominal Load (KVA)	Load Type
2	1150	Non-critical
3	490	Hybrid
4	400	Hybrid
5	500	Hybrid
6	150	Critical
7	400	Hybrid
8	450	Hybrid
9	350	Critical
10	500	Hybrid
11	550	Non-critical

2.7.2 Benchmark Centralized Controller

A centralized control system (CCS) is designed to act as a base case for comparison against the DCS. The CCS relies on the equal incremental cost criteria (EICC) algorithm in [67] for dispatching, whereas the EICC algorithm uses the mathematical formulation presented in 2.2.2 to solve for active power dispatch. For further comparison with the DCS, the use of a non-optimization algorithm, such as priority-based dispatch, is also considered as it has been considered in the literature for centralized microgrid control [29]. In this thesis, the priority-based dispatch technique is based on prioritizing the dispatch of the cheapest assets first. An asset is deemed cheaper than another asset if its cost curve lies on average below another asset's cost curve. Once the cheapest asset is fully utilized, the next cheapest asset is used.

2.8 Real-time and C-HIL Testing

The Controller-Hardware in the Loop (C-HIL) setup is used to test the proposed control system. The C-HIL consists of a real-time simulator, a network switch, and three hardware controllers. An overview of the C-HIL setup is shown in Figure 2.6. Each hardware controller is

equipped with BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz, 1 GB LPDDR2 SDRAM, and Gigabit ethernet for local area network (LAN) communications.

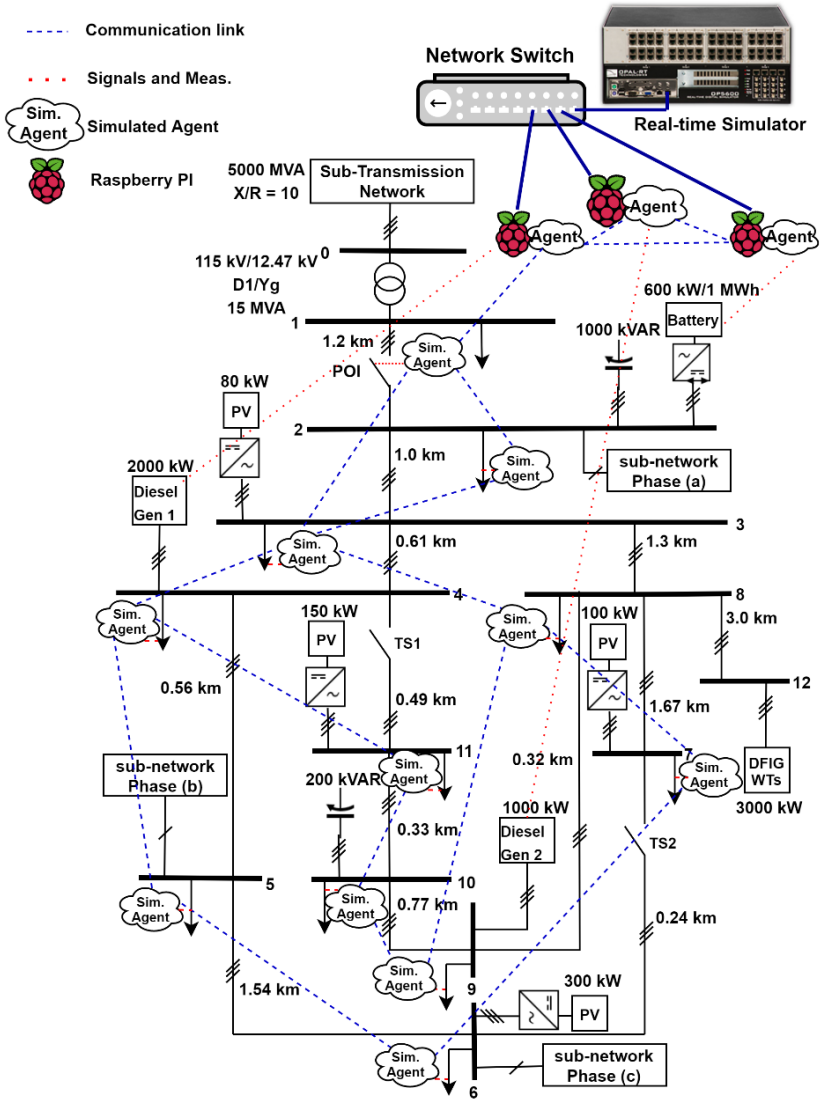

Figure 2.6 Schematic of the C-HIL setup

In total, 14 agents are participating in the tests. Three DER agents are running on three separate hardware controllers. On the other hand, the rest of the agents, which include ten Load agents and a POI agent, are simulated with the model on the real-time simulator, each running every 20 milliseconds.

The communication protocol used between hardware components is IEC 61850 GOOSE messaging [68]. An isolated network was created for the communications setup using a dedicated network switch. In addition, it can be noticed in Figure 2.6 how communications links, represented

by blue dashed lines, are drawn between agents. Microgrid communications assume a LAN. The network connects assets and makes it possible to send data between any two agents. The setup of the communication links is custom. In general, wherever there is a physical link, a communication link is also put. Also, extra redundant communication links are inserted, such as the ones between busses 2-3, 4-11, and 6-7. The extra communication link between busses 2-3 enhances the connectivity of the POI agent with the network, as it would be necessary considering the importance of the POI agent. On the other hand, the extra links between busses 4-11 and 6-7 makes the LAN meshed rather than radial, which would enhance the connectivity of the Load agents and consequently the convergence speed of the distributed algorithms.

2.9 Summary

In this chapter, the relevant control system math and theory were formulated in detail as well as the test setup. The mathematical formulations for economic dispatch and consensus theory were detailed. The microgrid control system's basic requirements for design and testing were listed. Finally, the tools to be used during the control system testing were presented. The next chapter will be about the design of the distributed control system algorithms, which use the formulations, that will be utilized to meet the microgrid control requirements.

Chapter 3

Design of the Distributed Control System

3.1 Introduction

This chapter details the algorithms used by the DCS to deliver on microgrid functions, namely the consensus algorithm and the proposed distributed priority queuing algorithm (DPQ). The algorithms are based on the mathematical formulation and theory presented in chapter 2. Depending on the application, the algorithms are driven by the specified microgrid objectives. A Monte Carlo simulation is carried out to study the convergence of the proposed DPQ algorithm considering different communication topologies.

3.2 Objectives of Distributed Control for Microgrids

The main purpose of implementing a distributed control system (DCS) is the apportioning of the microgrid control system (MGCS) among individual assets. In doing so, the risk of failure of the MGCS can be reduced. Furthermore, this division of control allows the design of modular control units, which when combined can make up the entire MGCS. This modular approach to design allows the MGCS to be plug and play, which makes the DCS reactive to changes in the microgrid topology.

In addition, it is of importance for microgrid economic operations to maximize renewable energy consumption while minimizing operational costs. To achieve this, the power output of renewable energy resources such as solar and wind is always at its maximum. Taking this into account, the DCS needs to properly coordinate controllable microgrid assets to ensure sound microgrid operation while fulfilling microgrid control core functions.

3.3 Applications of Distributed Control Theory for Microgrids

The DCS is applied to controllable assets such as synchronous machines, ESS, controllable loads and the POI circuit breaker. To fulfill the IEEE standard 2030.7 core functions, distributed algorithms based on graph theory are used to control the microgrid in grid-connected mode, islanded mode and during the transition.

Two distributed algorithms with unique features are utilized, namely the consensus algorithm and the distributed priority queuing (DPQ) algorithm. The consensus algorithm is used

to coordinate agents by weighted averaging. On the other hand, the DPQ algorithm works by selecting an agent given certain rules using a distributed data structure.

If the execution order of commands does not matter, the consensus algorithm can be utilized. However, some functions require sequential execution, such as load shedding. Since all agents run in parallel, the DPQ algorithm can be used to allow sequential execution by applying the action on one priority node. Next, the technical details of the algorithms and their applications are explained.

3.4 Consensus Algorithm Applications

Considering the consensus algorithm fundamentals presented in section 2.3.4, the algorithm is employed for various dispatch applications such as DERs output power control and load curtailment. Specifically, the innovation term of the consensus algorithm is updated to fit each application objective as will be explained next.

3.4.1 Power Tracking in Grid-connected Mode

The objective function used to meet a specified active power setpoint at the point of interconnection (POI) is formulated as follows:

$$\min \frac{1}{2}\left(\sum_{i=1}^{N_i} P_{Gi} - P_{POI}\right)^2 \tag{19}$$

Where P_{POI} is the measured active power at the POI and P_{Gi} is the generated power from asset *i*. The corresponding Lagrangian function is:

$$\mathrm{L} = \frac{1}{2}\left(\sum_{i=1}^{N_i} P_{Gi} - P_{POI}\right)^2 \tag{20}$$

Using (9) in section 2.2.2, L can be expressed as:

$$\mathrm{L} = \frac{1}{2}\left(\sum_{i=1}^{N_i} \frac{r_i(t)-b_i}{a_i} - P_{POI}\right)^2 \tag{21}$$

Let d_i be the sub-gradient of the Lagrangian in (21) to the incremental cost. The sub-gradient d_i is calculated as:

$$d_i(t) = \frac{\partial L_i}{\partial r_i} = \alpha\left(\sum_{i=1}^{N_i} \frac{r_i(t)-b_i}{a_i} - P_{poi}(\mathrm{t})\right) = \alpha\left(\sum_{i=1}^{N_i} P_{Gi}\,(\mathrm{t}) - P_{POI}(\mathrm{t})\right) \tag{22}$$

Where α is a constant. Using the calculated d_i in the consensus equation (12), the following is obtained:

$$r_i[t+1] = \sum_{j=1}^{N_i} L_{ij} r_j[t] + \varepsilon_i.\,\alpha\left(\sum_{i=1}^{N_i} P_{Gi}\,(\mathrm{t}) - P_{POI}(\mathrm{t})\right) \tag{23}$$

Since the objective is to have the total generation $\sum_{i=1}^{n} P_{Gi}$ (t) meet a specific power export setpoint, equation (23) can be re-written as:

$$r_i[t+1] = \sum_{j=1}^{N_i} L_{ij} r_j[t] + \varepsilon_i (P_{POI}^{*} - P_{POI}(\mathrm{t})) \tag{24}$$

Where P_{POI}^{*} represents setpoint to be reached as an objective. Once r_i *[t+1]* is calculated, it can be used to directly calculate the power setpoint using (10).

In the case of planned islanding, the objective is to reach zero net active power. Furthermore, the P_{POI} sign is negative as it is seen as a sink from the DER perspective. Therefore, equation (24) updates to:

$$r_i[t+1] = \sum_{j=1}^{N_i} L_{ij} r_j[t] + \varepsilon_i P_{POI}(\mathrm{t}) \tag{25}$$

Where (25) uses the objective of canceling the active power at the POI as the driving factor of the consensus algorithm.

During planned islanding, reactive power needs to be dispatched to cancel any reactive power exchange at the POI before islanding. Typically, the required reactive power dispatching from the DERs is minimal. The reactive power compensation comes mainly from compensator devices such as shunt capacitor banks. However, in most cases, compensator devices cannot be tuned to provide a specific reactive power setpoint as needed. In this case, the available reactive power from DERs can be dispatched to cancel any remaining reactive power exchange at the POI.

For reactive power, the objective of canceling the reactive power is formulated as follows:

$$\min \frac{1}{2} \left(\sum_{i=1}^{N_i} Q_{Gi} - Q_{POI}\right)^2 \tag{26}$$

Where Q_{Gi} is the reactive power generation by asset *i*. Q_{POI} is the measured reactive power at the POI. Finally, m is the number of assets with reactive power capability. In the case of planned islanding, the objective is to reach zero net reactive power. The following equation can be derived in a very similar manner to (25):

$$u_i[t+1] = \sum_{j=1}^{N_i} L_{ij} u_j[t] + \mu_i Q_{POI}(\mathrm{t}) \tag{27}$$

Where u_i is the consensus signal pertaining to reactive power-sharing at asset *i*. μ_i is the learning rate used by agent *i* on reactive power consensus. Next, u_i is used to calculate the reactive power generated locally according to the following equation:

$$Q_{Gi} = \begin{cases} \sqrt{{P_{Gmax}}^2 - {P_{Gi}}^2} & if\ u_i[t+1] > \sqrt{{P_{Gmax}}^2 - {P_{Gi}}^2} \\ \frac{u_i[t+1]}{N_i} & Otherwise \end{cases} \tag{28}$$

Where N_i is agent i's number of neighbors. Two observations can be noted from the above calculation of Q_{Gi}. First, active power dispatch takes precedence over reactive power dispatch. Second, the number of neighbors is taken into consideration when sharing reactive power dispatch between participating DERs.

3.4.2 Distributed Load Curtailment

For the application of distributed load curtailment, the objective is to match the total curtailments of loads with a requested power curtailment setpoint. Therefore, this function follows a similar derivation listed in 3.4.1 for power tracking. However, the second form of the consensus algorithm in (13) is utilized. Each load curtailment is treated as a power source as follows:

$$\lambda_i[\mathrm{t}+1] = \lambda_i[\mathrm{t}] + \sum_{j\in NEI_i} L_{ij}\big(\lambda_j[\mathrm{t}] - \lambda_i[\mathrm{t}+1]\big) + \varepsilon_i(-P_{load\_i}) \tag{29}$$

Whereas λ_i is the consensus variable calculated by agent i, P_{load_i} is agent i's load curtailment contribution, and ε_i is the learning rate used by agent i. On the other hand, the power at the POI is treated as a power sink as follows:

$$\lambda_i[\mathrm{t}+1] = \lambda_i[\mathrm{t}] + \sum_{j\in NEI_i} L_{ij}\big(\lambda_j[\mathrm{t}] - \lambda_i[\mathrm{t}+1]\big) + \varepsilon_i(P_{POI}) \tag{30}$$

Where P_{POI} is the measured power at the POI when planned islanding is considered. If $P_{POI} < 0$, then the consensus variable would be set to zero since loads cannot go negative. Using the above formulation, the total loads' curtailments $\sum_i P_{load\_i}$ matches the total requested power setpoint at the POI, P_{POI}.

3.5 Distributed Priority Queuing Algorithm

3.5.1 Motivation

The main limitation of the consensus algorithm is that it cannot be used for sequential execution. Therefore, the algorithm is ineffective at applications that require prioritization. For example, in the application of load shedding, the consensus algorithm cannot be used to determine the order of load shedding since it is based on the concept of averaging. It is an industrial practice to shed the largest non-critical loads first, and then the second-largest, and so on. To solve this problem, an algorithm is designed that implements a distributed data structure and to find the highest priority element given certain rules.

3.5.2 Definition and Application

A priority queue is a data structure where elements are processed according to a certain priority associated with them. The proposed algorithm is a naive implementation of the priority queue, where only one element with the highest priority is found. Once this latter element is dispatched for some purpose, the algorithm finds the next highest priority element. The challenge in this work is that finding the highest priority element needs to be done in a distributed fashion.

This algorithm is inspired by a protocol implemented in the literature where the authors implemented a network discovery algorithm in order to find the total number of nodes in the network [32]. However, it is important also to note that the scope of the application, implementation, and structure of the protocol in [32] are different from those of the algorithm proposed here. Hence, their protocol and the proposed algorithm herein cannot be compared. The protocol in [32] works by allowing each agent to build an accumulating database of other agents available on the network. During each iteration, each agent shares the database they have obtained up until the current time step, updates their database, and then shares it again with other agents. Each entry of the database is composed of a tuple $(ID, Timestamp, Value)$, where each agent was assigned a specific ID, a time stamp, and a value which represented estimated supply-demand imbalance. The problem with the implementation in [32] is its scalability. As the number of assets that need control increases, each asset controller will need to store a larger database. Building up databases would require substantial memory and communication bandwidth requirements. To overcome this issue, the proposed algorithm does not require an agent to build up a database. On the contrary, agents need only to find a single entry representing the agent with the highest priority. Therefore, only a single entry is stored and shared by each agent at any time. The proposed algorithm is called a distributed priority queuing algorithm (DPQ). The objective of this algorithm is to specify the highest priority entry and to duplicate it in all the agents' local memories.

In this thesis, the DPQ algorithm is employed for the application of load shedding. Therefore, priority refers to load shedding priority. Under this context, a high priority load indicates the least critical load with the largest consumption. Specifically, the following tuple is defined:

$$ND = \left(ID_i, \text{Load\_Type}_i, Loading_i, \text{Time\_stamp}_i\right) \tag{31}$$

Where the tuple is denoted as ND. The first entry indicates the agent ID. When a new load agent is registered in the microgrid network, it is assigned a specific ID that is unique to the agent. The agent registration process is not covered in this thesis and is assumed to be done by a separate

higher-level MAS entity. The second entry indicates the load type. Load types can be critical, hybrid and non-critical. Loading is sheddable consumption. Finally, the time_stamp indicates the time when the tuple was updated. The algorithm is listed in Table 3.1, specified only for the application of identifying the largest and least critical load in the microgrid. It can be argued that the algorithm can be reused for a different application than load shedding of similar nature of complexity.

Table 3.1 Distributed priority queuing algorithm

	Algorithm: Distributed Priority Queuing (Execution for each agent v_i)
	Initialize: ID_{local}, $Load\_Type_{local}$, $Time\_stamp_{local} = 0$, $ND_i = (ID_{local}, Load\_Type_{local}, Loading_{local}, Time\_stamp_{local})$, where ND_i is a tuple stored in local memory.
	Input: $Loading_{local}$, ND_j from each neighbor v_j where ND is a tuple containing (ID_j, $Load\_Type_j$, $Loading_j$, $Time\_stamp_j$)
	Output: $ND_i = (ID_i, Load\_Type_i, Loading_i, Time\_stamp_i)$ sent to each neighbor v_j
	Algorithm
	%% In the first part of the algorithm, each ND_j received by a neighbor is processed. Furthermore, the locally
	%% stored ND_i is compared against each received ND_j according to specified rules.
1:	**For** every ND_j received from a neighbor v_j :
2:	**If** $Load\_Type_j \neq$ critical
3:	**If** $Time\_stamp_j > Time\_stamp_i$
4:	$ND_i = ND_j$
5:	**Else if** $Time\_stamp_j = Time\_stamp_i$
6:	**if** $Load\_Type_i$ = non-critical **and** $Loading_i = 0$ **and** $Load\_Type_j$ = hybrid **and** $Loading_j > 0$
7:	$ND_i = ND_j$
8:	**Else if** $Load\_Type_i$ = hybrid **and** $Load\_Type_j$ = non-critical **and** $Loading_j > 0$
9:	$ND_i = ND_j$
10:	**Else if** $Loading_i < Loading_j$
11:	$ND_i = ND_j$
	%% Following the processing of each ND_j in the first part of the algorithm, ND_i is compared against agent %%v_i's characteristics (ID_{local}, $Load\_Type_{local}$, $Loading_{local}$, $Time\_stamp_{local}$) according to specified rules.
14:	**If** $ID_i = ID_{local}$ **and** $Loading_i \neq Loading_{local}$
15:	$ND_i = (ID_{local}, Load\_Type_{local}, Loading_{local}, Time\_stamp_{local})$
16:	**Else if** $Loading_{local} \neq 0$ **and** $Load\_Type_{local} \neq$ critical
17:	**If** $Load\_Type_i$ = non-critical **and** $Loading_i = 0$ **and** $Load\_Type_{local}$ = hybrid

18:	$ND_i = (ID_{local}, Load\_Type_{local}, Loading_{local}, Time\_stamp_{local})$
19:	**Else if** $Load\_Type_i$ = hybrid **and** $Load\_Type_{local}$ = non-critical
20:	$ND_i = (ID_{local}, Load\_Type_{local}, Loading_{local}, Time\_stamp_{local})$
21:	**Else if** $Loading_{local} > Loading_i$
22:	$ND_i = (ID_{local}, Load\_Type_{local}, Loading_{local}, Time\_stamp_{local})$
23:	Increment $Time\_stamp_{local}$

To explain the algorithm, the details are listed from the first-person perspective of agent *i*. The algorithm is initialized with an ND_i, which is the tuple entry stored in the local memory of agent *i*. ND_i is initialized with the agent's local parameters, which are ID_{local}, $Load\_Type_{local}$, $loading_{local}$ and timestamp of zero. At each algorithm iteration, it receives as input the $loading_{local}$ which is a measurement of load i's consumption, and ND_j entries, which are received from each neighbor agent *j*. As an output, the agent sends out the local ND_i entry post-processing to neighboring agents.

As for the main body of the algorithm, it is divided into two parts. In the first part, the algorithm processes the received ND_j entries and compares them to the local stored ND_i entry, potentially replacing the local ND_i entry by the received ND_j entry. The first part of the algorithm can be read from line 1 to 11. The rules for replacing ND_i with ND_j are as follows:

- Load *j* is not a critical load.
- ND_j has a newer timestamp than ND_i, which indicates that it is more up-to-date.
- ND_j is a non-critical load and ND_i is a hybrid load (higher priority) unless load *j* is zero. The inverse is true.
- The loading of agent *j* is larger than the loading of agent *i*.

In the second part, the algorithm compares the ND_i entry against agent *i*'s local parameters. Specifically, ND_i entries are replaced by local entries if any of the following rules are fulfilled:

- ND_i indicates the same *ID* as agent *i*'s local *ID*.
- Agent *i* is not a critical load and has a *loading* larger than 0.
- ND_i indicates a non-critical load with a *loading* of zero, while agent *i* is a hybrid load.
- ND_i indicates a hybrid load while agent *i* is a non-critical load.
- The *loading* of agent *i* is larger than the *loading* entry in ND_i.

The expectation is that upon the convergence of the algorithm, all the agents ND_i entries are the same, and are equal to the largest and least critical load.

3.5.3 Performance Analysis

In this section, the convergence analysis of the DPQ algorithm is presented, as well as how the convergence speed is affected by the agents' connectivity. This study is made to highlight and evaluate the speed of convergence of the DPQ algorithm, which would potentially impact the speed of accurate response in emergency scenarios, such as fast load shedding during unplanned islanding. A Monte Carlo simulation was conducted, whereas the number of nodes was increased from 3 to 100, and for each number of nodes, 500 simulations were made. In each simulation, the load sizes and types were randomized. These details are summarized in Table 3.2.

Table 3.2 Monte Carlo simulation setup

Parameter	Configuration
Load size	Random [0,1], in p.u.
Load Type	Random [critical, non-critical, hybrid]
Number of Nodes (N)	3 - 100
# of simulations per N nodes	500

Furthermore, four types of data sharing topologies were experimented with, shown in Figure 3.1. A one-hop connection means that each agent is connected to one neighboring agent. A two-hop connection means that each agent is connected to two neighboring agents. A randomly connected network is a network where an agent can have a random number of links to any number of agents. When the Monte Carlo features a randomly connected network, this randomization is repeated for each of the 500 simulations. Finally, in a fully connected network, each agent is connected to every other agent.

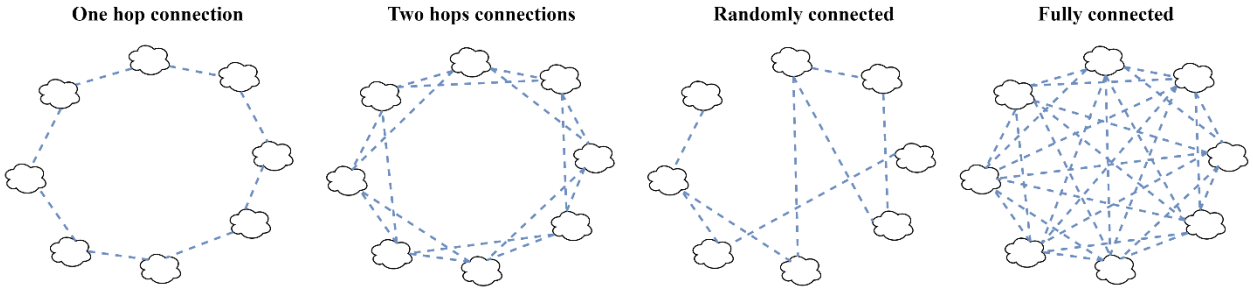

Figure 3.1 Communication topologies

Figure 3.2 shows the average number of iterations needed for the convergence of the DPQ algorithm under the four communication topologies. Additionally, around each DPQ curve, the upper and lower 98% confidence intervals are plotted. For further comparison, centralized Selection sort is plotted considering a fixed number of iterations equaling to n for every n number of nodes, as it would always require scanning through all the data to find the highest priority element. As can be noticed, the DPQ convergence is fastest with randomly and fully connected networks and becomes slower with two hops connection and one hop connection.

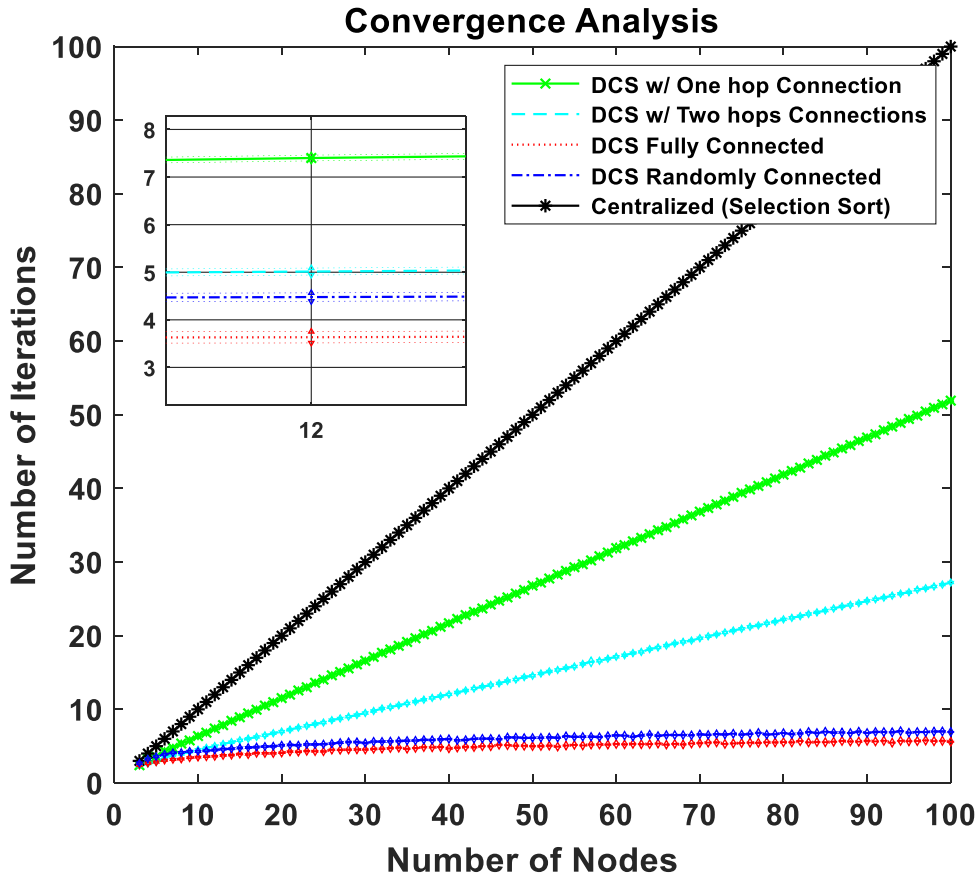

Figure 3.2 Convergence analysis of DPQ using different communication topologies

It can be further observed that using randomly and fully connected network topologies, the algorithm does well with scalability as the number of iterations very slowly rise with higher number of nodes. In addition, considering the subplot on Figure 3.2 which zooms in at 12 nodes, the confidence interval limits are in the close neighborhood of the mean values. The confidence interval margins are smallest with 1 hop topology, and slightly increase with 2 hops, random and fully-connected, respectively. This implies that the algorithm is deterministic in terms of the

required number of iterations to reach convergence given a certain of nodes. This deterministic feature can help the program issuing the load shedding to estimate with confidence the required time for convergence. Furthermore, in theory, the DPQ algorithm has a faster convergence rate than a centralized selection sort algorithm. This makes sense because multiple entities are simultaneously involved in the computation. However, the DPQ algorithm, requires the use of communications with each passing iteration, which slows down the overall process. Nonetheless, considering the limited number of iterations required to reach a solution, particularly by a fully connected or random network, the total communication time required is limited.

To study the impact of communication delays on the DPQ algorithm response time, Figure 3.3 is plotted considering 5, 10, 15 and 20 milliseconds communication delays.

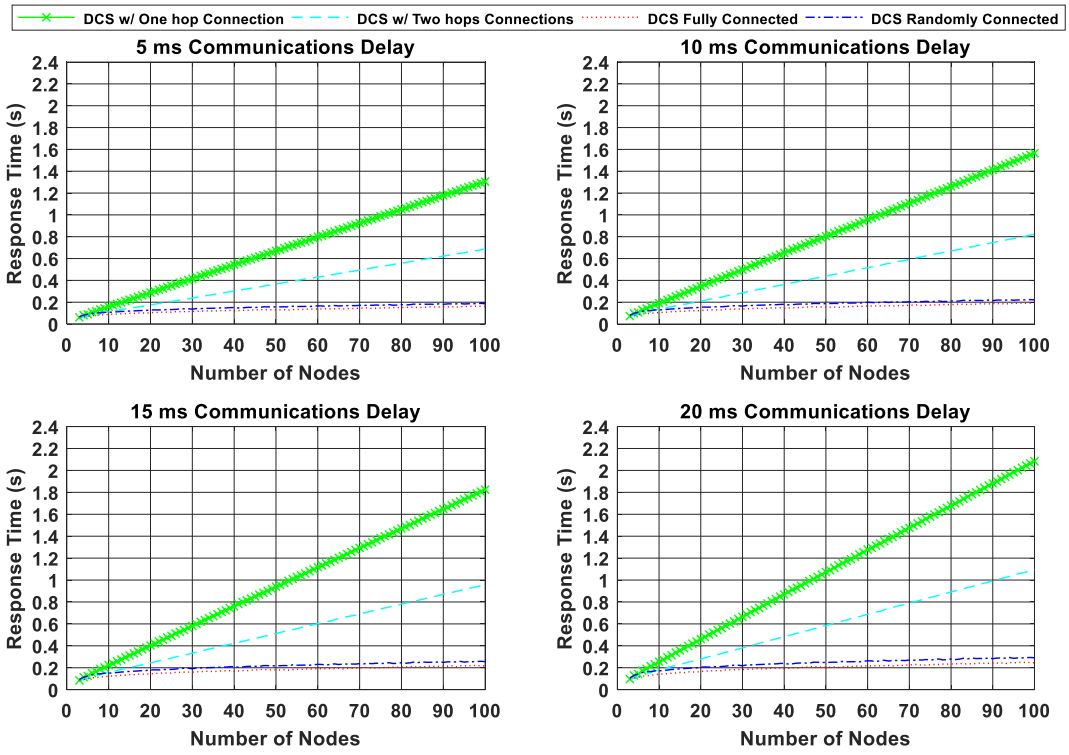

Figure 3.3 Response time of the DPQ algorithm considering communication delays

The response time of the DPQ algorithm is the total communication time plus the hardware computation time plus the program iteration time. The hardware computation time is the average recorded convergence time for the 500 Monte Carlo simulations per node. The program iteration time is set to 20 milliseconds, as per the C-HIL in section 2.8, multiplied by the average number of iterations. As can be observed, communication delays clearly impact the response time of the

DPQ algorithm. At the same time, the impact is dependent on the communication topology. The response time is fastest when a fully-connected network is considered and increases with randomly connected, 2 hops and 1 hop topologies, respectively. However, it must be considered that the fully connected network is expected to be the most expensive one to implement as it requires n*n-1 links, where n is the number of nodes. On the other hand, the two hops topology require n*4 links and the one hop topology requires n*2 links. Finally, it is worth mentioning that the results clearly show how advancements in communication technologies that would allow systems to have fully connected networks at large scale can significantly improve the control system performance.

The algorithm works for shedding one load at a time. If another load is needed to be shed, then the designated agent program must wait for the algorithm to converge again before shedding another load. This is further demonstrated with the successful use of the algorithm in the simulation of unplanned islanding in section 5.3.4, where two consecutive load sheds were issued.

3.6 Summary

This chapter detailed the distributed algorithms used to deliver on the microgrid applications. On the one hand, the use of the consensus algorithm was proposed to coordinate DER assets for power tracking and load assets for distributed curtailment. On the other hand, the DPQ algorithm was proposed for selecting an agent in a distributed fashion for the purpose of load shedding. The next chapter presents the agents' implementation and logic that will be coded with these algorithms.

Chapter 4

Distributed Control System Implementation

4.1 Introduction

This chapter presents the implementation multi-agent system framework that makes up the DCS. Three types of agents are proposed confirming to three general asset categories: POI agent, DER agent and load agent. Each DCS agent receives relevant asset measurements, utilizes the relevant distributed algorithms and finally uses the outputs of the algorithms to formulate the dispatch commands. Furthermore, each agent is encoded with a logic that bridges the algorithms to the practical context of microgrid control.

4.2 Hardware and Software Support for Distributed Control

The main hardware requirement for DCS implementation is the allocation of a DCS controller to every designated asset in the microgrid. The minimum controller hardware specifications are the same as what is specified for the controller used in the C-HIL setup in section 2.8. It needs to be equipped with an operating system such as Linux, memory storage, and to have a communication interface, such as an Ethernet.

The required hardware capability to store and execute the required DCS software is the same for all DCS controllers. A typical DCS controller is connected to the asset controller to send commands to it and receive measurements from it. Here, the importance of interoperability between the DCS controller and each asset type is emphasized. To help achieve this, several software frameworks are designed to adhere to the type and function of the asset being controlled. Finally, each DCS controller should be able to communicate with neighboring DCS controllers to execute distributed algorithms.

4.3 Distributed Control System Framework

The DCS framework is based on the designed multi-agent system (MAS) which specifies agents' types and functions. An agent of the MAS represents the control program running in the DCS controller. Every agent has a built-in MGCS transition logic that switches its operation strategy depending on the microgrid mode of operation. However, the agent logic needs only to

apply to the control requirements imposed by the asset it's assigned to, which reduces its program complexity.

Several agent types with different responsibilities are identified. Figure 4.1 shows the topology of the designed MAS. As can be observed, three main categories of agents are identified: Regulating Agents, Load Agents, and DER Agents. Agents belonging to the same category actively coordinate with each other to deliver relevant microgrid applications. First, regulating agents control microgrid assets that are critical for maintaining sound microgrid operations such as the microgrid circuit breaker (MCB). Also, its primary agent, namely the POI Agent, has the extended role of communicating on behalf of the microgrid with the distribution system operator (DSO). Furthermore, it is responsible for controlling the MCB and has other important functions that are vital for power management at the POI. On the other hand, DER Agents are assigned to controllable DERs such as synchronous machines and energy storage systems (ESS). DER Agents run dispatch functions depending on the microgrid mode of operation. Finally, Load Agents control the three types of loads specified in this thesis, non-critical, hybrid and critical.

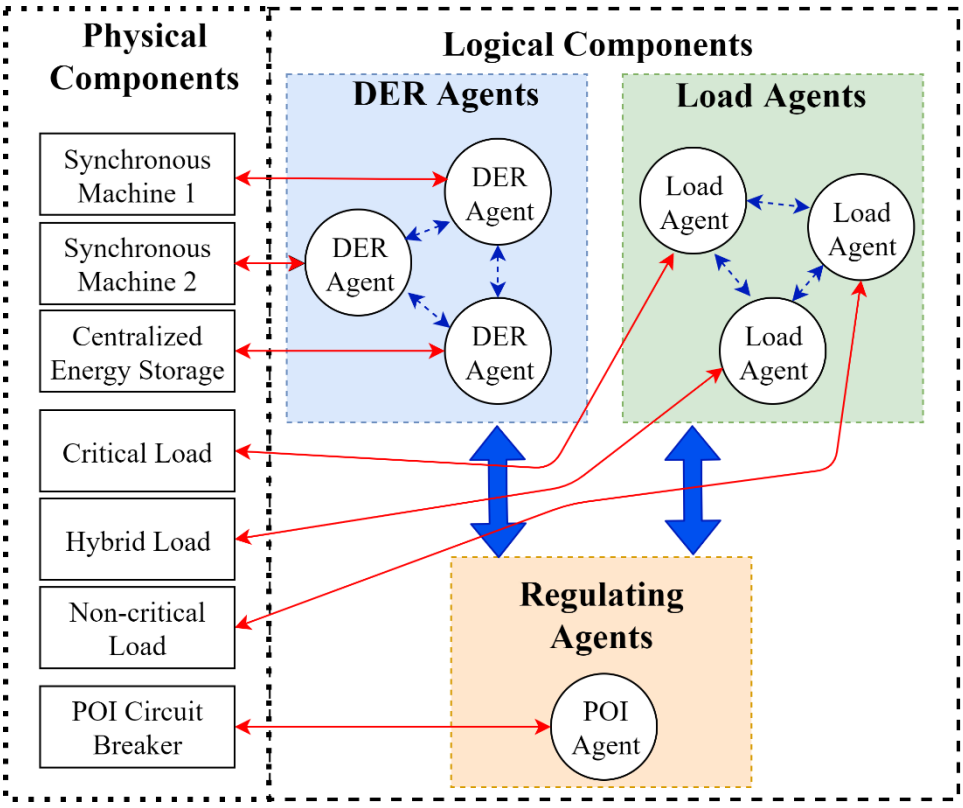

Figure 4.1 Multi-agent system framework and topology

4.4 Distributed Control System Agents

In this section, the designs of the POI agent, DER agent and the Load agent are presented. Their inputs, outputs and program functions are described. Furthermore, it is explained which agents execute which distributed algorithms, and how the execution is carried out.

4.4.1 POI Agent

The POI agent is the only agent with more than one unique function, hence being in a category of its own. First, the POI agent controls the microgrid circuit breaker (MCB) at the POI. Normally, circuit breakers are primarily controlled by protection relays. However, in a microgrid, the MCB needs to be further controlled by the MGCS during microgrid transition from grid-connected to islanded modes and vice versa. Second, the POI agent is responsible for power management at the POI. Third, the POI agent is the microgrid's point of contact with the distribution system operator (DSO). In a centralized control system, the DSO sends commands to the MGCC, which would then process the commands and controls the assets accordingly. On the contrary, the DCS does not have a MGCC, but rather a conglomerate of distributed controllers working together. Therefore, it is necessary to assign an agent in the MAS to receive commands from the DSO. Fourth, the POI agent sends shed and curtail commands to load agents, but it does not run the DPQ algorithm, nor does it initiate the DPQ algorithm at the load agents. As will be specified in more detail later, the POI agent only sends a logic true or false, expressing the need to lose some load. It does not specify which loads to shed or how much to shed. The latter two questions are answered by coordination between the load agents.

Figure 4.2 shows the overall program logic in a single loop, divided into three sections: 1. Signals and communication processing, 2. Grid-connected operation, and 3. Islanded operation. First, the POI agent processes communications and measurements. It receives from the DSO the following commands:

- Islanding command, which requests to island the microgrid.
- Active power setpoint P_{DSO}, which dictates how much active power should be regulated at the POI.

On the other hand, it receives from measurement units their recordings from both sides of the POI. Specifically, it expects active power, reactive power, frequency, voltage, and phase. Finally, it receives from neighboring load agents the communications necessary for running the consensus algorithm. Each neighbor load agent sends to the POI agent its number of neighbors and its locally calculated consensus.

All in all, considering 20 milliseconds per program iteration C-HIL setup in section 2.8, the POI agent is expected to exchange 67.2 Kbps of data. This calculation breaks down to 3.6 Kbps with the DSO, 32 Kbps of measurements from the MCB, and 31.6 Kbps of data in communications with other agents.

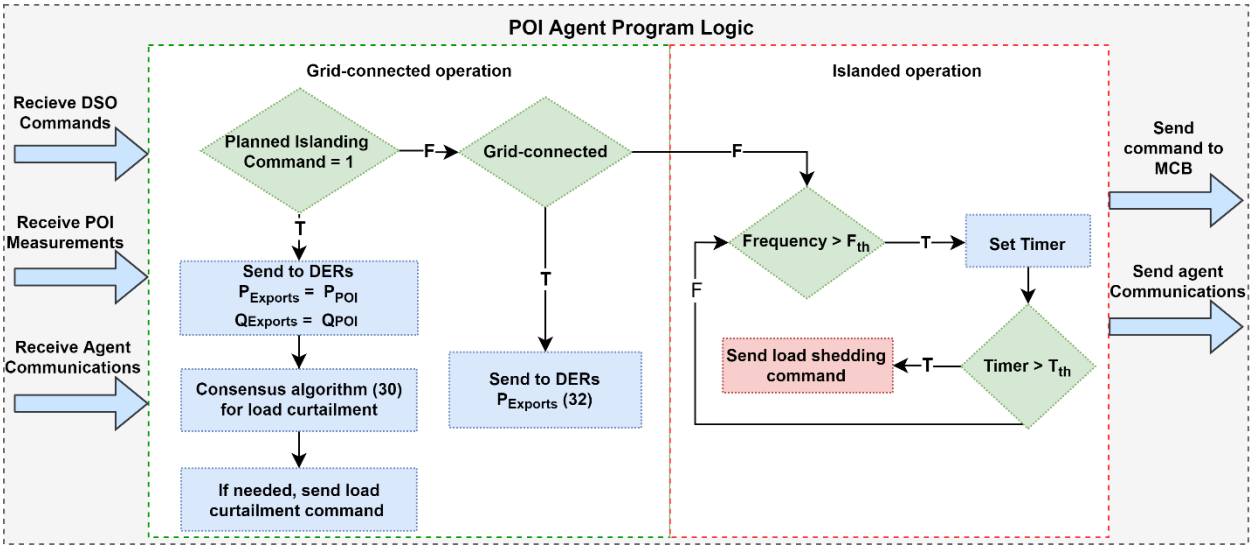

Figure 4.2 POI agent program flowchart

Next, the POI agent checks if an islanding command was received. If it is the case, then it 1. Sends active and reactive export setpoints to DERs, which are equal to the measured active and reactive power at the POI, 2. Uses (30) to calculate the consensus for load curtailment, and 3. Sends load curtailment command to load agents if necessary. The POI agent knows that load curtailment is necessary when the power at the POI stops approaching zero, which would indicate that the DERs have reached their capacity. Once the active and reactive powers at the POI near zero, the POI agent opens the MCB.

If it is not planned islanding, it is a grid-connected steady-state operation. In this case, the POI agent receives from the DSO a power export setpoint to be met at the POI. Accordingly, the program broadcasts to the DER agents the power that needs to be exported by the microgrid, which is

$$P_{Exports} = P_{POI} - P_{DSO} \quad (32)$$

Where $P_{Exports}$ is continuously transmitted to the DER Agents, P_{DSO} is the setpoint received by the DSO that is to be met at the POI, and P_{POI} is the measured power at the POI. Note that P_{DSO} is positive when importing power to the microgrid, as this is the reference adopted here.

Furthermore, in islanded mode, the POI agent is responsible for implementing wide-area under-frequency load shedding (UFLS). For example, in the case of unplanned islanding, microgrid generation deficiency can occur. In this case, the microgrid can remain stable only if fast load shedding is implemented. Since the electrical grid frequency is a reliable indicator of generation deficiency [69], it is used as a metric to shed load. Furthermore, a PLL is installed at the microgrid POI [70], which provides the frequency measurement to the POI agent. Note that the UFLS implementation is functionally similar to the 81U frequency relay. However, the execution is different. The idea of implementing fast UFLS in this context is to help in recovering the microgrid frequency upon large loss of supply before the frequency drops to critical tripping thresholds. The POI agent is responsible for properly timing load shedding commands. In the event of a frequency deviation $deltaF > F_{th}$, where F_{th} is termed Frequency Threshold, the POI agent starts a timer. If the frequency deviation maintains for a time threshold T_{th}, the POI agent sends a load shedding command to load agents. The shedding command is merely a Boolean value, and it does not specify which loads should shed and by how much. The execution of the shedding is decided by the load agents. Regarding the value of T_{th}, trial and error experiments have shown that it should be of low value of at least 20 milliseconds for the very first load shed. The maximum tolerable delay to the first load shed is difficult to predict as it depends on several variable factors including the inertia available in the microgrid at the time of the disturbance and the magnitude of the disturbance that led to the frequency drop. However, for subsequent load shedding, T_{th} is set to a higher value in the order of seconds. This is done in order to give time for the microgrid synchronous machines to respond to the frequency drop by means of their primary control. Furthermore, setting subsequent shedding timer thresholds to a low T_{th} can lead excessive load shedding which could potentially cause microgrid over-frequency.

4.4.2 DER Agent

The DER Agent is responsible for controlling a DER asset, where the DER could be a synchronous machine or a battery ESS. On the other hand, Renewable DERs are not assigned

agents and are assumed to be always working on the Maximum Power Point Tracking mode. A DER agent calculates the active and reactive power setpoints during steady-state and transition operation and sends them to the asset local controller. Other control inputs such as mode of operation, nominal frequency, and nominal voltage are set by constants. On the other hand, other asset-specific parameters such as cost function coefficients and generation limits are initialized by the program. The set of constant parameters and initializations are listed in Table 4.1.

Table 4.1 Initialization and constant parameters of the DER Agent Program

Parameter	**Description**
Cost function coefficients	Necessary for calculating the P setpoints for incremental cost criterion, specifically coefficients a and b.
P_{Gmax}	The upper percentage limit of the rated capacity. Set to 95% during steady-state and to 75% during transition.
P_{Gmin}	The lower percentage limit of the rated capacity. It is set to 30% for synchronous machines and -100% for ESS.
Active power control mode	Frequency-active power droop control.
Reactive power control mode	Voltage-reactive power droop control.
Nominal voltage	Asset nominal voltage rating.
Nominal frequency	60 Hz

Figure 4.3 shows the main program flow of the DER Agent. First, the agent processes received communications, summarized as follows:

- Asset local measurements: frequency and voltage.
- It receives from the POI agent:
  - Microgrid connection state (grid-connected/islanded).
  - Measured active and reactive power at the POI, for planned islanding.
  - Active power export setpoint (32), for grid-connected operation.
  - Islanding command.
- It receives from neighboring DER agents:
  - The calculated consensus that is relevant to the current mode of operation.
  - Connection state to the microgrid (enabled/disabled), used to build (14).
  - The number of DER agents connected to them used to build (14).

All in all, considering 20 milliseconds per program iteration and the C-HIL setup in section 2.8, the DER agent is expected to exchange 27.2 Kbps of data. This calculation breaks down to 10.4 Kbps with the POI agent, 6.4 Kbps of measurements from the DER, and 10.4 Kbps of data in communications with other DER agents.

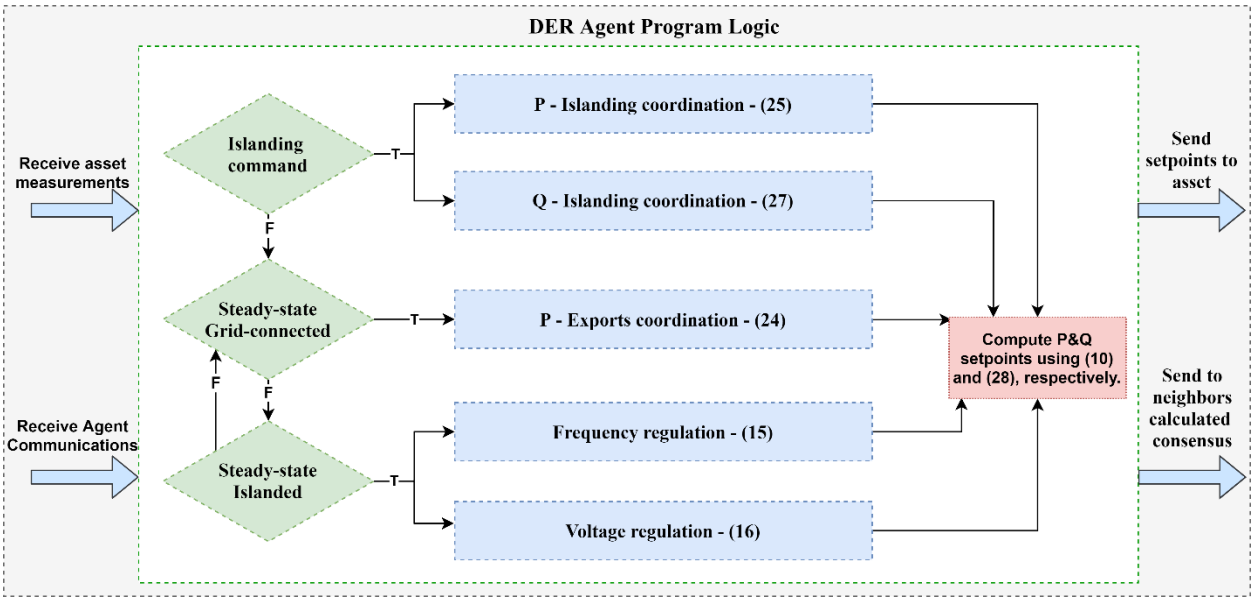

Figure 4.3 DER agent program flowchart

Next, the DER agent program processes the dispatch and transition logic. In every mode of operation, the consensus algorithm is utilized to coordinate with other neighbor DER agents. Therefore, depending on the mode of operation, the consensus is calculated. In grid-connected mode, the agent uses (24) to calculate the consensus for active power dispatch to meet an export power setpoint at the POI. If a planned islanding command is received, the agents coordinate active power dispatch using (25) and reactive power dispatch using (27). In islanded mode, the consensus for frequency regulation is calculated using (15), and a separate consensus for voltage regulation is calculated using (16). Afterward, the calculated consensus is used to determine the active power setpoint using (10), and the reactive power setpoint using (28). Finally, the agent dispatches the computed setpoints to the asset and sends its updated consensus to neighbor DER agents.

4.4.3 Load Agent

The load agent is responsible for controlling a unit of electrical load. A load could be non-critical load, hybrid load or a critical load, similar to the types listed in Table 2.4. A load agent has control options such as curtailing, shedding or restoration. The availability of these options

depends on the load type. Load curtailment is the partial disconnection of a load unit. Load shedding is the complete disconnection of a load unit. Load restoration is the complete reconnection of a load unit. In this thesis, the control options considered are curtailing and shedding, while the restoration is only listed for completeness. To remark, only the hybrid load type could be curtailed. On the other hand, non-critical and hybrid loads can be completely shed. Finally, critical loads cannot be curtailed or shed. A load agent is initialized with parameters about the load it controls as follows:

- ID; A unique number given to a load agent and assigned by a supervising entity upon agent registration.
- Load type; the nature and function of the load would dictate its type, which could be any of the three options as iterated above.
- Cost coefficient, γ; Entails the price to curtail.
- Minimum load.
- Maximum load.

Figure 4.4 details the program flow of the load agent. First, the program processes communications, summarized as follows:

- Measurement of the load consumption;
- It receives from neighbor load agents:
    - ND, (31), which is used by the DPQ algorithm. Each load agent has a calculated and locally stored ND.
    - Calculated consensus to be used by the consensus algorithm in (29).
    - The number of load agents connected to each of them used to build (14).
- It receives from the POI agent:
    - (If directly connected). Calculated consensus to be used by the consensus algorithm in (29).
    - (If directly connected). The number of load agents connected to it used to build (14).
    - Command to shed load.
    - Command to curtail load.

All in all, considering 20 milliseconds per program iteration and the C-HIL setup in section 2.8, the DER agent is expected to exchange 32.8 Kbps of data. This calculation breaks down to 0.8 Kbps with the POI agent(considering that the POI is not a direct neighbor that participates in the consensus algorithm), 3.2 Kbps of measurements from the load, and 28.8 Kbps of data in communications with other load agents.

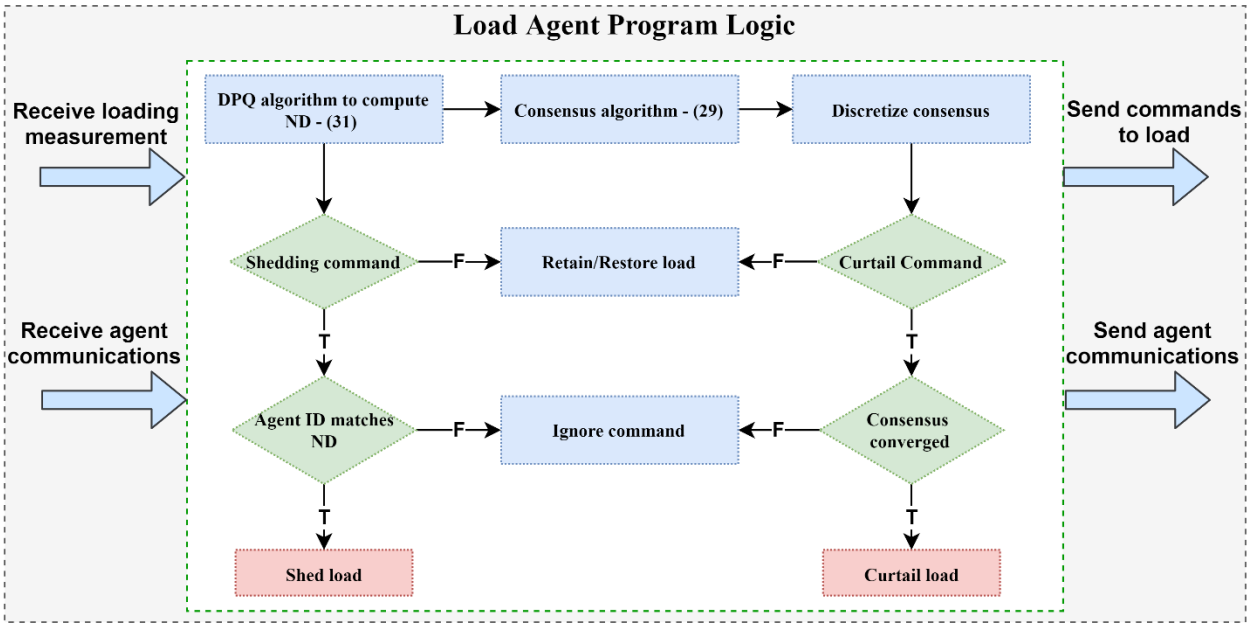

Figure 4.4 Load agent program flowchart

Second, the load agent runs the DPQ algorithm listed in Table 3.1 to compute and store ND presented by the tuple in (31). The DPQ algorithm runs repeatedly at every loop of the agent program. Therefore, ND is dynamically updated during run-time during at every program loop, making the result available when load shedding is required. In the C-HIL implementation presented in section 2.8, the program loops every 20 ms. During an under-frequency event, the POI agent sends a shedding command to all load agents. Consequently, the load agent with the ID that matches the ID entry in ND sheds. Once a load sheds, the agent controlling that load updates its data and the DPQ algorithm updates ND starting from the last saved ND. Accordingly, the ND result would be available again upon the convergence of DPQ for any subsequent load shedding.

Third, the load agent runs the consensus algorithm in (29) to determine how much to curtail. As a remark, P_{load_i} in (29) is load agent *i*'s curtailment amount and it is the term that needs to be determined. Initially, (29) calculates the consensus variable λ_i. To translate the consensus value of λ_i to a value of practical sense, it is discretized. In this implementation, the discretization is done

in steps of 10% to reflect the discontinuity nature of physical load consumptions. However, this can be generalized to fit the curtailment steps accepted by a certain load. To discretize, the percentage curtailment of the load is calculated as:

$$\text{Per} = \frac{\gamma_i \lambda_i}{P_{nom\_i}} \tag{33}$$

Where the consensus variable is λ, γ is the price coefficient of load i and P_{nom_i} is the nominal load value. It is expected that Per should be always less than 1, as the calculated load curtailment must be less than P_{nom}. Next, the normalized value calculated by (33) is rounded down to the nearest tenth, resulting in an updated $Per_{rounded} \leq$ Per. Finally, P_{load_i} is calculated as:

$$P_{load\_i} = Per_{rounded} * P_{nom\_i} \tag{34}$$

Next, if the load agent receives the command from the POI agent to curtail, it checks if the consensus converged. Convergence is checked if the difference between the consensus λ^t and λ^{t-1} is less than 1%. If both conditions are true, load i curtails P_{load_i}.

4.5 Summary

This chapter detailed the DCS implementation using the proposed multi-agent system framework. The implementation involved the detailed operation of each agent, the coordination between agents of the same type, and the cooperation between agents of different types. In general terms, The POI agent controls the MCB and is involved in transition coordination. The DER agents control dispatchable assets during steady-state and transition. The Load agents control the curtailment or shedding of their loads. The next chapter will test the proposed DCS implementation in a set of practical microgrid case studies.

Chapter 5

Testing of the Distributed Control System

5.1 Introduction

This chapter is dedicated to the testing and performance evaluation of the proposed DCS. A case study is constructed for several scenarios and relevant events are initiated during each test to assess the control response. Furthermore, the tuning of the consensus algorithm learning rate is performed, and its corresponding impact is studied. Finally, the DCS is compared against the centralized controller in terms of cost optimization during dispatch and the overall control system reliability.

5.2 Test set-up and metrics

To validate the operation of the DCS, the tests are conducted on the C-HIL testbed presented in section 2.8. The tests are constructed as case studies and are designed to showcase the operation of the proposed algorithms. The case studies involve:

1. Active power tracking at the point of interconnection in steady-state grid-connected mode subject to changes in renewable generation output and the distribution system operator (DSO) dispatch order.
2. Frequency regulation, voltage regulation and economic dispatch in steady-state islanded mode subject to large step load changes.
3. Planned islanding upon receiving an islanding request from the DSO. Load curtailment is utilized to match the supply and demand before islanding.
4. Unplanned islanding upon sudden microgrid disconnection. The resulting large mismatch between supply and demand is mitigated using fast load shedding.

In each case study, the performance of the DCS is analyzed as applicable using the following metrics:

- Response time (s). Applicable to step changes. It is defined as the time required for a parameter to change from the initial value to 90% of the final steady-state value.

- Settling time (s). Applicable to disturbances and step changes. It is defined as the time required for a parameter to reach a certain dead-band of the final steady-state value from the starting time of the event. The IEEE standard 2030.8 does not specify settling time dead-bands for power or frequency, as they depend on the customer contract [11]. For consistency, the active power dead-band is chosen to be 5%. However for the frequency, it is chosen as 2% as that is the first frequency threshold for the DER tripping within clearing time as summarized in Table 5.1.
- Overshoot (%). It is defined as a percentage for disturbances and step changes as the deviation minus the nominal value divided by the nominal value.
- Integral absolute error (IAE), is the total error in absolute value that is measured during tacking. It is used during POI power tracking in grid-connected mode. Physically, the IAE represents total energy in KWh.
- Steady-state error (%). It is the absolute percentage difference between the nominal value and the steady-state value of a parameter.

The voltage and frequency responses are checked against metrics defined in the IEEE standard 1547. The metrics are chosen assuming the DERs are within the abnormal performance Category III, summarized in Table 5.1. Measured frequency and voltage values that are between the over and under thresholds fall within the continuous operation region where ride-through is permitted regardless of any disturbances. In case a measurement crosses a specified under or over threshold, then the control system is permitted a certain clearing time to get that measurement back to its continuous region of operation before the DER trips. According to the IEEE standard 1547, the metrics apply to both islanded and grid-connected modes and it is the assumed case in this thesis. However, the conditions for islanded operation may be further relaxed in practice depending on contractual requirements [11].

Judging from Table 5.1, an appropriate settling time bandgap for the frequency disturbance response is 2%, which translates to range from 58.8 Hz to 61.2 Hz. On the other hand, an appropriate settling time bandgap for the voltage disturbance response is 10%, which translates to range from 0.9 p.u. to 1.1 p.u. Therefore, measuring the settling time for voltage and frequency is only meaningful when they cross their bandgaps.

Table 5.1 IEEE Standard 1547 DER voltage and frequency tripping criteria

Shall trip function	**Voltage (p.u. of 12.47 kV)**	**Clearing time (s)**
OV1	1.1	13.0
UV1	0.88	21.0
Shall trip function	**Frequency (Hz)**	**Clearing time (s)**
OF1	61.2	300.0
UF1	58.5	300.0

5.3 Case Studies, Results, and Discussions

5.3.1 Case Study – Grid-connected Mode

This case study showcases POI power tracking in grid-connected mode. To recall, the objective is to meet the DSO dispatch order P_{DSO} while the renewables maintain MPPT operation. The case study involves the POI agent and the DER agents. The POI agent receives the dispatch order from the DSO, calculates the real-time power setpoint $P_{Exports}$ and sends it to the DER agents, as indicated in Figure 4.2. In turn, the DER agents receive $P_{Exports}$ and use the consensus algorithm to coordinate the power export at the POI as illustrated in Figure 4.3.

In terms of initial conditions, the simulation started from steady-state grid-connected mode where the active power at the POI was regulated to P_{DSO} = 750 kW. Figure 5.1 shows the recorded active power probes at the DERs and the POI during the real-time simulation. The 2000 kVA diesel generator at bus 4 was running temporarily by agreement with the DSO during peak-hour operation, while the 1000 kVA diesel generator at bus 9 was shut down, as it is inherently a backup generator. The wind farm was producing 812.9 kW while the cumulative production of solar farms is 457.7 kW. The loads were at 70% of their max loading.

A major disturbance occurred at t = 100 seconds where the wind speed increased from 8 m/s to 11 m/s. From t = 100 s to t = 140 s, it can be noticed how the DCS was coordinating the DERs to maintain regulation of the P_{POI} after the disturbance, with the worst performance recorded to be at the time of the disturbance, and at t = 117.5 s where the consensus was still converging. The main factor that contributes to the speed of consensus convergence is the consensus learning rate set, which is set by trial and error at a relatively low value of $\varepsilon = 0.00001$ in order to have a non-oscillating control response from the DERs. On the other hand, due to the small number of

DER agents, communications do not present an issue for the consensus convergence as the DER agents communicate on average at a data rate of 27.2 Kbps, which is a small value compared to the available throughput of an ethernet network that is in the order of Mbps.

The integral absolute error between 100 s and 140 s was recorded to be 0.776 KWh. Consequently, due to excess renewable power and the need to regulate P_{POI}, the diesel generator P setpoint reached its minimum, while the ESS P setpoint neared zero at 16.9 kW. Therefore, to maintain efficient operation of the microgrid DERs, P_{DSO} was updated to P_{DSO} = -750 kW at t = 140 s to be exported by the microgrid. By t = 149 s, the active power at the POI was regulated to -750 kW.

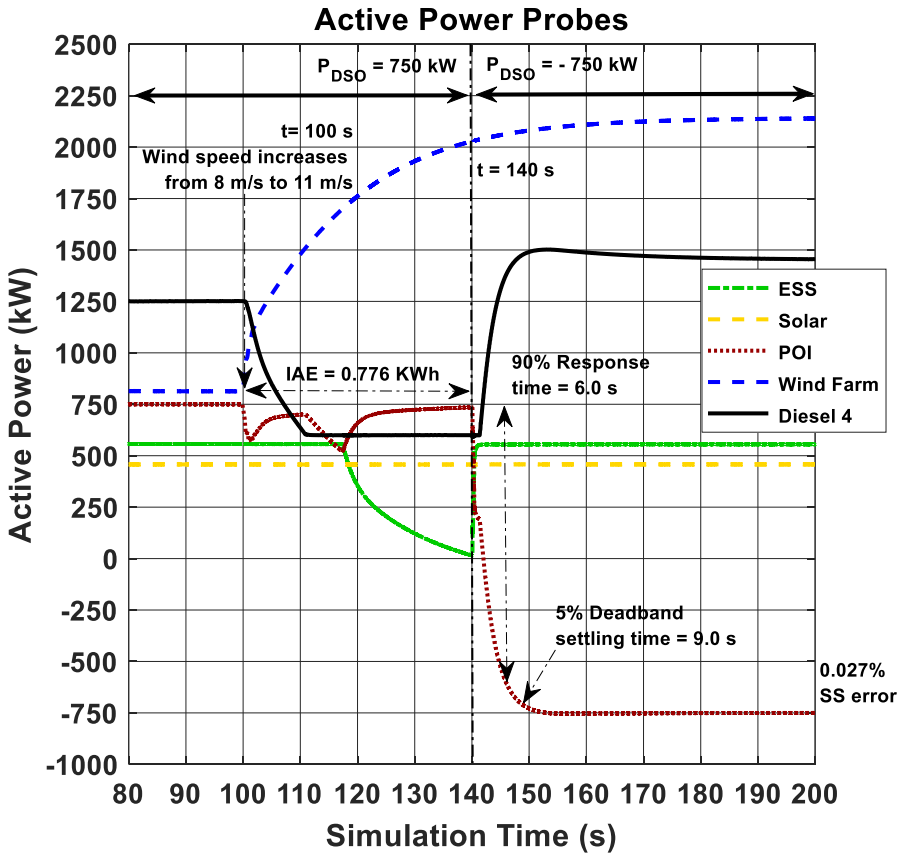

Figure 5.1 POI power tracking in grid-connected mode of operation

Next, the P_{POI} response from t = 140 seconds until steady-state is analyzed. The 90% response time was measured to be 6.0 seconds from P_{POI} = 735.2 kW at t= 140 s down to P_{POI} = -601.5 kW at t = 146 s. The settling time was recorded to be 9.0 seconds for P_{POI} to be within 5% of P_{DSO}. The final SS value was calculated to be within 0.027% of P_{DSO}, which translates to 270

watts, a negligible error. On the other hand, the 0% overshoot originates from the consensus algorithm chosen learning rate ε. However, it is possible to obtain an underdamped response with a higher value of the learning rate to increase the response time at the cost of incurring oscillations and an overshoot.

5.3.2 Case Study – Islanded Mode

This case study showcases the DCS response to a large step load change in islanded mode. The objective is to maintain frequency regulation, voltage regulation, and economic dispatch. The case study involves the DER agents only. The DER agents operate in steady-state islanded mode as illustrated in Figure 4.3.

In terms of initial conditions, the simulation started from steady-state islanded mode. Figure 5.2 shows the recorded active power measurements at all the DERs and loads, the POI voltage and the microgrid frequency. In islanded mode, both diesel generators were running as planned because islanding is an emergency scenario and the total supply must exceed the total demand. The wind farm was producing 813.2 kW while the cumulative production of solar farms was 474 kW. The loads were at 50% of their max loading. At t = 88 s, dispatch analysis was conducted, and results compared with the CCS. The results are listed in Table 5.2. As can be observed, the DCS dispatching of the machines and the ESS DER is nearly the same as that of the CCS dispatching.

Table 5.2 Islanded mode dispatch analysis at t = 88 s

Measurement	DCS P Results (kW)	CCS(EICC) P Results (kW)
Wind	815.2	815.2
Solar	467.9	467.9
ESS	560.9	562.6
D4 – 2000 kVA	600.2	600.2
D9 – 1000 kVA	505.7	504.3
Loads	-2211	-2211
Others[1]	-738.9	-738.9
Power Balance	0.0	0.0
Total Cost ($/h)	260.7949	260.7849

[1] Others include single phase components, DER filters and system losses.

A relatively large load up-step occurred at t = 100 seconds where the loads' consumption increased by 30% or 1327 kW. At the instant of the step change, the primary response from the DERs would kick in to halt the frequency drop, but without DCS frequency regulation, the frequency would go down to at least 58.8 Hz. During the transient from t = 100 s to t = 104.9 s, the DER agents ran frequency regulation-based dispatch to meet the new power balance. The active power measurement of P_{D4}, corresponding to the 2000 kVA diesel generator, was chosen to calculate the metrics as it had the slowest response. The response time from P_{D4} = 598.8 kW at t = 100 s to P_{D4} = 1370.3 kW was recorded to be 4.9 s. On the other hand, the 5% settling time was recorded to be 5.5 s at P_{D4} = 1387 kW. Furthermore, the voltage incurred a 4.98% overshoot but was regulated back to settle with 0.85% steady-state error from 1.0 per unit. At the same time, the frequency incurred a 0.81% overshoot but was regulated back to its nominal value. Both the voltage and frequency responses maintained within their continuous operating regions. At t = 140 s, another dispatch analysis is conducted and compared with the CCS. . As can be observed from Table 5.3, the DCS dispatching of the machines and the ESS DER is the same as that of the CCS. The reason is that since ESS and D9 are both at their maximum allowable operating points at t = 140 s, D4 is the only DER left with controllable power output margins. Hence it would be difficult for the CCS and the DCS not to arrive at the same dispatch for D4. Had more DERs required their dispatching to be solved for as in the previous case study, the DCS cost would have been higher, as it is unattainable for the DCS to arrive at same optimal results by the CCS within a short period. This fact is further illustrated in the consensus learning rate parametric analysis in section 5.4.3.

Table 5.3 Islanded mode dispatch analysis at t = 140 s

Measurement	DCS P Results (kW)	CCS(EICC) P Results (kW)
Wind	813.2	813.2
Solar	474.0	474.0
ESS	571.1	571.1
D4 – 2000 kVA	1456.0	1456.0
D9 – 1000 kVA	942.8	942.8
Loads	-3538.0	-3538.0
Others	-719.1	-719.1
Power Balance	0.0	0.0
Total Cost ($/h)	432.6071	432.6071

On another note, the results do not assume that the CCS and DCS share the same response time to arrive at their dispatch results. Furthermore, direct comparison results with the CCS curves are omitted as it is difficult to get a fair comparison with respect to the speed of response because there does not exist a standard centralized controller for which any other microgrid controller can be directly compared against. Consequently, comparison results with respect to the speed of response cannot be conclusive.

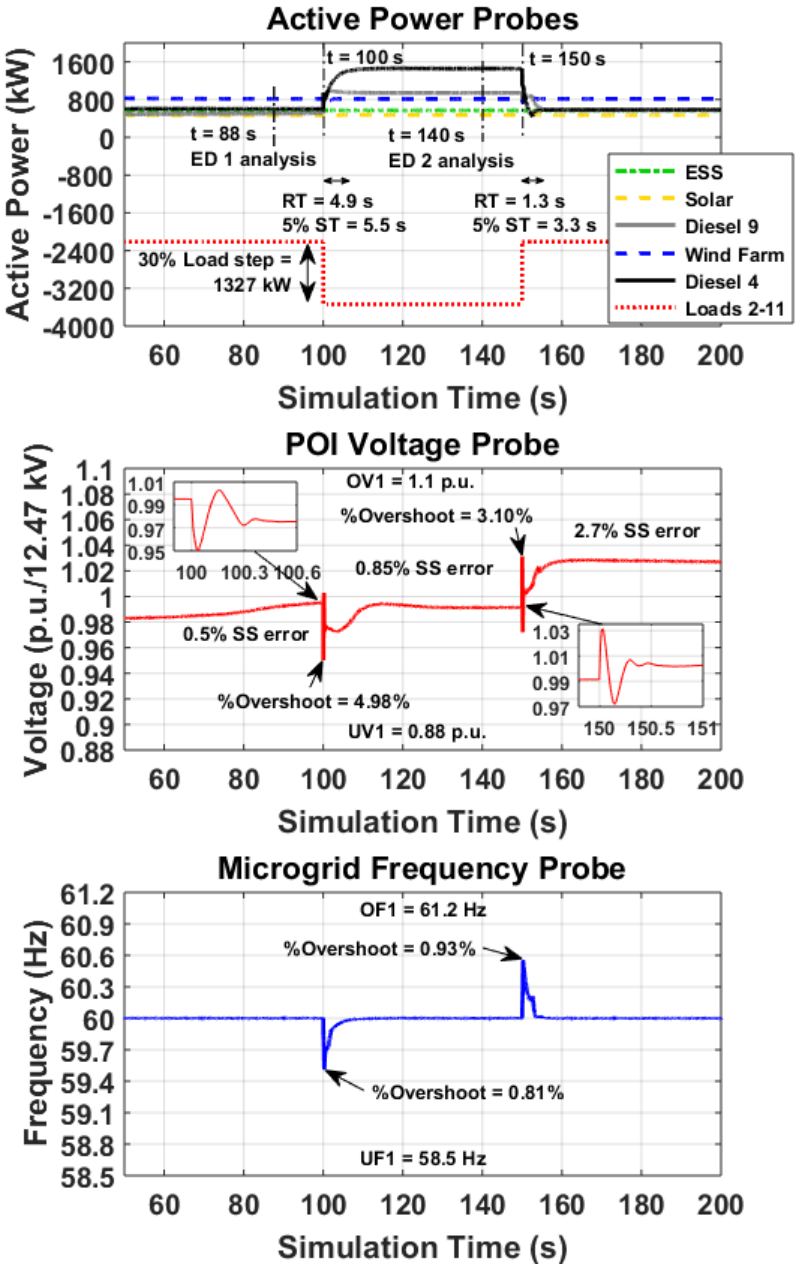

Figure 5.2 Step load change in islanded mode test results

At t = 150 s, a load down-step of the same magnitude occurred. The response time from P_{D4} = 1456 kW at t = 150 s to P_{D4} = 672.1 kW was recorded to be 1.3 s. On the other hand, the 5% settling time was recorded to be 3.3 s at P_{D4} = 555.75 kW. Furthermore, the voltage incurred a 3.10% overshoot but was regulated back to settle with a 2.7% steady-state error from 1.0 per unit. At the same time, the frequency incurred a 0.93% overshoot but was regulated back to its nominal value. Both the voltage and frequency responses maintained within their continuous operating regions.

In this case study, a learning rate of ε = 0.02 was used to obtain the frequency response. Note that this learning rate is larger than the one used for power regulation in section 5.3.1 in order to properly regulate the frequency to its nominal value. Under the same initial conditions as above, another study was conducted to monitor the effect of the learning rate ε_i on frequency response and to indicate how the 0.02 value was chosen. Figure 5.3 shows the results. A series of load changes, every 50 seconds, rise from 50% to 80% followed by a drop to 50%, were issued in islanded mode while increasing the learning rate by a magnitude of one for each other rise. Here, the general trend of the learning rate impact on the frequency response is emphasized. The results of this study do not apply to parameter responses other than the frequency, and the learning rates chosen are particular to the benchmark model used.

It is found that the use of a higher learning rate can generally improve the frequency response up to a certain limit. Using a very slow learning rate of value less than ε = 0.001 is not recommended as under-frequency or over-frequency thresholds could be breached. On the other hand, learning rates greater than ε = 0.01 can cause the algorithm to be highly sensitive to disturbances, which would result in highly varying power setpoints that could lead to system instability. Finally, ε = 0.02 was chosen as a good value since the best rates are observed to be between ε = 0.1 and ε = 0.01, and closer to ε = 0.01.

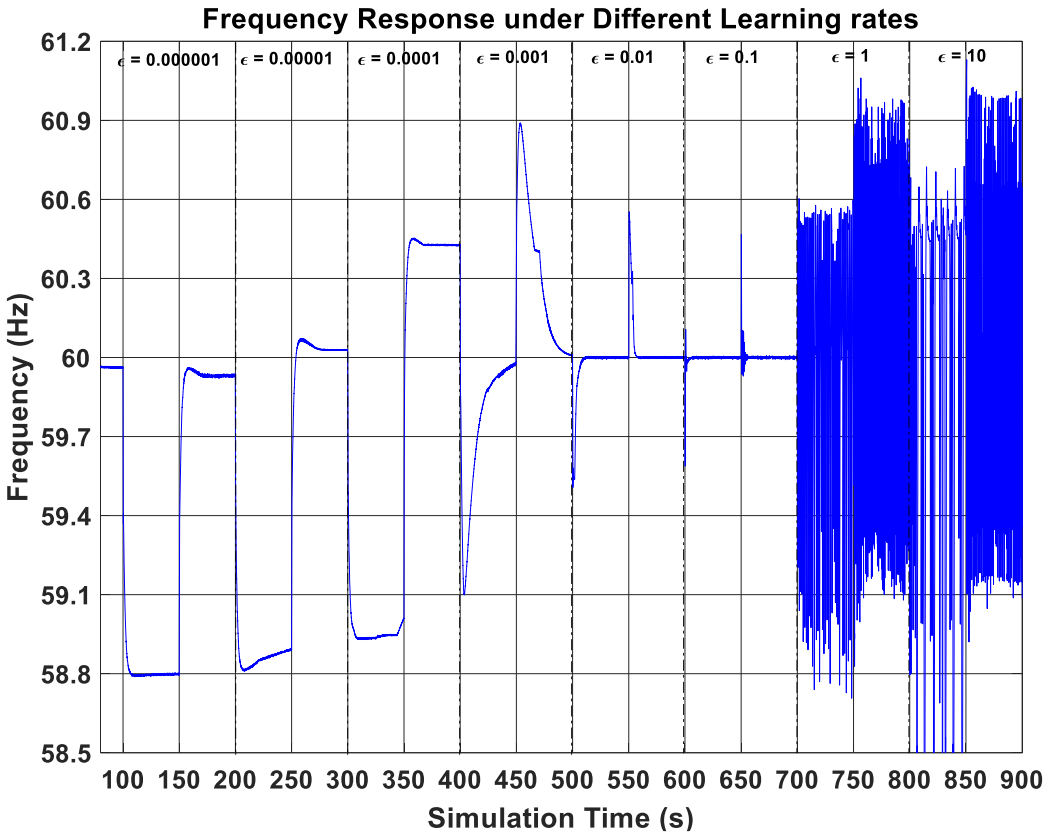

Figure 5.3 Frequency response under step load and variations of the learning rate

5.3.3 Case Study – Planned Islanding

This case study showcases planned islanding from grid-connected mode to islanded mode. To recall, the objective is to utilize the microgrid assets to zero P_{POI} and transition to steady-state islanded mode. The case study involves the POI agent, the DER agents, and the load agents. The POI agent receives the planned islanding order from the DSO, calculates the real-time power setpoint $P_{Exports}$, which is P_{POI}, and sends it to the DER agents, as indicated in Figure 4.2. In turn, the DER agents receive P_{POI} and Q_{POI} and use the consensus algorithm to coordinate zeroing the POI power as illustrated in Figure 4.3. In the meantime, the load agents use the consensus algorithm, with participation from the POI agent, to calculate each load's curtailment contribution, as shown in Figure 4.4. In this case study, equal price to curtail is assumed for all hybrid agents. If the available supply cannot zero P_{POI}, the POI agent sends a curtailment command to the load agents to curtail hybrid loads.

In terms of initial conditions, the simulation started from steady-state grid-connected mode. Figure 5.4 shows the recorded active and reactive power probes at the DERs, Loads and the POI,

POI voltage and the microgrid frequency. P_{POI} was regulated to P_{DSO} = 1500 kW. The 2000 kVA diesel generator at bus 4 was running for islanding. The wind farm was producing 816 kW. The case study happened in the evening time, where the loads were at 60% of their max loading and solar farms had zero power output.

At T1 = 62.1 s, the POI agent received an islanding command from the DSO. After this, the DER agents dispatched power until they reached their allowable margin at t = 74.66 s. At T2 = 75.77s, hybrid loads curtailed a total of 982.6 kW, offsetting P_{POI} to below zero. The 90% response time was 15.23 s, where P_{POI} dropped from 1504 kW to 150.4 kW. The 5% deadband settling time for P_{POI} to reach zero was 28.27s starting from the time of receiving the islanding command. The microgrid islanded at T3 = 112.60 s when P_{POI} and Q_{POI} reached the dictated tolerance of 0.1 kW. The total time to island is 50.5 s, which is acceptable for non time-critical process. However, choosing a higher tolerance would have significantly reduced the time to the island. Upon islanding, DER agents transitioned to steady-state islanded dispatch mode. The frequency incurred a very small overshoot of 0.033%, while the voltage went under negligible disturbance. The voltage settled with a 0.2% steady-state error from the 1 p.u. nominal value. Both the voltage and frequency responses maintained within their continuous operating regions.

Hybrid loads curtailment resulted in the decline of the consumption of each hybrid load by 40% of their nominal load, as detailed in Table 5.4. In this case study, it is expected that equal curtailment ratios would be achieved due to equal prices. This indicates that distributed curtailment is working properly. If any of the load agents had a higher set price, their curtailment would be less, as will be shown in section 5.4.2. The accuracy of matching the load curtailment amount by the residual P_{POI} at the time of curtailment is limited by multiple factors. First, the imposed 10% discretization of loads curtailment. Second, the time delay between the instant when the DERs outputs reached their capacity when zeroing the POI and the instant the POI agent issued the curtailment command. Currently set to zero, extending this time delay allows extra time for load agents to reach a more accurate consensus on load curtailment. Third, the slow learning rate used for distributed curtailment.

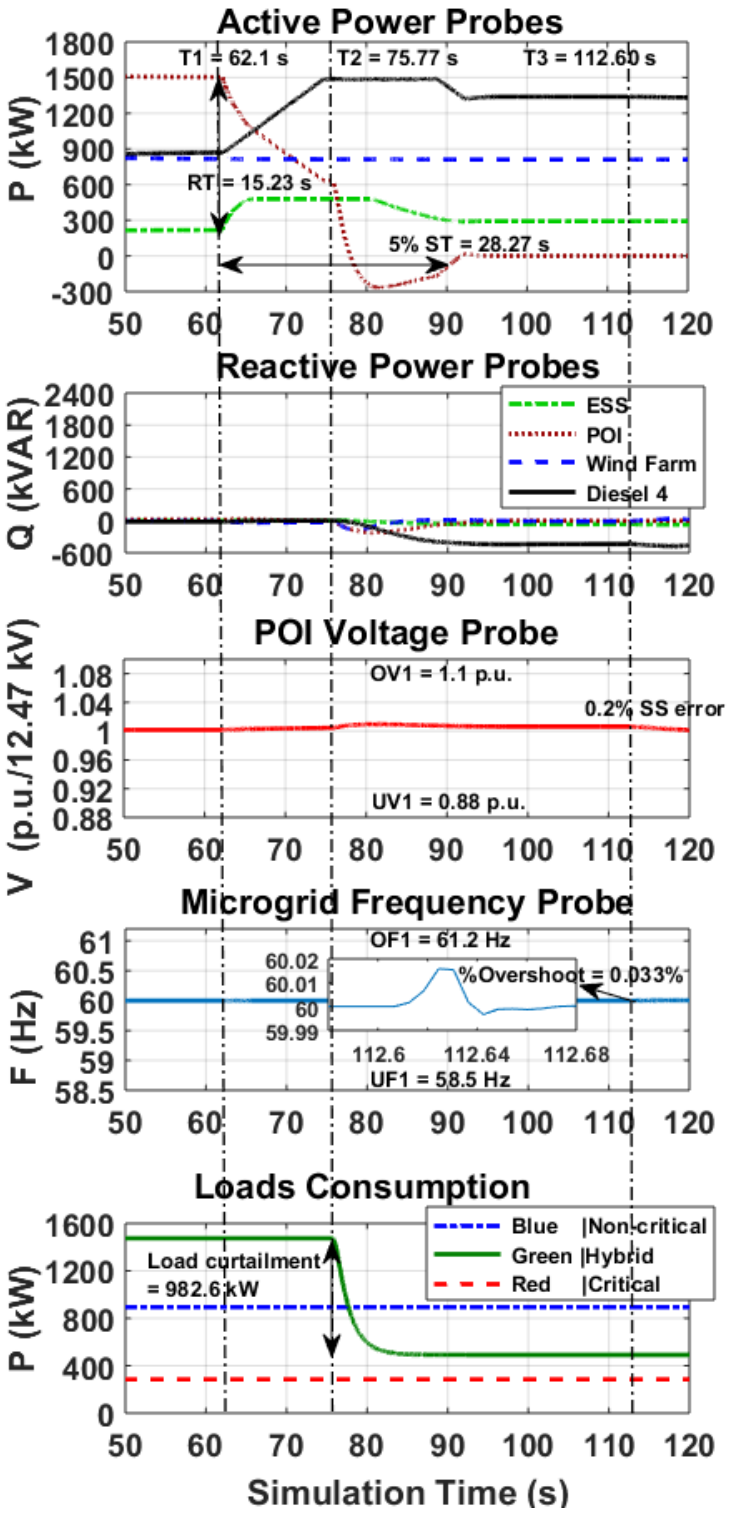

Figure 5.4 Planned islanding test results

Table 5.4 Loads curtailment during planned islanding

Load Bus #	Type	Before Curt. (kW)	After Curt. (kW)	Nominal P (kW)	Perc Curt. (%)
3	Hybrid	250.2	83.4	417	40
4	Hybrid	216	72	360	40
5	Hybrid	267	89	445.5	40
7	Hybrid	228	76	379.5	40
8	Hybrid	243	81	405	40
10	Hybrid	270	90	450	40

5.3.4 Case Study – Unplanned Islanding

This case study showcases unplanned islanding from grid-connected mode to islanded mode. To recall, the objective is to quickly recover the microgrid and to transition to steady-state islanded mode. The case study involves the POI agent, the DER agents, and the load agents. Upon abrupt microgrid disconnection, DER agents receive from the POI agent the new microgrid state, thereby changing their mode of operation to frequency and voltage regulation as illustrated in Figure 4.3. Often, the large supply-demand offset creates a steep frequency drop which the DERs cannot pick up, therefore the POI agent sends a load shedding command to load agents as shown in Figure 4.2. In turn, the load agents use the DPQ algorithm to shed the appropriate load as indicated in Figure 4.4.

In terms of initial conditions, this case re-iterates the same initial conditions as in the planned islanding case study. Figure 5.5 shows the recorded active and reactive power probes at the DERs, Loads and the POI, POI voltage and the microgrid frequency. At t = 61.77 s, the MCB abruptly tripped. The resulting unbalance was first met by the DERs primary response. However, as found by testing, the primary response would not pick up the frequency drop until it reaches 55 Hz. Unfortunately, by then, all protection relays would have tripped and there would be a blackout. To mitigate this issue, fast UFLS is utilized. The POI agent is programmed to watch for frequency deviation F_{th} = 0.3 Hz before starting its first load-shedding timer. Once the timer passes first timer threshold T_{th} = 20 ms, a load shedding command is broadcasted by the POI agent to the load agents. F_{th} and T_{th} are set by trial and error. Meanwhile, as can be followed from Figure 5.6, load agents used the DPQ algorithm to update the locally stored ND, whereas they all had $ND.ID = 2$. Figure 5.6 was built by watching the stored ND at each load agent at every step of the real-time

simulation. Therefore, at t = 61.92 sec upon receiving the shedding command from the POI agent, the load at bus 2 tripped because load 2 ID matched the stored ND ID, while the rest of the loads stayed connected. The time between abrupt tripping of the MCB and shedding of load 2 includes the time it took for the frequency to reach F_{th}, T_{th} and that each agent program iteration takes 20 ms to compute its states. Consequently, after the first shedding, it can be observed again in Figure 5.6 how in a sequence of 8 iterations the agents reach agreement once again on load 11, which is the second-largest non-critical load.

On the other hand, the POI agent reset the timer after the first load shedding command. However, for a subsequent load shedding, T_{th} is set to 3 seconds, which is a higher value than the first T_{th}. The response to shed the second load must be slower than the first load shed because time must be given for the DERs primary control to respond. Otherwise, over-frequency and further unnecessary load shedding are bound to happen. Three seconds later, the POI agent still read a frequency deviation larger than F_{th} = 0.3 Hz. Therefore, at t = 64.94 sec, the POI agent broadcasted another shedding command, which caused load 11 to trip.

Now, the control response is analyzed during unplanned islanding. At the time of the incident, the frequency incurred a 2.62% overshoot and crossing the first under-frequency threshold. However, the frequency 2% system settling was 1.32 s, therefore it was recovered within the 300 s clearing time back to the region of continuous operation. On the other hand, the voltage incurred a 9.89% overshoot, however, the voltage stayed within the region of continuous operation. Finally, the 5% settling time for P_{D4} active power response was recorded to be 14.65 s, after which the microgrid transitioned completely to islanded steady-state operation.

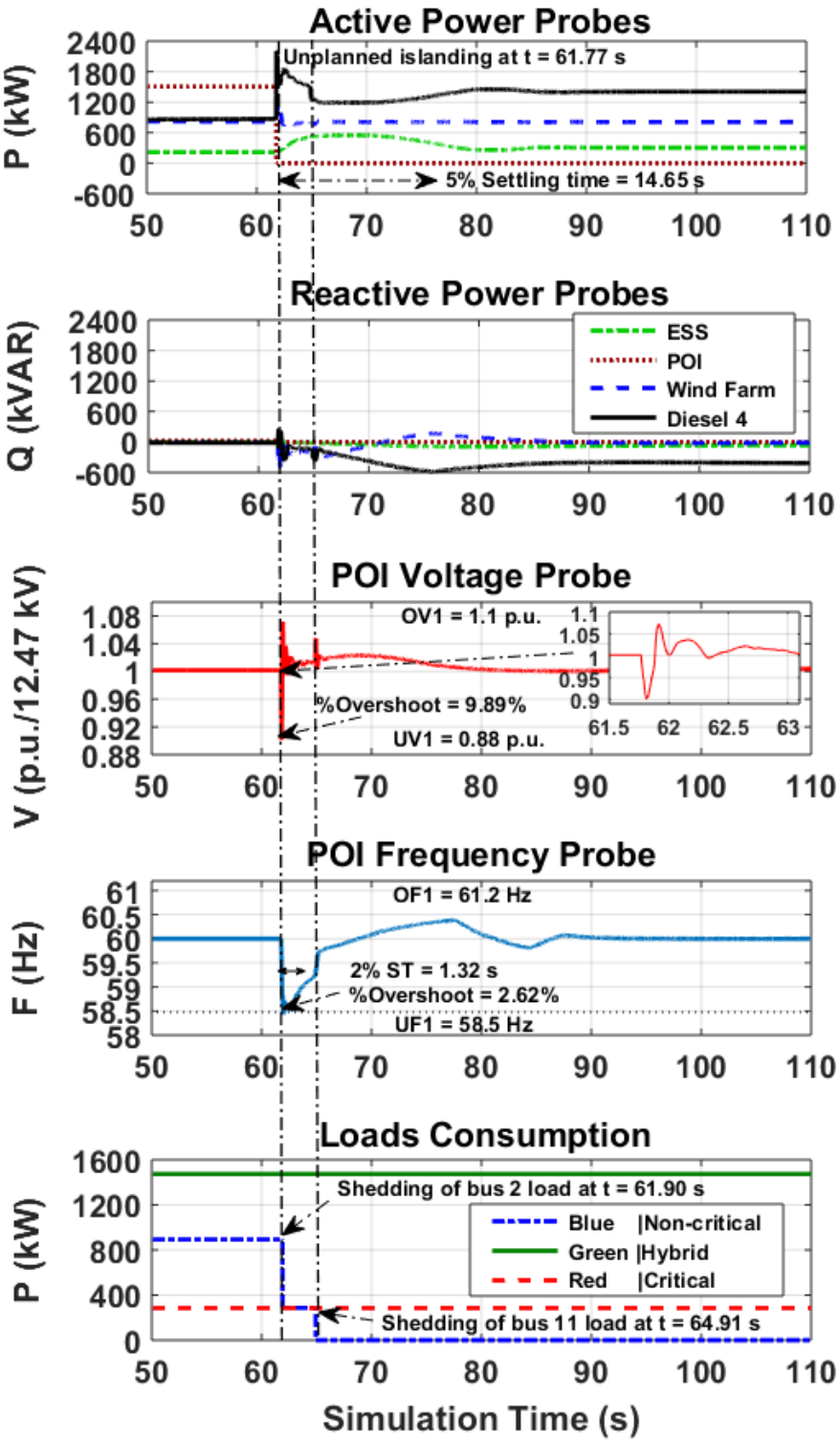

Figure 5.5 Unplanned islanding test results

Start at t = 61.918 s			Expected ID: 11							
Step (20 ms delay)			1	2	3	4	5	6	7	8
Load x	|Type	|Rating (kW)	Stored ND ID Entry at Load x							
Load 2	|Non-critical	|607	2	2	3	3	3	3	11	11
Load 3	|Hybrid	|250	2	3	3	3	3	11	11	11
Load 4	|Hybrid	|216	2	2	3	3	11	11	11	11
Load 5	|Hybrid	|267	2	2	2	5	5	11	11	11
Load 6	|Critical	|85	2	2	2	2	5	5	11	11
Load 7	|Hybrid	|228	2	2	2	3	3	5	5	11
Load 8	|Hybrid	|243	2	2	3	3	3	3	11	11
Load 9	|Critical	|200	2	2	2	3	3	11	11	11
Load 10	|Hybrid	|270	2	2	2	2	11	11	11	11
Load 11	|Non-critical	|287	2	2	2	11	11	11	11	11
Wait 3 seconds before next load shed										
Start at t = 64.935 s			Expected ID: 10							
Step (20 ms delay)			1	2	3	4	5	6	7	8
Load x	|Type	|Rating (kW)	Stored ND ID Entry at Load x							
Load 2	|Non-critical	|607	11	11	11	3	3	5	5	5
Load 3	|Hybrid	|250	11	11	3	3	5	5	5	5
Load 4	|Hybrid	|216	11	4	4	5	5	5	5	10
Load 5	|Hybrid	|267	11	11	5	5	5	5	5	5
Load 6	|Critical	|85	11	11	11	5	5	5	5	5
Load 7	|Hybrid	|228	11	11	11	11	5	5	5	5
Load 8	|Hybrid	|243	11	11	11	3	3	5	5	10
Load 9	|Critical	|200	11	11	10	10	3	3	10	10
Load 10	|Hybrid	|270	11	10	10	10	10	10	10	10
Load 11	|Non-critical	|287	11	11	10	10	5	5	10	10

9	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10

Figure 5.6 DPQ details, red font are tripped loads, blue font is the next load to be shed.

It should be noted that the POI agent does not wait for the agents to reach an agreement on shedding priority. Instead, the shedding command is sent out blindly, once each time the timer reaches T_{th} after its most recent reset, based on the estimation that the DPQ algorithm ran by the load agents has converged. The protocol is designed in this manner to allow for fast on-demand load shedding capability. Furthermore, the DPQ algorithm is always running in the background, regardless if there is an emergency event. This means that by the time an unplanned islanding event happens, the loads stored NDs are matched, and therefore the load with the matching ID is ready to be shed. The first load shed of the largest non-critical load is the one that mitigated the frequency drop that resulted from unplanned islanding. The second load shed of the second largest non-critical load helped in restoring the frequency and in keeping a reserve margin for load growth. Furthermore, the timer threshold T_{th} and the frequency threshold F_{th} have an important impact on the unplanned islanding response. If the first T_{th} is set to a value higher value, ex. 100 ms, it would

be observed that the third-largest load, which in this case would be load 10, sheds due to extended frequency excursion. However, as it is clear from this case, load 10 does not need to be shed. On the contrary, if T_{th} is set to a value lower than the time needed for agents to reach an agreement, then the DPQ algorithm could not converge, causing multiple load agents with matching $ND.ID$ to shed. In both cases, properly tuning the timer has important economic consequences. On the other hand, if the frequency threshold is set to a higher value, then the microgrid could go into a blackout as the load shedding would not be able to restore the frequency deviation in time. Finally, if the frequency threshold is set to a value lower than 0.3 Hz, then the MAS becomes too sensitive to frequency deviations, resulting in abrupt load tripping in islanded steady-state operation.

5.4 Further Analysis and Comparison with Centralized Control

5.4.1 Reliability Assessment

This study aims to point out the vulnerability of centralized control to the single point of failure and to quantify the impact of the DCS inherent redundancy. To study the reliability of a MGCS, the focus must be made on the reliability of the controller platform(s). Namely, the central controller in the CCS, and the DCS controllers in the DCS. Also, since both MGCSs are tested on the same microgrid model, the reliability indices of the physical components of the microgrid such as lines, transformers, and DERs, are invariant from the controllers' perspective. The only known work is in [53], where the authors attempted to compare the reliability of the DCS and the CCS using the state duration sampling approach by running Monte Carlo simulations as explained in Appendix D. However, in [53], the authors assumed continuous islanded mode of operation for the entirety of the Monte Carlo simulation. This is an unrealistic assumption since the islanded mode is considered an emergency mode, and typically runs for a short time only. This section adopts the methodology devised in [53] while considering the microgrid benchmark understudy in both islanded and grid-connected modes of operation.

From the CCS perspective, the failure of the centralized controller implies the failure of the microgrid regardless of its connection status with the grid. This is a valid assumption because, in grid-connected mode, the DERs cannot be left unmanaged. Furthermore, the failure of the centralized controller would also fail the load management function. This is potentially dangerous since the loads could exceed the power rating of the POI, without the central controller to manage

load-shedding mechanisms, the event can cause catastrophic damage. On the other hand, the failure of the CCS in islanded mode implies system blackout.

From the DCS perspective, the failure of one or more of the distributed controller agents does not cause microgrid failure. The agents can reconfigure their setpoints and maintain coordination with the remaining agents. For the DCS, it is only a matter of balancing the supply and demand. The failure of a DCS controller implies the disconnection of its corresponding DER from the microgrid. Besides, since the POI agent does not represent a DER, it is assumed that the failure of the POI agent causes islanding of the microgrid.

The simulation parameters are summarized in Table 5.5. The MTTF for the central controller is taken to be 8760 hours, expected to fail once per year, while the MTTF for each distributed controller is 4380, expected to fail twice per year [53]. On the other hand, the MTTR for all controllers is 12 hours. It is worth mentioning that it is possible in a DCS to have multiple controllers failing at the same time, though it is less likely. Nonetheless, the simulation considers multiple controller failures.

Table 5.5 MGCS controllers' enumeration and indices

System Type	Number of Controllers	Types of controllers	MTTF	MTTR
Centralized System	1	1 CCS Controller	8760	12
Distributed Control System	8	8 DCS Controllers	4380	12

The vulnerability is measured using two metrics, EENS and SAIFI. The goal is to have the two metric values converge to steady-state values in the Monte Carlo simulation. Every time a controller fails and repairs, new Time to Fail (TTF) and Time to Repair (TTR) are sampled from an exponential distribution function [71] and assigned in place of the original values. This process is chronologically repeated throughout the time of the simulation. Therefore, the simulation is ran for 1000 samples, where a new state starts once a controller fails and new TTF and TTR are sampled. Furthermore, the benchmark data load, solar and wind data are used based on the time of day and year from which a state transition is sampled. The data is in Appendix B.3. Figure 5.7 presents the convergence of the EENS metric, while Figure 5.8 presents the convergence of the SAIFI metric. Finally, Table 5.6 summarizes the results. Considering the assumptions in this study, it can be observed that there is a clear theoretical advantage in terms of the reliability of the DCS over the CCS, showing a 70.7% advantage in terms of the EENS, and 67.15% percent advantage

in terms of SAIFI. However, these results cannot be confidently be generalized from theory to practice, as having a redundant centralized controller is justifiable in terms of added cost in comparison to the DCS that is using 8 controllers. A redundant CCS controller would greatly enhance its reliability.

Table 5.6 Calculated metrics from Monte Carlo simulation

Reliability Metric	Centralized	Distributed	Advantage
EENS (MWh per year)	32.8	9.61	70.7 %
SAIFI (Inter. Per year)	1.076	0.3535	67.15 %

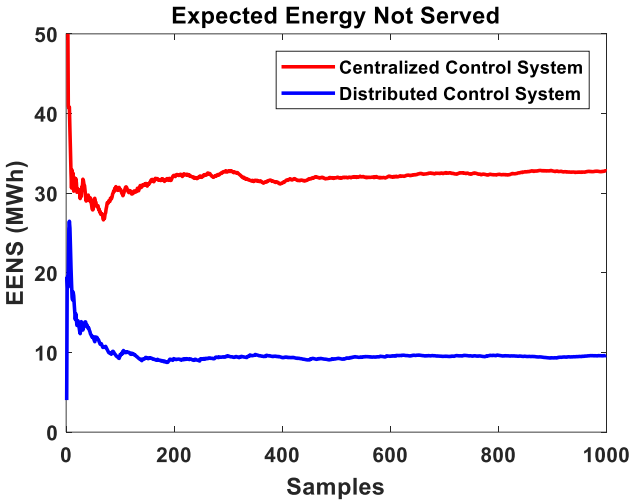

Figure 5.7 EENS results from the Monte Carlo Simulation of the two control systems

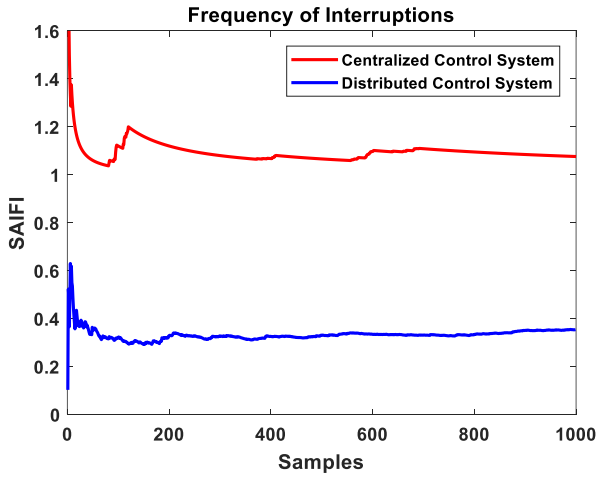

Figure 5.8 SAIFI results from the Monte Carlo Simulation of the two control systems

5.4.2 Optimization of Loads Curtailment

Load curtailment is utilized for a variety of applications. In this work, it is used to match the supply and demand before islanding. Therefore, the objective is to find the cheapest combination of loads to curtail. To construct a more insightful study, it will be assumed that all 10 benchmark loads are hybrid loads, which are curtailable, and have different costs. Table 5.7 below lists the load characteristics and each respective cost. The main goal of this study is to evaluate and compare the performance of three different dispatch algorithms. To remark, the three algorithms being: Rule-based Priority Dispatch as explained in section 2.7.2, Equal Incremental Cost Criterion (EICC) [67], and the consensus algorithm in (29) and (30). While the first two algorithms are centralized, the third is a distributed algorithm. The optimization algorithms are solved to a 10^{-5} tolerance.

Table 5.7 Loads ratings and costs

Bus #	Max Load (kVA)	Min Load (kVA)	Cost Parameter a $/h.kW^2
2	1012.5	0	0.001
3	417.0	0	0.005
4	360.0	0	0.008
5	445.0	0	0.004
6	142.5	0	0.010
7	380.0	0	0.007
8	405.0	0	0.006
9	332.5	0	0.009
10	450.0	0	0.003
11	477.5	0	0.002

Table 5.8 shows the results. First, the Consensus algorithm and the EICC algorithm are solved for almost half the cost as what is solved by Priority rule-based dispatch. The reason is simple, the EICC and Consensus can use the cost curve characteristic to further divide up the curtailments between available loads. The priority-based dispatch is the most expensive one as it only sorts which loads to curtail from largest to smallest. On another note, it can be noticed that the consensus algorithm had a 0.4% more cost than the EICC algorithm. The reason behind this is the choice of the consensus algorithm learning rate, which traded off having a much quicker

solution for the 0.4% increase in cost. This is more elaborated on in the next section, on the consensus learning rate parametric analysis.

In terms of hardware runtime, Table 5.8 shows that the priority-based and the consensus algorithm have approximately the same absolute runtime performance, each measured by running a timer clock from the beginning till the end of their program. The reason for that fast performance of the latter two algorithms is that the underlying code is very simple, involving mainly some rules and a single loop. However, due to the distributed nature of the consensus algorithm, the communication time must be considered. Therefore, the total time runtime required is 1.815 s, considering 5 ms communication delay per iteration, for 363 iterations. On the other hand, EICC takes on average x400 more runtime than the priority-based algorithm to run, but it converges 50 times faster on average than the consensus algorithm.

Table 5.8 Results of curtailment considering three dispatch algorithms

Load Bus # and Curtailment	Priority-based (Centralized)	EICC (Centralized)	Consensus (Distributed)
Load 2 (kW)	1012.5	477.9840	438.6547
Load 3 (kW)	0	95.5968	101.2280
Load 4 (kW)	0	59.7480	64.1934
Load 5 (kW)	0	119.4960	125.3299
Load 6 (kW)	0	47.7984	51.6064
Load 7 (kW)	0	68.2834	73.1091
Load 8 (kW)	0	79.6640	84.9009
Load 9 (kW)	0	53.1093	57.2158
Load 10 (kW)	0	159.3280	164.4955
Load 11 (kW)	387.5	238.9920	239.2662
Total Runtime (s)	0.00037	0.0335	0.00008 + 1.815[2]
Total Cost ($/hr)	662.7344	334.5888	335.9409

5.4.3 Consensus Learning Rate Parametric Analysis

The ability to converge fast enough using the consensus algorithm is important to consider speeding up the dispatch calculation time. In this study, it is explored how the consensus algorithm

[2] 1.815 seconds is the communication delay time, since each iteration by the consensus algorithm involves 5 ms of communication delay between DCS controllers.

behaves when using different learning rates. The goal is to investigate how to decrease the number of iterations required for convergence. Despite the low computational requirement needed by the hardware to run the consensus algorithm, the communication time plays an important role because the result of every iteration needs to be communicated between neighboring agents. Since the main player in the convergence speed is the consensus learning rate, the same load curtailment case study in the previous section is re-iterated using different learning rates. The results are recorded in Table 5.9. The total communication time is the number of iterations multiplied by 5 ms delay per iteration.

Table 5.9 Consensus learning rate parametric analysis results

Learning Rate ε	# of Iterations	Total Com. Time (s)	Consensus Cost ($/hr)	EICC Cost ($/hr)
0.00104	9207	46.035	351.9848	334.5888
0.001005	406	2.030	351.4360	334.5888
0.001	357	1.785	351.1377	334.5888
0.0002	363	1.815	335.9409	334.5888
0.0001	716	3.580	334.9688	334.5888
0.00005	1420	7.100	334.6895	334.5888
0.00001	7056	35.280	334.5931	334.5888
0.000001	70456	352.280	334.5888	334.5888

As can be noticed, there is a trade-off between the number of iterations required to reach 10^{-5} tolerance and the resulting curtailment cost. Specifically, it can be observed that while the lowest number of iterations can be achieved with $\alpha = 0.001$, the resulting cost of operation is $351.1377 which is 4.413% less than the optimal value that would be obtained by the EICC algorithm that is $334.5888. On the other hand, reaching the optimal value using the consensus algorithm would require $\alpha = 0.000001$ with a total of 70456 iterations. The total time required considering 5 ms communication delay per iteration, is 1.785 seconds for $\alpha = 0.001$, and 352.280 seconds for $\alpha = 0.000001$. Depending on the time criticality of the application, it might not be so compelling to consider the 5 extra minute's option to solve for a 0.4% cost improvement. Finally, using learning rates of values higher than 0.00104 may never converge to the required tolerance value due to sustained oscillations.

5.4.4 Distributed Control System Plug and Play

One of the features of the proposed DCS is its modular design which allows seamless plug and play. The DCS framework is designed to look at the control system from the subjective point of view of the asset. For example, one DER operating in a microgrid does not have any knowledge about its neighboring DERs specifications. It only receives two key pieces of information from its neighbors to allow it to run the consensus algorithm. One, whether the neighboring DER is in operation or not. Two, the consensus signal, which is needed for distributed coordination. If a new DER agent is plugged in the microgrid, the other DER agent does not need to change any of its source code to allow it to work with that new DER.

To demonstrate the flexibility of a DCS, a hypothetical case study is carried-out in grid-connected mode to show the automatic update of its connectivity matrix. The results are shown in Figure 5.9. During a time of peak demand, the utility calls for a reduction in the power exported from the microgrid from -1500 kW to -500 kW. Consequently, DG9 got connected to provide more power to the microgrid. Upon connecting, DG9 starts coordinating with its neighbors which are the other diesel generator at bus 4 (DG4), and the ESS. To establish this coordination, the connectivity matrix L of the DCS automatically updates to accommodate DG9. This update can be seen in Table 5.10 from L_1 to L_2. Once the utility demand of reduced power exports relaxes from -500 kW to -1600 kW, the larger diesel unit DG4 is disconnected as its services are no longer needed. Consequently, DG9 and the ESS adjust their operations accordingly and the connectivity matrix is updated again from L_2 to L_3. Note that each DER only sees a single corresponding row in L, and not the whole L. For example, in the beginning, DG4 local connectivity matrix was constructed to be [0.5 0 0.5], which says to DG4 that the second DER is disconnected, while the third is connected. Once DG9 is connected, this row is updated to be [0.33 0.33 0.33], indicating that three DERs now can be connected and they all participate in the consensus algorithm. This process can be generalized to any number of DERs.

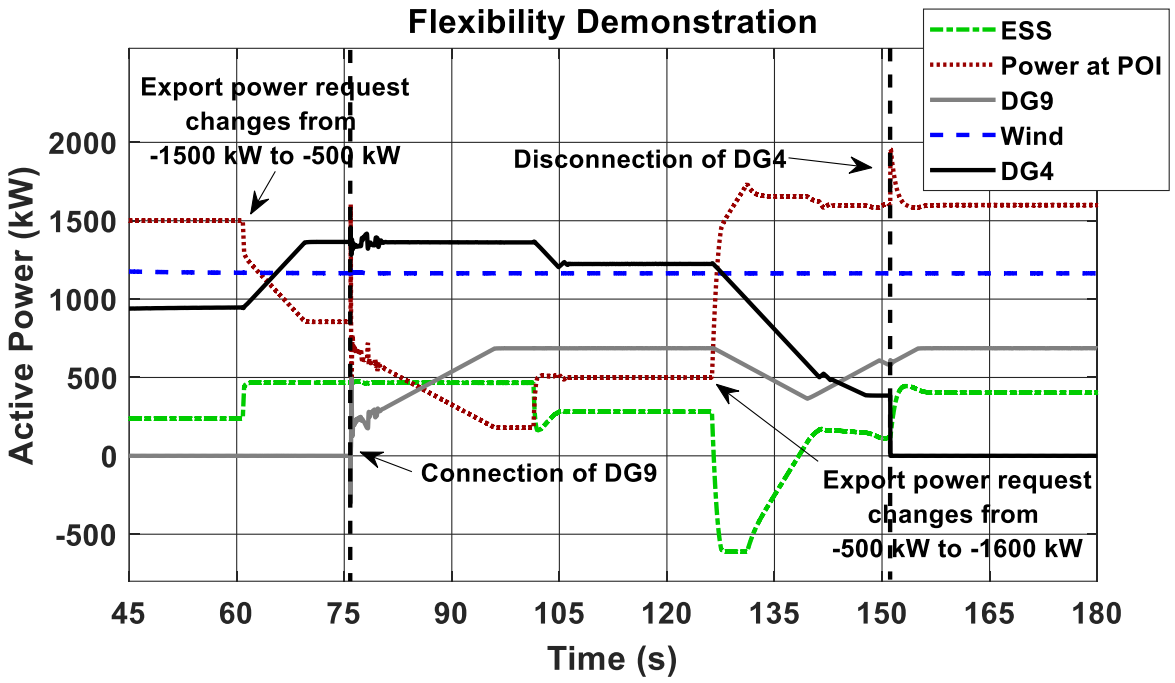

Figure 5.9 Demonstration on DCS flexibility

Table 5.10 Connectivity matrix during each phase of the simulation

Asset	Status	L_1			Status	L_2			Status	L_3		
DG4	1	0.5	0	0.5	1	0.33	0.33	0.33	0	0	0	0
DG9	0	0	0	0	1	0.33	0.33	0.33	1	0.5	0	0.5
ESS	1	0.5	0	0.5	1	0.33	0.33	0.33	1	0.5	0	0.5

On the other hand, a CCS does not by default support plug and play, and its flexibility is limited by the need to update its program to incorporate new assets [17, 55]. A conventional CCS operates by having complete knowledge of the microgrid. The optimization algorithm running in the CCS has all the assets characteristics registered in its database to perform proper optimization. Therefore, it is difficult to update the CCS program logic with new assets without the need to perform a complete system shutdown and perform an upgrade to the control system. This is particularly the case with the CCS that supports rule-based dispatch, where the rules are set up and tuned to operate according to the assets at hand. While this approach works well for sites that do not expect a future change in topology, it does not work as well with microgrids that would.

5.5 Summary

In this chapter, the ability of the proposed DCS to fulfill the required microgrid core functions was demonstrated. In the conducted case studies, the results were quantified using performance metrics and were shown to be reasonable in the context of microgrid control. Furthermore, the frequency and voltage results adhered to the IEEE standard 1547, not allowing DERs to trip during microgrid operation.

The case studies gave a detailed overview of the MGCS coordination capability. In grid-connected mode, the coordinated control of DER agents met the required active power requirements at the POI despite the presence of wind disturbance. Also, the control response was timely against step changes in control setpoints. In islanded mode, the DCS response to step load changes was smooth, showing its ability to regulate the frequency and voltage to their nominal values, while maintaining economic operation as compared to centralized control. In planned islanding, the DCS was able to carry out seamless islanding, whereas the time sequence of events was detailed and tradeoffs in control speed were explained. During planned islanding, distributed load curtailment by load agents was used to cancel any excess power at the POI, allowing the DER agents to properly continue microgrid control post islanding. Finally, during unplanned islanding, fast load shedding was executed promptly using the proposed DPQ algorithm to save the microgrid from instability.

In addition to economic operation, a comparison between distributed and centralized control was carried out in terms of reliability. Monte Carlo simulations showed that distributed control can cut the lost energy and customer interruptions to a third.

For the next step in control deployment, the DCS should be demonstrated on a power hardware-in-the-loop setup. This way, the magnitude and the timing of the control response and its ability to maintain the integrity of the assets under control would be better validated.

Chapter 6

Conclusions

6.1 Summary

Microgrids are a popular solution for controlling distribution level assets as a single unit. The end objective is to achieve robust control of the assets to increase the reliability and efficiency of the power system and to provide a platform for the proper integration of renewable DERs. Being an innovative solution, the MGCS should be reliable in its terms, economical and flexible for expansion. However, the microgrid control problem exhibits many challenges that need to be met by the MGCS. Also, the MGCS must be able to deliver steady-state and transition core functional requirements specified in the IEEE standard 2030.7. The conventional centralized control system falls short in being susceptible to a single point of failure and difficult to scale. On the other hand, when switching to a distributed control system, coordinating the transition function is a challenging endeavor that is rarely discussed in the literature.

Therefore, the thesis proposes a multi-agent system framework that can carry out microgrid core functions while maintaining the features promised by distributed control. The multi-agent system agents are assigned to dispatchable DERs, loads and the MCB. The agents consider a cooperative coordination strategy and rely on local measurements and neighbor communications to execute their functions. The agents are modular in their design because they are adaptable to the asset under control and can immediately engage in distributed coordination without the need for reprogramming. This adaptability is granted by the algorithms presented in the thesis. Specifically, the consensus algorithm allows the control to be flexible by automatically updating the L connectivity matrix. On the other hand, the distributed priority queuing (DPQ) algorithm only requires the load agent to receive the updated data tuple from each near neighbor.

The designed control system employs distributed logic and decision making. The consensus algorithm is employed for the applications of coordinating power tracking and load curtailment in grid-connected mode, and frequency and voltage regulation in islanded mode. On the other hand, the DPQ algorithm is to find and select the largest and least critical load for emergency shedding. This distributed logic is aided by centralized decision making for fast transition events. Specifically, in the application of islanding, the POI agent decides when is the

best time for shedding or curtailment. However, the POI agent is not familiar with the load agents, their ratings, their types, or their capacity to shed or curtail. Selecting which load is curtailed/shed and by how much, is decided in a distributed fashion by the consensus algorithm and the DPQ algorithm respectively.

Practical case studies were designed and conducted to strictly analyze the performance of the DCS under each microgrid mode of operation using quantifiable metrics. The case studies were conducted on a validated microgrid benchmark. Furthermore, the impact of using different learning rates for the consensus algorithm is studied for several applications including load curtailment and frequency regulation. The testing of the DCS was done on a controller hardware-in-the-loop testbed, whereas the microgrid model, load agents, and the POI agent ran on the real-time simulator, and the DER agents ran on hardware controllers. The control performance of the DCS is assessed in terms of response time, settling time, overshoot, integral absolute error during power tracking and steady-state error. Moreover, the voltage and frequency responses in islanded mode are compared against standard metrics. Finally, comparisons with centralized control were conducted on multiple aspects including reliability and economical operation.

6.2 Conclusions

The DCS framework proposed for the control of microgrid assets is capable of carrying out dispatch and transition functions. Through the conducted case studies, it was demonstrated that the DCS can meet the minimum requirements for microgrid control when operated in several scenarios, including the dispatch of microgrid assets in grid-connected mode, islanded mode, planned islanding and unplanned islanding. Results by case studies presented in Chapter 5 show capable control performance in the presence of disturbances and relevant initiating events, maintaining the frequency and voltage with the IEEE standard 1547 standard continuous operating regions. On the other hand, control performance metrics were calculated based on the recorded results and were shown to be acceptable.

Furthermore, it was demonstrated that the DCS can offer some advantages over the CCS while suffering from a few limitations. Table 6.1 lists the finding and conclusions of comparing the DCS and CCS on several points of comparison considering their facts and the demonstrated results. The DCS was shown to be advantageous with respect to increased reliability, low hardware computational requirements, improved scalability, and enhanced control system flexibility. On the

other hand, the DCS was found to be on par with centralized control on solving for cost-optimal dispatch setpoints. However, as the consensus solution approaches the optimal solution, the time it requires to reach that solution increases. In addition, the DCS presents a disadvantage regarding the dispatch total convergence time, because neighbor to neighbor communications are required by the DCS for each solution iteration which add delays; whereas in the CCS, the algorithm iterations are done on the same controller. This limitation also impacts the DCS capability of coordinating fast events. Therefore, the DCS required the use of a centrally acting controller, the POI agent, to execute fast and timely functions such as load curtailment and load shedding during islanding. Nevertheless, this limitation was considered when conducting the reliability assessment and hence it has a limited impact on the DCS reliability.

Table 6.1 Summary of comparisons between CCS and DCS

Point of comparison	**Centralized Control System (CCS)**	**Distributed Control System (DCS)**	**Findings and Conclusions**
Reliability	Single point of failure at the centralized controller. The failure of the central controller impacts the entire microgrid.	No single point of failure. The failure of a DCS controller impacts only the associated asset.	DCS has improved reliability indices EENS and SAIFI as demonstrated in section 5.4.1.
Cost of operation and resource optimization	Optimization algorithms such as EICC can find the optimal dispatch setpoints, while rule-based algorithms such as priority-based are not optimal.	Considering the consensus algorithm, optimal dispatch setpoints can be found when the consensus learning rate is small enough, otherwise, it is sub-optimal.	As demonstrated by the results in sections 5.3.2 and 5.4.2, and considering the choice of the consensus learning rate, the DCS is as costly as the CCS.
Hardware computational requirements	A single program is run by a single controller that solves the optimization problem and the entire microgrid control logic.	The optimization problem is divided among the DCS controllers. Every asset agent solves a sub-problem of the microgrid control logic.	Considering the use of an optimization algorithm, DCS solves much faster for its setpoints as demonstrated in section 5.4.2.
Dispatch algorithm total convergence time	The centralized optimization problem time is the total time it takes for the central controller	The distributed optimization problem time is the time it takes for the global solution	As demonstrated by the results in section 5.42, the DCS is slower than a CCS

	to solve for the new dispatch setpoints.	to converge among the DCS controllers.	due to overhead communication time.
Control system complexity and scalability	CCS dispatch algorithm complexity is quadratic. The algorithm solves for all the assets in the microgrid.	DCS dispatch algorithm complexity is linear. Complexity depends on the number of neighboring agents considered in the dispatch algorithm.	As demonstrated by the consensus convergence analysis in [22] and the DPQ convergence analysis in section 3.5.3, the DCS dispatch algorithm has less complexity and is more scalable.
Control system flexibility	The CCS controller requires global knowledge and predefined logic to run its algorithms. Therefore, the removal or addition of new assets require a system shutdown and upgrade.	The DCS controller requires local knowledge and neighbor communications. The addition or removal of a DCS controller does not impede the DCS operation.	DCS flexibility was demonstrated in section 5.4.4. The DCS advantage over CCS regarding control system flexibility is further asserted in [17, 55].

Seamless planned islanding was achieved by utilizing both dispatchable DERs and curtailable loads. It has been demonstrated that the consensus algorithm can be effective for optimizing load curtailment, given the proper choice of the consensus learning rate. Also, the proposed coordination strategy during unplanned islanding has shown promising capability in recovering the microgrid from the emergency using fast load shedding. The proposed DPQ algorithm allows timely distributed coordination between load agents to select the largest and least critical load. However, the POI agent is required for timing the shedding command sent to load agents.

Finally, it has been shown that multiple parameters used by the agent's algorithms require local tuning, which was done manually in this work. For example, finding the optimal consensus learning rate that can achieve a good system response and achieve fast convergence may require scanning a wide range of values. In cases such as frequency regulation, finding the optimal learning rate can only be achieved when tested online in the presence of the microgrid dynamics, which is more difficult to carry out.

6.3 Future Work

In this thesis, a framework for a distributed multi-agent system is proposed for microgrid control. Microgrids typically feature high penetration of renewable energy which, due to their output power uncertainty, can cause the control system output to deviate and even fluctuate. In the applied case studies, step changes were introduced to the meteorological data fed to renewables to test the control system response. In future work, the use of high resolution solar and wind data can instead be explored to test the control system against more severe power fluctuations.

In addition, thesis work can be extended by using an adaptive consensus learning rate to improve consensus convergence speed. As indicated in the parametric analysis in section 5.4.3, operational cost improvements can be achieved with a much smaller number of iterations. Having a smaller number of iterations in a distributed control system can potentially reduce the total convergence time since every iteration requires communications to be carried out between neighbors.

Another potential extension of the consensus algorithm is to consider the use of a multi-step consensus. This allows consensus to be reached for multiple time frames, which could be used to solve the unit commitment problem in a distributed fashion.

The use of local frequency measurements at the loads can be investigated to further improve the DCS framework. Specifically, the use of local frequency measurements at each load may relax the need to use the POI agent as a central entity for sending the shedding commands.

Furthermore, renewable resources are expected to have reactive power contributions to support the distribution system voltage. Therefore, another potential extension is to include in the MAS agents for renewable DERs to coordinate their reactive power production or consumption.

Appendix A

IEEE Standard 2030.7 Regarding Microgrid Controller Functions

IEEE standard 2030.7 specifies two core functions that are the dispatch and the transition functions. Figure A.1 below summarizes the dispatch modes and the transition logic. It entails six modes of operation, including two steady-state modes and four transitive modes. At every iteration, the microgrid control system checks the islanding status. If the system is islanded, it checks if it is islanded because of a blackout, in which case a black start procedure is initiated. If it is not a blackout, it checks for any reconnection requests. If no reconnection request is made, then it is a steady-state islanded operation, and it will issue the corresponding dispatch function. Otherwise, if the system is grid-connected, then it checks for any islanding requests. If an islanding request is made, then it issues a planned islanding dispatch. Otherwise, if it is islanded without an islanding request, after which it has already been declared as a grid-connected system, then un-planned islanding phenomena occurred. In this case, an emergency dispatch order is issued. If everything is ok, then it is a steady-state grid-connected operation, and the corresponding dispatch command is issued.

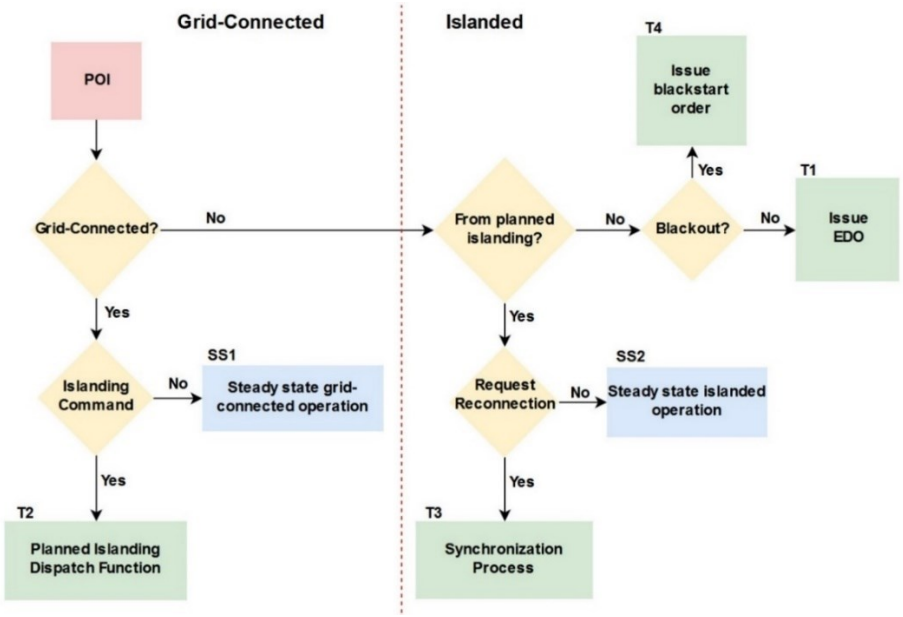

Figure A.1 MGCS transition logic

Appendix B

Benchmark Model Validation and Data

B.1 Model Context and Structure

The benchmark topology originated from a remote German rural distribution network with inherited problems [72]. The problems were voltage drop down to 0.95 p.u., not served energy and overheated transformer problems during winter nights. The network was chosen to study the behavior of a distribution microgrid with large penetration DERs and particularly renewable energy sources. Next, a set of benchmarks was proposed to allow the modeling and study of DER integration problems [73]. The benchmark brochure was finally published in 2014 [65]. The brochure contains both European and North American adaptations, of low, medium and high voltage benchmarks. Therefore, from the time and experience put in the making of this benchmark, it is the ideal choice for microgrid studies and microgrid control systems. Further modeling details are present in [74]. It is remarked that this benchmark has a few challenges. For instance, having single-phase elements introduces unbalanced phases in the simulation. Also, relatively long feeder lines result in voltage drops that need to be compensated.

B.2 Benchmark Validation

The task is to validate the powerflow results of our implementation against the published benchmark results. The distribution network was developed on the Simulink platform, where the Simscape Power GUI tool was used to generate the power flow results. Simscape load flow results are presented in Appendix A, in Table B.3 and Table B.4 for the three-phase sections and single-phase sections, respectively. They also included the discrepancies of the results for every entry with the benchmark powerflow results. Table B.1 and Table B.2 list the total average discrepancies between the Simulink results and the benchmark published results for the three-phase sections and single-phase sections, respectively. As shown, the errors are within acceptable threshold values. Henceforth, it is concluded that the benchmark is validated according to the power-flow results published in [74].

Table B.1 Three-phase load flow Simulink results and benchmark results

Voltage		Current	
Magnitude (LL RMS)	Phase (Degrees)	Magnitude (A RMS)	Phase
Average Percentage Error from [65]	Average Absolute Error from [65]	Average Percentage Error from [65]	Average Absolute Error from [65]
(%)	(Degrees)	(%)	(Degrees)
1.64	2.52	4.58	2.68

Table B.2 Single-phase load flow Simulink results and benchmark results

Voltage		Current	
Magnitude (LN RMS)	Phase (Degrees)	Magnitude (A RMS)	Phase
Average Percentage Error from [65]	Average Absolute Error from [65]	Average Percentage Error from [65]	Average Absolute Error from [65]
(%)	(Degrees)	(%)	(Degrees)
1.81	2.77	4.80	2.71

Table B.3 Powerflow results of three-phase busses

Bus	Voltage					From bus to bus	Current				
	Phase	LL RMS	Percentage E from [65]	Angle	Absolute Error from [65]		Phase	RMS	Percentage E from [65]	Angle	Absolute Error from [65]
		[kV]	%	[degree]				[A]	%	[degree]	
0	AB	115	0.00	0	0	HV bus -0	A	198.5	3.74	-31.56	2.58
0	BC	115	0.00	-120	0	HV bus -0	B	201.52	3.46	-152.68	2.57
0	CA	115	0.00	120	0	HV bus -0	C	196.65	3.60	87.11	2.42
1	AB	12.31	1.44	22.25	2.89	0 - 1	A	1368.1	0.59	-2.79	2.93
1	BC	12.33	1.52	-97.47	2.81	0 – 1	B	1326.23	0.68	-122.8	2.78
1	CA	12.37	1.59	142.32	2.91	0 – 1	C	1277.93	0.16	116.82	2.74
2	AB	12.00	1.56	20.85	2.81	1 – 2	A	247.8	3.77	-3.99	2.9
2	BC	11.99	1.72	-98.72	2.76	1 – 2	B	289.41	4.95	-125.75	2.72
2	CA	12.10	1.55	141.10	2.85	1 – 2	C	254.87	3.50	114.96	2.69
3	AB	11.82	1.66	19.87	2.74	2 – 3	A	169.86	4.25	-4.51	2.87
3	BC	11.77	1.83	-99.55	2.71	2 – 3	B	241.06	5.50	-125.17	2.63

3	CA	11.90	1.73	140.37	2.81	2 – 3	C	185.9	4.22	114.85	2.6
4	AB	11.78	1.67	19.63	2.73	3 – 4	A	69.19	3.87	-2.51	2.87
4	BC	11.72	1.92	-99.73	2.69	3 – 4	B	91.85	5.68	-124.85	2.57
4	CA	11.87	1.82	140.19	2.81	3 – 4	C	57.41	4.40	117.4	2.7
5	AB	11.75	1.76	19.45	2.72	4 – 5	A	41.49	3.96	-0.37	2.87
5	BC	11.69	1.85	-99.85	2.69	4 – 5	B	77.97	5.85	-124.44	2.71
5	CA	11.84	1.82	140.08	2.82	4 – 5	C	43.55	4.84	118.28	2.69
6	AB	11.74	1.76	19.36	2.71	5 – 6	A	6.94	3.89	1.84	2.9
6	BC	11.66	1.93	-100.09	2.67	5 – 6	B	13.89	5.39	-118.88	2.67
6	CA	11.80	1.83	139.98	2.8	5 – 6	C	36.73	5.12	119.66	2.26
7	AB	11.68	1.77	19.06	2.69	8 – 7	A	13.89	4.51	1.53	2.85
7	BC	11.60	1.94	-100.33	2.67	8 – 7	B	27.78	5.87	-119.26	2.63
7	CA	11.75	1.84	139.73	2.78	8 – 7	C	13.89	4.12	121.6	2.63
8	AB	11.71	1.93	19.26	2.7	3 – 8	A	89.86	4.62	-4.6	2.85
8	BC	11.63	1.94	-100.19	2.67	3 – 8	B	110.6	5.77	-124.07	2.64
8	CA	11.77	1.83	139.84	2.78	3 – 8	C	110.74	4.36	114.74	2.8
9	AB	11.69	1.68	19.17	2.7	8 – 9	A	62.19	4.71	-5.68	2.85
9	BC	11.61	1.94	-100.28	2.67	8 – 9	B	62.14	5.84	-125.39	2.63
9	CA	11.75	1.84	139.76	2.78	8 – 9	C	69.19	4.53	113.59	2.62
10	AB	11.65	1.77	19.03	2.68	9 – 10	A	48.45	4.80	-7.76	2.85
10	BC	11.58	1.95	-100.46	2.65	9 – 10	B	41.5	5.92	-128.57	2.64
10	CA	11.71	1.84	139.60	2.77	9 – 10	C	55.48	4.60	111.57	2.63
11	AB	11.64	1.77	18.99	2.68	10 – 11	A	27.63	4.82	-8.87	2.86
11	BC	11.57	1.95	-100.48	2.66	10 – 11	B	27.63	5.94	-129.44	2.64
11	CA	11.70	1.93	139.56	2.76	10 – 11	C	20.83	4.62	107.85	2.64

Table B.4 Power flow results of single-phase busses

Bus	Subnetwork Bus	Voltage					From bus to bus	Current			
		Phase	LL RMS	Percentage E from [65]	Angle	Absolute Error from [65]		RMS	Percentage E from [65]	Angle	Absolute Error from [65]
			[kV]	%	[degree]			[A]	%	[degree]	
2	1	A	6.94	1.56	-9.09	2.92	MV bus 2 – 1	36.73	3.49	-28.99	2.94
2	2	A	6.93	1.56	-9.12	2.94	1 – 2	34.66	3.59	-28.64	2.94

2	3	A	6.93	1.42	-9.14	2.95	2 – 3	20.12	3.82	-28.07	2.94
2	4	A	6.92	1.56	-9.14	2.95	3 – 4	1.39	2.21	-27.33	2.96
2	5	A	6.93	1.56	-9.12	2.94	2 – 5	4.16	2.21	-34.91	2.92
2	6	A	6.93	1.56	-9.12	2.95	5 – 6	2.08	1.96	-34.96	2.95
2	7	A	6.93	1.56	-9.13	2.94	2 – 7	8.33	4.00	-27.3	2.94
2	8	A	6.93	1.56	-9.13	2.95	7 – 8	6.94	4.36	-27.33	2.95
2	9	A	6.92	1.56	-9.14	2.95	3 – 9	3.46	2.37	-31.86	2.93
2	10	A	6.92	1.56	-9.14	2.96	9 – 10	1.39	2.21	-27.34	2.95
2	11	A	6.92	1.56	-9.14	2.96	3 – 11	8.33	4.13	-27.32	2.95
2	12	A	6.92	1.56	-9.15	2.96	11 – 12	6.94	4.36	-27.35	2.96
5	1	B	6.71	2.04	-130.66	2.66	MV bus 5 – 1	36.74	6.62	-150.56	2.69
5	2	B	6.7	2.05	-130.7	2.68	1 – 2	34.67	6.81	-150.21	3.69
5	3	B	6.7	1.90	-130.71	2.70	2 – 3	20.12	6.91	-149.64	2.69
5	4	B	6.69	2.05	-130.71	2.70	3 – 4	1.39	5.30	-148.9	2.70
5	5	B	6.7	2.05	-130.7	2.68	2 – 5	4.17	5.57	-156.48	2.67
5	6	B	6.69	2.19	-130.71	2.67	5 – 6	2.08	5.05	-156.53	2.70
5	7	B	6.7	2.05	-130.7	2.69	2 – 7	8.33	7.07	-148.87	2.68
5	8	B	6.7	1.90	-130.71	2.69	7 – 8	6.94	7.43	-148.9	2.69
5	9	B	6.69	2.05	-130.72	2.69	3 – 9	3.46	5.17	-153.43	2.68
5	10	B	6.69	2.05	-130.71	2.70	9 – 10	1.39	5.30	-148.91	2.70
5	11	B	6.69	2.05	-130.71	2.71	3 – 11	8.33	7.21	-148.89	2.66
5	12	B	6.69	2.05	-130.72	2.71	11 – 12	6.94	7.60	-148.92	2.71
6	1	C	6.82	1.73	109.97	2.65	MV bus 6 – 1	36.73	5.12	90.06	2.66
6	2	C	6.81	1.73	109.92	2.65	1 – 2	34.66	5.22	90.41	2.66
6	3	C	6.8	1.88	109.91	2.67	2 – 3	20.12	5.45	90.98	2.67
6	4	C	6.8	1.88	109.91	2.67	3 – 4	1.39	3.73	91.72	2.68
6	5	C	6.8	1.88	109.91	2.64	2 – 5	4.17	3.99	84.14	3.36
6	6	C	6.8	1.88	109.91	2.65	5 – 6	2.08	3.48	84.09	3.66
6	7	C	6.8	1.88	109.91	2.65	2 – 7	8.33	5.71	91.76	2.67
6	8	C	6.8	1.88	109.91	2.66	7 – 8	6.94	5.95	91.73	2.68
6	9	C	6.79	2.02	109.9	2.66	3 – 9	3.46	3.90	87.19	2.65
6	10	C	6.8	1.88	109.9	2.67	9 – 10	1.39	3.73	91.72	2.68

6	11	C	6.8	1.73	109.91	2.68	3 – 11	8.33	5.71	91.74	2.68
6	12	C	6.79	1.88	109.9	2.68	11 – 12	6.94	6.12	91.71	2.69

B.3 Benchmark Model Data

The benchmark model features two load categories, residential and commercial. The coincidental peak loads are 2650 kVA for the residential load, and 2290 kVA for the commercial load. Figure B.1 shows the daily load profile of each load, adopted from [65].

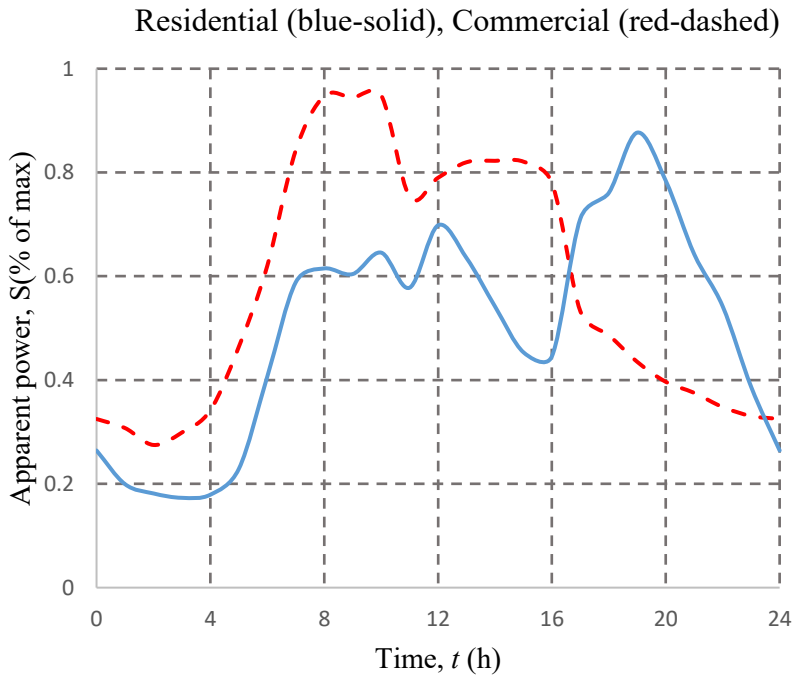

Figure B.1 Daily load profiles

On the other hand, Figure B.2 and Figure B.3 below show the wind speed histogram and solar irradiation yearly profile as used for the reliability assessment study, adopted from [75].

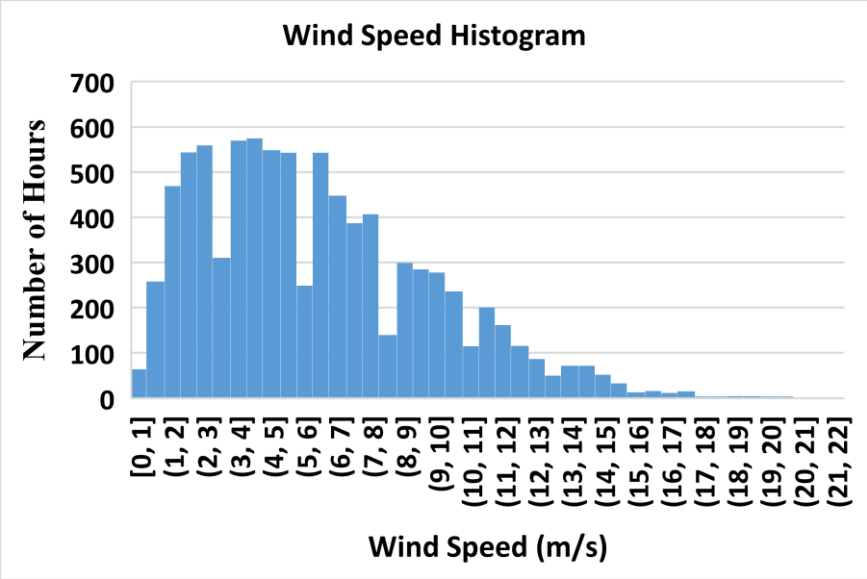

Figure B.2 Yearly wind speed histogram

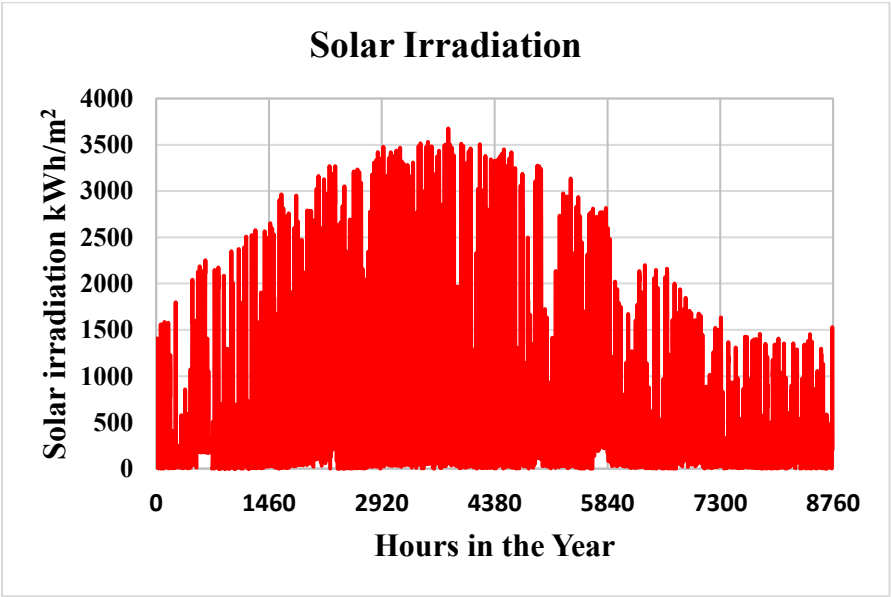

Figure B.3 Yearly solar irradiation

Appendix C

Microgrid Assets Models and Primary Controls

C.1 Substation Model

The substation consists of a 3-phase voltage source representing the sub-transmission grid, a 3-phase transformer, and a medium voltage circuit breaker. The sub-transmission grid is modeled as a 115 kV Y_g three-phase swing voltage source. It has a 3-phase short-circuit power of 5000 MVA and a 10 X/R ratio. The 3-phase transformer is a 15 MVA D_1/Y_g transformer that drops the voltage from 115 kV to 13.8 kV. Finally, the 3-phase medium voltage circuit breaker is the point of interconnection which can be controlled to switch to islanded mode or grid-connected mode.

C.2 Diesel Generator Model

The diesel generator model expects active power, reactive power, and voltage and frequency setpoints from the controller. Figure C.1 depicts the primary control function. Primary control is based on droop control. The frequency deviation is multiplied by a droop gain, which translates to an increase in active power when the frequency drops, and vice versa. The resulting active power setpoint is fed into a modeled engine, which in turn provides mechanical power input to the Simscape synchronous machine model. On the other hand, the voltage and reactive power reference are fed to the automatic voltage regulator through a PI controller, which in turns provide the field voltage for the synchronous machine model.

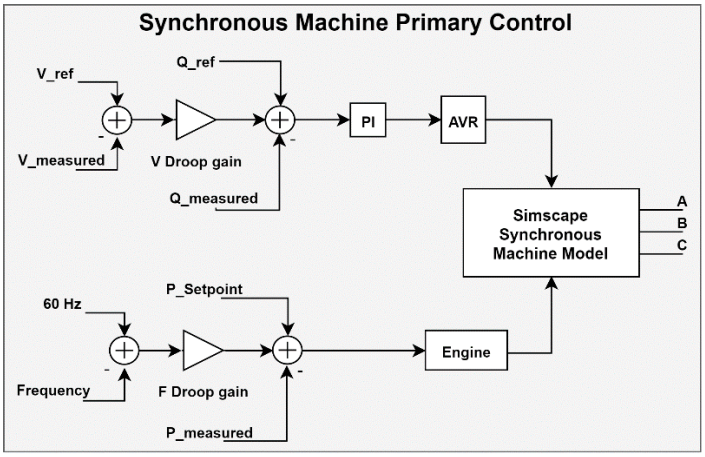

Figure C.1 Synchronous machine primary control schematic

C.3 PV System Model

The PV model expects solar irradiance and reactive power setpoint as inputs. Figure C.2 below is a block diagram of the solar primary control model. Solar output power is based on an MPPT emulator that takes in solar irradiance and outputs solar power setpoint that feeds a current-controlled current source. The current source is parallel with a capacitor that feeds the universal bridge. The universal bridge modeling is based on an average-model converter for efficient computation. A converter control system is used to control the switching of the universal bridge. The converter has a VDC regulator and a VAR regulator.

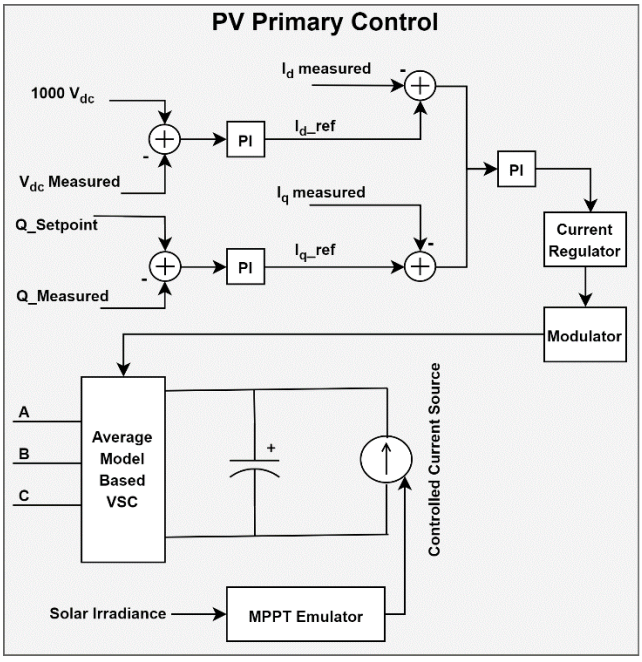

Figure C.2 PV primary control schematic

C.4 Wind Farm Model

The wind farm model expects wind speed and reactive power setpoint. The wind model is based on Simscape average doubly-fed induction generator (DFIG) model. It consists of a wound-rotor induction generator, followed by AC-DC-AC converters stack. Please refer to [76] for further modeling details.

C.5 Battery Energy Storage System Model

The battery energy storage system (ESS) model expects active and reactive power setpoints to operate. Figure C.3 shows the primary control of the ESS. The battery model is based on a

lithium-ion battery Simulink model. The battery feeds an average model-based VSC. The converter control system consists of a current regulator, voltage regulator, and a modulator. The current regulator is fed by a frequency-power control-based I_d reference. The voltage regulator is fed by a reactive power control based I_q reference.

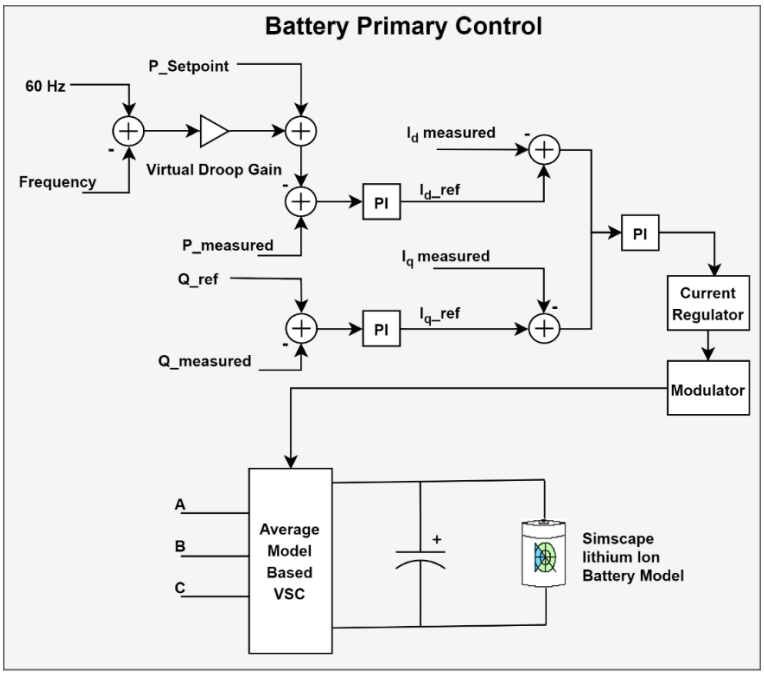

Figure C.3 Energy storage primary control schematic

C.6 Load Model

The 3-phase load is modeled as three single-phase current sources. Each current source is controlled by a sinusoid of which magnitude and phase need to be specified. The magnitude of the sine wave is that of the peak current and is calculated as $I = \frac{S}{V}$, where S is the apparent power, and V is the phase-ground voltage measurement of the bus connected to the current source. On the other hand, the sine wave phase is calculated as the bus voltage phase, measured using a PLL, minus the cosine inverse of the designed load power factor.

There are two advantages to adopting this load model. First, this model allows the user to dynamically change the loading magnitude of each phase during the simulation. Second, since the control of each load phase is separate, it allows the modeling of unbalanced loads, which is the case with the benchmark model.

Appendix D

Reliability Study Methodology

The reliability assessment method used in the thesis is based on Monte Carlo simulations as explained in [71] and is readapted from [53]. The state duration sampling approach is adopted. Since the failure of a controller and its repair time is stochastic, the state transition of each controller is chronologically sampled from an exponential distribution function. An exponential distribution is used as it makes the most sense when sampling failure of components; hence, the more time passes the more likely a component would fail. The state for a local controller i is modeled as follows:

$$S_i = \begin{cases} 1 & Up\ state \\ 0 & down\ state \end{cases} \tag{35}$$

Next, the state transition and duration are represented by two values, Mean Time to Fail (MTTF) and Mean Time to Repair (MTTR), whereas the sampling is done as follows:

$$TTF = -MTTF \ln (U_F) \tag{36}$$

$$TTR = -MTTR \ln (U_R) \tag{37}$$

TTF is the time to fail. TTR is the time to repair. U_F and U_R are random numbers from a uniform distribution between [0, 1]. TTF and TTR are repeatedly sampled throughout the length of the simulation. Specifically, once a controller is "repaired", another controller with a different TTF and TTR replaces it. The idea is that if the simulation runs enough times, the metrics of interest for the control system would converge to values that can be used for comparison.

References

[1] M. F. A. Jajeh, S. Q. Ali, G. Joos, and I. Novickij, "Islanding of a Microgrid Using a Distributed Multi-agent Control System," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, 29 Sept.-3 Oct. 2019 2019, pp. 6286-6293, doi: 10.1109/ECCE.2019.8912499.

[2] "IEEE Standard for the Specification of Microgrid Controllers," *IEEE Std 2030.7-2017,* pp. 1-43, 2018.

[3] E. Wood. "Puerto Rico Issues Final Microgrid Rules as Hurricane Season Looms." MICROGRID KNOWLEDGE. https://microgridknowledge.com/final-microgrid-rules-puerto-rico/ (accessed.

[4] (2018). *REGULATION ON MICROGRID DEVELOPMENT*.

[5] (2015). *Electric Utility Innovation Toward Vision 2050*.

[6] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine,* vol. 8, no. 1, pp. 18-28, 2010, doi: 10.1109/MPE.2009.934876.

[7] W. C6.22, "Microgrids 1 Engineering, Economics, & Experience," *CIGRE,* 2015.

[8] K. Dubé, "Cutting-Edge Energy Solutions for Lac-Mégantic's New Downtown," ed. Lac-Mégantic: Hydro Quebéc, 2018.

[9] R. Uluski *et al.*, "Microgrid Controller Design, Implementation, and Deployment: A Journey from Conception to Implementation at the Philadelphia Navy Yard," *IEEE Power and Energy Magazine,* vol. 15, no. 4, pp. 50-62, 2017, doi: 10.1109/MPE.2017.2691239.

[10] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids," *IEEE Power and Energy Magazine,* vol. 5, no. 4, pp. 78-94, 2007, doi: 10.1109/MPAE.2007.376583.

[11] "IEEE Standard for the Testing of Microgrid Controllers," *IEEE Std 2030.8-2018,* pp. 1-42, 2018, doi: 10.1109/IEEESTD.2018.8444947.

[12] (2010). *Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid*. [Online] Available: https://www.hsdl.org/?view&did=18739

[13] M. Yazdanian and A. Mehrizi-Sani, "Distributed Control Techniques in Microgrids," *IEEE Transactions on Smart Grid,* vol. 5, no. 6, pp. 2901-2909, 2014, doi: 10.1109/TSG.2014.2337838.

[14] F. Nejabatkhah and Y. W. Li, "Overview of Power Management Strategies of Hybrid AC/DC Microgrid," *IEEE Transactions on Power Electronics,* vol. 30, no. 12, pp. 7072-7089, 2015, doi: 10.1109/TPEL.2014.2384999.

[15] M. Shahidehpour, Z. Li, S. Bahramirad, Z. Li, and W. Tian, "Networked Microgrids: Exploring the Possibilities of the IIT-Bronzeville Grid," *IEEE Power and Energy Magazine,* vol. 15, no. 4, pp. 63-71, 2017, doi: 10.1109/MPE.2017.2688599.

[16] A. Rojas and T. Rousan, "Microgrid Control Strategy: Derived from Stakeholder Requirements Analysis," *IEEE Power and Energy Magazine,* vol. 15, no. 4, pp. 72-79, 2017, doi: 10.1109/MPE.2017.2690520.

[17] D. E. Olivares *et al.*, "Trends in Microgrid Control," *IEEE Transactions on Smart Grid,* vol. 5, no. 4, pp. 1905-1919, 2014, doi: 10.1109/TSG.2013.2295514.

[18] Z. Cheng, J. Duan, and M. Chow, "To Centralize or to Distribute: That Is the Question: A Comparison of Advanced Microgrid Management Systems," *IEEE Industrial Electronics Magazine,* vol. 12, no. 1, pp. 6-24, 2018, doi: 10.1109/MIE.2018.2789926.

[19] R. H. Lasseter, "Certs Microgrid," in *2007 IEEE International Conference on System of Systems Engineering*, 16-18 April 2007 2007, pp. 1-5, doi: 10.1109/SYSOSE.2007.4304248.

[20] H. E. Joseph *et al.*, "The CERTS Microgrid Concept, as Demonstrated at the CERTS/AEP Microgrid Test Bed," 2018.

[21] J. A. P. Lopes, C. L. Moreira, and A. G. Madureira, "Defining control strategies for MicroGrids islanded operation," *IEEE Transactions on Power Systems,* vol. 21, no. 2, pp. 916-924, 2006, doi: 10.1109/TPWRS.2006.873018.

[22] Z. Zhang and M. Chow, "Convergence Analysis of the Incremental Cost Consensus Algorithm Under Different Communication Network Topologies in a Smart Grid," *IEEE Transactions on Power Systems,* vol. 27, no. 4, pp. 1761-1768, 2012, doi: 10.1109/TPWRS.2012.2188912.

[23] H. Farhangi and G. z. Joós, *Microgrid planning and design : a concise guide*, Hoboken, New Jersey: Wiley-IEEE Press, 2019. [Online]. Available: https://doi.org/10.1002/9781119453550.

[24] A. Gomez-Exposito, A. Conejo, and C. Canizares, *Electric Energy Systems Analysis and Operation*, Second ed. CRC Press, 2018, p. 6.

[25] M. F. Ryan and M. Chris, "Energy Manager Design for Microgrids," CERTS, 2005.

[26] Q. Jiang, M. Xue, and G. Geng, "Energy Management of Microgrid in Grid-Connected and Stand-Alone Modes," *IEEE Transactions on Power Systems,* vol. 28, no. 3, pp. 3380-3389, 2013, doi: 10.1109/TPWRS.2013.2244104.

[27] M. Ross, C. Abbey, F. Bouffard, and G. Jos, "Multiobjective Optimization Dispatch for Microgrids With a High Penetration of Renewable Generation," *IEEE Transactions on Sustainable Energy,* vol. 6, no. 4, pp. 1306-1314, 2015, doi: 10.1109/TSTE.2015.2428676.

[28] C. M. Colson, M. H. Nehrir, and S. A. Pourmousavi, "Towards real-time microgrid power management using computational intelligence methods," in *IEEE PES General Meeting*, 25-29 July 2010 2010, pp. 1-8, doi: 10.1109/PES.2010.5588053.

[29] C. Sun, J. N. Paquin, F. A. Jajeh, G. Joos, and F. Bouffard, "Implementation and CHIL Testing of a Microgrid Control System," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 23-27 Sept. 2018 2018, pp. 2073-2080, doi: 10.1109/ECCE.2018.8557761.

[30] D. E. Olivares, C. A. Cañizares, and M. Kazerani, "A Centralized Energy Management System for Isolated Microgrids," *IEEE Transactions on Smart Grid,* vol. 5, no. 4, pp. 1864-1875, 2014, doi: 10.1109/TSG.2013.2294187.

[31] Q. Fu, A. Nasiri, V. Bhavaraju, A. Solanki, T. Abdallah, and D. C. Yu, "Transition Management of Microgrids With High Penetration of Renewable Energy," *IEEE Transactions on Smart Grid,* vol. 5, no. 2, pp. 539-549, 2014, doi: 10.1109/TSG.2013.2286952.

[32] G. Binetti, A. Davoudi, F. L. Lewis, D. Naso, and B. Turchiano, "Distributed Consensus-Based Economic Dispatch With Transmission Losses," *IEEE Transactions on Power Systems,* vol. 29, no. 4, pp. 1711-1720, 2014, doi: 10.1109/TPWRS.2014.2299436.

[33] G. Hug, S. Kar, and C. Wu, "Consensus + Innovations Approach for Distributed Multiagent Coordination in a Microgrid," *IEEE Transactions on Smart Grid,* vol. 6, no. 4, pp. 1893-1903, 2015, doi: 10.1109/TSG.2015.2409053.

[34] D. K. Molzahn *et al.*, "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems," *IEEE Transactions on Smart Grid,* vol. 8, no. 6, pp. 2941-2962, 2017, doi: 10.1109/TSG.2017.2720471.

[35] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE,* vol. 95, no. 1, pp. 215-233, 2007, doi: 10.1109/JPROC.2006.887293.

[36] A. Bidram, A. Davoudi, F. L. Lewis, and J. M. Guerrero, "Distributed Cooperative Secondary Control of Microgrids Using Feedback Linearization," *IEEE Transactions on Power Systems,* vol. 28, no. 3, pp. 3462-3470, 2013, doi: 10.1109/TPWRS.2013.2247071.

[37] S. Yang, S. Tan, and J. Xu, "Consensus Based Approach for Economic Dispatch Problem in a Smart Grid," *IEEE Transactions on Power Systems,* vol. 28, no. 4, pp. 4416-4426, 2013, doi: 10.1109/TPWRS.2013.2271640.

[38] Y. Xu and Z. Li, "Distributed Optimal Resource Management Based on the Consensus Algorithm in a Microgrid," *IEEE Transactions on Industrial Electronics,* vol. 62, no. 4, pp. 2584-2592, 2015, doi: 10.1109/TIE.2014.2356171.

[39] Z. Wang, W. Wu, and B. Zhang, "A Fully Distributed Power Dispatch Method for Fast Frequency Recovery and Minimal Generation Cost in Autonomous Microgrids," *IEEE Transactions on Smart Grid,* vol. 7, no. 1, pp. 19-31, 2016, doi: 10.1109/TSG.2015.2493638.

[40] L. P, J. Zhao, J. Yao, and S. Yang, "A Decentralized Approach for Frequency Control and Economic Dispatch in Smart Grids," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 7, no. 3, pp. 447-458, 2017, doi: 10.1109/JETCAS.2017.2708900.

[41] Y. Xu, W. Liu, and J. Gong, "Stable Multi-Agent-Based Load Shedding Algorithm for Power Systems," *IEEE Transactions on Power Systems,* vol. 26, no. 4, pp. 2006-2014, 2011, doi: 10.1109/TPWRS.2011.2120631.

[42] M. R. V. Moghadam, R. T. B. Ma, and R. Zhang, "Distributed Frequency Control in Smart Grids via Randomized Demand Response," *IEEE Transactions on Smart Grid,* vol. 5, no. 6, pp. 2798-2809, 2014, doi: 10.1109/TSG.2014.2316913.

[43] D. Shi *et al.*, "A Distributed Cooperative Control Framework for Synchronized Reconnection of a Multi-Bus Microgrid," *IEEE Transactions on Smart Grid,* pp. 1-1, 2018, doi: 10.1109/TSG.2017.2717806.

[44] W. Xie, C. Gu, Y. Zhang, W. Liu, W. Liu, and W. Gu, "Distributed Cooperative Droop Control for Seamless Islanding of an Autonomous Microgrid," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 20-22 Oct. 2018 2018, pp. 1-4, doi: 10.1109/EI2.2018.8582351.

[45] T. Logenthiran, R. T. Naayagi, W. L. Woo, V. Phan, and K. Abidi, "Intelligent Control System for Microgrids Using Multiagent System," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 3, no. 4, pp. 1036-1045, 2015, doi: 10.1109/JESTPE.2015.2443187.

[46] A. Beck, *Introduction to nonlinear optimization : theory, algorithms, and applications with MATLAB* (MOS-SIAM series on optimization). Philadelphia: Society for Industrial and Applied Mathematics (in English), 2014.

[47] S. D. J. McArthur *et al.*, "Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems,* vol. 22, no. 4, pp. 1743-1752, 2007, doi: 10.1109/TPWRS.2007.908471.

[48] A. Gibbons, *Algorithmic graph theory*. Cambridge [Cambridgeshire] ;: Cambridge University Press (in English), 1985.

[49] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control,* vol. 54, no. 1, pp. 48-61, 2009, doi: 10.1109/TAC.2008.2009515.

[50] Y. Xu and W. Liu, "Novel Multiagent Based Load Restoration Algorithm for Microgrids," *IEEE Transactions on Smart Grid,* vol. 2, no. 1, pp. 152-161, 2011, doi: 10.1109/TSG.2010.2099675.

[51] J. Schiffer, T. Seel, J. Raisch, and T. Sezi, "Voltage Stability and Reactive Power Sharing in Inverter-Based Microgrids With Consensus-Based Distributed Voltage Control," *IEEE Transactions on Control Systems Technology,* vol. 24, no. 1, pp. 96-109, 2016, doi: 10.1109/TCST.2015.2420622.

[52] "IEEE Guide for Electric Power Distribution Reliability Indices," *IEEE Std 1366-2012 (Revision of IEEE Std 1366-2003),* pp. 1-43, 2012, doi: 10.1109/IEEESTD.2012.6209381.

[53] Z. Cheng, J. Duan, and M. Chow, "Reliability assessment and comparison between centralized and distributed energy management system in islanding microgrid," in *2017 North American Power Symposium (NAPS)*, 17-19 Sept. 2017 2017, pp. 1-6, doi: 10.1109/NAPS.2017.8107366.

[54] ABB. Microgrid Controller 600

[55] D. E. Olivares, C. A. Cañizares, and M. Kazerani, "A centralized optimal energy management system for microgrids," in *2011 IEEE Power and Energy Society General Meeting*, 24-28 July 2011 2011, pp. 1-6, doi: 10.1109/PES.2011.6039527.

[56] "IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces," *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003),* pp. 1-138, 2018, doi: 10.1109/IEEESTD.2018.8332112.

[57] A. S. Vijay, S. Doolla, and M. C. Chandorkar, "Real-Time Testing Approaches for Microgrids," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 5, no. 3, pp. 1356-1376, 2017, doi: 10.1109/JESTPE.2017.2695486.

[58] F. Huerta, J. K. Gruber, M. Prodanovic, and P. Matatagui, "Power-hardware-in-the-loop test beds: evaluation tools for grid integration of distributed energy resources," *IEEE Industry Applications Magazine,* vol. 22, no. 2, pp. 18-26, 2016, doi: 10.1109/MIAS.2015.2459091.

[59] S. Buso and T. Caldognetto, "Rapid Prototyping of Digital Controllers for Microgrid Inverters," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 3, no. 2, pp. 440-450, 2015, doi: 10.1109/JESTPE.2014.2327064.

[60] M. H. Cintuglu, T. Youssef, and O. A. Mohammed, "Development and Application of a Real-Time Testbed for Multiagent System Interoperability: A Case Study on Hierarchical Microgrid Control," *IEEE Transactions on Smart Grid,* vol. 9, no. 3, pp. 1759-1768, 2018, doi: 10.1109/TSG.2016.2599265.

[61] V. Nasirian, Q. Shafiee, J. M. Guerrero, F. L. Lewis, and A. Davoudi, "Droop-Free Distributed Control for AC Microgrids," *IEEE Transactions on Power Electronics,* vol. 31, no. 2, pp. 1600-1617, 2016, doi: 10.1109/TPEL.2015.2414457.

[62] C. Yoo, W. Choi, I. Chung, D. Won, S. Hong, and B. Jang, "Hardware-in-the-loop simulation of DC microgrid with Multi-Agent System for emergency demand response," in *2012 IEEE Power and Energy Society General Meeting*, 22-26 July 2012 2012, pp. 1-6, doi: 10.1109/PESGM.2012.6345678.

[63] J. Bélanger, P. Venne, and J.-N. Paquin, "The what, where, and why of real-time simulation," *Planet RT,* pp. 37-49, 01/01 2010.
[64] C. Dufour and J. Bélanger, "On the Use of Real-Time Simulation Technology in Smart Grid Research and Development," *IEEE Transactions on Industry Applications,* vol. 50, no. 6, pp. 3963-3970, 2014, doi: 10.1109/TIA.2014.2315507.
[65] C. T. F. C6.04.02, "Benchmark systems for network integration fo renewable and distributed energy resources," *CIGRE Technical Brochure 575,* 2014.
[66] "Approximate Diesel Fuel Consumption Chart." Generator Source. https://www.dieselserviceandsupply.com/Diesel_Fuel_Consumption.aspx (accessed.
[67] A. S. Debs, *Modern Power Systems Control and Operation*. Boston, MA: Springer US (in English), 1988.
[68] S. M. Blair, F. Coffele, C. D. Booth, and G. M. Burt, "An Open Platform for Rapid-Prototyping Protection and Control Schemes With IEC 61850," *IEEE Transactions on Power Delivery,* vol. 28, no. 2, pp. 1103-1110, 2013, doi: 10.1109/TPWRD.2012.2231099.
[69] "IEEE Guide for the Application of Protective Relays Used for Abnormal Frequency Load Shedding and Restoration," *IEEE Std C37.117-2007,* pp. 1-55, 2007, doi: 10.1109/IEEESTD.2007.4299516.
[70] MathWorks, "PLL (3ph)." [Online]. Available: https://www.mathworks.com/help/physmod/sps/powersys/ref/pll3ph.html?searchHighlight=power_pll&s_tid=doc_srchtitle#responsive_offcanvas
[71] Y. G. Hegazy, M. M. A. Salama, and A. Y. Chikhani, "Adequacy assessment of distributed generation systems using Monte Carlo Simulation," *IEEE Transactions on Power Systems,* vol. 18, no. 1, pp. 48-52, 2003, doi: 10.1109/TPWRS.2002.807044.
[72] B. Buchholz and B. Buchholz, "Advanced planning and operation of dispersed generation ensuring power quality, security and efficiency in distribution systems," (in eng), *CIGRE 2004 Session,* 2004.
[73] K. Strunz, "Developing benchmark models for studying the integration of distributed energy resources," in *2006 IEEE Power Engineering Society General Meeting*, 18-22 June 2006 2006, p. 2 pp., doi: 10.1109/PES.2006.1709568.
[74] K. Rudion, A. Orths, Z. A. Styczynski, and K. Strunz, "Design of benchmark of medium voltage distribution network for investigation of DG integration," in *2006 IEEE Power Engineering Society General Meeting*, 18-22 June 2006 2006, p. 6 pp., doi: 10.1109/PES.2006.1709447.
[75] "Weather Dashboard for Montéal." https://montréal.weatherstats.ca/ (accessed.
[76] R. Gagnon. "Wind Farm - DFIG Average Model." Hydro-Quebec. https://www.mathworks.com/help/physmod/sps/examples/wind-farm-dfig-average-model.html (accessed.