Distributed Real-Time Control Schemes for Microgrids Considering Uncertain Renewable Sources With Provisions for Resilient Operation

MARCOS EDUARDO CRUZ-VICTORIO

Distributed Real-Time Control Schemes for Microgrids Considering Uncertain Renewable Sources With Provisions for Resilient Operation

Marcos Eduardo Cruz Victorio

A thesis presented for the degree of
Doctor of Philosophy

Department of Engineering
The University of Durham
United Kingdom

April 2022

Distributed Real-Time Control Schemes for Microgrids Considering Uncertain Renewable Sources With Provisions for Resilient Operation

Marcos Eduardo Cruz Victorio

Abstract

This thesis presents the integration and improvement of different technologies for optimal operation of microgrids, where the objective is to minimise cost of supply and maximise their renewable hosting capacity in a reliable and seamless manner. To this end, this work has proposed a hierarchical control framework for coordinating energy transactions between different stakeholders within the microgrid, and different microgrid clusters with distributed renewable resources integrated, in an economical and reliable manner.

The stability of the proposed primary control layer, and corresponding voltage bus, has been investigated using Lyapunov's direct method to ensure it remains stable under all operating conditions. Meanwhile, the resiliency of the underlying communication network to maintain a fully connected network for the distributed control nodes has been verified mathematically using Graph theory by ensuring the underlying Laplacian matrix for the network always remains singular and without repeated zero eigenvalues, even following failures in any one individual communication node. This is done to ensure that the proposed control system is capable of reliably maintaining electrical service under normal and faulty conditions.

The hierarchical control framework relies on implementation of Artificial Intelligence (AI)-based techniques, mainly using Multi Agent Systems implemented within the Java Agent Development Framework environment to simulate a distributed control architecture for achieving real-time control of multiple resources within the microgrid (and clusters of microgrids). Meanwhile, using Machine Learning (ML), accurate forecast models have been developed for both electricity price forecast and wind speed (for distributed renewable resources) within the control architecture for optimising energy transactions between relevant stakeholders within the microgrid.

To verify the accuracy and compare the results of ML-based forecast models, this thesis proposes the use of statistical methods to ensure any differences between different forecast models are due to the model structure rather than randomness in the data. This comparison is carried out in addition to usual statistical comparison of different forecast models in terms of total absolute error, squared error and total cost.

At the hardware level, the hierarchical control system has been implemented using a connected network of Raspberry Pi computers acting as individual distributed control nodes with the microgrid simulated in a real-time simulator environment realised by an OPAL-RT (series OP5700) real-time simulator to test the viability of the control system to respond to a microgrid cluster system in real-time.

Supervisors: Dr. Behzad Kazemtabrizi and Dr. Mahmoud Shahbazi

Acknowledgements

Le doy las gracias a mi familia que esta lejos de mi, pero que volveremos a vernos pronto, por su apoyo en esta aventura.

I would like to express my gratitude to my supervisors Dr. Behzad Kazemtabrizi and Dr. Mahmoud Shahbazi for their guidance in this journey during my Masters and my PhD. It has been an honour to work under your leadership. I hope we can work together again in the future.

Finally, I would like to thank the Mexican government for their support that made this project possible.

This work is supported by the CONACYT-SENER energy sustainability programme with scholarship number 447877.

I would like to thank Prof Hongjian Sun from Durham University Department of Engineering and his group for helping with the hardware support for this paper. The hardware was acquired under EU-TESTBED grant number 734325.

Contents

Declaration

The work in this thesis is based on research carried out at the Department of Engineering, University of Durham, England. No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is the sole work of the author unless referenced to the contrary in the text.

Some of the work presented in this thesis has been published in journals and conference proceedings - the relevant publications are listed below.

Publications

Part of this thesis has been published in the following conference papers:

Marcos Eduardo Cruz Victorio, Behzad Kazemtabrizi, Mahmoud Shahbazi, "Real-time Cost Optimisation for Power Management in Microgrids Using Multi-Agent Control" in 2019 9th International Conference on Power and Energy Systems (ICPES)

Marcos Eduardo Cruz Victorio, Behzad Kazemtabrizi, Mahmoud Shahbazi, "Distributed Real-Time Power Management in Microgrids Using Multi-Agent Control with Provisions for Fault-Tolerance" in 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)

Marcos Eduardo Cruz Victorio, Behzad Kazemtabrizi, Mahmoud Shahbazi, "Price Forecast Methodologies Comparison for Microgrid Control with Multi-Agent Systems" in 2021 IEEE Madrid PowerTech

Likewise, the following journal publications are under review:

Marcos Eduardo Cruz Victorio, Behzad Kazemtabrizi, Mahmoud Shahbazi, "Statistical Evaluation of Wind Speed Forecast Models for Microgrid Distributed Control"

Marcos Eduardo Cruz Victorio, Behzad Kazemtabrizi, Mahmoud Shahbazi, "Real-time Resilient Microgrid Power Management based on Multi-Agent Systems with Price Forecast"

List of Figures

List of Tables

Nomenclature

$\mathcal{T}$	*abc* to *dq*0 reference frame transform
S_S	Sending bus Apparent Power
I_S	Sending bus current
V_S	Sending bus voltage
V_R	Receiving bus voltage
P_S	Sending bus Active Power
Q_S	Sending bus Reactive Power
V_{out}	Voltage measurement for the primary controller's outer loop
I_{out}	Current measurement for the primary controller's outer loop
P_{out}	Calculated active power for the primary controller's outer loop
Q_{out}	Calculated reactive power for the primary controller's outer loop
P_{ref}	Active power reference for the primary controller
Q_{ref}	Reactive power reference for the primary controller
$\mathcal{J}$	Set of microgrid's generation sources
j	Microgrid's source index
c_j	Cost function of source j
P_j	Power from source j
i	Hour index for optimisation
$\mathcal{I}$	Set of time steps for price optimisation
N	Total period for optimisation

$P_{j_{min}}$	Minimum power from source j
$P_{j_{max}}$	Maximum power from source j
SOC	State of Charge
SOC_{min}	Minimum State of Charge
SOC_{max}	Maximum State of Charge
P_L	Electric Load Power
η	Capacity and round trip efficiency proportionality constant
X	State space vector for Lyapunov stability
D_X	Domain of the state space vector for Lyapunov stability
$\mathbf{f}$	interconnected system function
$\dot{\mathbf{X}}$	interconnected state space
θ_e	Statistical parameters of the grid's electricity price probability density function
Θ	Markov Chain for the electricity price
n_e	number of elements in the Markov Chain
θ_r	Markov Chain random variation
$\mathcal{Z}$	Random variable with uniform distribution between $[0, 1]$
$\mathcal{D}$	Electricity price data
$\overline{\pi}_e$	Estimated mean of the electricity price
σ_e	Estimated standard deviation of the electricity price
p_i	real electricity price at hour i given data $\mathcal{D}$
π_i	electricity price forecast for hour i
w_{am_i}	Weights for the Auto-regression MCMC price model
W_{am}	Vector of weights for the Auto-regression MCMC price model
$\overline{\pi}_a$	Mean MCMC estimated electricity price from same weekday data as π_i
$\overline{\pi}_b$	Mean MCMC estimated electricity price from season data
$\overline{\pi}_c$	Mean MCMC estimated electricity price from entire data

H_k	k-th iteration of the Hessian Matrix estimate for the Quasi-Newton method
k	Iteration index of the Quasi-Newton method
f_{QN}	Quasi-Newton cost function
x_{QN}	Quasi-Newton input variable
g_k	Quasi-Newton optimal direction for gradient descent
ϵ	Quasi-Newton minimum change of the cost function tolerance
D_{ann}	Delay vector of the NARNET
W_{ann}	Neuron weight matrix of the NARNET
B_l	Bias vector of the NARNET at layer l
f_a	Activation function of the ANN
Y_{ws}	Wind speed time series
A_l	Artificial Neural Network layer l output
$\mathcal{V}$	Lyapunov Candidate function
$\mathcal{G}$	Network's Graph representation
$V_{\mathcal{G}}$	set of vertices in a graph
$E_{\mathcal{G}}$	Set of Edges in a graph
$\mathcal{N}_{\mathfrak{i}}$	Number of neighbours of vertex $\mathfrak{i}$
$\mathfrak{i}$	index for row elements in graph theory
$\mathfrak{j}$	index for column elements in graph theory
$\mathcal{L}$	Laplacian matrix of a network
$\mathfrak{D}$	Degree matrix of a network
$\mathfrak{d}_{\mathfrak{ij}}$	Degree matrix elements
$\mathcal{A}$	Adjacency matrix of a network
$\mathfrak{a}$	elements of adjacency matrix of a network
$\mathfrak{n}$	Number of vertices in a graph
$\mathcal{K}$	Set of sub-graphs a network
$\mathfrak{s}$	sub-graph in a network
$\mathfrak{C}$	L-polynomial of a graph

f_{logsig}	Logistic Sigmoid activation function
$f_{sigmoid}$	Sigmoid activation function
f_{ReLU}	ReLU activation function
d	number of elements in delay vector D_{ann}
$\mathfrak{q}$	number of neurons in a hidden layer
$\mathfrak{u}$	size of input for the ANN
F_{WAp}	Weighted average price Forecast model
F_{Np}	NARNET price Forecast model
σ_0	MCMC standard deviation estimation of electricity grid price using same hour data
σ_1	MCMC standard deviation estimation of electricity grid price using same hour and same week day data
μ_0	MCMC electricity price using same hour data
μ_1	MCMC electricity price using same hour and same weekday data
T	Total period of samples for wind speed forecast model optimisation
W_{AR}	Vector of weights for the AR wind speed forecast model
$W_{AR,i}$	elements of vector W_{AR}
d_{AR}	delay size of AR wind speed forecast model
D_{AR}	set of delays used in the AR wind speed forecast model
d_i	index for the D_{AR} set
$Y_{ws}(t)$	samples of wind speed at time t
$\mathbf{e}$	squared errors of forecast
$\mathbf{d}$	differential error function
γ	Autocovariance function
$\mathcal{H}$	lag vector for DM-test
h	lag index
$\mathbf{h}$	number of elements in H
$\mathcal{P}$	MCMC Probability distribution function
α	MCMC acceptance ratio

DM_0	calculated DM-test value
DM_1	calculated DM-test with the Harvey adjustment
$\mathbf{v}_d$	variance estimation for DM-test
$\mathbf{K}$	Degrees of freedom for DM-test
$\mathbf{t}$	integration variable for DM-test
$\mathbf{p}$	calculated p-value for DM-test
P_{wt}	Wind turbine power Output
V_{wt}	Wind speed
V_{min}	Minimum generation wind speed
V_{rated}	Rated wind speed
V_{cut}	Cut-out wind speed
P_{max}	Maximum wind turbine power output
C_p	Coefficient of performance
τ	Period of generation
τ_{max}	Duration of generation period
$\mathbb{E}$	Expected power generation
v_{w_i}	Wind speed realisation

AI	Artificial Intelligence
ML	Machine Learning
DER	Distributed Energy Resource
RES	Renewable Energy Source
PCC	Point of Common Coupling
ADN	Active Distribution Network
TSO	Transmission System Operator
DNO	Distributed Network Operator

DSO	Distributed System Operator
MAS	Multi Agent System
ESS	Energy Storage System
PI	Proportional Integral
SOC	State of Charge
MPC	Model Predictive Control
p-value	probability value
AR	Auto-Regression
ARX	Auto regressive External
MCMC	Markov Chain Monte Carlo
ANN	Artificial Neural Network
NARNET	Non-linear Auto-Regression Network
WA	Weighted Average
PDF	Probability Density Function
ARMA	Auto Regression Moving Average
QN	Quasi-Newton
ReLU	Rectified Linear Unit
AMS	Agent Management system
DF	Directory Facilitator
RMA	Remote Monitoring Agent
ACL	Agent Communication Language

TCP	Transmission Control protocol
HTTP	Hypertext Transfer Protocol
UDP	User Datagram Protocol
MTP	Message Transport Protocol
FIPA	Foundation of Intelligent Physical Agents
HIL	Hardware In the Loop
EPHCC	Electric Power HIL Controls Collaborative
FC	Fuel Cell
MT	Micro Turbine
DM	Diebold-Mariano
NARX	Non-linear Auto-Regression with Exogenous Variables
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
CDF	Cumulative Distribution Function
CI	Confidence Interval
JADE	Java Agent Development framework
GrA	Grid Agent
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
PS	Pattern Search
MSDP	Multi-Stage Decision Programming

RESRA	Renewable Energy Source Regulator Agent
WSFA	Wind speed Forecast Agent

Chapter 1

Introduction

1.1 Motivation

With the continuous global population growth, and the ever increasing of the electricity demand and carbon emissions, different types of energy resources are being integrated into the electric system. In this context, the microgrid has emerged as a viable option to integrate new small-scale electricity generation known as distributed generation. The concept of a microgrid as a building block of the smart grid has appeared in recent years to improve the conventional model of energy transmission in terms of reliability, demand side management and scalability [2].

A microgrid is a system capable of generating or storing its own energy and supplying its own electric demand autonomously, where each internal source is known as a Distributed Energy Resource (DER) [3]. The microgrid's sources use either conventional generation sources or Renewable Energy Source (RES), and are connected to the grid in a Point of Common Coupling (PCC).

The study of the microgrid is often divided in two main parts, the physical layer, which includes all the electrical components, and the information layer, which describes the control system that regulates all the components of the physical layer to achieve as many objectives as necessary. This layer may be organised as a central energy management system or a collection of controllers connected by a communication network. Given that organisation is done at different levels and for different time scales, the control system of the microgrid is usually divided in a hierarchy, where the bottom levels are the fastest and follow the references provided by the upper control levels.

In this sense, the optimal operation of the microgrid depends on the objectives pursued by the control system, the configuration and resources available to the microgrid. Finding the solution to the optimisation problems solved in the control system can be done offline or online. In offline optimisation, the references for the control system are calculated in advance to operation, as opposed to online optimisation, where the references are generated continuously during operation. Online optimisation allows the microgrid's control system to respond immediately to signals from the physical layer. This type of control is known as real-time control.

Optimal operation of the microgrid in economic terms is required to promote the utilisation of distributed generation. This can be done in a reliable way with the use of real-time control in a distributed control system, with the implementation of electricity grid price forecast.

Moreover, to increase the efforts to combat climate change, the electricity sector is encouraged to integrate renewable generation, for example, with wind turbines. While there has been an increase of large-scale wind farms applications, the regions suitable for their application is limited. These limits include social, economic or environmental reasons, for which small-scale wind generation in the form of microgrids is needed as well as part of the renewable generation integration.

However, wind energy is inherently variable and such variations are often hard to control. For this reason any reliable power generation scheduling regime for wind-integrated power systems relies on the accuracy of wind speed forecast. This is done in order to maintain an optimum and stable operation which could account for unforeseen wind variations.

This thesis is aimed to contribute to the global effort to combat climate change by the development of a stable, reliable, and optimal real-time control system for the integration of conventional and renewable distributed generation in the form of active distribution networks with the use of artificial intelligence.

1.2 Research Objectives

The main objective of this research is to find methods to streamline the implementation of distributed generation at the medium and low voltage levels, in the distribution circuit, in order to combat climate change by reducing the amount of conventional generation needed and simultaneously making the electricity service cheaper and more reliable.

The main objective will be addressed by the design and development of a microgrid control system with the following characteristics:

- Hierarchical structure to maximise scalability
- Stable power output considering the non-linear response of the circuit
- Free from single points of failure
- Fault tolerance against distributed controller faults and communication system faults
- Optimal power schedule in terms of total cost
- Real-time operation capabilities

To verify that the control system has the characteristics detailed above, the appropriate test cases will be defined and simulated accordingly, presenting the simulation response, and the corresponding analysis in each chapter through the thesis.

1.3 Original Contributions

Given the research objectives outlined and guided by the motivation presented above, the following contributions are presented in this thesis:

- The design and testing of a tertiary and secondary control layer based on Multi-Agent System, suitable for multi-microgrid stochastic multi-objective optimal operation in real-time with renewable energy integrated. Part of these results are presented in [4–6].
- The development, testing and comparison of suitable forecast methods for electricity price and wind speed, using non-linear auto-regressive artificial neural networks, for facilitating distributed real-time control in microgrids. Additionally, a forecast method comparison is presented for statistical validation and computational cost comparison. Part of these results are presented in [7, 8].
- The implementation of a real-time distributed hierarchical control framework together with its stability verified using Lyapunov's stability analysis. The conditions for asymptotic stability are presented for distributed generation source with the proposed primary control layer. This conditions are then added as constraints in the optimisation problem to ensure stability and optimal operation of the microgrid. Conditions to extends the stability analysis to the microgrid and the rest of the control system are discussed as well. Some of these results are accepted for publication in [6].
- The design, implementation and testing of a suitable restoration service to increase the fault-tolerance capabilities of the control system of microgrid clusters and within a single microgrid, verified with the use of graph theory. Some of these results are under review for publication in [6].

Table 1.1: Peer-reviewed Publications of this research.

Research Publications
1. Cruz Victorio, M. E., Kazemtabrizi, B. & Shahbazi, M. (2021), Price Forecast Methodologies Comparison for Microgrid Control with Multi-Agent Systems, 14th IEEE PES Power Tech Conference. Madrid Virtual, IEEE.
2. Cruz Victorio, M. E., Kazemtabrizi, B. & Shahbazi, M. (2020), Distributed Real-Time Power Management in Microgrids using Multi-agent Control with Provisions of Fault Tolerance, 29th IEEE International Symposium on Industrial Electronics. Delft, Netherlands, IEEE, 108-113.
3. Cruz Victorio, M. E., Kazemtabrizi, B. & Shahbazi, M. (2019), Decentralised Real-time Optimisation of Power Management in Microgrids Using Multi-Agent Control, 9th International Conference on Power and Energy Systems. Perth, Australia, IEEE, Piscataway, NJ, 1-6.
4. Cruz Victorio, Marcos Eduardo, Kazemtabrizi, Behzad & Mahmoud, Shahbazi (2022). Statistical Evaluation of Wind Speed Forecast Models for Microgrid Distributed Control. IET Smart Grid

Table 1.2: Presentations regarding this research.

Conference titles and corresponding events
1. Real-time Cost Optimisation for Power Management in Microgrids. RT20. Online event organised by OPAL-RT Technologies in Paris.
2. Design of Distributed Control Systems for Microgrids RT21. Online event organised by OPAL-RT Technologies in Paris.

Parts of this thesis have been published in international conferences and a journal article as described in Table 1.1. Additionally, this research has been presented in the international events described in Table 1.2.

1.4 Thesis outline

The chapters of this thesis are organised as follows:

Chapter 2 presents the general literature background of the methods and techniques applied for the development of the control systems and corresponding analysis. It starts with the general properties of a control system suitable for microgrids, followed by the methods required to achieve stable and optimal operation with the use of forecast methods and Multi Agent Systems. It finishes with the description of the hardware used for real-time simulation to validate the proposed control system.

Chapter 3 presents the stability and resilient operation of the microgrid, given the non-linear response of the electric circuit and its interaction with the primary controllers. The stability of operation is demonstrated with the use of the Lyapunov stability analysis between the primary controllers and the physical layer. A restoration service at the secondary control level is analysed with the use of graph theory to establish an acceptable level of resiliency and fault tolerance for the proposed hierarchical control system. The appropriate test case, simulations and results are presented in terms of stability, electrical regulation and fault-tolerance.

Chapter 4 presents the study of variables required in advance to achieve optimal real-time control in a distributed environment, namely, electricity price and wind speed. As these variables are needed in advance, suitable forecast methods are needed, given limited information and computational resources. Forecast models based on auto-regression and artificial intelligence are selected for their ability to generate the relevant inputs locally. Test cases are presented for electricity price and wind speed, analysing the performance of the models in terms of error and statistical significance difference.

Chapter 5 presents the proposed microgrid hierarchical control system and its performance in terms of the global objectives of a single microgrids, namely, total cost and renewable generation maximisation, analysed by the relevant simulated test cases. Additionally, the tertiary control layer is presented for microgrid-cluster operation. Simulation results from a microgrid-cluster test case that integrates the entire control hierarchy is used to validate the multi-objective capabilities of the proposed microgrid distributed control.

Finally, Chapter 6 concludes the thesis and explores the possible future work.

Chapter 2

Background

This chapter serves as a literature review of the necessary techniques, methods, algorithms and technologies in the design of a control system capable of meeting all the requirements set in the previous chapter.

In this chapter, the nature of distributed generation and its advantages and disadvantages are discussed first [9], before presenting the hierarchical control structure and paradigms of such control system [10–13]. This is followed by a review of the reliability and stability methods employed to guarantee the microgrid's ability to maintain power supply [14, 15], followed by the forecast method required for optimum operation of and power management within the microgrid [16–18].

To this end, suitable forecast methods will be discussed [18]. Furthermore, in this chapter, the underlying processes for realising distributed control the microgrids relying Artificial Intelligence units, otherwise known as *agents* will also be discussed in this chapter [19–21]. The introduction to such agents will be presented and the chapter ends with the discussion of the technologies employed to simulate and test the integration of all the different components for realising a hierarchical control framework for optimum and reliable operation of microgrids [22].

2.1 Distributed Generation and Microgrids

In conventional power grids, large central generation generate power in the order of kilovolts. The voltage is then elevated to hundreds of kilovolts and send to the transmission network. Power in the transmission network travels large distances using the tranmission lines. The Transmission System Operator (TSO) is the entity that performs the planning and management of this part of the power system [23, 24].

Power reaches substations that reduce voltage and then goes into the distribution circuit at medium and lower voltage. The consumers are connected to the distribution circuit and are only capable of controlling their own load, in the sense that they choose when to use electric power. It follows that in conventional power systems, the direction of the power flow from central generation to consumers in a vertically oriented unidirectional manner. This type of electric networks are managed by a Distributed Network Operator (DNO) [24, 25].

With the application of modern communication and automation technologies, the components of the distribution network, such as microgrids, become "active", in the sense that they can interact the main grid beyond controlling their own demand, forming an Active Distribution Network (ADN) [9]. As the operator of the circuit must also consider the management of active components and the communication network of the ADN, the operator of the ADN is known as Distributed System Operator (DSO) [25].

The smart grid is the future electrical grid that is capable of two-way communication between generation and consumption by means of smart technology, by implementation of AI into the power management and modern communication technologies. The DERs, microgrids, and ADNs are the building blocks of the smart grid [3, 26, 27].

The implementation of these types of distribution circuits leads to consumers being capable of sending power through the distribution grid, either to the main grid or another consumer with the use of DERs [24]. With this new type of power generation, new control systems for the active components must be developed to maximise the benefits of distributed generation.

2.2 Microgrid Distributed Control

Most control frameworks applied to microgrids are of a hierarchical nature, using up to three levels of control, separated by the different time frames required for the control tasks for microgrid management [28]. The primary control is usually intended for the physical regulation of microgrids [11]. Depending on the type of control, the secondary can be used to regulate the power flow of the microgrid's sources [10, 11], for example, for load sharing, frequency restoration or supply cost minimisation. The tertiary level can be used to regulate the power flow between the main grid and the microgrid or between microgrids. The properties and objectives of each control layer will be explored in the next section of this chapter.

The term "microgrid control" in this thesis is used to represent any control scheme for such applications as optimum generation scheduling coupled with real-time control of distributed generation assets within a microgrid to ensure demand is met at all times, subject to the microgrid's physical and the operational boundaries. The control schemes can be implemented either as centralised, decentralised or in a distributed fashion [28, 29].

In a centralised control framework, the operation of the entire microgrid relies on a single controller, often called central energy management system [10]. In this type of control, all the information of the system must be shared with a central controller that provides all the control signals for the rest of the system, which limits the privacy of users of this type of control, compared to the other control frameworks [30]. Additionally, this type of control tends to be computationally expensive and leaves the entire primary controllers dependant on a single secondary controller [10], which means that the entire operation of the microgrid is vulnerable to a single point of failure [6].

In decentralised control, there is no communication between the different components of the microgrid, and regulation relies on the electrical state of the physical layer [12], which limits the type of objectives that can be achieved in this type of control. Given that the decentralised control relies on physical measurements of the electrical system, there is no control on the information available between controllers with regards to use of power generation and load. In other words, in this type of control there is no mechanism to manage the amount of information available to with regards to load profile.

To combat these vulnerabilities, distributed secondary control emerges as an alternative for microgrid control [6]. In a distributed secondary control, there are separate control units that coordinate with each other through a separate communication network in the form of a computational cloud to realise the power management objectives. With the use of the cloud, each distributed controller can be designed to manage and separate locally maintained information and information made available to other participants in the distribution network. Distributed control has the advantage of increased privacy for owners of distributed generation and end users compared to the other types of control paradigms [30].

With recent efforts to move towards the ADN, with more DERs connected in the distribution network, control systems also have to "reflect" this new nature. To this end, Multi Agent System (MAS) is proposed as a scalable distributed control method [31–33]. The MAS approach will be further explored in the followings sections. Scalability is the ability of the power system to easily change or upgrade the amount of its components, in the context of microgrids, is the ability to increase or reduce the amount of distributed generation it contains [34].

Distributed control also adds scalability to the structure of the microgrid, as any DER can be added or removed from the microgrid without the need to redesign the

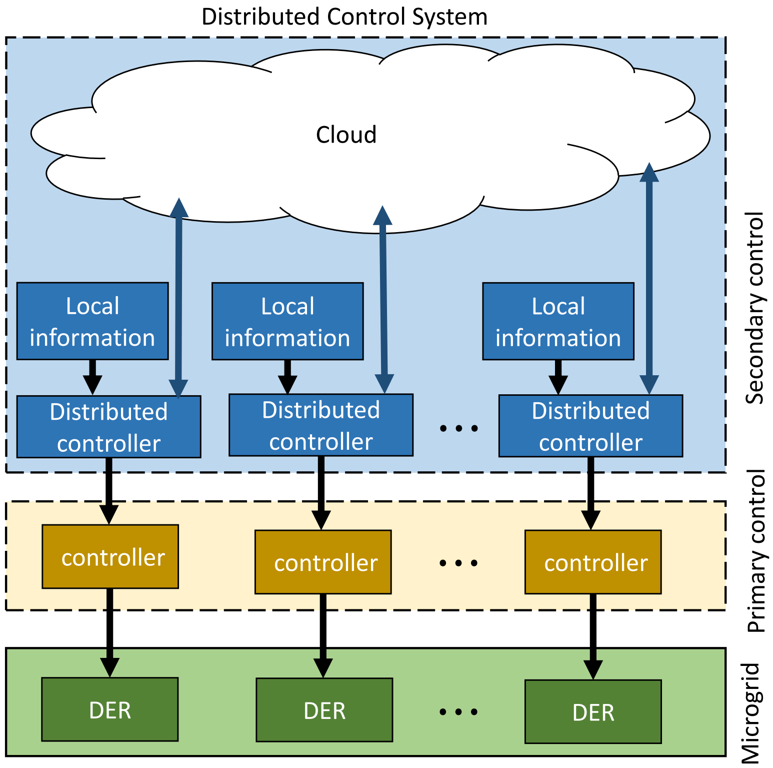


Figure 2.1: Distributed control system model implementation in real-time simulation.

control scheme, as the system adjusts itself [22, 35, 36].

The hierarchical and distributed control applied to microgrid control in this thesis, along with the physical layer, is illustrated in Figure 2.1. Distributed control is selected as the control paradigm of the secondary control as it follows the nature of distributed generation.

The organisation, benefits and challenges of microgrid hierarchical distributed control will be detailed in the following sections. In general terms, distributed control is considered ideal, as it offers a good trade-off between communication requirements, reliability, scalability and independence of the sources within the microgrid, or across different microgrids [37].

2.3 Hierarchical Control for Microgrids

Some of the most pressing challenges of research in microgrid control are the improvement of stability, scalability and cost effectiveness of the microgrid's generation resources[38]. A dynamical system is stable if it can remain or oscillate around an equilibrium point or state of motion [39], in the context of power system, stability can be measured on the variations of electrical magnitudes, such as voltage and frequency around a defined range. This chapter introduces a hierarchical microgrid control scheme aimed at addressing all these challenges.

The microgrid control presented in this thesis is organised in a hierarchy, where each level has specific objectives and time response capabilities, with control references for each layer supplied by the immediate preceding top layer. This structure is based on the ISA-95, an international standard for hierarchical organisation between enterprise and control systems, which allows the implementation and organisation of additional grid services [10–13].

Each of these layers will be presented in detail in the following subsections. The control hierarchy is illustrated in Figure 2.2.

The primary control layer directly regulates the voltage of the individual DERs, having one controller per resource, to maintain the power flow required considering stability and voltage regulation limits of the DERs in the distribution circuit [4, 5, 22].

The Secondary control layer is used to plan and coordinate an optimum power schedule for the DERs within each microgrid [10, 11]. Many microgrids have a centralised secondary control to maximise the economic benefits from the use of DERs, by solving the entire microgrid's optimisation problems, for example, supply

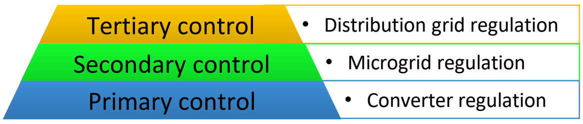


Figure 2.2: Hierarchical Control for microgrids.

cost minimisation, in a centralised way. Although, this kind of control structure tends to be computationally expensive and leaves all the primary controllers dependant on a single secondary controller, which means that the entire operation of the microgrid and privacy of end users [30] are vulnerable to a single point of failure. Because of this, the secondary control layer can be centralised, decentralised or have elements of both [10, 13, 37].

The tertiary control level is responsible for realising control at the distribution system level at the top most level, which includes, offering services between the main grid and the microgrid or other microgrids in a microgrid cluster, forming a hierarchical framework for coordinating and controlling several Microgrids simultaneously [40].

To achieve optimal operation at the secondary and tertiary control levels in real-time control, it is necessary for the microgrids to have information about the future electricity grid prices and weather conditions to generate its power schedule [6, 41]. This is due to start-up cost for controllable DERs, the states of charge for the Energy Storage System (ESS) and stochastic generation for the RES, which make real-time operation, and in turn total cost, of the microgrid dependant on its previous states [41].

Therefore, it is essential that decisions taken at the secondary and tertiary levels are supported by accurate forecasts of relevant variables according to their local objectives [6, 8]. Moreover, it is essential that each source in the microgrid has its own local forecast model, according to its needs.

Using local forecast models in the distributed control architecture also improves the security of local information, as the information itself is not required to be shared directly with the rest of the system. For example, a distributed controller in the secondary control layer may communicate the power references used for its local DER to a primary controller, without the need to inform a central energy management system any of the characteristics of said DER [10].

2.3.1 Primary Control Layer

Most primary controllers use two control loops to regulate the voltage and current of the DER. The primary control of [42] is an example of a control with an inner and outer control loop, where the outer loop sends a reference for the inner loop, making a more stable, accurate and fast control. The most common communication-less decentralised control is the droop control [43], however it cannot simultaneously offer good voltage regulation and power sharing, having to trade-off one for the other [38]. The main difference of the proposed controller for this thesis from conventional droop control is that the proposed controller doesn't rely on the frequency for power regulation. I will be shown in chapter 5 that this allows the primary control to follow the power references from secondary control to achieve cost minimisation. Additionally, this feature of the primary controller avoids the need of under-frequency load shedding schemes, that could result in cascading failures of the power system [44].

2.3.1.1 Proposed Primary Controller

The designed primary controller of each DER, in this thesis, is composed of two Proportional Integral (PI) control loops, an inner loop for voltage control and an outer loop for power control [4, 45]. This type of control allows the regulation of disturbances in the voltage before their effects affect the power regulation, given the difference in the required speed of regulation [45]. The control diagram is ilustrated in Figure 2.3.

As this control layer is designed for distributed control to ensure stability and availability of the electricity service, its differences with other similar approaches will be detailed in chapter 3.

In this thesis, it is assumed that all DERs are converter interfaced. The inner loop regulates voltage and phase angle based on the reference provided by the outer loop. The outer loop is used to direct the DER to send or receive a specific amount of

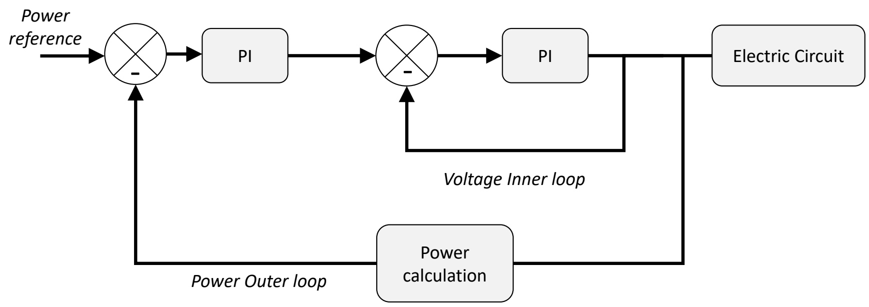


Figure 2.3: Primary cascade controller design diagram.

active and reactive power, given the power references from the secondary control layer [4].

The AC voltage signal used in the inner loop is converted to dq0 reference frame with the Park's Transform [46]. The dq0 reference frame is used to convert the AC signal into 3 constant variables at steady state, which are significantly easier to regulate using PI control.

Park's Transform, as defined by (2.1) and (2.2), is applied using a local clock to provide the $\omega\ t$ reference for PI control regulation of voltage signals, which outputs the signal reference for the converters [4].

$$V_{dq0} = \mathcal{T} V_{abc} \tag{2.1}$$

$$\mathcal{T} = \begin{bmatrix} sin(\omega t) & sin(\omega t - \frac{2\pi}{3}) & sin(\omega t + \frac{2\pi}{3}) \\ cos(\omega t) & cos(\omega t - \frac{2\pi}{3}) & cos(\omega t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.2}$$

The DQ0 transformation enables the determination of the direction of the power flow from a single point of measurement and provide a measurement of instantaneous active and reactive powers which could not have been possible to obtain from a single direct measurement of voltage and current in the time domain [46, 47]. In other words, this is done locally to obtain all the relevant information the electrical

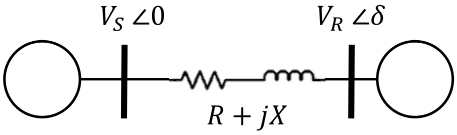


Figure 2.4: Short line model diagram for the power flow equations used for primary control.

state of the DER. The stability of the proposed primary controller is analysed in chapter 3.

The primary control layer follows the equations of power flow between two buses in a short line model based on the lines of the distribution grid, shown in Figure 2.4, described by:

$$S_S = 3V_S I_S^* \tag{2.3}$$

where S_S is the apparent power sent, V_S sending bus voltage and I_S is the sending bus current and $\cdot^*$ indicates the complex conjugate operation. This leads to:

$$P_S + jQ_S = 3V_{S\phi}\Big(\frac{V_{s\phi} - V_{R\phi}\angle\delta}{Z}\Big)^* \tag{2.4}$$

where P_S is the active power sent, Q_S reactive power sent, V_R receiving bus voltage, δ phase angle, R is the resistance of the line, $Z = R + jX$ is the impedance of the line and ϕ represents each phase.

In the case of the distribution grid and in microgrids, the lines are essentially resistive, which means that $R >> X$. This allows the decoupling active and reactive power in 2.5 in voltage amplitude and voltage phase angle control expressed by the following equation:

$$P_S + jQ_S \approx 3(\frac{V_{S\phi}^2 - V_{S\phi}V_{R\phi}cos\delta}{R} + j\frac{V_{S\phi}V_{R\phi}sin\delta}{R}) \tag{2.5}$$

The instantaneous active and reactive powers are calculated using the bus phase current and voltages with (2.6) and (2.7) [47] and are fed back in the outer loop :

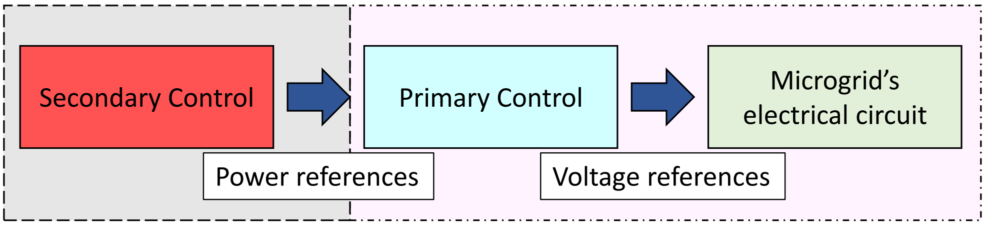


Figure 2.5: Conceptual diagram of the hierarchical control system used in this thesis. The grey region indicates the part of the system analysed with the use of graph theory for control resiliency and the pink region delimits the elements analysed with the use of Lyapunov theory for stability.

$$P_{out} = (v_a i_a + v_b i_b + v_c i_c) \tag{2.6}$$

$$Q_{out} = \frac{1}{\sqrt{3}}[(v_b - v_c)i_a + (v_c - v_a)i_b + (v_a - v_b)i_c] \tag{2.7}$$

where v_a, v_b and v_c are the phase voltages measurements and i_a, i_b and i_c are the phase current measurements, for simplicity, all of these are represented as V_{out} and I_{out} in Figure 2.3.

The primary control level in this thesis has the objective of regulating the voltage and phase angle of each DER while maintaining the frequency constant, following the power references from the secondary control [4, 6] as shown in the blue region in Figure 2.5.

Each primary controller regulates the voltage and frequency response of each DER and ESS to be maintained within the UK standard, of $400 - 6\% + 10\% V_{line}$ and $50 \pm 1\% Hz$.

While the controller can maintain microgrid operation with only voltage references, in hierarchical control, a power schedule supplied as the active and reactive power references, P_{ref} and Q_{ref}, to the input of the primary controllers to achieve the objectives of the secondary control layer [4, 6], which is presented next.

2.3.2 Secondary Control Layer

This control layer realises the power management of the microgrid by regulating the primary control layer.

This control layer mainly optimises the use of the combination of DERs and ESS in the microgrid, for example, for load sharing within the microgrid, or minimising total supply cost of microgrid over one day. However, optimising only for the cost of distributed generation may neglect other aspects such as reliability and scalability, which compromise the operation of the distributed resources and the supply of energy [10] [48], therefore, for long term cost optimisation, these aspects of the microgrid operation must be considered. The secondary control can also be optimised in terms of maximum use of installed renewable generation [49].

Compared to larger power plants, the formulation of the microgrid supply cost minimisation problem can be simplified compared to the former, given the difference in the physical response of small-scale generation. For example, cost minimisation models for grid-connected microgrids can neglect shutdown cost, minimum up and down time, and ramping rates, focusing instead on start up cost and operating cost for DERs and the dynamic behaviour of the State of Charge (SOC)s of the batteries [50]. To minimise costs with the use of batteries, energy arbitrage, which essentially is a buying and selling strategy to take advantage of energy prices and generate profit, has been proposed for the UK, as well as grid balancing for renewables [51].

For the case of distributed control, these simplification of the supply cost minimisation problem can be solved by integrating Model Predictive Control (MPC). MPC, also known as Receding Horizon control, is a control method where the plant is simulated to calculate its state in the future to select the control inputs required to reach such state [52].

MPC requires that all system outputs be measured with the same sample rate, this could be expensive if different parts of the system respond with dynamics at different time scales [53]. More specifically, in MPC schemes, the simulation of the plant is done offline, this means that the response of the system to some given input is calculated in advance, which greatly reduces the amount of calculations necessary to forecast future states of the microgrid sources [54]. The MPC method combined with a forecast method can be used in a secondary distributed control for the optimisation of a components of the microgrid, such that the following objectives are achieved.

2.3.2.1 Microgrid Cost Optimisation

The operation of the secondary layer as a whole has the objective of minimising the total supply cost considering buying and selling from/to the grid, from the perspective of the owners/operators of the DERs and ESS, over some period, for example, one day [6, 55]. To minimise the total supply cost of the entire microgrid the following optimisation problem is solved by the secondary control layer [29, 30, 56]:

$$\min_{P_j} \quad \sum\nolimits_i \sum\nolimits_j c_j((P_j(i)), \forall j \in \mathcal{J} \land \forall i \in \mathcal{I} \tag{2.8a}$$

where c_j is the cost function of each source j of DERs and ESS, $\mathcal{J}$ is the set containing the generation sources. P_j is the active power sent by source j at each hour i and $\mathcal{I}$ is the set containing the time steps.

Given the physical limitations of generators, the optimisation function includes the following constraints [55]:

$$P_{j_{min}} \leq P_j \leq P_{j_{max}} \tag{2.9}$$

$$SOC_{min} \leq SOC \leq SOC_{max} \tag{2.10}$$

$$SOC(i+1) = SOC(i) - \eta P(i) \tag{2.11}$$

$$P_L(i) - \sum\nolimits_j P_j(i) = 0, \forall j \in \mathcal{J}, \forall i \in \mathcal{I} \tag{2.12}$$

As mentioned previously, the solution of the cost optimisation problem defines the power references for the primary controllers. Apart from the cost minimisation, microgrids control can also be used to implement renewable generation optimisation [41], as presented next. The calculation of the optimal power schedule for these objectives is done by a distributed secondary control layer realised by a MAS application.

2.3.2.2 Renewable Generation Optimisation

The use of RES has emerged as a solution to reduce carbon emissions due to the conventional electrical generation. Microgrids can facilitate the integration of these resources for small-scale resources in locations where large-scale renewable integration is limited due to geographical, social or economic reasons [57].

However, an important challenge with maximisation of renewable generation in a microgrid is its stochastic nature, which causes errors in the power schedule for this type of generation[6, 41, 55]. Consequently, a regulation system must be put in place to increase the reliability of the renewable generation.

In this thesis, this is achieved by coupling the renewable generation with a battery, such that a reliable power schedule can be produced with this types of sources to account for the variable nature of the RES generation output [41]. Integrating RES with a suitable battery storage system has two main challenges, first, the battery has limited capacity to compensate for the difference between the scheduled power and the actual power generation of the RES. The second challenge is that batteries have a limited number of cycles of charge and discharge before their performance deteriorates significantly, for which it is important to use the batteries only when it is necessary [55].

These challenges can be mitigated by reducing the error between the scheduled power and the actual power generation [8, 41]. It is therefore critical for the distributed control system to have the most accurate wind speed forecast model (in case of a wind-based RES), to maximise the use of renewable generation and at the same time minimise the use of batteries to increase their lifespan. The battery management to regulate wind-based RES is considered in chapter 5.

2.3.3 Tertiary Control Layer

Tertiary control layer is considered to be the highest control level and is used to coordinate the interaction of several microgrids in a single electrical circuit as a microgrid cluster [10]. This is done by applying smart pricing to the cluster, creating incentives for the use of distributed generation, minimising the load for the main grid [30].

Given that the tertiary control layer coordinates the operation of several microgrid accross the distribution circuit [58], this level of control can operate at a level equivalent to the DSO level [10]. The microgrid coordination achieved by this layer is normally realised by solving a suitable multi-objective optimisation problem [58]. This level of control is presented in chapter 5.

2.4 Control Paradigms

As mentioned in previous sections of this chapter, microgrid hierarchical control can be designed as centralised power management unit, with all the control tasks being realised at a single point that concentrates all the input signals, information, optimisation and generation of the control signals [10]. It can also be designed as decentralised with no communication between generation sources and relying on obeying generalised rules based on the voltage and frequency of the distribution circuit [12], or distributed which contains elements of both centralised and decentralised control architectures [59].

The secondary control layer can be designed to mimic the centralised control schemes of a utility transmission grid in which a single control unit regulates the operation of the entire system and interactions between stakeholders within the system [60]. The secondary control can also be designed as fully decentralised, taking advantage

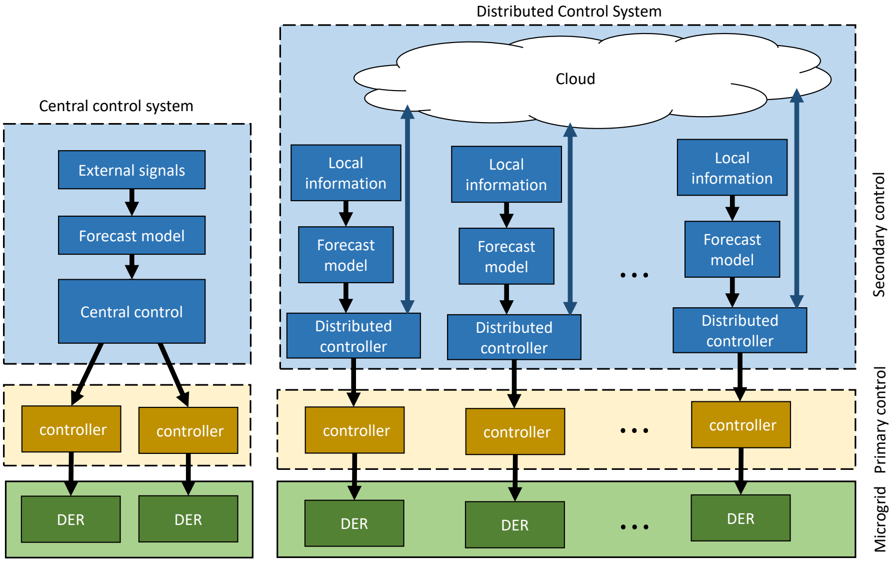


Figure 2.6: Comparison of the elements for Microgrid control hierarchy for centralised and distributed control.

of the already decentralised nature of DERs with their own autonomous controllers, where there is no need for a central overseer and the system is regulated only at the bus level [12]. The third type of secondary control is the distributed control, which combines elements of centralised and decentralised [6, 59]. In microgrid distributed control, each stakeholder, for example a DER, has some level of autonomy and communicates with other entities in the microgrid, such that the network as a whole reaches a global objective, such as minimum cost for DERs [55, 56]. Figure 2.6 illustrates the hierarchy control layer concept for central and distributed control.

The main difference between distributed control and fully decentralised microgrid control is that in the distributed control the controllers are connected in a communication network additional to the electrical network [4]. In the decentralised approach there is no communication between the controllers in the secondary control layer [12]. This is illustred in Figure 2.5.

In the case of a secondary distributed control, the forecast models used to solve the microgrid cost optimisation must have the highest possible accuracy at the least computational cost without requiring external signals to operate [8]. A slow forecast model might increase the delay in the microgrid response, deviating it from the

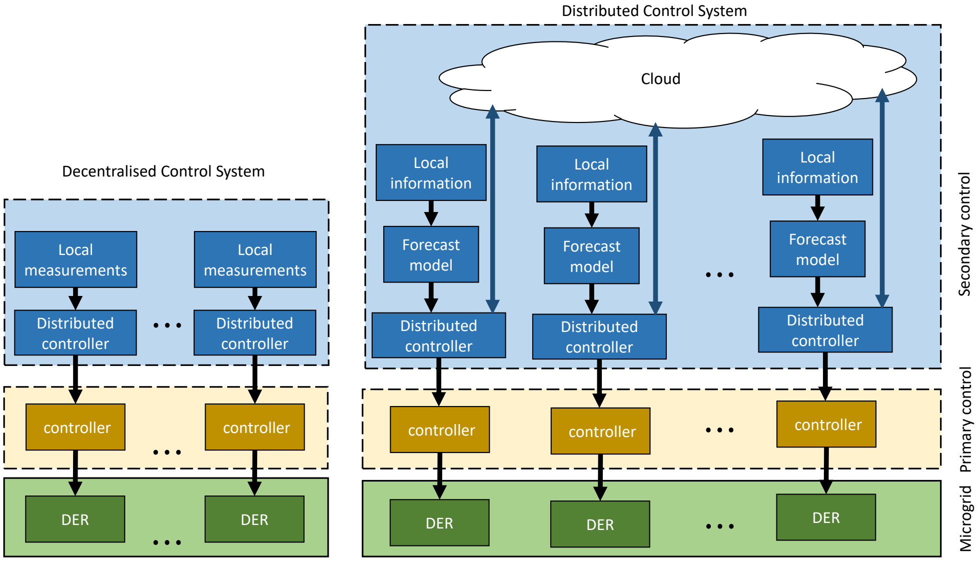


Figure 2.7: Comparison of the Microgrid control hierarchy for decentralised and distributed control.

optimal point of operation, while dependency on external signals would effectively centralise the control to the signal, defeating the purpose of distributed control.

Despite the fact that centralised secondary controls become vulnerable to single points of failure, completely decentralised systems heavily rely on the transmission of information as either voltage or frequency signals, which require the DERs and loads to operate as inertia-based systems, which limits their ability to respond to sudden load changes or nonlinear loads [10]. The distributed control offers a solution that implements the best of both approaches and it is therefore selected as the control paradigm in this thesis.

To minimise the computational requirements of the communication network in a distributed environment, some systems such as the one presented in [61] use an event-triggered communication mechanism instead of periodically sending information between the controllers of the distributed system. Therefore, these systems are vulnerable to cascade a fault in the communication network when a sender node is lost and the receiving nodes wait indefinitely for its message. There is therefore a need for distributed control methods that are capable of maintaining stable and reliable operation under normal and faulty conditions and simultaneously achieve optimal operation [5]. The following section introduces the reliability and stability requirements for the microgrids and the methods employed to analyse them.

2.5 Reliability and Stability

In any system that changes over time, such as any power system, is defined as a dynamic system. A dynamic system is defined by its states X and how they change over time. A dynamic system described by its states X and a function $\dot{X} = f$ has an equilibrium point in X_{eq} if $f(x_{eq}) = 0$. A system is Lyapunov stable if after a disturbance, starting from equilibrium, the states of the system remains in the vicinity of x_{eq}. If the system can become arbitrarily close to x_{eq} then the system is said to be asymptotically stable. If the system is asymptotically stable for x_{eq} from any point in the domain of x, then the system is said to be globally asymptotically stable. If after the disturbance the system does not become close to any equilibrium point, the system is said to be unstable [14, 15].

While the microgrid must enable the optimal power scheduling for cost minimisation, it must also be able to maintain stable operation, load and DER variations notwithstanding [58]. Consequently, the limits of operation of the primary control are defined for asymptotic stability operation, to guarantee that the primary control is stable, using Lyapunov stability analysis.

For the case of non-linear systems, there are two methods based on Lyapunov analysis [14]. In the first method, known as the indirect method, the system is linearised around the equilibrium point, however this method is valid only for stability in the small-signal case. The second method, known as the direct method, is based on verification of Lyapunov's theorems of the equilibrium point in combination with Lyapunov functions. The functions used to verify the stability of a nonlinear system in the direct method are named Lyapunov functions.

Given that the second method models the non-linear properties of the system, the second method is useful to analyse the stability of the system for the large-signal case as well. Similarly, with this method is possible to draw conclusions about the global stability of the systems [14]. Because of these properties, the Lyapunov's second method, or direct, is selected to analyse for the stability of the microgrid in combination with the proposed control scheme. This method will be further explored in chapter 3.

Resilient operation in the context of microgrids refers to the ability of the system withstand or adapt to adverse conditions, either in the physical layer or in the information layer [39]. Reliability, while it can be measured using certain indices such as loss of energy or load expectation [62], ability to withstand extreme conditions [3], in general terms, reliability refers to the ability of the system to operate as expected [48, 63]. Most of research has been focused on the faults in the physical layer as they impose a threat to life and property [64, 65].

Systems that are fault-tolerant to certain type of faults have been developed to combat these threats, taking advantage of distributed control. In [22] the use of graph theory for the study of changing communication networks of MAS applications. This is the case for fault-tolerance in the MAS components and communication links. In [64] multiple controllers called replicas are used for the same resource to maintain the availability of the microgrid's control system due to delays in real-time control, this however increases the number of controllers required for each microgrid's actuator. In [66] the use of graph theory is used in a MAS application in real-time control to show stable operation, however this approach only covered communication interruptions. In [65] a similar approach is used for power line faults in offline simulations.

Distributed control allows a higher level of resilient operation, as faulty controller does not necessary lead to system breakdown [56]. However, there is still a need to verify that distributed control schemes can maintain stable operation in real-time controllers under faulty conditions for any single component of the control system [6].

To guarantee resilient operation of the secondary control in the event of loss of one of the nodes of the communication network of the secondary control, the restoration service [5] is implemented in this thesis. This ability of the proposed control is analysed in terms of graph theory, with the MAS communication network modelled as a connected graph and applied to real-time control scenarios. The mechanisms for resilient microgrid operation of each layer are explained in the following subsections.

It is crucial that interactions between stakeholders within a microgrid remain optimal as they continue to grow in size and complexity [67]. Most importantly, microgrids should be able to continue to facilitate integration of distributed generation systems in line with the global decarbonisation goals of energy systems. To this end, the two main concerns of the distributed generation system are their economic efficiency and their reliability [58]. Suitable control methods must be designed to maximise both the economic efficiency and reliability at the same time. These control methods should be capable of generating the optimal power schedule, considering uncertainties due to electricity price variations as well as occurrence of faults in a reliable and efficient manner [5, 6, 58]. For improved reliability, [68] presents distributed controls that maintain operation in the event of any component outages.

A lot of effort has been put into designing control strategies to optimise the power schedules of microgrids, however, there is a need to verify that such control systems are reliable in terms of stability during normal operation and fault conditions, while at the same time remain optimal [5, 6, 58].

Each layer of the microgrid control design in this thesis is designed to guarantee resilient and stable operation while maintaining optimal cost [5]. The stability of the primary layer is verified with the use of Lyapunov's stability analysis while the secondary layer implements the appropriate price forecast methods based on Auto-Regression (AR) and fault-tolerance mechanisms [6]. The resilience of the secondary layer is verified mathematically using graph theory.

Defining the conditions in which a control system is stable is of critical importance in microgrids because they must be capable to regulate the voltage in low inertia conditions that make it more vulnerable to changes in the internal generation and demand. While the small signal analysis has been used in microgrid stability analysis, the large-signal analysis can properly define the nonlinear transient response of the system and define the stability limits of the system [69].

The use of the Lyapunov method to guarantee the stability of a microgrid controller can be found in control systems such as in [70]. Similarly, in [38] the Lyapunov stability method is applied to a single DER as a general tool of analysis for microgrids of any size for small signals in droop control, used in decentralised control. The Lyapunov analysis is used in this thesis to guarantee stability for large enough disturbances in large enough time scales that may not be captured in individual simulation cases, which could neglect such dynamics otherwise for distributed control. In other words, the stability is guaranteed independently from the simulation model.

For the resilience of the communication network formed by the MAS, the system must remain fully connected even after the loss of one of the nodes [5]. This can be mathematically verified with the use of graph theory, in particular with the use of the characteristic polynomial, also known as L-polynomial, of the connected graph modelling the underlying communication system for the agents within the MAS system [71, 72].

To verify the stability and resiliency of the control system proposed, the power flow of the line model used and the primary control applied is studied to guarantee the power availability in the microgrid.

The analysis is carried out by solving the analytical response of the primary controllers, which is composed of an inner and outer loop. For the inner loop, the study of the transfer function is done to establish its stability. The outer loop is analysed using Lyapunov's second method to determine the conditions for local asymptotic stability, given as the domain of the state variables used. Figure 2.5 shows in pink the area analysed for stability.

The second part of power availability of the microgrid is its resilient operation under normal and under faulty conditions. In the case of distributed control, the resilient operation refers to the ability of the control system to continue operation when one of its components becomes faulty without breakdown of the rest of the system [48, 63]. This can be verified with the use of graph theory for the connectivity of the microgrid's communication network [5, 6], which is analysed in chapter 3. Figure 2.5 shows in grey the area of the control system analysed for resilient operation using graph theory.

Once the design of the distributed control system allows the operation ofs the microgrid even in the event of loss of communication between its components, the microgrid control must also be able to achieve optimal operation based on a power management goal [5, 58].

To achieve optimal operation microgrid control system must be able to realise forecasts about the relevant variables affecting it, in this case grid price, to account for the non-linear behaviour of its components [58, 73]. The next section describes suitable forecast methods for distributed control as they rely only on historical data that can be stored and shared locally in each controller, such that they do not centralise the control system.

2.6 Forecast Methods

To achieve optimal operation at the secondary and tertiary control layers and maximise the economic benefits of distributed generation, it is necessary for microgrids to have information about available generation in the DERs in advance [58, 73]. This is the case due to start-up costs for controllable DERs, the state of charge for the case of ESS and stochastic generation for the case of RES, which make the optimal operation of the microgrid dependant on its previous, current and future states [6].

The use of forecast methods for the optimal operation of microgrids has been reported previously for centralised control, such as in [74] and in [75], where the power generation forecast were realised by an Auto regressive External (ARX) model for one-minute ahead predictions, and in [75] where an ARX model is used as part of a day-ahead electricity price forecast.

However, to maximise reliability of the microgrids, the forecast methods should be compatible with the distributed control paradigm, such that the forecast methods do not introduce single points of failure as in centralised forecast methods [8]. To achieve optimal operation in a distributed environment, auto-regression forecast models are proposed as a suitable solution. Additionally, the fast time response is needed to allow real-time control when interacting with the primary control layer [5–8].

AR models are statistical models, based on linear-regression for time-series forecasting. They do not implement any physical description of the model, but rather, are entirely based on the past values of the series to predict the future values [16, 17]. They are however, for the same reason, suitable for distributed control as forecasting models [18].

One type of such AR models is the persistence model, which is based on the idea that the changes in two consecutive samples of a time series are small enough to obtain a good approximation of the next sample, considering only the previous sample to forecast the next one [76]. For its simplicity, this model serves as a point of comparison with other forecast models [65, 77, 78].

Markov Chain Monte Carlo (MCMC) methods are Bayesian methods used to estimate the parameters that describe a Probability Density Function (PDF) [79]. MCMC methods have been used with distributed generation to estimate solar generation [80], and in [81] for wind speed forecasting. In [75, 82], a combined approach of machine learning with maximum likelihood estimation is utilised to predict day ahead market clearing prices, while in [50] scenarios are predicted to minimise total supply cost for a microgrid. In [83] price prediction is used for optimal load control and is established that there is a high correlation of the hour prices of one day and next day, and prices of the same week day.

Applications of Monte Carlo and Artificial Neural Network (ANN) methods have appeared in the literature, such as in [84], where a centralised energy management system used a deep neural network as a simulation tool to extract samples of a multi microgrid system's response to price signal, with the objective of minimising peak to average ratio power and maximise energy profits for the DSO.

ANNs are among the latest forecast models used as time-series forecasting models [75, 82, 85–87]. They are very powerful forecast tools because of the flexibility of their architecture and training, which allows the use of virtually any time-series data as their input.

ANNs are organised in groups of connected layers of neurons, with a non-linear functions applied to the output of each layer, known as activation functions [8]. In other words, the activation function is the non-linear transformation applied to the data that allows the ANN to model the non-linear properties of the data.

Some of the most common activation functions used in forecast models include the Logistic Sigmoid function, Tangent Hyperbolic function and Rectifier Linear [88, 89].

The Non-linear Auto-Regression Network (NARNET) is a type of ANN which is designed to generate future time steps in a non-linear time-series, without the need of any other external input [90]. With regards to the way of organising the layers of the NARNET, there are three main ways, having a single layer, layers connected in series and layers connected in parallel [82, 87].

The characteristics of the NARNET make it ideal for forecasting in a distributed control setting because the control would not be subject to vulnerabilities caused by dependence of external signals [8].

For each of the forecast requirements needed to achieve optimal operation of the microgrid, the following subsections describe the methods used.

2.6.1 Electric Price Forecast Methods

In this thesis two main price forecast methodologies are tested and compared for UK wholesale electricity price forecast. The transactions realised in future case studies are based on this results, given some adjustment to reflect the prices and costs of the electricity retail market.

The first one is an AR model built with the MCMC method ,which is further optimised with numerical and heuristic solvers, referred in this thesis as Weighted Average (WA) model [5, 17].

Price forecasting with AR models requires some assumptions based on the observed data: Price is considered to be normally distributed for a specific hour [91], and a high correlation exists between an hour price, and the prices for the same hour for the previous day and seven days ago as the demand tends to follow these patterns [83].

MCMC models have been applied in renewable generation [80, 81], as well for modelling load uncertainties in microgrid optimisation [92]. The Metropolis-Hasting [93] is used for its easy implementation for the MCMC.

The second family of methods are based on the NARNET model mentioned previously, which are computationally expensive for training, although execution only requires a small amount of computational resources [8]. The reason is that in order to find the optimal model and to take into account randomness in the training process, many iterations of the training need to be done.

Examples of ANNs that have been used for time series forecasting are found in [75, 82, 87]. For the case of intra-day price prediction, the entire day values can be considered as a single vector or wavelet [75] to model features that the ANN can learn.

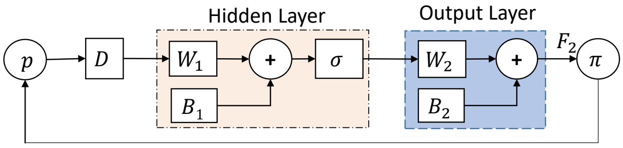


Figure 2.8: NARNET basic architecture. The price variables are used to indicate that the input can be the historical actual price p and the output π is the forecast.

Both of the price forecast methodologies are described in the following subsections.

2.6.1.1 Weighted Average Model

This method is a combination of an auto-regression model, and statistical parameters added via the MCMC. The contribution of each term in this model is defined as a weighted average. Each of the weights is obtained by optimising the model in terms of total error, using the Quasi-Newton (QN) Method. The QN algorithm is described in appendix A.

MCMC methods, as described in [79], are a common tool for dynamic price forecast in distributed generation. MCMC methods are used in this thesis to estimate mean and standard deviation of electricity prices at every hour. The WA combines this estimation with the past prices using an Auto-regression component to generate the price forecast.

2.6.1.2 Non-linear Auto-Regressive Neural Network Model

The second family of price forecast methods analysed in this thesis is a type of artificial intelligence in the form of a NARNET. The key difference between the NARNET and a conventional ANN is the use of the delay and feedback, to use the i-th value of a time series to obtain the $i+1$-th value.

The three architectures analysed in this thesis are shown in Figure 2.9. The main difference between the different architectures is the number and organisation of the hidden layers inside the network. In all cases, the NARNET models a time series as a function of its past values, in this case, the electricity price [94]:

$$\Pi_{day}(t) = f_{ann}(\pi(t-1), \ldots, \pi(t-d)) \tag{2.13}$$

where d is the number of elements in the delay vector D_{ann}, which functions as the memory in the NARNET, Π_{day} is vector of price estimates π for a day at time t, and f_{ann} is the NARNET mathematical model. The first block in a NARNET is the update of the delay vector D_{ann} that contains the prices of each past day as its elements:

$$D_{ann}(t) = [p_{t-1}, \ldots, p_{t-d}]^T \tag{2.14}$$

where p_t are the prices at each time t. As the NARNET operates, the values shift positions to the next time step, eliminating the oldest information first, while the vector D_{ann} is completed with the network's feedback loop. The mathematical model of the layers in the NARNET, and the model of the NARNET itself is known as the forecast model's transfer function. The transfer function for each hidden layer is described by the logistic sigmoid function[95]:

$$f_{Sigmoid}(W_1 D_{ann}) = \frac{1}{(1 + e^{-W_1 D_{ann}})} \tag{2.15}$$

Where W_1 is the matrix of weights of the hidden layer, with a number of rows equal to the number of neurons in the layer. The open loop transfer function of a NARNET with a single hidden layer is:

$$f_{series} = W_2 f_{Sigmoid}(W_1 D_{ann}(t) + B_1) + B_2 \tag{2.16}$$

where W_2 is the matrix of weights in the output layer of the NARNET, B_1 is the bias vector of the hidden layer and B_2 is the bias vector in the output layer.

As NARNETs are also used for wind speed forecasting in this thesis, more of it details will be presented at the end of this section.

Complementary to accuracy and error metrics, total cost is another way to assess the effectiveness of a forecast model when applied to a specific test case [96]. The total cost resulting from the use of the price forecast methods, along with the operation of the rest of the control system is presented in chapter 5.

2.6.2 Wind Speed Forecast Methods

In this subsection, the families of wind speed forecast methods applicable to distributed control are presented. Similar to grid price forecast methods, these methods are based on AR methods and ANN models.

As explained previously for the AR models, the persistence model serves as the simplest auto regressive model that can be implemented, often used as a point of comparison with another forecast model [65, 77, 78]. An evaluation of the performance of the persistence model can also be used as an indication of the magnitude of variation in the data is, when studying forecast models applied to different data sets. The following subsections provide a description of the suitable methods for real-time distributed control. The mathematical formulation of these models is described in chapter 4.

2.6.2.1 Persistence Model

The persistence model is basically the assumption that the difference between one sample and the next is small when the sampling frequency is sufficiently small, and therefore the wind speed one step ahead will remain mostly unchanged [78].

The effectiveness of the persistence model depends on the location and sampling frequency. This model would not be suitable for generating forecasts with time horizons larger than one hour. This model will be used as a baseline to compare the rest of the wind speed forecast models [65].

2.6.2.2 ARMA model

The Auto Regression Moving Average (ARMA) model uses two principles to realise its forecast. This models combines the past values with information about past errors to realise its forecast. This methods allows the modelling of complex behaviours with relatively low amount of weights used during execution [97]. However, as past errors are required, this model needs to run for a number of time-steps equal to the amount of past errors required [17]. In contrast, auto-regression only requires information about the past behaviour of the intended variable to be predicted and not past performance of the model itself, which simplifies optimisation of the model for minimal error and operation.

2.6.2.3 Auto-regression Model

This model also produces the forecast of the wind speed only with past values, with the difference that in this thesis it is optimised using the Quasi-Newton solver to minimise the total error, from a given starting point. The advantages of this method are the convergence of the solution and the quicker generation of the solution compared to the ANNs [8].

This forecast model is defined as follows:

$$F_{AR}(t) = \sum\nolimits_{d_i} W_{AR}(d_i) Y_{ws}(t - d_i), \forall d_i \in D_{AR} \tag{2.17}$$

Where F_{AR} is the AR wind speed forecast, realised by the sum of the wind speed samples Y_{ws} multiplied by the $w_{AR,i}$ elements in the vector of weights W_{AR} for the AR model. The elements of this sum are indexed by the d_i hours in the set $D_{AR} = [t - 1 \ldots t - d_{AR}]$, where d_{AR} is the delay size of this model.

The weights $w_{AR,i}$ are optimised with the Quasi-Newton method by solving the following least squares problem:

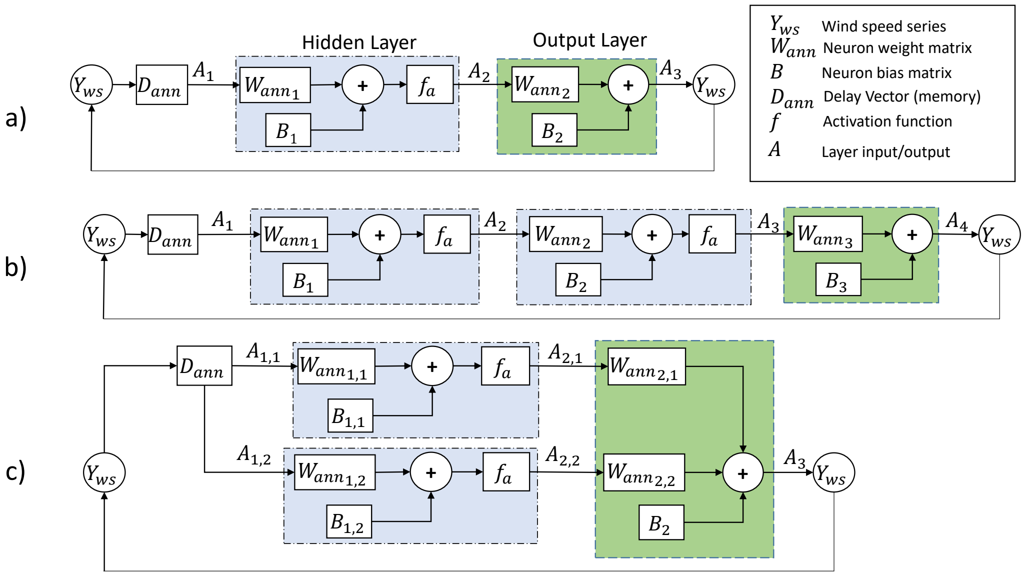


Figure 2.9: NARNET Architectures used for this study. a) single hidden layer. b) series hidden layers. c) parallel hidden layers.

$$\min_{W_{AR}} \sum\nolimits_t (Y_{ws}(t) - F_{AR}(t))^2, \forall t \in T \tag{2.18}$$

In equation 2.18, all the weights $W_{AR,i}$, grouped in the W_{AR} vector, are tuned to minimise the total summation of the squared errors applied to the entire historical sample data available of T samples.

2.6.2.4 NARNET Model

As mentioned previously, in this type of ANN, the output is feedback to the input, with an internal matrix called Delay that stores information and serves as a memory in time-series forecasting. This network then produces the forecast based only on past values of the time series, in this case, past values of wind speed [7, 8, 77].

In this study the connections between the layers is defined by three different architectures tested shown in Figure 2.9.

Each hidden layer in the NARNET has the following function:

$$A_{l+1} = f_a(W_l A_l + B_l) \tag{2.19}$$

where A_{l+1} is the output of the layer l, A_l is the input of the layer, ω_l is the weight matrix of the layer, B_l is the bias of the layer and A_1 is the input of the NARNET. Each hidden layer has an activation function f_a at the end, which encodes the non-linear properties of the data, in other words, is the function that allows the NARNET to forecast non-linear time series. The output layer function is [98]:

$$A_{l+1} = W_l A_l + B_l \tag{2.20}$$

In this study, three activation functions are used for the hidden layers of the NARNET forecast model:

$$f_{Logsig} = \frac{1}{1+e^{-x}} \tag{2.21}$$

$$f_{Sigmoid} = \frac{2}{1+e^{-x}} - 1 \tag{2.22}$$

$$f_{ReLU} = max(0, x) \tag{2.23}$$

where f_{Logsig} is the Logistic Sigmoid function, $f_{Sigmoid}$ is the hyperbolic tangent Sigmoid function and f_{ReLU} is the Rectified Linear Unit (ReLU) function [88, 89, 98].

The input A_1 at a specific time t is stored in the NARNET as part of an internal vector D_{ann} that also contains past values of the input, in other words, D_{ann} represents the "memory" of the NARNET. In this case, D_{ann} contains past values of Y_{ws} up to t used to generate the wind speed at time $t+1$, also for the following generation transfer functions, the suffix indicate the sizes of the vectors and matrices involved, rather that enumerating the layers in which they belong, this is is done to illustrate how these elements are affected by the parameters set on the model.

Given the complexity of the NARNET, and the randomness introduced in the training of the ANN, this method may produce different results in terms of accuracy for the same size of the internal matrices each time the training is done [7].

NARNET training method

The NARNET is a machine learning method driven by historical data. The model produces different results each time it is run even when the inputs are the same because of the randomness induced in the training process [7, 8].

The training process is randomised for two reasons: to mitigate the bias from the selection of the training, testing and validation sets and secondly to better explore the accuracy the model can achieve depending on the sets mentioned before. In other words, this mechanism aids in the generalisation of the model for new data.

The training algorithm for the NARNET is the Levenberg-Marquardt algorithm, which minimises the Mean-Squared Error (MSE) of the NARNET bewteen the training values, which are the actual wind speeds, and the output values by adjusting the weights and biases at each layer. The minimisation problem solved by the training algorithm is defined as follows:

$$\min_{\omega,B} \frac{\sum_t (Y_{ws}(t) - F_k(t))^2}{N_T}, \forall t \in T \tag{2.24}$$

The method uses the back-propagation algorithm to obtain the Jacobian of the errors with respect to the weights and updates the weights using a gradient descent method similar to the quasi-Newton and Newton methods [98]. The training method can be summarised as follows. The network is initialised with all the weights and biases randomised with values between -1 and 1, using a uniform distributed PDF. From this starting point the weights and biases are updated using the Levenberg-Marquardt back-propagation algorithm, following the direction of steepest descend until the stop criterion is met. The stop criterion is met when any of these conditions is met: **1)** The epochs have reached a maximum of 1000, an epoch is completed when the algorithm uses all the training data once. **2)** The performance goal of $MSE = 0$ is achieved. **3)** The gradient of the performance is $1x10^{-5}$ or lower. **4)** The maximum validation check of 6 epochs is reached.

In this training method, the entire data set is divided in three sets, namely, the training set, the validation set and the testing set. The training set is the set of samples used directly to obtain the weights and biases updates of the NARNET, therefore the weights and biases are heavily biased by the training set.

To prevent over-fitting of the NARNET, the validation set is used as a separate calculation of the performance of the NARNET. When the performance from the validation check is continuously deteriorating up to the maximum of 6 epochs, the training stops. This allows the NARNET model to be validated while avoiding over-fitting [8].

However, the validation set indirectly influences the weights of the NARNET during training. For this reason the test set is not used during training but as its name implies, it used to test the performance of the NARNET once it is training. The training, testing and validation set are randomly selected from the entire data set in ratios of 70%, 15% and 15% accordingly. The entire training method for the NARNET is summarised in Figure 2.10.

Once the variables required for optimal scheduling of the microgrid are available to the control system, the secondary control layer must solve its optimisation using the methods discussed next.

2.7 Optimisation Solvers

Power flow management must be done in the microgrid to coordinate the different generation sources such that regulation is maintained during operation [4]. However, power management should also use the generation sources such that their operation is optimal given one or more microgrid objectives based on the stakeholders involved.

In this thesis, optimisation is a major component of power schedule generation for cost minimisation as well as in the optimisation of accurate renewable generation.

The main optimisation tools are based on convex programming, dynamic programming, stochastic programming, robust programming, and heuristic methods [2].

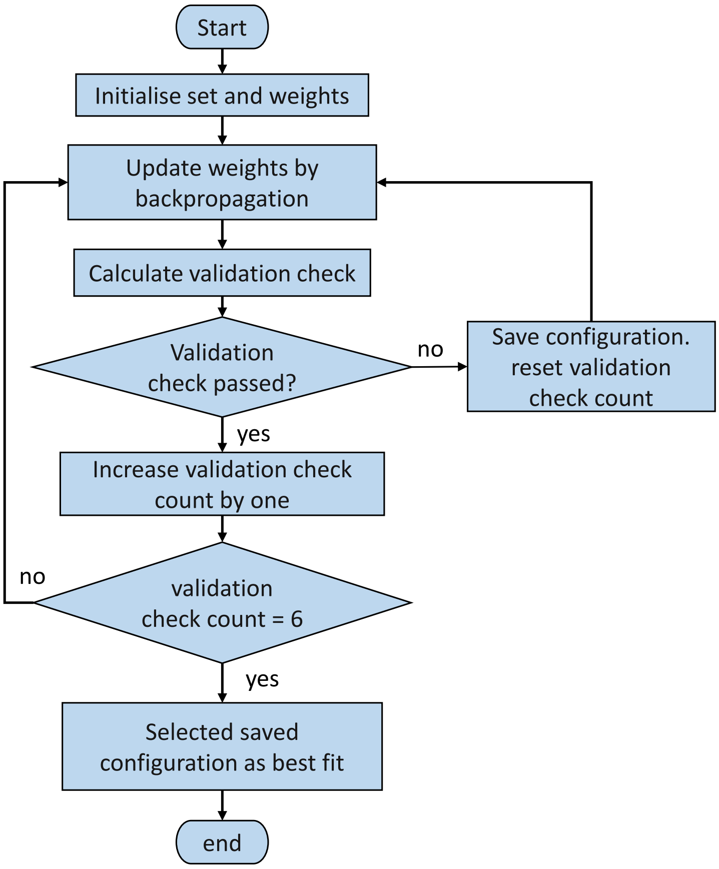

Start
Initialise set and weights
Update weights by backpropagation
Calculate validation check
Validation check passed?
no
Save configuration. reset validation check count
yes
Increase validation check count by one
no
validation check count = 6
yes
Selected saved configuration as best fit
end

Figure 2.10: Summary of the training process of the NARNET model.

The heuristic methods have the benefit that they can handle the non-linear problems of optimal power management and are relatively easy to program [4, 55]. The disadvantage of heuristic methods is that, given their use randomness to solve the optimal power management, the computation time required to generate the solutions and the solution itself is random [4]. This means that the same inputs can generate different solutions and take different amounts of time to reach such solution.

To maintain the ability of the microgrid's control system to continuously regulate the microgrid in a reliable way, tot only is important to show the ability of the secondary control to generate optimal power schedules, but also its ability to be deployed in a real-time control system and in a distributed control environment. This may not be achieved with the heuristic or any other centralised offline optimisation solver

[4]. Comparison in terms of total cost achieved by different optimisation solvers, and computational time used, is presented in chapter 5.

For the case of the optimisation of the forecast models both heuristic and numerical solvers can be used, both of which are useful in the verification of the solution found to maximise accuracy achieved [8, 99].

Further in the thesis, the use of Genetic Algorithm and Quasi-Newton solvers are used to optimise the Auto-regression forecast models [8]. The auto-regression models used in this thesis can be optimised using linear programming, given the nature of the minimisation problems required to be solved. A gradient descent method is employed in the optimisation of the Artificial Neural Networks in the training algorithm for electrical grid price and wind speed forecast [98]. Results for the optimisation in terms of accuracy of the forecast methods used in this thesis is presented in chapter 4.

The optimisation requirements for microgrid control can also be solved by the MAS approach, with the benefits in control distribution and computational speed [4, 6]. The MAS approach is described in the next section.

2.8 Multi Agent Systems

To realise the hierarchical control system in a distributed environment, the use of individual units of artificial intelligence called agents is proposed as mentioned previously in this chapter in the form of a Multi-Agent System [100, 101].

The agents are organised in virtual groups called containers, which are connected in a communication network. This networks of agent containers that are used to achieve a more complex task is what composes the basic structure of the MAS platform.

Most MAS platform applications contain three standard agents for operation: The Agent Management system (AMS) agent that deals with the creation of agents,

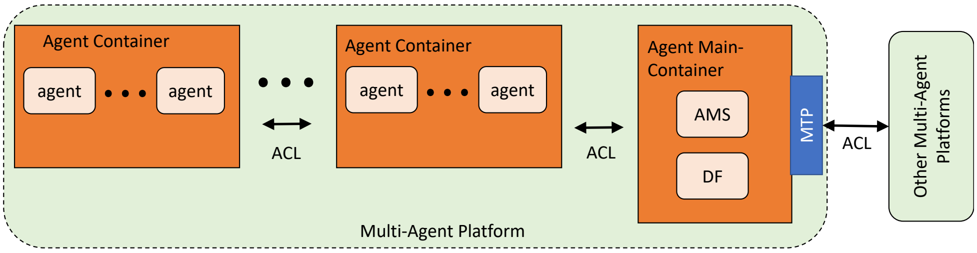


Figure 2.11: Basic organisation of the elements in a MAS application. The arrows indicate the communication between the agents. The platform can be distributed in as many computers (hardware) as required. [1]

Directory Facilitator (DF) agent that coordinate agent service offers and request, and the Remote Monitoring Agent (RMA) which deals with the interface across the platform. The container that hosts the AMS and the DF of the MAS platform is known as the main-container, only one main-container can be active in the platform at a time [6, 102]. The rest of the containers contain the agents used for a specific application, such as realising the microgrid's distributed control. Figure 2.11 illustrates the basic elements of the MAS platform.

The agents have three main properties that allow the realisation of global optimisation tasks in a distributed environment. This properties are: **1)** The agents follow simple algorithms called behaviours depending on the information available to them. **2)** The agents have the ability to send and receive message from other agents using the Agent Communication Language (ACL) protocol. **3)** they are easy to multiply, such that the global tasks requirement can be realise by a combination of multiple agents [22, 102].

MAS applications include management of: natural resources [103], single microgrids [19] or a combination of microgrids [20, 21]. Plug-and-play design of MAS ensures that any component can be added or removed at any point in the system without re-engineering the controls, making it suitable for distributed control [104, 105]. Agents need to act coordinately, either by consensus to carry on most of their actions [106], or by following a leader agent [107–109].

Examples of the MAS approach in microgrid control are found in [22, 110] where the MAS was used for a consensus protocol for power flow distribution between a central controller and distributed primary controllers. However, these control systems have a central node of operation for the forecast methods required, which represents a single point of failure. To avoid this vulnerability, in this thesis independent forecast methods are available at each node with local information as presented in [5].

Given their optimisation capabilities, MAS applications are used to minimise power losses or maximise the economic benefit in microgrids, benefiting from separating control tasks in different sub problems, for asynchronous and parallel operation [33, 111]. These simplified sub problems can be solved by individual agents for optimal power flows [112].

This arrangement prevents single points of failure and maximise the scalability and adaptability of the control system, such that the distributed generation sources can be added or removed from the microgrid.

Most of distributed controls that apply MAS, do it by programming the agents either by cooperation or competition [10, 113]. This defines the rate at which agents communicate, in this sense, event-triggered agents require less computational resources for communication than state-dependent agents, as some agents may not need to be updated at the same rate as others in the system [114], state-dependent agents heavily rely on high frequency communication [115].

For the resilient operation of the MAS implementation in this thesis, the platform is distributed in separate hardware devices. The device hold an agent container connected to each of the primary controllers for the microgrid control, along with the capacity to launch the main-container. The devices are connected in a ring communication topology, such that a main-container is available at all times in the platform [5]. The analysis of the resilient operation of the MAS communication network is presented in chapter 3.

In the context of hierarchical control, optimal operation of the microgrids with

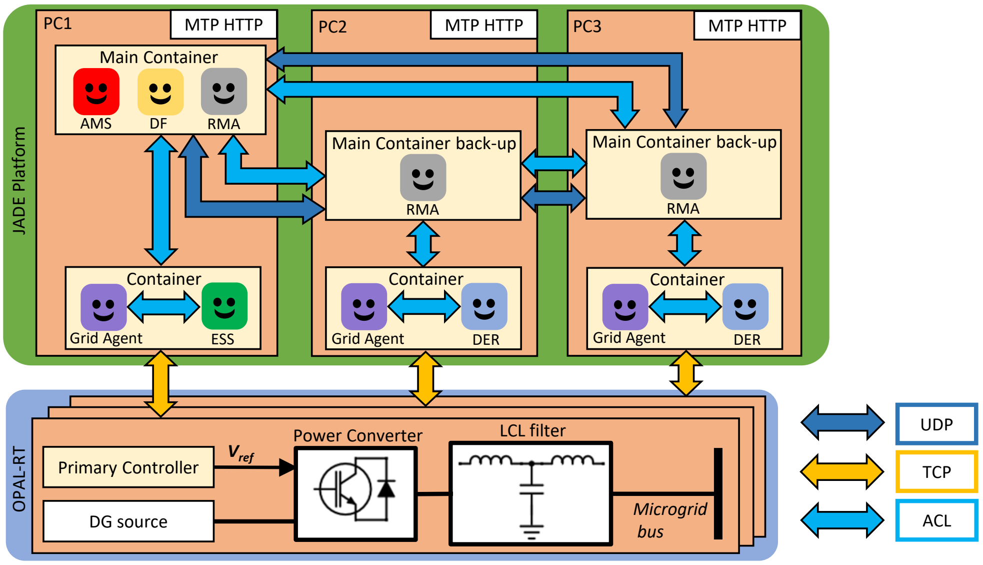


Figure 2.12: MAS implementation in real-time simulation for distributed control.

the use of MAS is achieved by breaking the control objectives in simple tasks that are carried over by individual agents [4, 5]. The combined operation of the agents generate the required forecast signals as well as the power schedule, which is used as a reference point for the primary control layer. Figure 2.12 represents the MAS application and its interaction in the microgrid control system. Results for the optimal operation of the microgrids with the MAS applications, given their different objectives, are found in chapter 5.

2.8.1 Communication Protocols

A characteristic feature of the distributed control is the communication network employed to maintain organisation of the control systems. Depending on the purpose of the communication task, different communication protocols are used.

the backbone of the communication between the agents described previously is the ACL messages [102]. These messages not only carry the information passed from one agent to another, but can also describe the type of message and description of the sender and intended receiver. Because of this, the ACL messages allow the formation of the communication network even when information about the receiver, for example, address, is not known from the sender point.

Additionally to the ACL messages, in the context of microgrid distributed control, the Transmission Control protocol (TCP) is used to communicate the devices that realise the MAS platform, either between or with the primary control layer, requiring the definition of Hypertext Transfer Protocol (HTTP) addresses for each container, for containers in the same device, the containers may share the same HTTP address, using different ports [6, 102].

The main communication protocol between the secondary control layer, realised by the MAS platform, and the primary level is the TCP/IP, through a defined socket for connection, which is composed by an IP address and port.

For the case of the restoration service of the MAS platform, User Datagram Protocol (UDP) packet monitoring is implemented. With this service, the devices that realise the MAS platform are pinging the connection to each other [32]. This service and its contribution to resilient operation are presented in chapter 3.

For the case of the tertiary control layer, with the Message Transport Protocol (MTP) defined by the Foundation of Intelligent Physical Agents (FIPA), is possible to send ACL messages over different MAS platforms. To send a ACL message between two agents in different platforms, an MTP address is required additionally to the TCP/IP address [32, 33].

The communication components that allow sharing messages between agents in separate computers and in separate MAS platforms can be visualised in Figure 2.13.

2.9 Simulation Tools

All of the methods introduced in this chapter to solve the different problems of optimal and reliable microgrid management need to be tested with conditions as

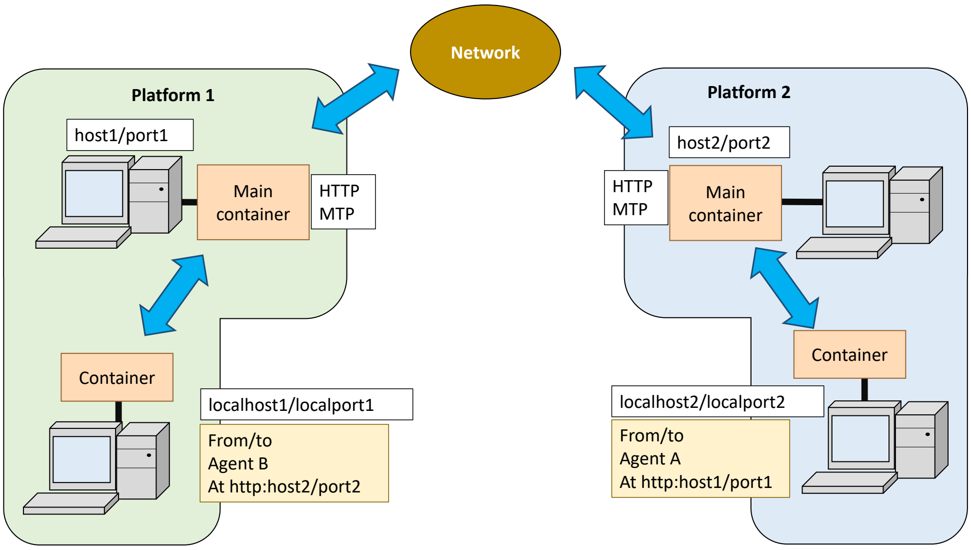


Figure 2.13: Elements used in MAS communication between MAS platforms. The blue arrows represent a communication link of ACL messages between agents.

realistic as possible. This subsection will describe the hardware used at each part of the design to test it.

In the context of control systems testing, Hardware In the Loop (HIL) simulation, refers to a type of simulation that combines hardware and software components into the simulation model, where the software contains a virtual or mathematical representation of the model, and the rest of the software and the hardware represent the control system. This type of simulation is ideal to test the ability of the control system to respond to real-time conditions in terms of operation under normal and faulty conditions [73].

The control system developed through this thesis is tested in an OPAL-RT real-time simulator connected to a communication network of Raspberry Pis as the secondary control [4–6]. The reason to use this configuration is to test the ability of the system to respond at real-time in terms of electrical variables and test the communication algorithms against realistic technical limitations to make the tests of the control system as close to real operation as possible.

Inside the real-time simulator, the microgrid electric circuit is modelled using the components provided by the Simscape Library by MATLAB and microgrid blocks developed by the Electric Power HIL Controls Collaborative (EPHCC) [4]. Similarly, the primary controllers are modelled inside the real-time simulator.

The secondary and tertiary control layers are realised by MAS platforms in external devices [5, 6]. The simulator is configured to be used as a TCP/IP server to allow the communication between the devices and the simulated model. Depending on the test case, the devices are communicated using TCP/IP or UDP communication protocols. For the test cases of this thesis, the devices are Raspberry Pis and external PCs.

2.10 Summary

In this chapter, the necessary technologies and methods for the design for the control system are presented.

It is established that a distributed control hierarchy is the main focus of the control schemes employed in this thesis. This type of control is selected for its scalability in the number of distributed generation sources and their respective controllers in the microgrid. This includes the ability to maintain the rest of the microgrid components in the case of unforeseen changes in the control system under faulty conditions.

Solutions for economically and environmentally optimal operation without the use of centralised power management are introduced in this chapter as suitable distributed forecast methods, in the form of artificial intelligence integrated in the control system. Two methods of artificial intelligence employed for this purpose are introduced in this chapter, the Multi-Agent System and the Artificial Neural Networks. Additionally, other technologies used to compare the performance of these methods are also introduced in this chapter.

Finally, this chapter presents the real-time simulation technologies that are employed in the rest of the chapters in this thesis.

Chapter 3

Distributed Real-Time Control Stability and Reliability

3.1 Introduction

With the growing demand for decarbonisation of energy systems. To this end, there has been an increasing effort to integrate small-scale distributed generation sources, which will include renewable as well as conventional generation resources, with the electric demand locally in form of microgrids.

To achieve stable and optimal operation, many microgrid applications rely on the use of centralised power management systems, however, this approach leaves the operation of the microgrid with the risk of single point of failure vulnerability. Consequently, in this chapter (and in this thesis), the control schemes for microgrids are designed with a distributed architecture.

The proposed distributed control system analysed in this chapter is the hierarchical control based on MAS, as introduced in chapter 2. The microgrid and the distributed control system implemented in this thesis for real-time control is shown in Figure 3.1.

Given that the microgrid is used to implemented and coordinate the use of distributed generation, combined with the electrical load over time, it is necessary to have a

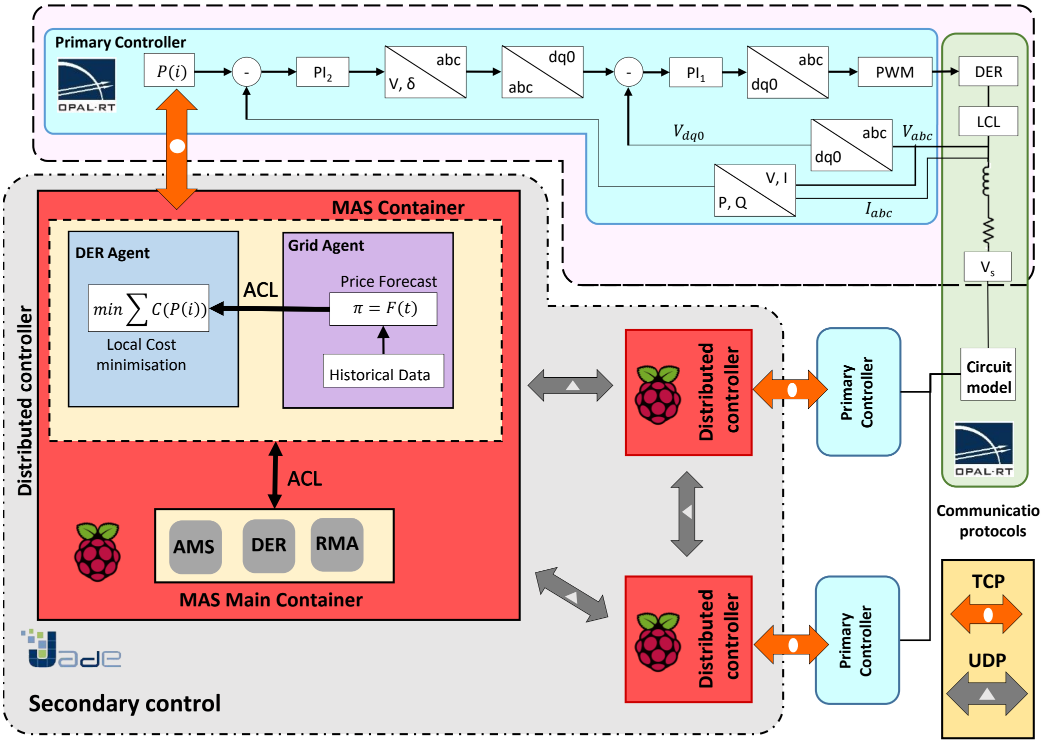


Figure 3.1: Distributed control System model implementation in real-time simulation. The pink region delimits the Lyapunov stability analysis. The grey region, which is the MAS platform, delimits the region of reliability analysis by graph theory. The green block represents the physical layer, while the rest of the blocks represent the hierarchical control system.

control system capable of maintaining and regulate voltage stability, such that end users have a safe and reliable electrical service.

In this chapter, the necessary equations to guarantee stability of the electrical circuit, achieved by the distributed control system used in this thesis are presented. The stability is analysed in the physical layer, represented by the green area in Figure 3.1, considering the interaction between buses, and the interaction between the primary controllers with its corresponding DER.

This analysis is done with the use of Lyapunov stability theory [14], deriving the conditions of voltage and current for stable operation given the design of the circuit and primary controllers. The Lyapunov second method is selected for this analysis given that it considers the non-linear components of AC power flow in the short line model and therefore includes the large-signal case. Because of this, the conclusions drawn from this method can be extended to cases outside the specific simulation test cases in this thesis.

Additionally, graph theory is employed for verifying the resiliency of the secondary control layer. This is done by analysing the connectivity of the MAS communication network, modelled as a connected graph using its underlying characteristic polynomial.

The characteristic polynomial, also known as the L-polynomial is used to verify the connectivity of the MAS network under normal and faulty conditions. This is done to further validate the stability of operation of the microgrid distributed control system proposed.

The reliability and resilient operation of the proposed control system is validated in this chapter with a simulation test case. The test case consists of a real-time simulation to demonstrate the operation of the system under normal and faulty conditions. Simulation results will be presented at the end of this chapter.

3.2 Stability Analysis

Although distributed generation offers several benefits to the electricity grid, such as minimising power line congestion, reducing transmission losses by bringing generation closer to consumption, and allowing the integration of small-scale renewable generation, it is most important to guarantee that such systems are reliable.

This means that there is a need for verifying that the control systems operating distributed generation are stable during transient and steady state of operation. The verification for stability of the microgrid with the proposed hierarchical control system in this thesis is done at the primary control layer and its interaction with the physical layer.

By delimiting the electrical conditions at which the interaction between buses is stable, the primary control layer can be designed to operate each of the buses within

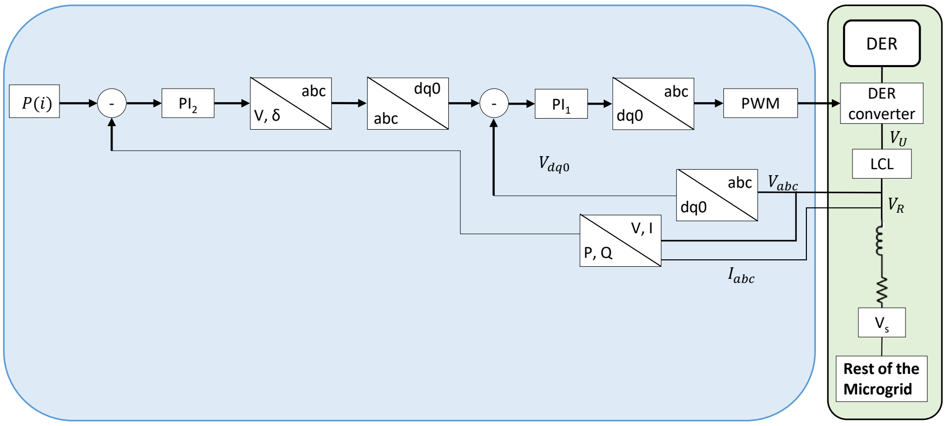


Figure 3.2: Primary control of each DER shown in the blue region and the physical layer of the DER shown in the green region.

some given range. In this case the buses are regulated to operate within the UK standard of $400V_{ph}$ at $50Hz$ for the three phase system in the distribution circuit.

Each primary controller, which is shown in Figure 3.2, is designed as a PI control with a double feedback loop, an inner loop for the regulation of the voltage of the DERs and ESSs and an outer loop for the regulation of the power output.

This is similar to the concept of droop control, where droop control is designed to mimic the inertia of rotational machines of the transmission circuit, in the sense that the frequency is dependant on the active power and the voltage is dependant on the reactive power. The paradigm of droop controllers is that the power lines of the distribution circuit are used as a communication network in fully decentralised control.

In other words, in decentralised droop control the DERs communicate information regarding their generation without the need of an additional communication network, by affecting the frequency at which the circuit operates. In contrast, in this thesis the distributed control paradigm is employed, which uses a separate communication network to realise the microgrid control. Given that the stability of the microgrids depends essentially on the response of the primary control layer and its modelling, the main differences of the conventional droop control and the control used here are listed in Table 3.1.

Table 3.1: Main properties of the hierarchical control of the droop control and the proposed control in a microgrid

	Droop Control	**Proposed Controller**
Main effect of physical layer	Frequency is affected by active power and Voltage is affected by reactive power	Voltage is affected by active power and phase angle is affected by reactive power
Type of Communication	No communication, it relies on local measurements	A separate communication network is maintained by the controllers

In the primary controllers of this thesis, the power output control is based on the power between the source and the rest of the system, which is regarded as simple short power line model between two buses from the point of view of the outer loop in each of the primary controllers.

The controller is modelled in terms of the voltage and phase angle interactions with the active and reactive power P and Q between two buses as follows:

$$P = \frac{V_S}{R^2 + \omega^2 L^2}(RV_S - RV_R cos\delta - \omega LV_R sin\delta) \tag{3.1}$$

$$Q = \frac{V_S}{R^2 + w^2 L^2}(RV_R sin\delta + \omega LV_S - \omega LV_R cos\delta) \tag{3.2}$$

Where V_S is the sending bus voltage, V_R is the receiving bus voltage, *delta* the phase angle of the receiving bus, ω the angular frequency of the system, R the resistance of the power line connecting the buses and L the impedance of the power line. Given the non-linear components of these equations, the Lyapunov's second method introduced in Chapter 2 is selected to delimit the conditions for stability of the primary controllers.

With the use of the Lyapunov's second method its possible to establish the limits to which the system will reach stability for a given set point. Secondly, the current response is analysed to delimit the speed response between the secondary level and primary level. The stability criterion of this method is presented in the next subsection, similarly, details on how this method is used in this thesis are found in Appendix C.

Once the primary controllers are guaranteed to maintain voltage stability of the system during transient and steady state, there is a need to ensure that the secondary control is resilient. The resilient operation of the secondary control layer is guaranteed by implementation of the restoration service of the communication network formed by the distributed controllers in the secondary control layer. This is analysed by calculating the characteristic polynomial of the communication network before and after faults using graph theory. These polynomials indicate if the network remains fully connected and operational, which is the main advantage of distributed control over centralised control schemes.

The stable design of the control layers provide a foundation to the entire control system on which the optimal operation can be integrated. The stable and resilient operation required for the reliable operation of the microgrid is analysed in this chapter.

3.2.1 Lyapunov Stability

The following subsections will outline the necessary conditions to guarantee the steady state stability of the system analytically using Lyapunov's direct method, which is required for non-linear systems, as it is in this case, analysing the system as a large-signal model. This analysis is done over the pink region in Figure 3.1.

The system stability can be analysed by analysing the stability of each control loop, shown in the blue region in Figure 3.2, in the primary control layer as follows:

3.2.1.1 Inner Loop

The inner loop regulates the voltage of the power converter with the aid of the LCL filter depicted in the green region of Figure 3.2. The dynamics of the inner loop consider the effect of the LCL filter and the line connecting the two buses. The stability is established by the stability of the current with respect to the voltage output from the power converter V_U and the bus voltage V_S:

$$V_U = L_U \frac{dI_U}{dt} + V_C \tag{3.3}$$

$$V_C = \frac{1}{C} \int (I_U - I_S) dt \tag{3.4}$$

$$V_C = L_S \frac{dI_S}{dt} + RI_S + V_S \tag{3.5}$$

where I_U is the converter output current, I_S is the bus current, L_U the inductance at the converter side, L_S the inductance at the bus side, C the capacitance and V_C the capacitor voltage of the LCL filter. Transfer functions for I_S are:

$$T_1 = \frac{I_S}{V_U} = \frac{1}{L_U L_S C s^3 + R L_U C s^2 + (L_U + L_S)s + R} \tag{3.6}$$

$$T_2 = \frac{I_S}{V_S} = -\frac{L_U C s^2 + 1}{L_U L_S C s^3 + R L_U C s^2 + (L_U + L_S)s + R} \tag{3.7}$$

As there are no sign changes in the terms of the denominators, and all terms are positive, all roots have negative real parts, which guarantees open loop stability with respect to V_S. In the same manner, it is possible to calculate gains for the PI control such that the system is closed loop stable with respect to V_U. A step by step calculation is shown in appendix B.

3.2.1.2 Outer Loop

As the inner function is stable, the outer loop must be asymptotically stable [15] to establish stability of the entire cascade control [116]. The outer loop models the dynamics of power flow between buses with the following equation:

$$\begin{aligned} S_S = 3\frac{V_S}{R^2 + \omega^2 L^2}((RV_S - RV_R cos\delta - \omega L V_R sin\delta) \\ +i(RV_R sin\delta + \omega L V_S - \omega L V_R cos\delta)) \end{aligned} \tag{3.8}$$

Where S_S is the apparent power sent, V_S sending bus voltage, V_R is the receiving bus voltage, δ is the phase angle between V_R and V_S, ω is the angular frequency, R is the resistance of the line and L is the inductance of the line.

Selecting V_R and δ as the variables and treating the rest of parameters as constant, the following system of equations can be formulated for the active power P and reactive power Q sent from the converter to the bus:

$$P(V_R, \delta) = K_1(RV_R cos\delta + \omega L V_R sin\delta - RV_S) \tag{3.9}$$

$$Q(V_R, \delta) = K_1(\omega L V_R cos\delta - RV_R sin\delta - \omega L V_S) \tag{3.10}$$

$$K_1 = \frac{3V_S}{R^2 + \omega^2 L^2} \tag{3.11}$$

By selecting the appropriate state variables, X_1 and X_2, the close loop integral control is formulated a follows:

$$V_R = X_1 + V_S \tag{3.12}$$

$$-\delta = X_2 \tag{3.13}$$

$$\begin{aligned} P(X_1, X_2) = K_1(X_1(RcosX_2 + \omega L sinX_2) \\ +V_S(RcosX_2 + \omega L sinX_2) - RV_S) \end{aligned} \tag{3.14}$$

$$\begin{aligned} Q(X_1, X_2) = K_1(X_1(\omega L cosX_2 - RsinX_2) \\ +V_S(\omega L cosX_2 - RsinX_2) - \omega L V_S) \end{aligned} \tag{3.15}$$

$$\begin{aligned} \dot{X_1} = -K_1(X_1(RcosX_2 + \omega L sinX_2) \\ +V_S(RcosX_2 + \omega L sinX_2) - RV_S) \end{aligned} \tag{3.16}$$

$$\begin{aligned} \dot{X_2} = -K_1(X_1(\omega L cosX_2 - RsinX_2) \\ +V_S(\omega L cosX_2 - RsinX_2) - \omega L V_S) \end{aligned} \tag{3.17}$$

which has a solution in $\dot{X}(0) = 0$, where $X = [X_1 X_2]^T$. To demonstrate Lyapunov local Asymptotic Stability by the direct method, a candidate function $\mathcal{V}$ must have the following properties [14, 117]:

$$\mathcal{V}(0) = 0 \tag{3.18}$$

$$\mathcal{V}(X) > 0, X \neq 0, X \in D_X \tag{3.19}$$

$$\mathcal{V}(X) \to \infty, ||X|| \to \infty \tag{3.20}$$

$$\dot{\mathcal{V}}(0) = 0 \tag{3.21}$$

$$\dot{\mathcal{V}}(X) < 0, X \neq 0, X \in D_X \tag{3.22}$$

The function $\mathcal{V}$ and domain D_X are selected as follows:

$$\mathcal{V}(X) = \frac{1}{2}X_1^2 + \frac{1}{2}X_2^2, \qquad X \in D_X \tag{3.23}$$

$$D_X = \{X \in \mathbb{R} | V(X) \leq \alpha, \ 0 \in D_X, \ D_X \; is \; continuous\} \tag{3.24}$$

As $\mathcal{V}$ has infinitely many solutions, D_X is constrained such that it delimits a closed contour inside one of the closed levels curves at value α around the origin, such that the domain D_X only contains one solution for $\mathcal{V}(X) = 0$ and $\dot{\mathcal{V}}(x) = 0$ at $X = 0$, to comply with the properties of a Lyapunov's candidate function.

Calculating the derivative and substituting the time derivatives from the system in (3.12) the following is obtained:

$$\begin{aligned} \dot{\mathcal{V}}(X) =& X_1\dot{X}_1 + X_2\dot{X}_2, X \in D_X \\ =& -K_1(X_1(X_1(RcosX_2 - \omega LsinX_2)+ \\ & V_S(RcosX_2 - \omega LsinX_2) - RV_S)- \\ & X_2(X_1(\omega LcosX_2 + RsinX_2)+ \\ & V_S(\omega LcosX_2 + RsinX_2) - \omega LV_S)) \end{aligned} \tag{3.25}$$

given (3.23) and (3.25), the conditions in (3.18) are true, and therefore $\mathcal{V}$ is a Lyapunov function, and therefore, that the primary control is locally asymptotically stable around the equilibrium point in $X = 0$. The stability limits of the voltage imply a stability limit in the power that can be exchanged from one bus to another in the grid, as the voltage drop is directly proportional to the current in the cable and in turn, proportional to the power flow. Additionally, given that $\dot{X}$ has infinitely many solutions, which means $D_X \neq \mathbb{R}$, it can also be concluded that systems based on this line models do not have a global asymptotic stable equilibrium point.

Figure 3.3 shows a rendering of $\dot{\mathcal{V}}(X)$ with the level curves on top, showing that the derivative is concave with a single root at 0 for a domain that delimits any of the closed level curves.

Figure 3.4 shows the level curves of $\dot{V}$ overlaid with the trajectories of $\dot{X}$ to show that the system is locally asymptotically stable if the starting disturbance point is inside one of the closed level curves for an arbitrarily small error as all trajectories, shown as the blue arrows, point towards $X = 0$. The step by step calculations for the outer loop stability is presented in appendix C.

3.2.1.3 Current Stability

Once the delimitation in the domain of the references for the primary controller is established, the next step is guaranteeing that there is enough time between the response of each control layer, such that the control layers do not interfere with each other. This is done by establishing the speed limits between the inner control loop and the outer control loop, as the latter receives its reference point from the secondary control layer.

To guarantee that the two loops do not interfere with each other, the time domain response of the line current $I(t)$ between the buses is calculated:

$$\sqrt{2}V_S sin(\omega t) = RI(t) + L\frac{dI(t)}{dt} + \sqrt{2}V_R sin(\omega t + \delta) \tag{3.26}$$

With Laplace transform equation:

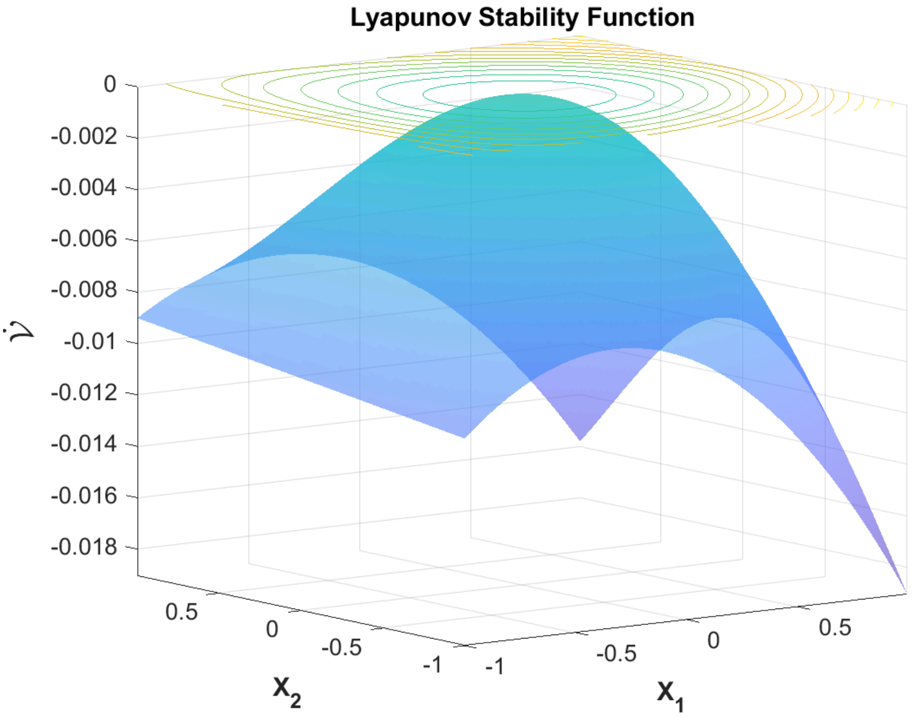


Figure 3.3: Visualisation of the properties of (3.25) to verify that (3.23) is a Lyapunov function. Given a closed level curve (shown at the top) as the domain D_X the conditions $\dot{\mathcal{V}}(0) = 0$ and $\dot{\mathcal{V}}(X) < 0, X \neq 0, X \in D_X$ are verified.

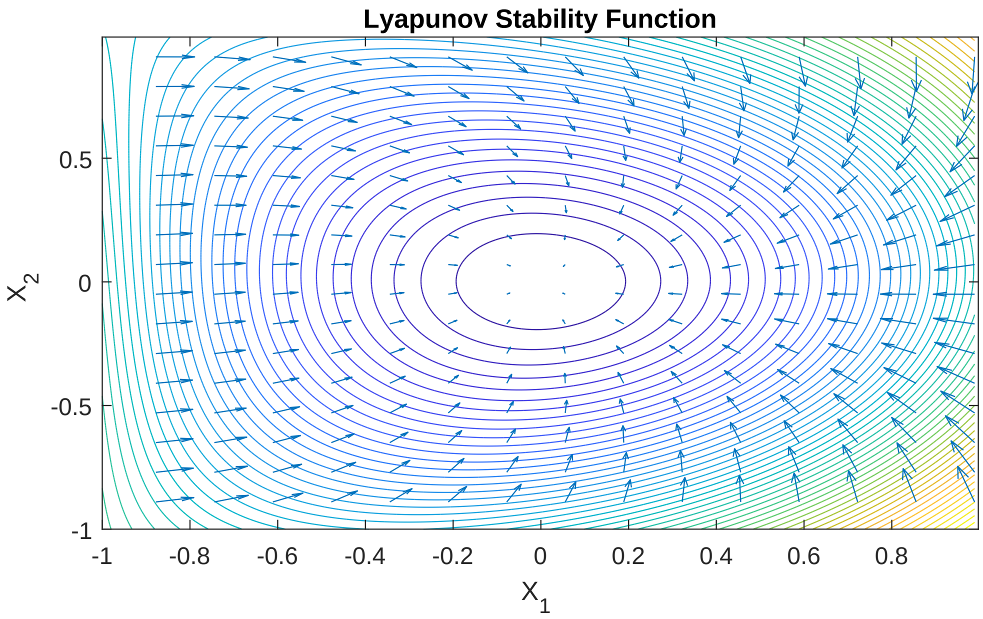


Figure 3.4: Time trajectories of the state space of the outer loop represented as a field space. The level curves are the Lyapunov function evaluated with respect to the state space.

$$I(s) = \frac{K_2 S}{s^2 + \omega^2} + \frac{K_3}{s^2 + \omega^2} + \frac{K_4}{\frac{R}{L} + s} \tag{3.27}$$

Where K_2, K_3 and K_4 are constants that are obtained by solving the following matrix:

$$\begin{bmatrix} L & 0 & 1 \\ R & L & 0 \\ 0 & R & \omega^2 \end{bmatrix} \begin{bmatrix} K_2 \\ K_3 \\ K_4 \end{bmatrix} = \begin{bmatrix} 0 \\ -\sqrt{2}V_R sin(\delta) \\ \sqrt{2}V_S\omega - \sqrt{2}V_R cos(\delta) \end{bmatrix} \tag{3.28}$$

With rest initial conditions, the current in time domain is expressed as follows:

$$\begin{aligned} i(t) = \frac{\sqrt{2}}{L^2\omega^2 + R^2}&((L\omega(V_R cos\delta - V_S) - RV_R sin\delta)cos(\omega t) \\ &+ (R(\omega V_S - V_R cos\delta) - V_R sin\delta L\omega)sin(\omega t) \\ &+ (L(\omega V_S - V_R cos\delta) + RV_R sin\delta)e^{-\frac{R}{L}t}) \end{aligned} \tag{3.29}$$

It can be observed in (3.29) that the transient state depends only on the last term, lasting 5 R/L seconds, after which the current stabilises. If the current and voltage are stable, then it follows that the output power will also be stable, following the reference set by the secondary controller. This sets an upper limit on how fast the outer loop can operate and a lower limit on how slow the inner loop must be to prevent interference between each other, in other words, to guarantee stability, the secondary control must be slow enough to allow the primary control to stabilise the current before setting a new power reference.

The equations described in this subsection are derived in appendix D.

3.2.1.4 Interconnected Stability

Although the stability analysis is done for the interaction of one bus with its neighbour, this aids in the definition for the conditions of stability in the microgrid, seen as a series of interconnected nonlinear dynamical systems.

The following theorem is applied for interconnected systems of the form:

$$\dot{\mathbf{X}}_1 = \mathbf{f_1}(\mathbf{X}_1(t), \mathbf{X}_2(t)) \tag{3.30}$$

$$\dot{\mathbf{X}}_2 = \mathbf{f_2}(\mathbf{X}_1(t), \mathbf{X}_2(t)), t \geq 0 \tag{3.31}$$

With initial conditions $X_1(0) = X_{10}, X_2(0) = X_{20}$, Where $\mathbf{f}$ are the functions that describe the interconnected system with state space $\dot{\mathbf{X}}$. If $\mathbf{X}_2$ is input to state stable for $\mathbf{X}_1$, and bounded with respect to X_1 uniformly in X_{20}, then the interconnected system is bounded [118]. In this case, the system is bounded by D_X to guarantee asymptotic stability.

Provided each DER/ESS controller in the microgrid fulfils the conditions set for asymptotic stability for each bus, no element in the circuit is unstable, load and power references vary sufficiently slow and there is a bus with constant voltage (infinite bus), namely, the main grid, a radial microgrid following this control system will always reach steady state stability.

The reason lies in the interaction of the last bus in the system with its neighbour, which can be seen as an interconnected nonlinear system of the form set in 3.30. If all elements are individually asymptotic stable, it follows that the last bus (bus_n) of the circuit, counting from the infinite bus, can always set its voltage difference with its neighbour to follow the power reference after a finite time, stabilising it. This process is repeated with the bus_{n-1} until it reaches the bus connected to the infinite bus, which is stable by definition, which in turn makes the interconnected system bounded by D_X.

Additionally as the states of any element are always within the domain D_X, and provided the power references change sufficiently slow, any power oscillations will diminish until power stability is reached throughout the microgrid, as each system is asymptotic Lyapunov stable.

As long as all the previous conditions described are met, the primary controllers will reach stable operation, however the control system must also be resilient at the secondary control level. The next section will briefly review the mathematical theory used to assure the connectivity in the communication network of the secondary control and therefore guarantee its resilience.

3.3 Resilient Operation

In addition to verify the reliability of the microgrid's control regarding transient and steady state operation, it is also important to verify that in the case of a faulty node, the remaining components of the microgrid continue stable operation. For the case of a distributed control system, this needs to be true for the fault of any single component, which is done by implementing the restoration service described next.

The restoration service is implemented in the MAS platform which realises the secondary control layer of the microgrid. This service allows any of the distributed computers in the communication network to deploy the main container if the computer hosting it is down or unexpectedly disconnected from the network. This means that any single computer that realises the secondary control layer can be restored in case of faulty operation. The region in which this service has impact is delimited by the grey region in Figure 3.1.

This service enumerates the computers forming the platform based on the order in which they connect and forms a UDP communication ring between them. The oldest computer in the network becomes the current leader of the platform by deploying the main-container.

If the leader computer is disconnected from the platform, the next oldest computer takes the leadership of the platform by incorporating the host address and port of the previous leader and launching a back-up of the main container.

This mechanisms allows a seamless fault-tolerance operation from the point of the view of any other computer in the platform, as from their point of view, the main-container simply updated its address.

In the case where there is no computer available that can launch the main container, the platform will be lost and the primary control will no longer receive updates for their power references. In this scenario, the microgrid itself maintains operation at a stable point, although without a smart power management system.

Another possible scenario covered by the restoration service is that the fault occurs in the communication link and not in the computer itself. In this case the secondary control is divided in two platform and each communication network can operate independently, however, this means that each platform is more vulnerable to faults as each can only rely on a fraction of actual computers available for subsequent faults.

The ring topology, in the formation of the communication network, is selected as the default configuration of the communication network given that it requires the minimum communication links between computers that allow redundancy. In other words, the loss of any single computer does not cause disconnection of the remaining computers in the network, which is verified by the mathematical properties of the network with the use of graph theory as shown in the following section.

3.3.1 Secondary Control Layer Resilience

In this section the resiliency of the MAS communication network in the secondary control layer against faults is analysed by modelling the underlying MAS communication network as a connected graph. First, the necessary elements of graph theory are introduced, followed by the design of the restoration service in the MAS platform.

3.3.1.1 Graph Theory Overview

Any communication network can be represented by a graph, which in turn has mathematical properties that can be analysed to check the connectivity of the network, regardless of its size. This is necessary to guarantee that the distributed control can maintain operation in the event of loss of any of its communication nodes.

A graph $\mathcal{G}$ can be represented by vertices $V_{\mathcal{G}}$ and edges $E_{\mathcal{G}}$ between them. They are represented by the equation:

$$\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}}) \tag{3.32}$$

For the set of vertices $V_{\mathcal{G}}$:

$$V_{\mathcal{G}} = \{\mathfrak{v}_1, \ldots, \mathfrak{v}_{\mathfrak{n}}\} \tag{3.33}$$

Elements of $E_{\mathcal{G}}$ are denoted as the pair of vertices the edge connecting $\mathfrak{v}_j$ from $\mathfrak{v}_i$:

$$(\mathfrak{v}_i, \mathfrak{v}_j) \in E_{\mathcal{G}} \tag{3.34}$$

For every graph $\mathcal{G}$ we can define the adjacency matrix $\mathcal{A}$ of size $\mathfrak{n} \times \mathfrak{n}$ with elements $\mathfrak{a}_{\mathfrak{ij}}$:

$$\mathfrak{a}_{\mathfrak{ij}} = \begin{cases} 1 & (\mathfrak{v}_{\mathfrak{i}}, \mathfrak{v}_{\mathfrak{j}}) \in E_{\mathcal{G}} \\ 0 & (\mathfrak{v}_{\mathfrak{i}}, \mathfrak{v}_{\mathfrak{j}}) \notin E_{\mathcal{G}} \end{cases}, \tag{3.35}$$

And the degree matrix $\mathfrak{D}$ of size $\mathfrak{n} \times \mathfrak{n}$ with elements $\mathfrak{d}_{\mathfrak{ij}}$:

$$\mathfrak{d}_{\mathfrak{ij}} = \begin{cases} \mathcal{N}_{\mathfrak{i}} & \mathfrak{i} = \mathfrak{j} \\ 0 & \mathfrak{i} \neq \mathfrak{j} \end{cases}, \tag{3.36}$$

where $\mathcal{N}_{\mathfrak{i}}$ is the amount of neighbours for each node $\mathfrak{v}_{\mathfrak{i}}$, in other words, the number of elements of $E_{\mathcal{G}}$ formed with $\mathfrak{v}_{\mathfrak{i}}$. These matrices are used to build the Laplacian matrix $\mathcal{L}$ of the graph, defined as:

$$\mathcal{L} = \mathfrak{D} - \mathcal{A} \tag{3.37}$$

The eigenvalues of the Laplacian, which have information about the graph, can be found by finding the roots of its L-polynomial, also known as characteristic polynomial, according to the following:

$$|\mathcal{L} - \lambda I| = 0 \tag{3.38}$$

where I is the identity matrix. The roots in terms of λ are the eigenvalues. Each L-polynomial has $\mathfrak{n}$ eigenvalues from λ_1 to $\lambda_{\mathfrak{n}}$. A fully connected graph only contains one eigenvalue equal to zero, given that the Laplacian matrix is singular. For the case of the union of disjoint sub-graphs with individual L-polynomials $\mathfrak{C}_{\mathfrak{s}}$ in the set $\mathcal{K}$ of sub-graphs, the resulting L-polynomial $\mathfrak{C}$ is:

$$\mathfrak{C} = \prod \mathfrak{C}_{\mathfrak{s}} \qquad \forall \mathfrak{s} \in \mathcal{K} \tag{3.39}$$

Given that each sub-graph contains the eigenvalue $\lambda = 0$ as a solution of the characteristic polynomial, the number of solutions $\lambda = 0$ for a given characteristic polynomial indicate the number of separate sub-graphs in the corresponding graph [119]. In other words, the smallest exponent of the L-polynomial reflects the number of separate networks in the system. If this exponent is 1, then the communication network is fully connected. The next subsection describes the mechanism implemented in the MAS to guarantee this mathematical property in the microgrid control under normal and faulty conditions.

3.3.1.2 Restoration Service

The proposed MAS control has this service installed to maintain the operation of the microgrid even during faulty conditions in one part of the system, which increases the reliability of the system overall, as the rest of the system remains in operation.

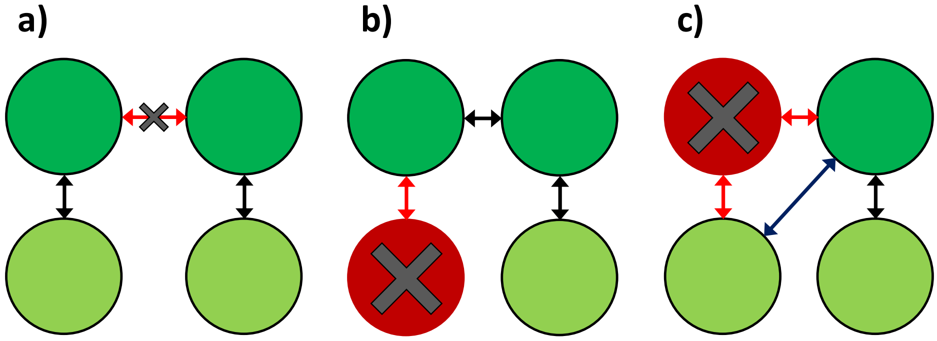


Figure 3.5: Graph representation of modes of failure of the MAS communication network covered by the restoration service. the functioning nodes are represented by the green nodes and in red the faulty nodes and communication links.

The fault-tolerance mechanism could be triggered in any of the following cases, which are represented in Figure 3.5: **a.** The communication link between containers is broken; **b.** Loss of agent container in the MAS; or **c.** Loss of the main container in the MAS.

For the first case, each side of the communication link will assume that the other side is no longer in the system. The side without a main container will launch a copy of the main container from the backup and assume leadership of the remaining container or containers in the network. This case results in two independent MAS applications working in the same microgrid.

In the second case, the primary controller keeps the last instruction sent by the secondary control at the corresponding lost container. The TCP/IP port would become available to be controlled by another container if there is a communication link available between the MAS and the primary control. If this is not the case, then the primary controller will follow the last instruction until the fault is cleared.

In the third case, the next backup in the UDP ring launches a copy of the main container and assumes the address of the original main container. In this scenario, only the main container backups in the UDP ring will be notified, while the transition is seamless for the rest of the control system. There is also no effect from the primary controller point view as this type of fault can only be detected from the UDP ring.

This mechanism allows for fault-tolerance without a central controller, which is a common argument against distributed and decentralised controllers in terms of reliability. The hierarchical control with all the mechanisms described for optimal power schedule and resilient operation will be implemented in the test case for its evaluation.

3.4 Simulation Test Case

First, a simulation model is presented to compare the droop control with the proposed approach. This test is done mainly to verify that the proposed control can achieve the same level of stability of the electric circuit, with the additional benefits of directly controlling the power references of the primary controllers, independently from the frequency of the electric circuit.

For this test, the frequency is set across the microgrid's voltage according to the power/frequency slope and the total power either received or sent by the main grid. The power of each DER source is then controlled based on the local frequency measurement and a frequency reference of $50Hz$ using a PI control. This is expressed by:

$$V_{DER} = V_{ref} - K_{droop_P}(f_{ref} - f) \tag{3.40}$$

$$\delta_{DER} = \delta_{ref} - K_{droop_Q}(-Q_{DER}) \tag{3.41}$$

Where V_{DER} is the DER voltage, V_ref is the reference voltage, K_{droop_P} is the active power PI gain, f_{ref} is the reference frequency, f is the frequency of the system, δ_{DER} is the DER phase angle, δ_{ref} is the DER phase angle reference, K_{droop_Q} is the reactive power PI gain, and Q_{DER} is the measured reactive power of the DER. This control rule is intended to follow the principle that in mainly resistive lines, active power depends on bus voltage, and the droop characteristic that frequency depends on power [10]. The reactive power control is added with a reference of 0, to account for the reactance component in the lines.

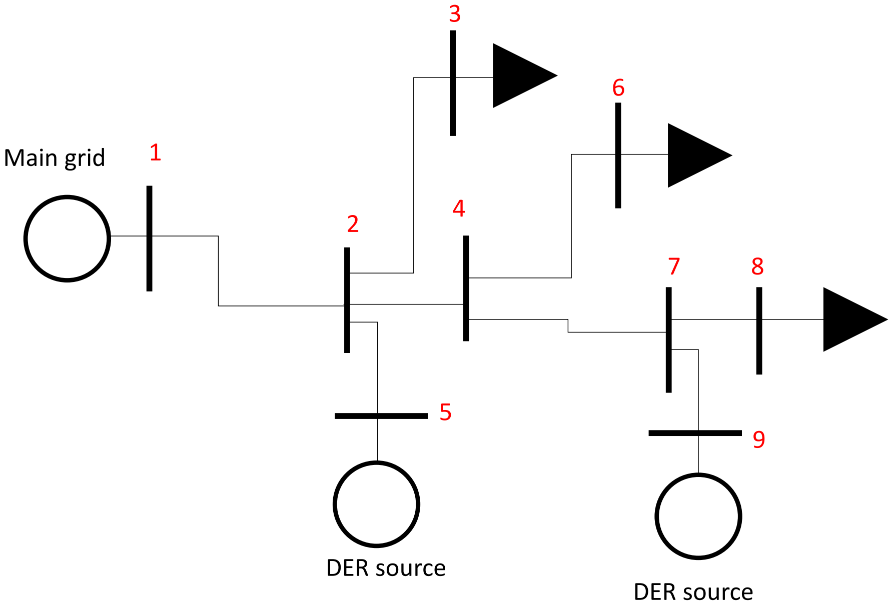


Figure 3.6: Circuit diagram for control comparison test.

Droop control		
Power reference of main bus	1.1kW	
Power/frequency slope	0.0015	
Active power PI	0.5	10
Reactive power PI	0.2	5
Proposed control		
Inner loop PI gains	0.2	12
Active power PI gains	0	10
Reactive power PI gains	0	5

Table 3.2: Simulation Parameters for control Comparison test.

The test case circuit diagram is shown in3.6. All of the lines shown have an impedance of $Z = 1 + i0.3\Omega$. The load at bus 3 starts at $2.3kW$ and then after 4 seconds drops to $2kW$, at $t = 5s$. The load at bus 6 has the same initial conditions, however, at $t = 2s$ its load drops to $1.7kW$. For the load in bus 8, the initial value is $2kW$, changing at $t = 6s$ to $1.4kW$. An additional $0.4kVARs$ load is added in each load bus. Table 3.2 shows the parameters used for the simulation.

Additionally, a real-time simulation model is presented in this section to test and validate the stability and resilient operation of a microgrid with the proposed control system. The simulation model consists in a real-time microgrid model of the physical layer and primary control layer implemented on an OPAL-RT real-time simulator, combined with a communication network built with two Raspberry Pis and a PC.

The real-time simulation is used to verify the viability of the control system in a much more realistic scenario than the offline optimisation and simulation. Additionally to this, the real-time interaction between the MAS platform and the real-time simulator allow to verify the ability of the control scheme proposed to remain stable under normal and faulty conditions.

Simulation results show a stable operation in terms of voltage and frequency in both conditions, which validates the stability analysis presented in this chapter. This is done to address the gap in the demonstration of real-time control system in real-time simulation [66].

The physical layer of the microgrid simulation model tested is composed of 3 controllable DERs, a Fuel Cell (FC) a Micro Turbine (MT) and a battery as the ESS as shown in Figure 3.7 with the same cost function parameters as presented in [5]. The percentages show the load distribution and the numbers represent the buses that are being measured in the next section. The power lines and load over time are the same as described in [5].

As mentioned previously, the MAS platform is realised in a network of one PC and two Raspberry Pi that are connected to an OPAL-RT real-time simulator. Each device hosts an agent container, and depending on the order in which they connect to the MAS platform, each contains the agent main container or a main container backup. The network formed by the devices forms an UDP ring between the main container and the main container back-ups as described in [5].

The power schedule generated during the simulation is the result of the optimisation process realised in the secondary control layer. while this is part of the simulation process, the methods used to generate the power schedule are explored in chapters 4 and 5 of this thesis.

Three 24-hours scenarios are run, in the first scenario, all the DERs operating continuously; in the second scenario, a fault is induced at the ESS secondary control at 12:00 to test the fault-tolerance mechanism of the secondary control layer and the stability maintained by the primary control layer; in the third scenario, a fault is induced in the main container to observe the restoration service.

The first scenario is intended to show, through real-time simulation, the stability achieved using the proposed control system, including the asymptotic stability of of the primary control layer when interacting with the secondary control layer. Additionally it shows the global stability of the microgrid, following the conditions set for the stability of the primary controllers interconnected by the electrical circuit.

The second scenario is intended to show that the conditions verified in the first scenario are still valid when one bus no longer receives power references and the primary controller affected is only set to maintain voltage stability. This case represents distributed controller faults, communication faults or simply components being disconnected from the microgrid. From the point of view of the secondary control layer, this scenario verifies the resilient operation of the secondary control layer when a container is lost or disconnected from the MAS platform.

The third case is designed to show the ability of the proposed control system to maintain resilient operation in the scenario that the main container of the MAS platform is lost unexpectedly due to communication or controller failure. This scenario shows that from the point of view of the primary control layer, these types of failure do not affect the operation of the microgrid's circuit layer.

These scenarios are selected to represent all possible types of failures of single components of the proposed control system to highlight the fault-tolerance ability and stability of said system. While multiple failures are possible, the probability of such events is considered too low compared to the case of single failure; and therefore are left out of the scope of the fault-tolerance design in this thesis.

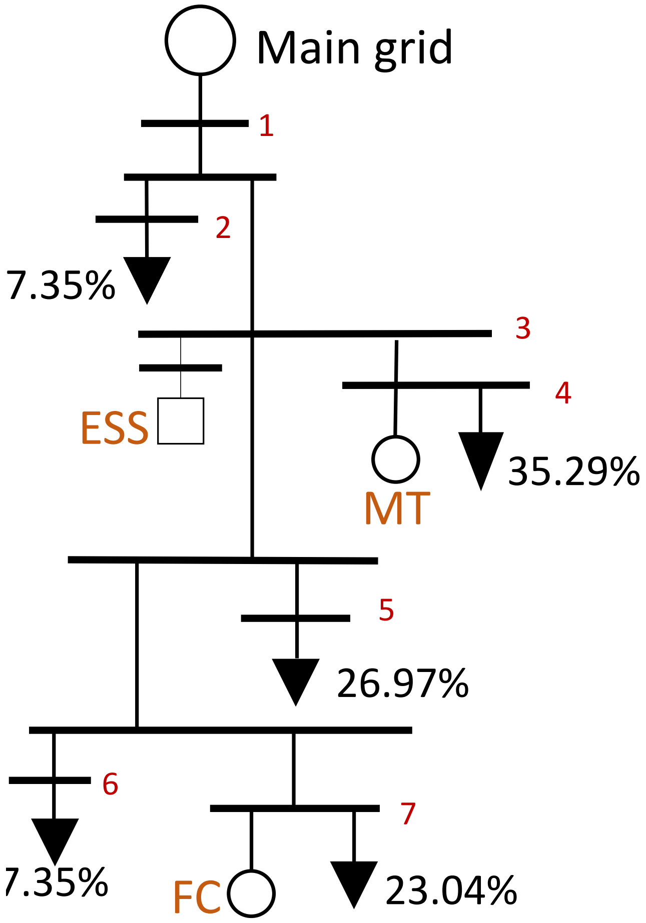


Figure 3.7: Microgrid Circuit test model.

3.5 Simulation Results and Discussion

3.5.1 Droop and proposed control comparison results

Results for the simulation test case are presented for the power response of each source and the frequency response comparison as shown in the figures 3.8, 3.9 and 3.10.

It can be seen that the proposed controller can achieve stability within similar time frames. For the case of the droop control, the power supplied by each source is dependant on the distance from the main grid. The reason is that the power is dependant on the bus voltage and frequency deviation. Additional considerations are

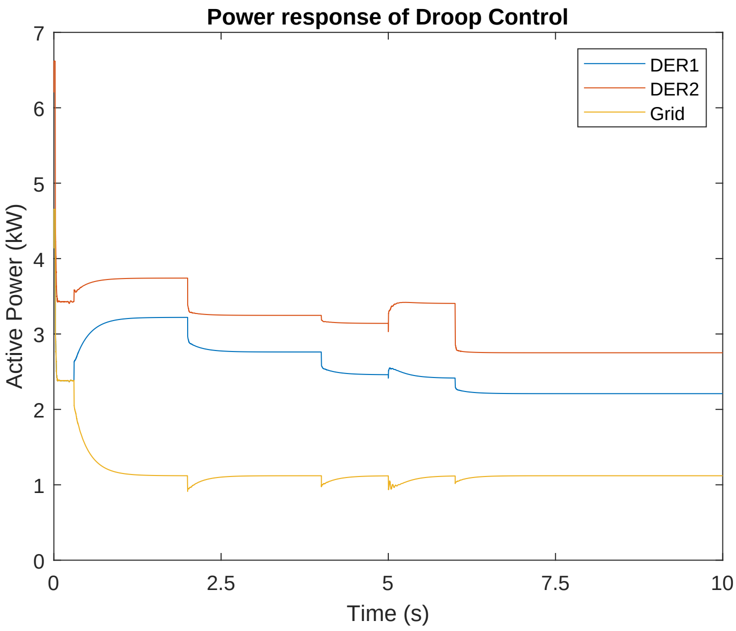


Figure 3.8: Active power response of the DERs and main grid in the droop control case.

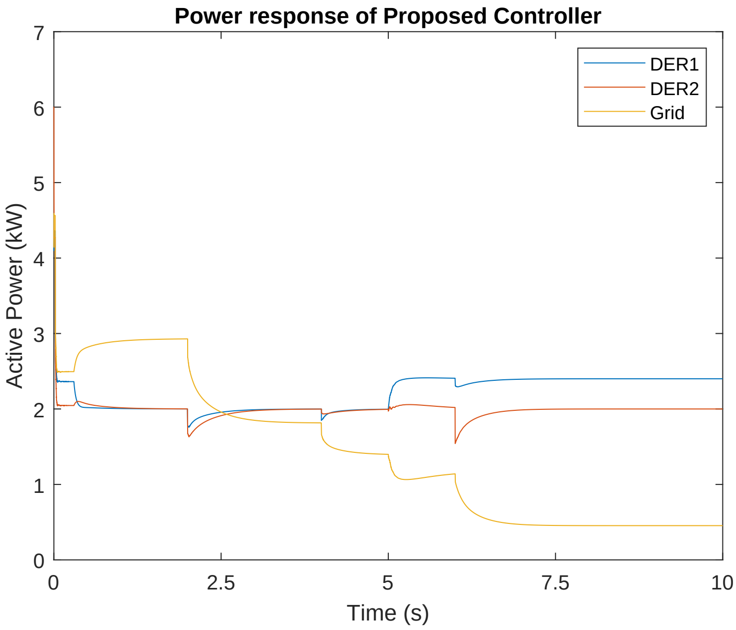


Figure 3.9: Active power response of the DERs and main grid in the proposed control case.

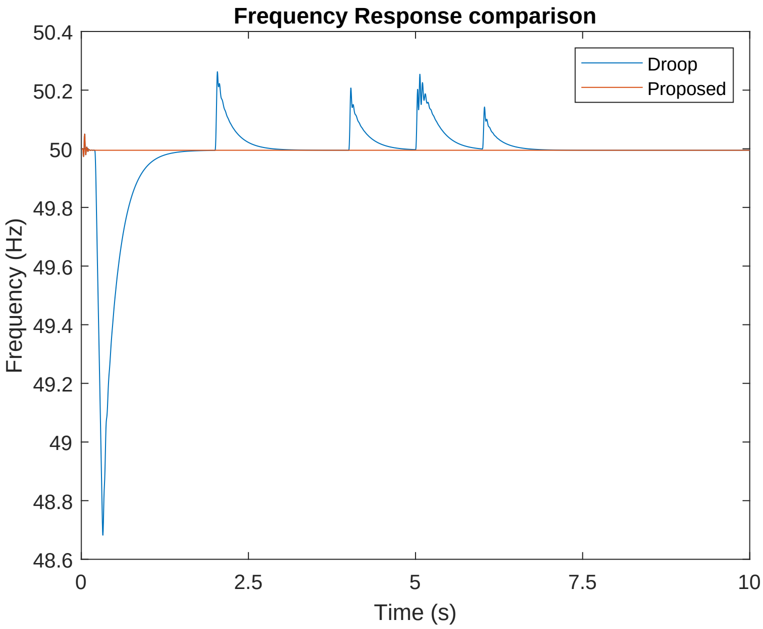


Figure 3.10: Frequency response comparison for the droop control and the proposed control for the same simulation case.

required to adjust power supplied by each DER, given a cost function or maximum power capacity for proportional power sharing.

For the case of the proposed controller, it can be seen that the DERs can be controlled to supply the same power (first 5 seconds of simulation) or different powers (last 5 seconds of simulation) directly by supplying the power references from the distributed secondary control. This can be done independently from the load, frequency and bus voltages connected to each DER.

This is useful in cases where it may become beneficial for the microgrid to temporary allow more power to be drawn from the main bus, for example, when the grid prices drops below the internal cost, and then simply adjust the power references when grid prices become larger and cost of internal generation, allowing cost minimisation, or in general, any other microgrid objective apart from power sharing.

It can also be seen that the proposed controller does not depend on the frequency, and can be set to maintain the required frequency throughout the simulation, allowing a much more stable frequency compared to the droop control.

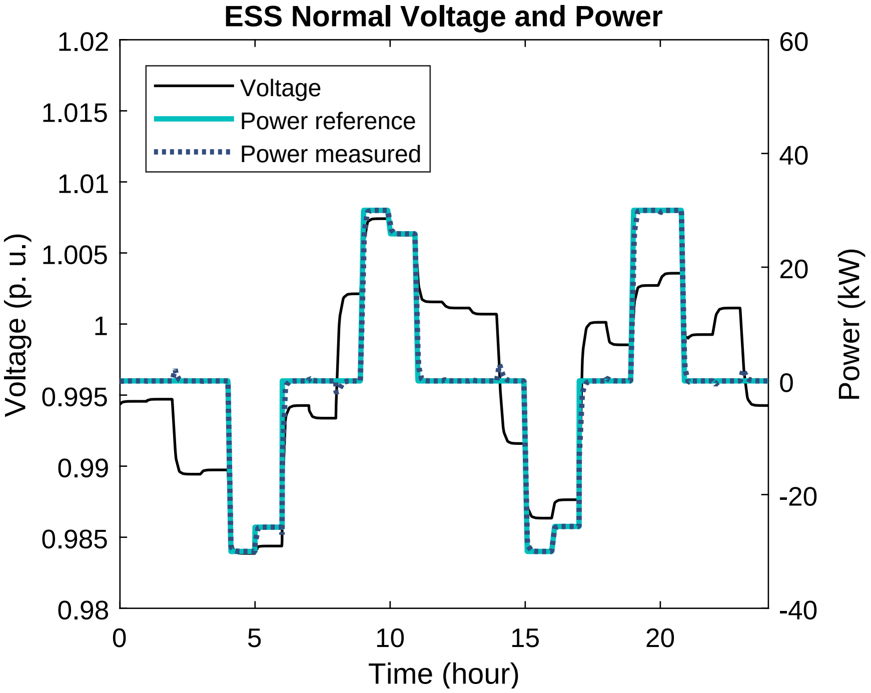


Figure 3.11: ESS control response under normal operation.

Results for the real-time simulation test case are presented for the system under normal conditions and the system under fault conditions. In all cases, the control system is capable of maintaining stability of operation.

3.5.2 System Performance Under Normal Conditions

It can be observed in Figure 3.11 that the primary control layer regulates the voltage according to the changes of the load and accommodates for the power references from the secondary control simultaneously.

As is can be seen in the Figure 3.12 the voltages for the buses across the microgrid remain within the current UK standard of voltage deviation tolerance of -6 % and +10% for the distribution circuit.

Each of the primary controllers adjust the bidirectional active power flow and maintains the reactive power and frequency constant by adjusting the phase angle as shown in Figure 3.13.

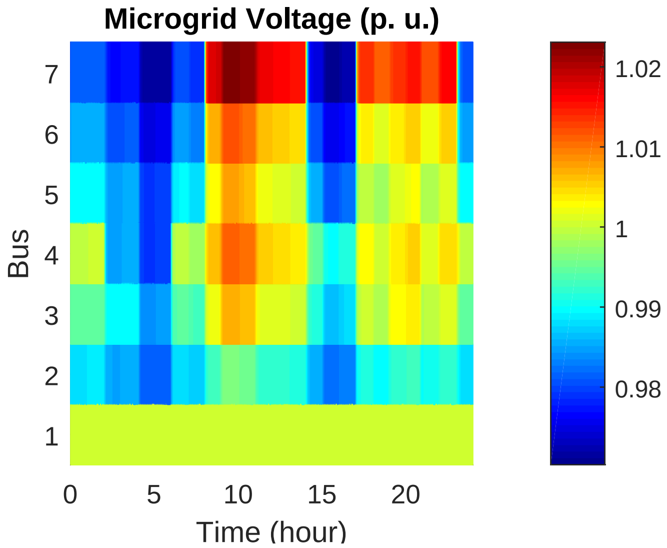


Figure 3.12: Microgrid normal Voltage response.

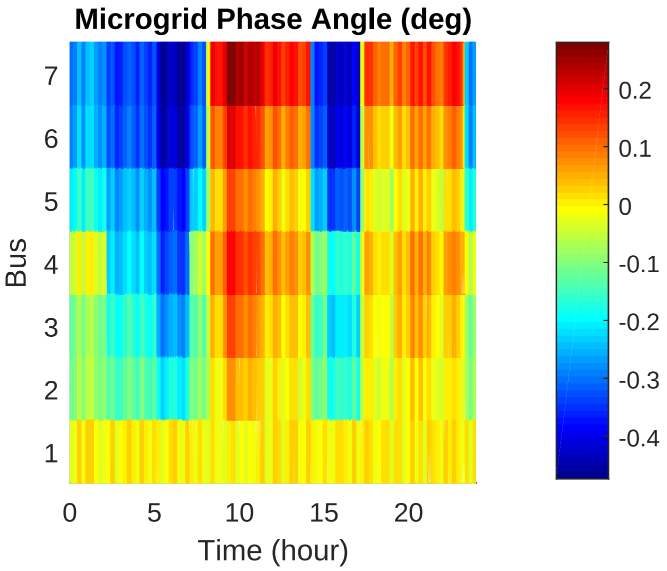


Figure 3.13: Phase angle response across the microgrid under normal operation.

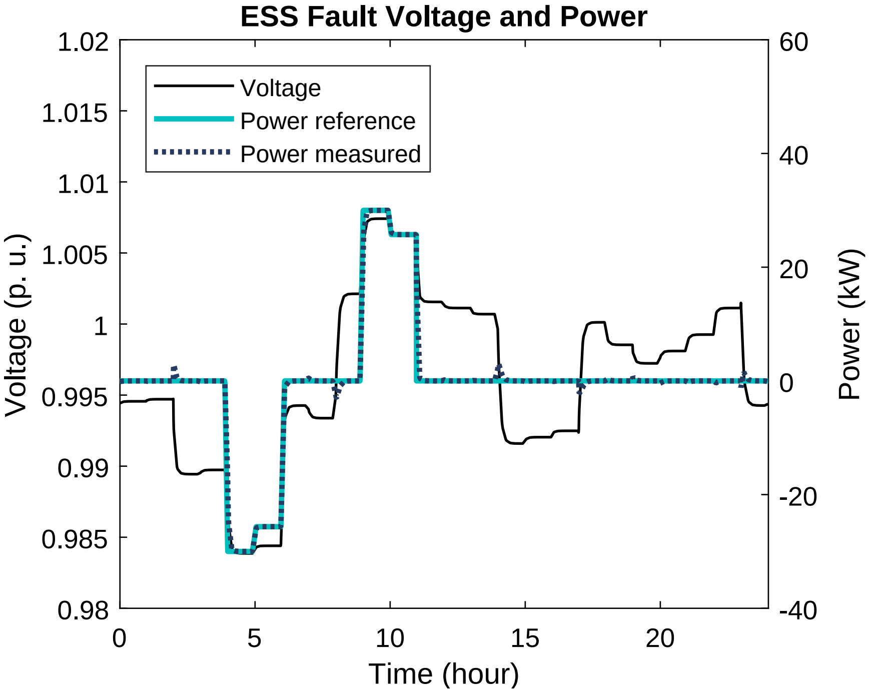


Figure 3.14: Secondary ESS control response with fault at noon.

For the secondary control layer, the calculation of the L-polynomial applying equation (3.38), yields:

$$\lambda^6 - 12\lambda^5 + 51\lambda^4 - 92\lambda^3 + 69\lambda^2 - 18\lambda = 0 \qquad (3.42)$$

Which has only one eigenvalue equal to zero, and therefore the communication network is fully connected.

3.5.3 System Performance Under Fault Conditions

For the fault case, the voltage variations are smaller as there is no power transfer from the ESS to the microgrid after the fault of the secondary control at noon. However, the voltage is stable for the rest of the day as seen in Figure 3.14.

Moreover, In Figure 3.15 it can be seen that the bus 3 voltage decreases as a result of the ESS not being used, and that the rest of the buses are adjusted for the remaining power flows. A similar adjustment can be seen for the phase angle in

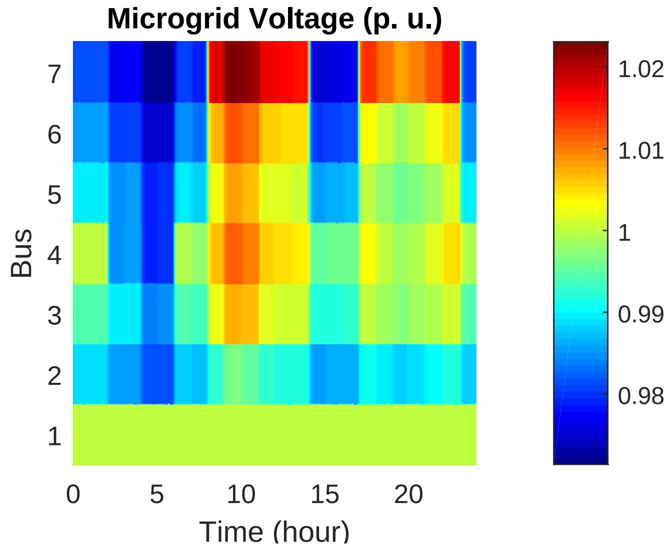


Figure 3.15: Microgrid Voltage response under fault conditions.

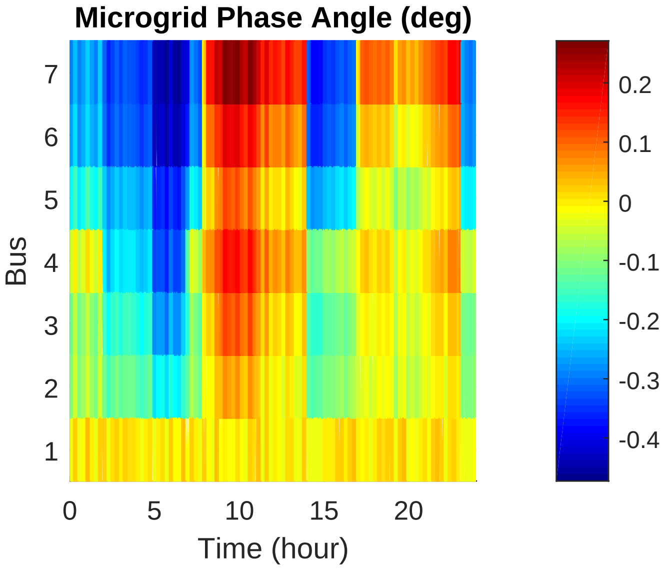


Figure 3.16: Phase angle response across the microgrid under ESS fault conditions.

Figure 3.16. The microgrid remains stable and within the UK standard range in both cases in the transient state and steady state for the entire simulation.

With the loss of the ESS container, the L-polynomial of the communication network of the secondary control becomes:

$$-\lambda^5 + 10\lambda^4 - 33\lambda^3 + 40\lambda^2 - 23\lambda = 0 \tag{3.43}$$

Which, as in the previous case only has one eigenvalue equal to zero and therefore indicates that the rest of the network remains fully connected. In other words, the distributed controllers that remain in the secondary control layer are able to communicate with each other, maintaining their ability to achieve optimal operation in the rest of the microgrid. In the case that the fault is cleared, the controller can rejoin the MAS platform and resume its operation. This demonstrates that the proposed model allows resilient operation under normal and faulty conditions.

3.5.4 Main-container Restoration

A fault of the main-container is induced in the system to test connectivity of the containers after applying the service restoration to the secondary control layer. From the L-polynomial calculated after the fault of the main-container, it can be seen that the network remains fully connected. As depicted in its graph in Figure 3.17. Activation of the service is only reflected in the ACL messages from the AMS, as the creation of the copy is informed to other back-ups as shown in Figure 3.18.

After the fault,the network reforms to exclude the faulty node, with an L-polynomial of:

$$-\lambda^5 + 8\lambda^4 - 20\lambda^3 + 8\lambda^2 - 5\lambda = 0 \tag{3.44}$$

which also has only one root equal to zero. This indicates that the restoration service prevents breakdown of the MAS system and therefore maintains stable operation of the secondary control layer. As no effect can be observed from the point of view of the physical layer or primary controllers, the real-time response of this case is the same as the system performance under normal conditions.

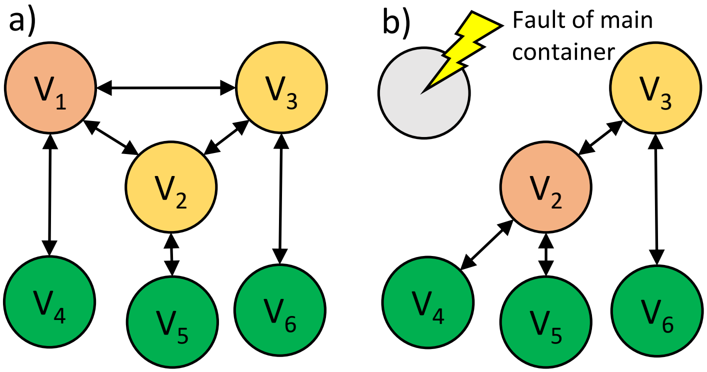
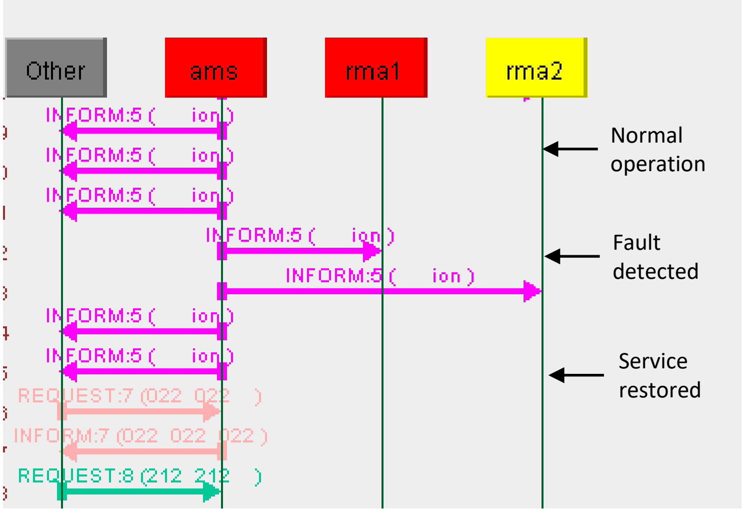


Figure 3.17: Graph of the MAS network. a) before and b) after main-container fault in orange, back-ups in yellow, and containers in green.



INFORM:5 (ion)
INFORM:5 (ion)
INFORM:5 (ion)
INFORM:5 (ion)
INFORM:5 (ion)
INFORM:5 (ion)
INFORM:5 (ion)
REQUEST:7 (022 022)
INFORM:7 (022 022 022)
REQUEST:8 (212 212)

Normal operation

Fault detected

Service restored

Figure 3.18: ACL messages of the AMS showing service restoration.

3.6 Summary

This chapter presented the methods to guarantee that the system is stable and reliable during normal and faulty conditions. The test cases in this chapter refer to the performance of the control system designed in terms of voltage regulation and fault-tolerance, for the primary and secondary control respectively.

The conditions for stability for the primary level control are analysed with the direct Lyapunov method considering both the resistive and reactive components in a short-line model. The system developed in this thesis has shown that it may continue stable operation within allowed margins even in events of DER disconnection and instantaneous changes in the load, which is equivalent to connection and disconnection of loads over time.

The control system has also shown that is capable to operate at real time and can transition between loads, within the UK standard of voltage variations for the mains grid with varying loads.

Simulation results of the secondary control layer shows that the control system is not computationally expensive even with the UDP fault-tolerance mechanism. However, the ping between the nodes, that is, adjusting the time tolerance of the restoration service, can be adjusted to accommodate the computational resources available and communication requirements.

Chapter 4

Forecast models for Optimal Operation in Distributed Control

4.1 introduction

As stated in previous chapter, a central theme of this thesis, and a crucial part of recent developments in smart grids technology, is the distributed nature of the underlying control scheme applied to the microgrid. In particular, distributed control schemes are a better match for a microgrid with distributed energy resources integrated, which is the system of choice in this thesis. On the other hand, the goal of any control scheme is to realise a technically and economically viable operation within the microgrid. To this end, any cost minimisation framework applied within the control scheme has to take account for the non-linear properties of the distributed generation resources (for example in form of start-up costs) and energy storage systems (for example in form of State of Charge) integrated within the microgrid.

While internal generation by itself contributes to alleviate congestion in the main grid and could contribute to maintaining the bus voltages within an acceptable range, for the grid-connected microgrids, there is the possibility to interact with the main grid in such a way the the internal generation is used to minimise the supply cost from the point of view of the microgrid.

Microgrids can also contribute to the efforts of electricity decarbonisation by integrating small-scale renewable generation, such as small-scale wind generation, in regions where large-scale renewable generation is not feasible for social, economic or environmental reasons.

However, in microgrid's real-time distributed control, for the case of price, information such as electricity price is not available to the microgrid before the microgrid applies its power schedule, which affect total cost over time. For renewable generation, in this case wind power generation, maintaining a reliable power schedule requires information about future wind speeds combined with controllable energy generation for regulation.

For this reason, the use of accurate forecasting methods becomes necessary to achieve optimal operation in the microgrid, and in this chapter, electricity grid price and wind speed forecast methods are presented. However, in order to maintain distributed control (as the control scheme of choices in this thesis), the forecast methods cannot cause single points of failure in the system, in other words, suitable forecast methods should be as independent as possible from external signals to maintain the benefits of distributed control in terms of reliability.

The forecast methods presented in this chapter are designed to address these requirements for realising real-time distributed control with the use of artificial intelligence. More specifically, two methods namely, auto-regression and artificial neural networks, are presented in this chapter for purposes of developing forecast models for both electricity price and wind speed, which are fast enough to be deployed in a distributed control system.

For the price forecast methods, the auto-regression method is combined with the MCMC method. In all cases, the parameters of the AR methods are calculated as the solution for the least squares minimisation errors of the models, with the use of the Quasi-Newton method. The ANN method is the NARNET architecture. These forecast methods are chosen for their ability to generate the forecast signal without the need of external signals that could centralise the control system and in turn generate a single point of failure, and because these methods offer a good trade off between accuracy and computational power, that allows them to be deployed in a distributed communication network and operate in the real-time simulation environment.

While many efforts have been channelled to develop forecast models with high accuracy, there are very few studies about the statistical significant difference between the presented wind speed forecast models. In other words, many authors often do not explain if the difference in performance is due to difference in the actual model architectures, size of the models, or if the difference is due to other factors, such as randomness in the data used [16, 76, 78, 85, 86, 88, 120–128].

This chapter presents suitable wind speed forecast models for distributed control. The performance of the models is analysed in terms of total error and statistical significance difference between the models, considering a suitable PDF of the errors given that the inherent variations of wind speed, accentuated by the sampling rate of the data, generate heavier tails than the normal distribution.

As mentioned previously, the determination of statistical significance difference could be as important as the performance of the forecast model. The statistical evaluation method in this thesis is based on the Diebold-Mariano (DM) test, which is used to compare the performance of two forecast models between them and against true values [129]. In this case, the DM-test is applied to historical wind speed data and the wind speed forecast models. With this comparison, it is possible to determine if the difference in performance between two models is due to a significant difference between the architecture of each model and the size of the models, or if the difference is due to randomness in the data.

The DM-test can also be used to compare several runs of similar models, especially in the case of NARNETs that may produce different results for the same parameters and data set, due to the inherent randomness in the training and testing stages of the model.

While in previous works a statistical significance difference test was carried out, the actual parameters of the compared ANNs, which define the amount of calculations required to realise a forecast, are not mentioned. Therefore, the models are not compared in terms of computational requirements [126], whereas it is also important for the application of distributed control to have the most accurate model at the lowest computational requirement possible. In [130] no variation of the activation function, (the non-linear function of the ANN's layers that allows the ANN to learn non-linear patterns) was introduced, which limits the evaluation of the ANN forecast model. Similar limitations in the wind speed forecast model comparison are found in [99].

The statistical evaluation has been carried out previously with electricity prices using Non-linear Auto-Regression with Exogenous Variables (NARX) neural networks [89, 131], and for the statistical comparison of solar irradiance forecast models [132]. However, the NARX model is dependant on external signals for the exogenous variables, which make this model unsuitable for distributed control, as opposed to the NARNET which performs the forecast with local information.

In the case of DM-test in the context of wind speed forecast, while presented in [77], there was no consideration in the PDF of the error distribution,while in [65, 133] the DM test of the error residuals was applied to a normal distribution when converted into probability of the hypothesis tested.

This chapter includes the further development of the forecast models presented in [5, 7], applied to wind speed forecast suitable to be implemented in a distributed control setting at a local distributed controller level. The forecast models are therefore part of the secondary control of a distributed control system applied to a hierarchical microgrid control [40] (seen in Figure 3.1).

This chapter includes the further development of the forecast models presented in [5, 7], applied to wind speed forecast suitable to be implemented in a distributed control

setting at a local distributed controller level. The forecast models are therefore part of the secondary control of a distributed control system applied to a hierarchical microgrid control [40] (seen in Figure 3.1). To realise the operation of distributed control scheme, distributed controllers need to dedicate computational resources to maintain the communication network connecting the controllers. Additionally to this, to achieve real-time control, the distributed controller must have access to the forecast signals within a few milliseconds. In consequence, the forecast models are required to use the least amount of computational resources, while maintaining the highest accuracy possible. For this reason, the amount of weights of a forecast model is of interest as a point of comparison for models with a similar performance in terms of total error.

To validate the forecast models, the accuracy of the models is measured in the appropriate test case in terms of Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), while the total cost of operating the microgrid is analysed in combination of the cost optimisation process presented in the next chapter. For the case of the grid price forecast methods, they are compared in this chapter in terms of total error, combined with the cost minimisation achieved, which will be explored in the next chapter. For the case of the wind speed forecast models, they are evaluated in terms of RMSE with a persistence model and the ARMA model as a point of reference [97]. Additionally, the DM-test is evaluation is also performed for all the wind speed forecast models.

4.2 Price Forecast Method

To achieve supply cost minimisation in the microgrid, it is necessary that the distributed internal generation is used only when the accumulated internal cost is smaller than the cost of using energy directly from the grid. Given that the DERs have non-linear components in the cost with respect to time, such as start-up cost, it becomes necessary that the electricity spot price is known in advance, which in a distributed control framework is translated as the necessity to integrate forecast methods into the control scheme that are compatible with said distributed paradigm.

Two families of methods that are suitable for such distributed control framework are presented next for price forecast, both of which are AR based as introduced in chapter 2.

4.2.1 Weighted Average Model

This method is an algebraic combination of an auto-regression component of the electricity prices, equivalent to the "short-term memory" of other AI based methods; and the use of the MCMC simulation to add a statistical component to the price forecast which serves as the "long-term memory" of the method, as it summarises the historical behaviour of the price data in a few statistical term. The contribution of each of the terms used in this method is defined by the weighted average of terms, which is optimised to minimise error using the Quasi-Newton method [79].

In this price forecast method, the MCMC method is used to estimate the parameters θ_e of a PDF $\mathcal{P}$ given a data set $\mathcal{D}$, in this case, estimate the mean electricity price $\overline{\pi}_e$ and its corresponding standard deviation estimate σ_e, by the formation of a chain $\Theta = [\theta_1 \dots \theta_{n_e}]$. Following the Metropolis-Hasting algorithm, from a point θ_j, and a proposed set of parameters $\theta'_j = \theta_j + \theta_r$, where θ_r is a random variation in any direction, the next element is selected as:

$$\theta_{j+1} = \begin{cases} \theta'_j & \mathcal{Z} < e^{(Ln(\alpha))} \\ \theta_j & otherwise \end{cases} \tag{4.1}$$

Where $\mathcal{Z}$ is a random variable with uniform distribution in the range $[0, 1]$, and α is the acceptance ratio, defined as:

$$\alpha = \frac{\mathcal{P}(\mathcal{D}/\theta')\mathcal{P}(\theta')}{\mathcal{P}(\mathcal{D}/\theta)\mathcal{P}(\theta)} \tag{4.2}$$

where $\mathcal{P}(\mathcal{D}/\theta')$ and $P(\mathcal{D}/\theta)$ are the likelihoods of θ' and θ, given the price data $\mathcal{D}$, $\mathcal{P}(\theta')$ and $\mathcal{P}(\theta)$ are the priors on θ' and θ. The priors are the parameters evaluated in the normal distribution with respect to θ_0, which is selected empirically as the starting point of the chain Θ.

The chain continues to grow until the change in its average is below a given tolerance, in this case 10^{-6}, or maximum number of iterations of 4000, this stop criterion is used to guarantee convergence. Finally, the estimated parameters are calculated as:

$$[\overline{\pi}_e \quad \sigma_e]^T = \theta_e = \frac{1}{n_e}\sum_j \theta_j, \qquad \forall \theta_j \in \Theta \tag{4.3}$$

Where n_e is the number of elements θ_j in the chain, and θ_e is the vector containing the estimated parameters that describe the PDF of the electricity price. It is important to notice that given that the elements of the chain are vectors, the summation is done as an element-wise operation.

The estimated price π_i using this method is obtained as follows:

$$\pi_i = w_{am_1} p_{i-24} + w_{am_2} p_{i-168} + w_{am_3}\overline{\pi}_a + w_{am_4}\overline{\pi}_b + w_{am_5}\overline{\pi}_c \tag{4.4}$$

where p_{i-24} is the price for the previous day, p_{i-136} is the price for the previous week, $\overline{\pi_a}$ is the average price for the same day of the week, $\overline{\pi_b}$ the average price of the season and $\overline{\pi_c}$ is the average price of the entire data.

The averages $\overline{\pi_x}$ are estimated from the UK data set using the MCMC with Metropolis-Hastings method, and the weights w_j are obtained by solving a least squares regression problem:

$$\min_{W_{am}} \sum\nolimits_i (p_i - \pi_i(W_{am}))^2, \forall i \in \mathcal{D} \tag{4.5a}$$

$$\tag{4.5b}$$

where W_{am} is a vector containing the weights of (4.4) for the hours i in the data set $\mathcal{D}$.

While there are many different MCMC methods [79, 134], the method used in this thesis is the Metropolis-Hasting method. This MCMC sub-method describes the acceptance ratio that generate the elements of the Markov chain:

$$\alpha = \frac{\mathcal{P}(D/\theta')\mathcal{P}(\theta')}{\mathcal{P}(D/\theta)\mathcal{P}(\theta)} \tag{4.6}$$

where a is the acceptance, $\mathcal{P}(\mathcal{D}/\theta')$ is the likelihood of the proposed θ', given the data $\mathcal{D}$, $\mathcal{P}(\theta')$ is the prior on θ', $\mathcal{P}(\mathcal{D}/\theta)$ is the likelihood of the current θ and $\mathcal{P}(\theta)$ the prior on θ.

Given that the likelihood is computed as the product of all the evaluation of the data under the PDF with the current and proposed parameters, the computation may underflow, therefore the log-likelihood of the acceptance is taken instead. The next element θ_{j+1} of the Markov chain is then chosen as:

$$\theta_{j+1} = \begin{cases} \theta_j' & \mathcal{Z} < e^{(Ln(\alpha))} \\ \theta_j & otherwise \end{cases} \tag{4.7}$$

where $\mathcal{Z}$ is a random number taken from a uniform distribution function. After a long enough chain is generated, the final parameters for the posterior are Monte Carlo approximated as:

$$\theta_e = \frac{1}{n_e} \sum \theta_j, \qquad \forall \theta_j \in \Theta \tag{4.8}$$

where n_e is the number of elements in the chain $\Theta = [\theta_1 \ldots \theta_{n_e}]$, and θ_e contains the mean hour price and standard deviation estimation.

MCMC stop criterion: To optimise computational resources of the control system during real-time operation, two stop criterion are added to the MCMC method.

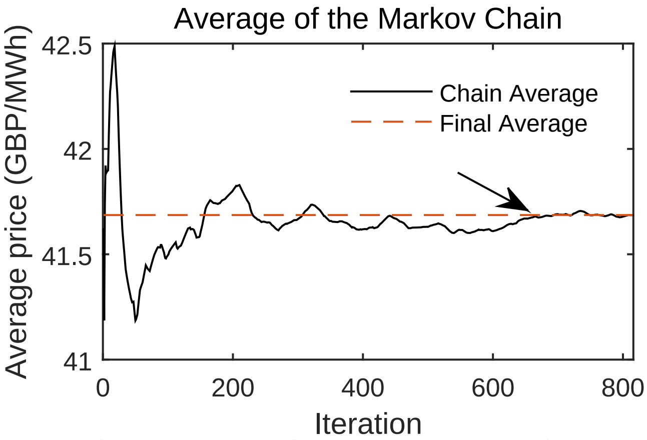


Figure 4.1: Evolution of the average of the markov chain for the price of hour 1, the arrow shows the end of the burn-in stage.

The first stop criterion limits the relative change of the average value of the Markov chain after each iteration to a minimum of 1×10^{-6}, therefore the chain stops growing when the following inequality is true:

$$\frac{|\overline{\pi}_j - \overline{\pi}_{j+1}|}{\overline{\pi}_j} < 10^{-6} \tag{4.9}$$

The second stop criterion is reaching maximum cycles per calculated hour price, in this study the limit is empirically chosen as 4000 for each element estimated using this method.

In Figure 4.1 the convergence of the average of the markov chain is shown for the price estimation of one hour. The arrow points the end of the burn-in stage, elements in this stage are discarded for the price estimation.

4.2.2 NARNET model

The second family of AR methods for the price forecasting is the NARNET. As the NARNET allows more variation in the ways the mathematical price forecast model is constructed, as well as the variation of the training parameters. This section will review the different configurations of the NARNET, referred as architectures.

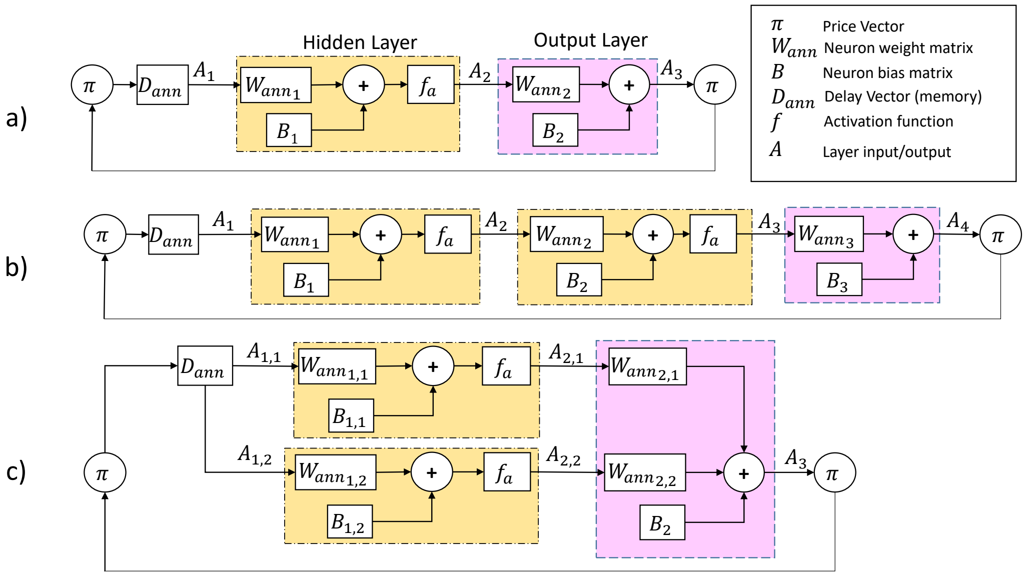


Figure 4.2: NARNET architectures used for the grid price forecast method.

As mentioned in chapter 2, the NARNET model is a type of ANN that considers past values of some input to generate the next values of said input, which is considered as part of a time-series. This network then produces the forecast based only on past values of the time series, in this case, past values of electricity grid prices. For this forecast method, the day prices are treated as the input vector for the ANN, which in turn outputs the next day price prediction, using a given delay size. The delay matrix inside the NARNET serves as the "short-term memory" while the neurons serve as the "long-term memory" equivalent in other AI based methods.

In this study the connections between the layers is defined by three different architectures tested shown in Figure 4.2, which are the **a)** single hidden layer, **b)** the hidden layer in series and **c)** the parallel hidden layers.

These architectures can also be expressed as transfer functions, as presented in chapter 2, as follows:

The general transfer functions of the entire NARNET, with d delay size, $\mathfrak{q} \times 1$ input size and $\mathfrak{u}$ neurons in each layer is described as follows for the single hidden layer [7, 98]:

$$F_{single} = W_{\mathfrak{q}\times\mathfrak{i}}(f_a(W_{\mathfrak{u}\times d\mathfrak{q}} D_{ann,d\mathfrak{q}\times 1} + B_{\mathfrak{i}\times 1})) + B_{\mathfrak{q}\times 1} \tag{4.10}$$

For the series hidden layers:

$$F_{series} = W_{\mathtt{q}\times\mathtt{u}}(f_a(W_{\mathtt{u}\times\mathtt{u}}(f_a(W_{\mathtt{u}\times d\mathtt{q}}D_{ann,d\mathtt{q}\times 1}) + B_{\mathtt{i}\times 1}) + B_{\mathtt{q}\times 1})) + B_{\mathtt{q}\times 1} \tag{4.11}$$

For the parallel hidden layers:

$$\begin{gathered} F_{parallel} = W_{\mathtt{q}\times\mathtt{u}}(f_a(W_{\mathtt{i}\times d\mathtt{q}}D_{ann,d\mathtt{q}\times 1} + B_{\mathtt{i}\times 1}))+ \\ W_{\mathtt{q}\times\mathtt{u}}(f_a(W_{\mathtt{u}\times d\mathtt{q}}D_{ann,d\mathtt{q}\times 1} + B_{\mathtt{u}\times 1})) + B_{\mathtt{q}\times 1} \end{gathered}$$

In equations (4.10), (4.11) and (4.2.2), the subscripts show the size of each matrix to illustrate the complexity of the NARNET, in particular from the matrix product $\omega_{\mathtt{i}\times d\mathtt{q}}D_{ann,d\mathtt{q}\times 1}$.

Recalling that D_{ann} is the delay vector, or "memory" of the NARNET, defined as:

$$D_{ann}(t) = [p_{t-1}, \ldots, p_{t-d}]^T \tag{4.12}$$

As the NARNET operates, the values in D_{ann} shift positions with the next time step, eliminating the oldest information first, while D_{ann} is updated through the network's feedback loop.

For the MAS application, an agent implements one of the NARNET models or the WA model for the price forecast, which is then sent to the corresponding DER or ESS agent to generate the power references for the primary control. The performance in terms of total cost of using either of the forecast models is analysed in chapter 5.

The NARNET models used for price forecast in this thesis are trained using the Levenberg-Marquardt back-propagation algorithm, by randomly separating 15% of the time steps for training, 15% for validation and the rest for evaluation, according to the training algorithm presented in chapter 2.

4.3 Wind Speed Forecast Method

One of the most important factors that limit the amount of renewable generation used, compared to conventional generation, is the reliability that the energy is available as scheduled. Accurate power schedule from renewable generation is therefore necessary to further promote this type of electricity generation sources.

As mentioned previously, the forecast methods in this section are suitable for wind speed forecasting in a distributed control architecture for microgrids. All of the methods presented in this section are ultimately data-driven and differ in how the data is handled in each case, which would generate different output forecasts outputs. These models will be analysed in terms of accuracy and statistical significance difference.

The forecast methods in the next subsections are used to model the wind speed as follows:

$$Y_{ws}(t) = F(t) + \epsilon(t) \tag{4.13}$$

where Y_{ws} is the historical wind speed and t is time, F represents one of the forecast model in the following subsections and ϵ is the error at time t, in other words, the difference between the real wind speed and the wind speed forecast for the same t. Each of the forecast models are described next.

4.3.1 Persistence model

The persistence model is the simplest forecast model and is defined as:

$$F_1(t) = Y_{ws}(t-1) \tag{4.14}$$

where F_1 is the wind speed forecast. The model is basically the assumption that the difference between one sample and the next is small when the sampling frequency is sufficiently small, and therefore the wind speed one step ahead will remain mostly unchanged. The effectiveness of the persistence model depends on the location and sampling frequency. This model would not be suitable for generating forecasts with time horizons larger than one hour. This model will be used as a baseline to compare the Neural Network models, with the use of the DM-test.

4.3.2 ARMA model

This method was been recently used in wind speed forecast in scenarios of limited information, making it suitable for distributed control systems, and therefore is useful as point of comparison for other forecast models [97].

This model combines the Auto-regression model with a correction of the average depending on the error obtained in previous evaluations of the model. The model is defined as follows:

$$F_2(t) = \mathcal{W}_0 + \sum_{i\in\mathcal{P}} \mathcal{W}_a(i) Y_{ws}(t-i) + \sum_{j\in\mathcal{Q}} \mathcal{W}_b(j)\epsilon(t-j) \tag{4.15}$$

where F_2 is the ARMA model, $\mathcal{W}$ are the model parameters, $\mathcal{W}_0$ is a constant, $\mathcal{W}_a$ is a vector containing the weights for the auto-regression model and $\mathcal{W}_b$ contains the elements for the moving average model, $\mathcal{P} = \{1, \ldots, \mathcal{P}_{ARMA}\}$, $\mathcal{P}_{ARMA}$ refers to the order of the auto regression component, $\mathcal{Q} = \{1, \ldots, \mathcal{Q}_{ARMA}\}$ and $\mathcal{Q}_{ARMA}$ is the order of the moving average model. The parameters of this model and the AR model are obtained with the Quasi-Netwon method, which will be discussed in the next subsection. This model will also be used as a point of comparison against the other models in terms of accuracy and with the use of the DM-test.

4.3.2.1 Auto-regression Model

This model also produces the forecast of the wind speed only with past values. with the difference that it does not require the information of the previous errors. The advantages of this method are the convergence of the solution and the quicker generation of the solution compared to the NARNET.

This forecast model is defined as follows:

$$F_3(t) = \sum\nolimits_i \mathcal{W}_c(i) Y_{ws}(t-i), \qquad \forall i \in D_{AR} \tag{4.16}$$

where F_3 is the AR wind speed forecast, realised by the weighted sum of wind speed samples Y_{ws} with a lag order, or delay size, d_{AR}. In 4.16 the set $D_{AR} = [1 \ldots d_{AR}]$ multiplied by the vector containing the W_i optimised weights, from $Y_{ws}(t - d_{AR})$ to $Y_{ws}(t-1)$.

The weights $\mathcal{W}$ are optimised with the Quasi-Newton method by solving the following least squares problem:

$$\min_{\mathcal{W}} \sum\nolimits_t (Y_{ws}(t) - F_k(t))^2, \forall t \in T \tag{4.17}$$

where the set $T = [1, \ldots, N_T]$ represents the times at which the historical data is taken, for N_T total available samples, for the k-th forecast model.

In equation 4.17, all the weights are tuned to minimise the total summation of the squared errors applied to the entire set of historical sample data available.

In this case, the fitting algorithm of the model starting with all the weights set to 1 and the weights are updated in the direction of highest decrease of equations 4.17 and 3.33, also known as the cost function, until the stop criterion is met. The stop criterion is met in this case in any of the following cases: **1)** the cost function is equal to 0. **2)** The change in the cost function is equal or below the $1x10^{-5}$. **3)** The change in the weights is equal or below $1x10^{-6}$. **4)** Iterations have reached a maximum of 1200.

4.3.3 NARNET model

The NARNET for wind speed forecast is the same model described previously for the price forecast method. The difference between these models are the configuration and sizes of the inputs and parameters of the ANN, and that the input of this

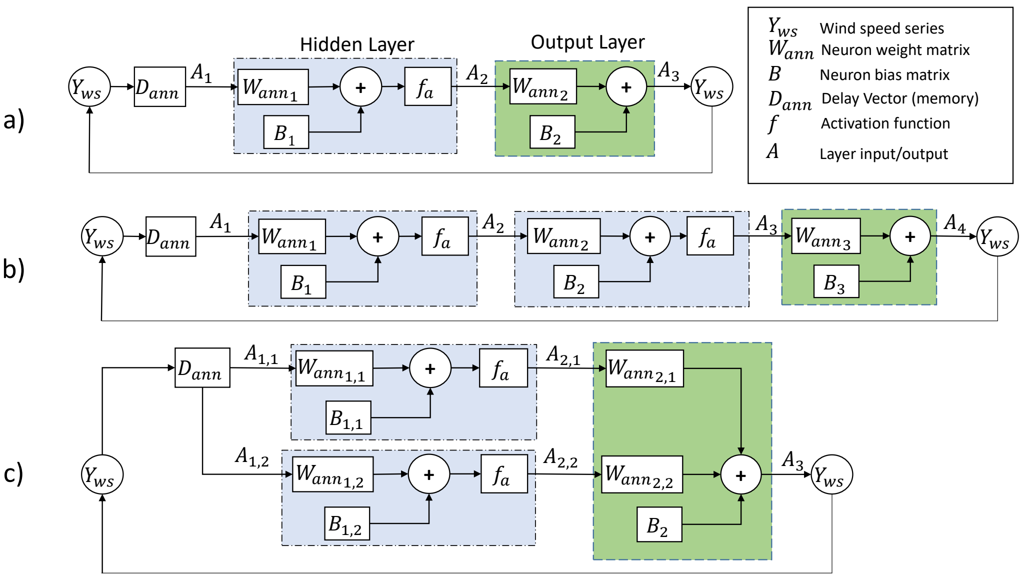


Figure 4.3: NARNET Architectures used for wind speed forecast. a) single hidden layer. b) series hidden layers. c) parallel hidden layers.

NARNET is a scalar, in other words, it receives the wind speeds for every time steps as opposed to the vector input of the price forecast method. The architectures used for wind speed forecast are illustrated in Figure 4.3.

4.4 DM-test for Forecast Model Evaluation

The DM-test is used in this thesis for hypothesis testing, in this case, the DM-test is used to calculate the probability of the null hypothesis. Before presenting the calculation required to perform the test, a conventional evaluation of the wind speed forecast methods is realised, then each wind speed forecast method is compared against the others and the true wind speed values to determine if the differences in accuracy are due to randomness in the true data set or because a forecast model is actually more accurate than others.

To compare the forecast models using the DM-test, each model is first evaluated in terms of RMSE, following the results from [7]. The RMSE is defined as follows for the k-th forecast model:

$$RMSE = \sqrt{\frac{\sum_t (Y_{ws}(t) - F_k(t))^2}{N_T}}, \forall t \in T \tag{4.18}$$

Then, the wind speed forecast models are analysed in terms of statistical significant difference, repeating the experiment and performing the DM-test of every realisation of each of the models presented in the previous section.

Here, the null hypothesis is that the forecast models have the same accuracy, and the difference in the performance is due to randomness in the dataset, or because the compared models are nested. A model F_x is nested in F_y when $F_y = F_x + F_z$, for any F_z function. Therefore, a rejection of the Null hypothesis concludes that there is a statistically significant difference between the models and a better performance is due to a better model.

The DM-test is essentially based on the ratio of the mean and covariance of the errors, encapsulated in the difference of errors between models. The DM-test is based on the principle that the errors from a valid forecast model are covariance stationary, which means that the residuals, or errors, of a valid model should behave as a random series. This test also assumes that the time series requiring forecast is infinite, and therefore, any set used for test calculation is a sample of the entire series.

The test requires 3 sets as inputs and provides a probability value (p-value) that the null hypothesis is true. The inputs are the true wind speed Y_{ws}, and the output of the two models that are being compared. When the p-value is sufficiently small, the null hypothesis is rejected. When this occurs, a statistically significant difference in accuracy between the forecast models is concluded.

From each pair of forecast model outputs F_a and F_b, the squared errors $\mathbf{e}$ are calculated as:

$$\mathbf{e}_a(t) = (Y_{ws}(t) - F_a(t))^2, \forall t \in T \tag{4.19a}$$

$$\mathbf{e}_b(t) = (Y_{ws}(t) - F_b(t))^2, \forall t \in T \tag{4.19b}$$

The loss differential function d is defined as:

$$d(t) = \mathbf{e}_a(t) - \mathbf{e}_b(t) \tag{4.20}$$

From this new set, the sample mean of the differential $\overline{d}$ is obtained. To determine if a deviation from zero of this value is statistically significant, this value is divided by an estimation of the standard deviation of the differential series.

Next, the autocovariance γ of each lag up to h is calculated:

$$\gamma(h) = Cov(d(t), d(t+h)) \tag{4.21}$$

where Cov is the covariance function. Then, the variance is estimated as [129]:

$$v_d = \frac{\gamma(0) + 2\sum_i \gamma(i)}{N_T}, \forall i \in H \tag{4.22}$$

where $H = [1 \dots h]$, which represents the set of lags of the autocovariance. Finally, the DM-test output is calculated as:

$$DM_0 = \frac{\overline{d}}{\sqrt{v_d}} \tag{4.23}$$

This test is adjusted using the Harvey adjustment, which corrects the test for heavy-tailed distributions of the loss differential. This correction is used because the data set contains outliers with respect to the normal distribution function given the nature of wind speed, therefore, the normal distribution function does not accurately describe the loss differential distribution [129]:

$$DM_1 = \sqrt{\frac{(N_T + 1 - 2h + N_T^{-1}h(h-1))}{N_T}} DM_0 \tag{4.24}$$

The p-value p is obtained by calculating the student's-t Cumulative Distribution Function (CDF) of DM_1:

$$p = \int_{-\infty}^{DM_1} \frac{\Gamma(\frac{K+1}{2})}{\sqrt{K\pi}\Gamma(\frac{K}{2})}(1 + \frac{\mathbf{t}^2}{K})^{-\frac{K+1}{2}} d\mathbf{t} \tag{4.25}$$

where $\Gamma(\cdot)$ is the Gamma function, $\mathbf{t}$ is the integration variable and K is the number of degrees of freedom [135]. For this test, $K = N_T - 1$. p represents the probability that the null hypothesis is true. In this case, if $0.05 \leq p < 0.10$ the significance is regarded as weak and if $p < 0.05$ the forecast models are regarded as statistically significantly different, in either of these cases, the null hypothesis is rejected. The null hypothesis is not rejected in any other case.

Given that a specific time horizon requirement in a microgrid application depends on factors such as available storage and the cost functions of other resources, the forecast methods will be evaluated at the prediction of one step ahead, such that is possible to realise the model evaluation between any combination of two models. While it is possible to increase the time horizon as done in [7], all of these methods continuously update the forecast to always be one step ahead as time progresses.

4.5 Test Cases

The test cases in this section are used to evaluate the accuracy of the proposed price and wind speed forecast methods. The price forecast models are evaluated mainly in terms of total RMSE and MAPE. For the case of wind speed forecast, the DM-test evaluation is included for these methods as well. As the total cost achieved by the microgrid power management system also depends on the optimisation capabilities of such system, the cost analysis is analysed in chapter 5.

4.5.1 Electricity Price and Wind Speed Data

The price forecast methods are tested to generate one day ahead prices, using the UK wholesale market. The UK price data used is from Nord Pool, for each hour over the period from the 29-12-2017 to 27-02-2020. During this time, price has sunk and spiked several times, and the monthly averages have decreased as seen in Figure 4.4.

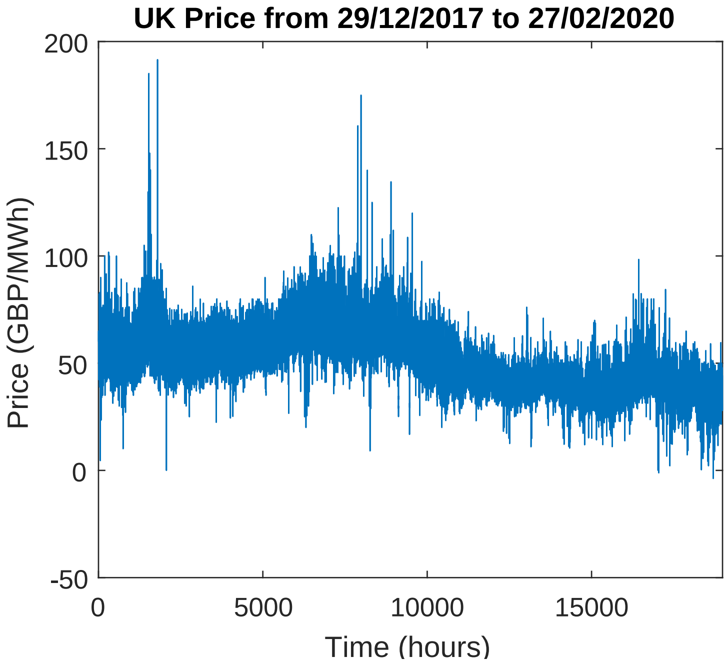


Figure 4.4: UK price data for the wholesale electricity market used for the price forecast methods.

While this set is used for the wholesale electricity market, rather than the retail market, this case study is designed to evaluated realistic price variations that could affect a day-ahead price scheme for microgrids, which could operate at either level of the electricity market, depending on the size.

The wind speed data used in this chapter is the historical wind speed data of four years, 2016, 2017, 2018 and 2019, taken every 30 minutes, at the Auchencorth Moss Atmospheric Observatory in Scotland, UK. The original dataset was obtained from [136]. The samples in this dataset are measured in m/s, with a total of 58655 samples.

For this study, the wind speed samples with values equal or higher than $30m/s$ were removed, with the remaining dataset having 58448 samples, of which 16499 represent the data for 2016, 17014 for 2017, 17514 for 2018 and 7628 for 2019, which is 99.6% of the original dataset. This set is composed of 97.45% of the possible samples, given that there are samples missing in this set. No form of extrapolation was used to fill the gaps given that the natural wind speed variations is enough to treat

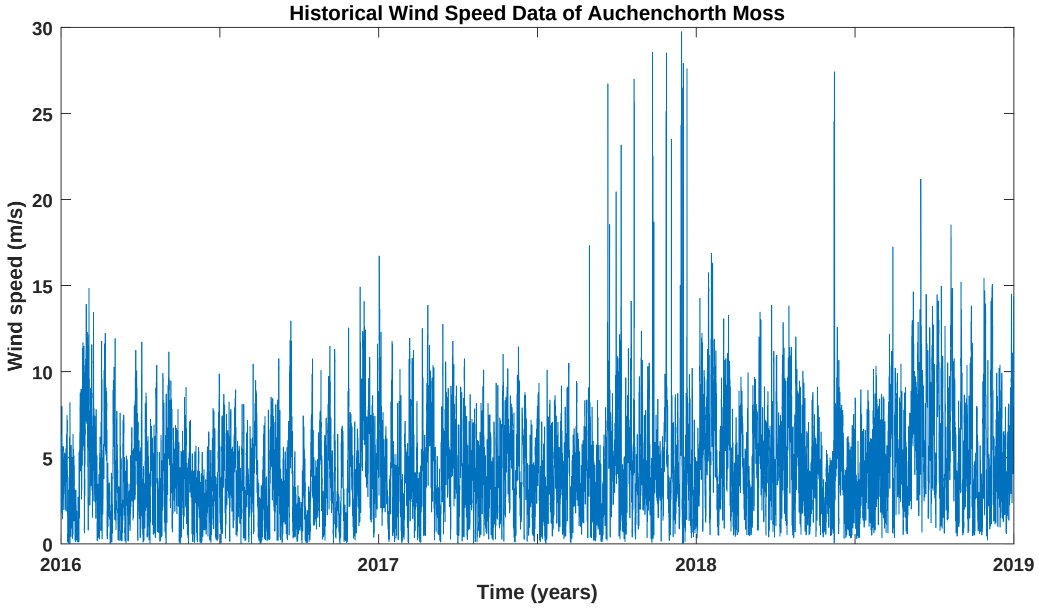


Figure 4.5: Auchencorth Moss wind speed, South East Scotland (55°47’36” N, 3°14’41” W). This dataset represents 99.6% of the total data available for this period.

the samples as consecutive. For reference, 87 samples of this set are above $15m/s$ (0.1485%) and 58568 (99.8515%) are between $0m/s$ and $15m/s$. The variation of the wind speed over this period is illustrated in Figure 4.5.

This set was selected, among other reasons, because it is freely available, has a good resolution compared to others which average for days or months, is sufficiently long to allow the statistically significantly difference test which relies in the variance of the error in a time series and becomes more reliable with more data and finally because the location provides data applicable for microgrid applications of onshore wind speed forecast, as opposed to large offshore wind farms, which have different environmental conditions.

4.5.2 Forecast Model Training

Each of the models was fitted, optimised or trained according to the data and the models described in previous sections. The hardware used for this study includes the Durham University super computer known as Hamilton. One node of Hamilton was used, comprising 24 CPU cores with 2x Intel Xeon E5-2650v4 model processors at 2.6GHz and 64GB RAM. Other hardware used for training includes a desktop with an i7-6700 CPU at 3.40GHz with 16 GB of RAM and a desktop with an i5-7500 CPU at 3.40 GHz with 8 GB of RAM. The software for training and fitting is MATLAB R2019a (9.6.0.1099321) 64-bit and the DM-test was done in the Scientific Python Development Environment, Spyder 4.1.5, with Python 3.8.5 64-bit.

The NARNET model architectures were trained with different number of delay sizes ranging from 20 (10 hours) up to 672 (2 weeks), number of neurons from 5 to 265 and the three activation functions discussed previously, with the most relevant ones presented in the results section. The selection of these parameters was based on [7], and the time required to realise the training and performance of the model.

The ARMA model, used as point of comparison, was optimised for the order $\mathcal{P} = 5$ and $\mathcal{Q} = 30$, compared with [97], that used orders $\mathcal{P} = 2$ and $\mathcal{Q} = 1$ and following the autocorrelation of the data set, this is considered enough to maximise the performance of the ARMA model.

The AR model was optimised for different sizes of delay, varying from 1 to 240. The ARMA and AR models are fitted using the data from 2016 to 2018 and the data from 2019 is left for error testing. For the case of the NARNET, the subsets are selected at random, as explained earlier. A total of 95 realisation where done for the single layer architecture, 45 realisation for the Parallel architecture and 18 for the Series architecture.

The difference in the amount of realisations done is due to the training time and performance in terms of RMSE. However, each combination of architecture and activation function was realised at least 6 times to account for randomness in the NARNET training. A total of 400 forecast models were computed, and every pair combination was evaluated with the DM-test, the number of realisations are summarised in Table 4.1.

Model	Architecture	Number of realisation
NARNET	single	95
	series	18
	parallel	45
AR	-	240
ARMA	-	1
persistence	-	1

Table 4.1: Summary of wind speed forecasts model realisation.

4.5.3 Case studies

For the first case study of price forecast, the MCMC price forecast method is used to generate a one week price estimation, from the 27/09/2019 to 03/10/2019, using the price data from 16/05/19 to 26/09/19 in GBP/MWh, obtained from Nord Pool. This is done to obtain the mean and standard deviation of the posterior at each hour for a normal distribution and for each hour of the same weekday, using an heuristic model considering as starting values the hour prices of the previous day as follows:

$$\pi_i = \frac{p_0 + \mu_0 + \mu_1}{3} \tag{4.26}$$

where π_i is the MCMC price estimation, p_0 is the price at the same hour of the previous day, μ_0 is the mean MCMC estimation over the entire data of the same hour and μ_1 the mean estimation with the same week day and same hour.

The standard deviation σ_2 used for the confidence interval is calculated as:

$$\sigma_2 = \frac{\sigma_0 + \sigma_1}{2} \tag{4.27}$$

Where σ_0 is the MCMC estimation using the entire data of the same hour and σ_1 the estimation using the data with the same hour and same week day.

For the second case study of the price forecast method, the electricity price methods are tested for four data scenarios: a) the entire price data set, b) the entire price data set without the 20 most significant outlier days, c) weekdays only and d) weekdays only without the outliers. Each of these scenarios was run 10 times for each configuration of the tested price forecast methods. For the case of the WA, the forecast is optimised with each solver. For the NARNET model, the forecast is tested with varying number of neurons in the hidden layers between 5 or 10 and the size of delay in each of the architectures shown in Figure 4.2 is tested for one week of delay and two weeks. In this case the price forecast models are those presented in the corresponding WA model and NARNET model subsections in this chapter.

The electricity price forecasts obtained are evaluated calculating the MAPE and RMSE. Two evaluations functions are required to better dimension the accuracy of the methods, as the data contains values close to zero [75, 137, 138]:

$$MAPE = \frac{100\%}{N_e} \sum\nolimits_i \frac{|p_i - \pi_i|}{p_i}, \forall i \in N_e \tag{4.28}$$

$$RMSE = \sqrt{\frac{\sum_i (p_i - \pi_i)^2}{|N_e|}}, \forall i \in N_e \tag{4.29}$$

where p_i is the hour real price and π_i is the hour price from (4.4), and N_e the set containing the period with available price estimations.

For the wind speed case study, the wind speed forecast models presented in this chapter are used to predict the half-hour wind speed data in 2019 for Auchencorth Moss, UK. The same models are used to generate the forecast with all the data from 2016, 2017 and 2018 to verify the statistical significant difference among the forecast models with the use of the DM-test described previously. The results are shown in RMSE and in terms of the p-values from the DM-test, using the wind speed historical data as mentioned before in the training subsection.

4.6 Results and discussion

4.6.1 Electricity Price Forecast Results

For the first case study, the predicted and the real prices are shown in Figure 4.6. The blue region this figure shows the 95% Confidence Interval (CI), corresponding to $\pi_1 \pm 1.96\sigma_2$, shows the estimation limits in terms of variations from the mean.

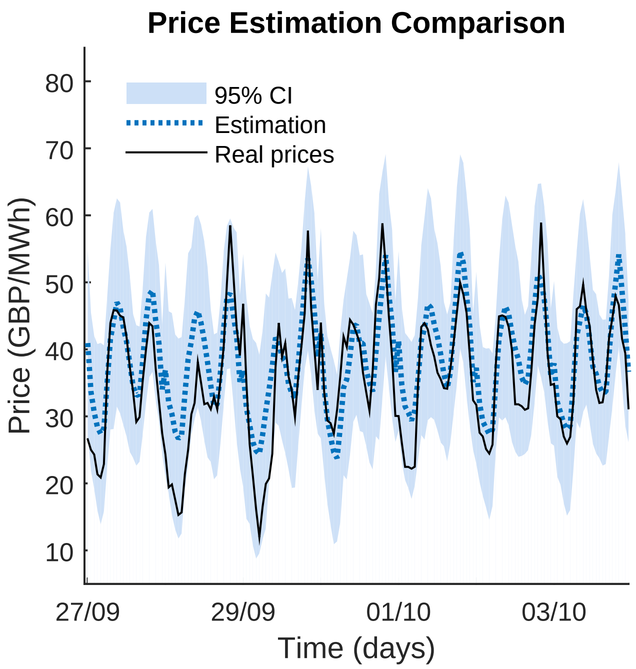


Figure 4.6: Comparison of the price estimation for one week.

Table 4.2: Summary of accuracy for the simple AR model for price forecast.

Price	**Model**	**MAPE**	**RMSE**
Average	μ_0	0.646%	0.299
	π_i	1.725 %	0.824
Week	μ_0	20.335%	7.214
	π_i	15.953 %	5.765

Table 4.2 summarises the accuracy of the method using the error functions, a comparison is shown between the price signal μ_0 and π_i for the average prices and for the prices of the week 27/09/2019 to 03/10/2019.

For the second case study, the price forecast accuracy results for the WA, based on QN, and NARNET methods are presented as follows:

After running each case 10 times for each set of parameters, the total MAPE and RMSE is calculated. The results obtained are plotted in Figure 4.7. The results of each set of parameters show that the NARNET can outperform the WA when the outliers are not included. When the entire dataset is taken into account, the MAPE increases in two orders of magnitude.

To focus on the best solutions of this case study, a zoom is done to Figure 4.7,

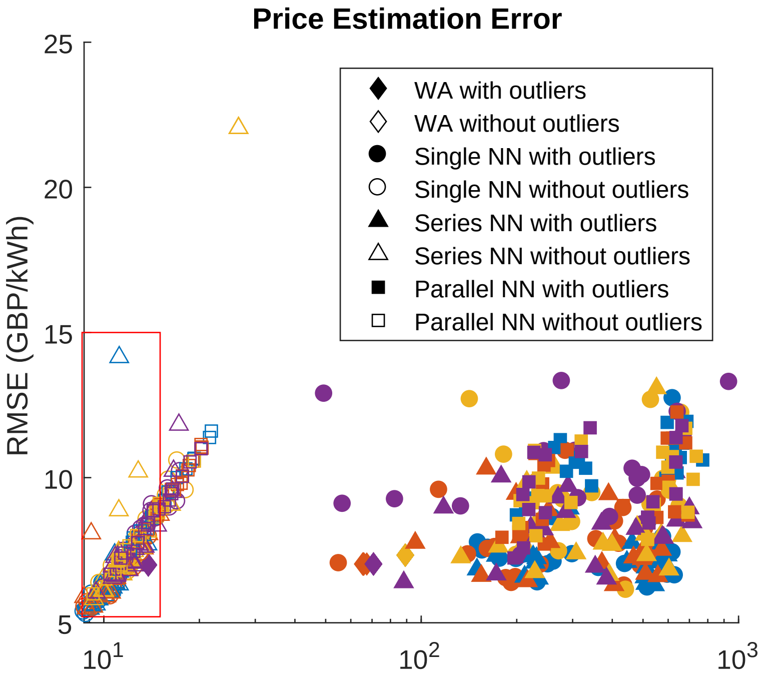


Figure 4.7: Individual Results of each method and set of parameters for the price forecast in the second test case. The red rectangle is shown as a zoom in Figure 4.8 to better visualise the best forecast models in terms of total error.

marked as a red rectangle in this figure, and the solvers re coloured as seen in Figure 4.8. It can be seen that in terms of individual runs, the best solution is provided by the single layer with 5 neurons with a week of delay for the NARNET, and the QN solver for the WA method. The best WA solutions have weights $W = [0.49\ 0.32\ 0.18\ 0\ 0]$ and $W = [0.48\ 0.34\ 0.96\ 0.03\ -0.82]$ for the constraint and unconstrained case.

The errors over the price dataset for each of these two best cases without the outliers and weekends for each method are shown in the Figure 4.9 and Figure 4.10. The individual errors are shown as blue circles deviating from the actual price in orange.

While they show a similar level of correlation between the target and output, the WA deviates more at the higher prices, increasing the final RMSE and MAPE scores. In both cases the error increases for the lower values, however, this may be caused by the prices in the data set that tend to zero.

The summary of error distributions is shown in Figure 4.11, which compares the

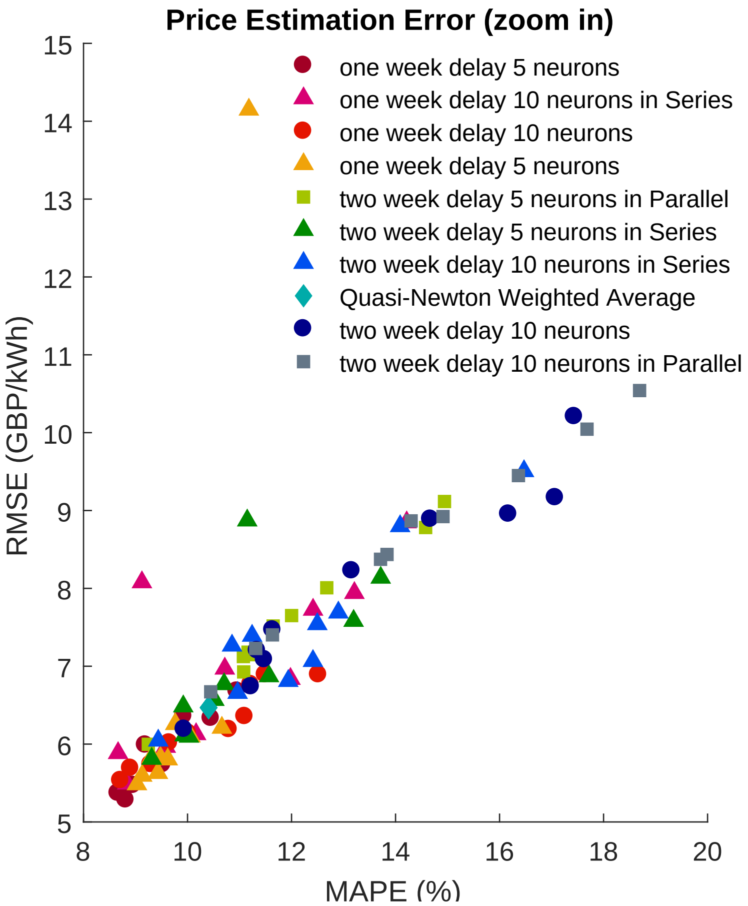


Figure 4.8: Zoom of the Figure 4.7 for the errors in for the price forecast models of the second test case.

QN solver for the WA and the NARNET using all the data and without the outliers. It can be seen that for both methods and data sets the errors have a normal distribution, and the difference in score depends on the number and deviation of large errors shown as blue circles.

Table 4.3 shows the average MAPE and RMSE of the combined results of the different parameters of each method and the best individual score. It can be seen that in general, the WA method is better as it has less variations in each run compared to the NARNET, but the latter has the best individual score, with the neural network with a single layer of 5 neurons and one week of delay achieving a MAPE of 8.67 % and a RMSE of 5.88 GBP/KWh.

It can be concluded that for the case of the UK price data from 29/12/2017 to

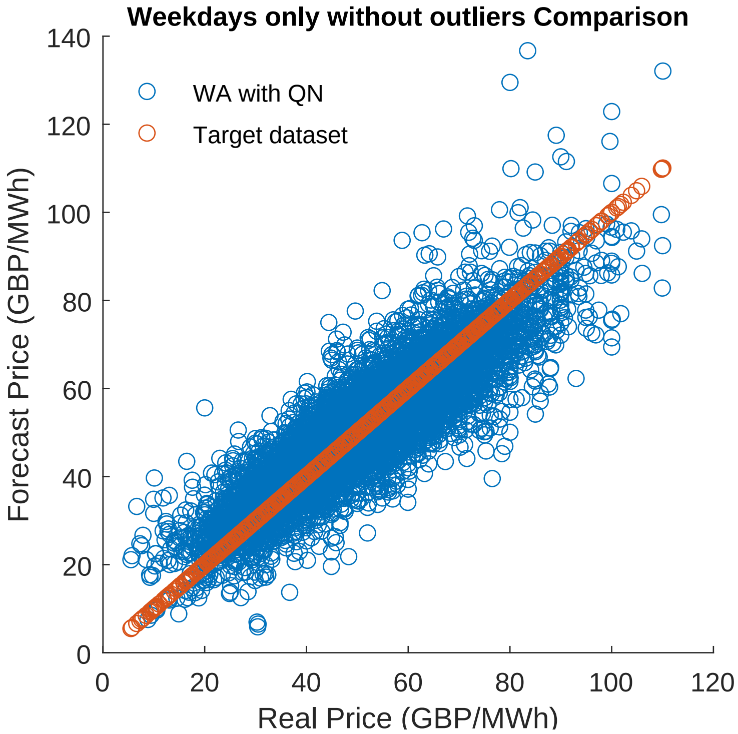


Figure 4.9: Error distribution of WA regression Model solved with QN.

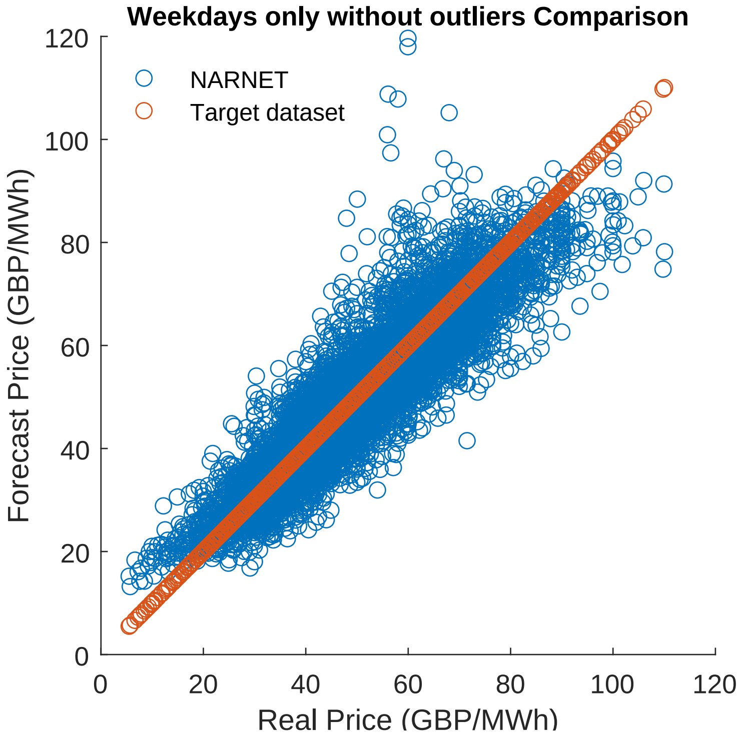


Figure 4.10: Error distribution of the NARNET.

Table 4.3: Error results of each method.

Method	Average Score		Best Score	
	MAPE	RMSE	MAPE	RMSE
WA Regression model	28.20904	7.11	10.41	6.47
Single Layer NARNET	189.44	8.00	8.67	5.37
Series NARNET	187.93.96	7.75	8.67	5.88
Parallel NARNET	223.98	9.41	9.25	6.00

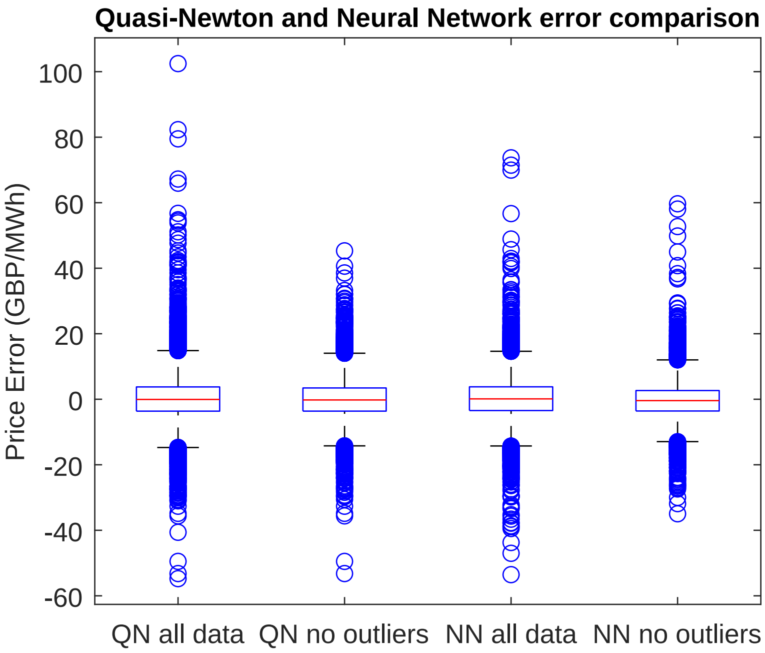


Figure 4.11: Error boxplot of best settings for each method.

27/02/2020, the NARNETs with one hidden layer and with 2 hidden layers in series with one week of delay achieved the best MAPE score at 8.67 %, with 5.37 GBP/kWh and 5.88 GBP/KWh RMSE respectively.

The WA solution shows that the previous day has the highest weight if each weight must be positive, the season is the most significant term without this constraint.

4.6.2 Wind Speed Forecast Results

For the wind speed forecast case, the following results are presented:

The RMSE of all the realisations varies between $0.74485m/s$ and $1.1267m/s$. Most of the models performed better than the persistence model at $0.9437m/s$ RMSE

Table 4.4: DM-test p-values between the best realisation of each forecast model. Si refers to Single, Se to Serial, Pa to parallel NARNET, AR to Auto regression, ARMA to Auto regression with moving average, and base to persistence model. The models are sorted by RMSE, shown in the second row.

Model	Pa	Si	Se	AR	ARMA	Base
RMSE	0.7448	0.7730	0.8289	0.8721	0.888	0.9437
Pa	1	0.277	8.9E-10	1.7E-17	2.57E-17	0
Si	0.277	1	1.06E-7	8.16E-19	8.73E-20	7.32E-43
Se	8.9E-10	1.06E-7	1	2.53E-7	4.56E-8	3.41E-134
AR	1.7E-17	8.16E-19	2.53E-7	1	1.69E-5	2.17E-118
ARMA	2.57E-17	8.73E-20	4.56E-8	1.69E-5	1	7.91E-113
Base	0	7.32E-43	3.41E-134	2.17E-118	7.91E-113	1

as expected, except for 7 realisations , all of which correspond to the combination of single layer with the Tansig activation function which is the best architecture found in [7], although, with a different dataset. However, other runs of the same NARNET with variations in the size of delay and number of neurons performed much better.

Figure 4.12 illustrate the error histograms for the best realisation of the NARNET, AR and ARMA models for all the data set to illustrate their error distributions.

In Figure 4.13, the forecast models are used only on the 2019 data to illustrate the suitability of the models over the test data.

The results in terms of RMSE are shown as an histogram to illustrate the variations between the same forecast model are shown in Figures 4.14 and 4.15. The results of the DM-test are shown in Figures 4.17 and 4.18. The RMSE results of the best individual realisation and corresponding p-values are shown in Tables 4.4 and 4.5.

The best realisations of each NARNET architecture, AR model, ARMA model and the persistence model, are shown in Table 4.4 with their respective p-values from the DM-test. While the null hypothesis is not rejected for the comparison between the parallel and series NARNET model of this table, the size of the ω matrices of the parallel NARNET are in total 8 times smaller than the series model W_{ann} matrix as described in Figure 4.18.

In Figure 4.14 the realisations are grouped by NARNET architecture and compared

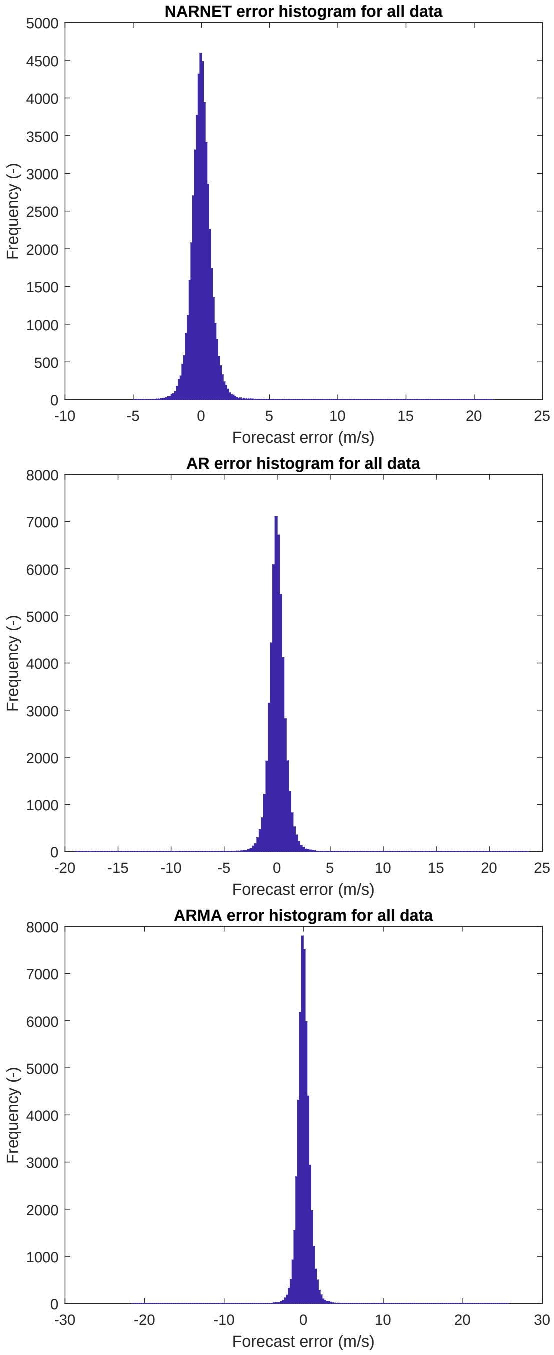


Figure 4.12: Error histogram to validate the forecast methods.

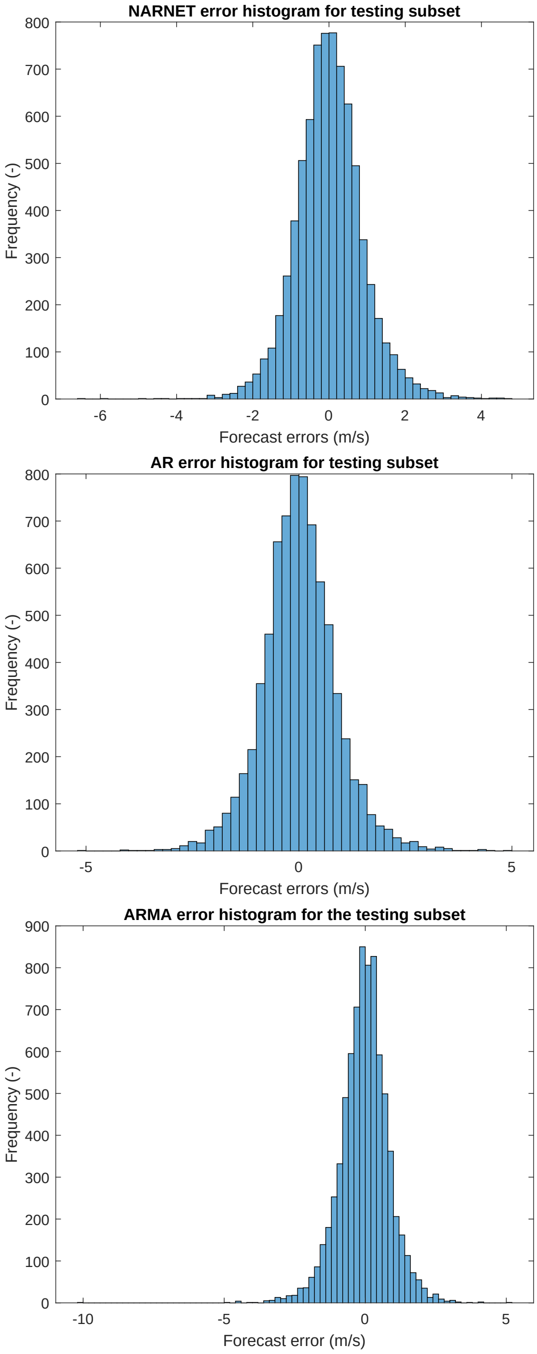


Figure 4.13: Error histogram to validate the forecast methods.

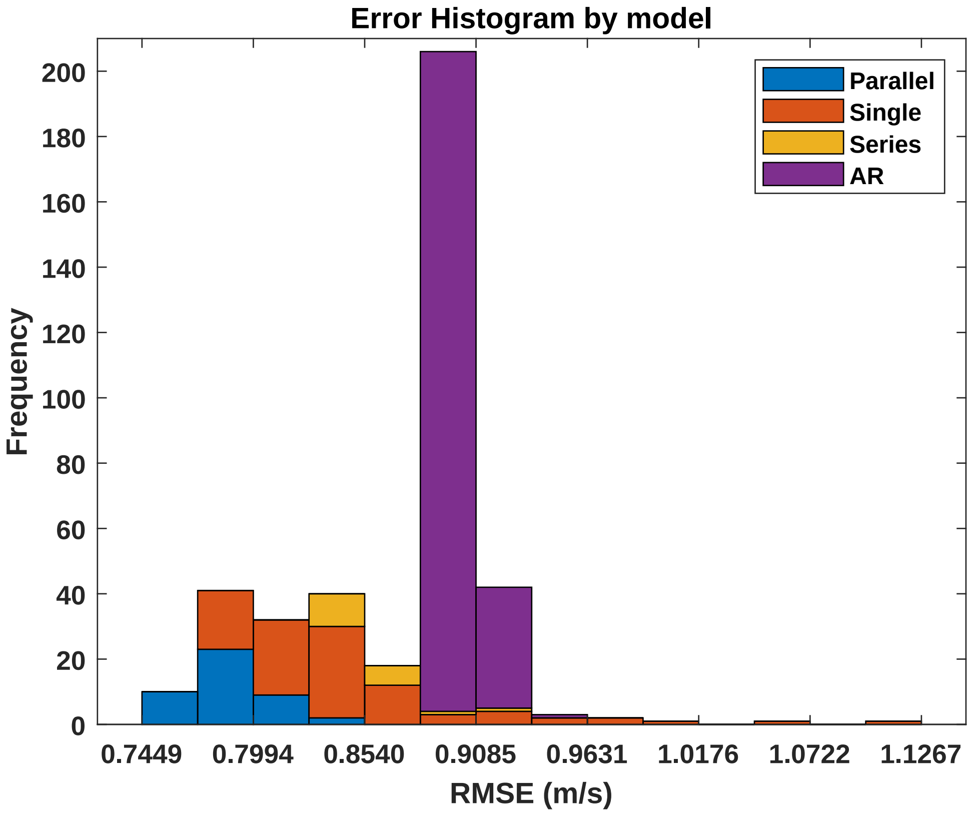


Figure 4.14: histogram of the total RMSE by forecast model. The persistence model, and ARMA model are omitted as they only have one iteration each.

with the AR model. It is notable that all the AR model realisations are grouped within 2 bins of the histogram, even when the delays range up to 240 samples, equivalent to 5 days of data.

For the case of the NARNET models, it is notable that the distribution that account for the lowest RMSE is the parallel architecture and that the series architecture has the largest variation.

In Figure 4.15 the same results of Figure 4.14 for the NARNET realisations are shown, grouped in an histogram by activation function. It can be seen that the ReLU function seems to have the best performance in terms of distribution. However, looking at the top 12 best forecast model realisation in Figure 4.18, it can be noted that the best activation function in terms of RMSE is the Logsig, while the Tansig

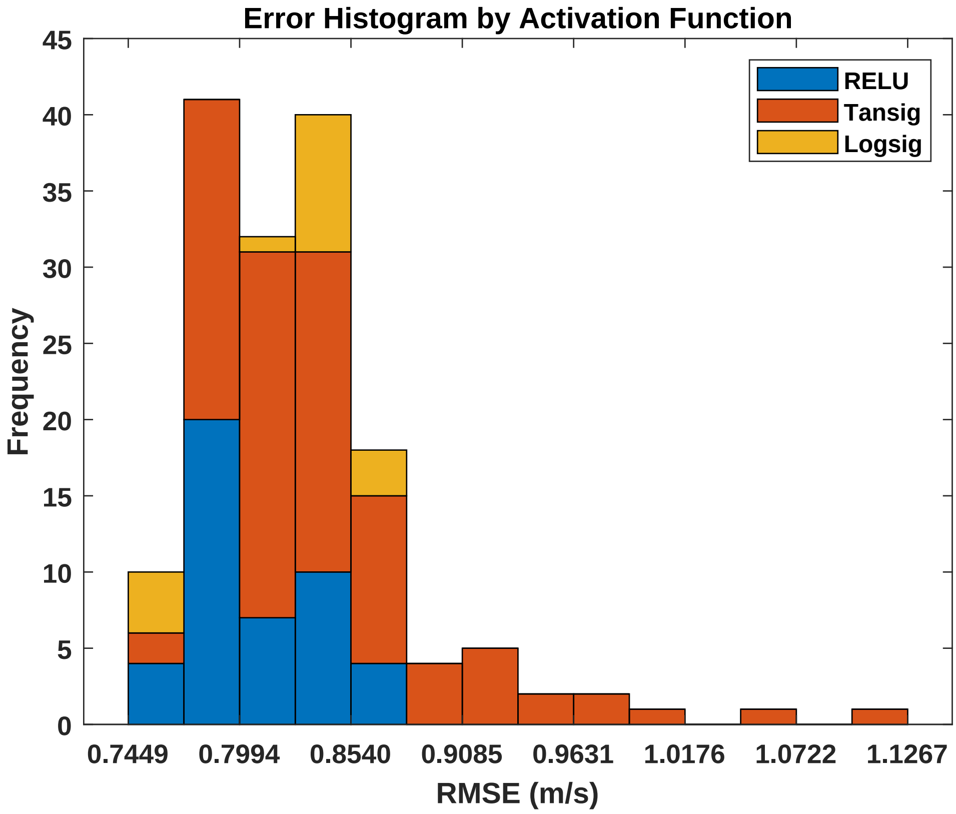


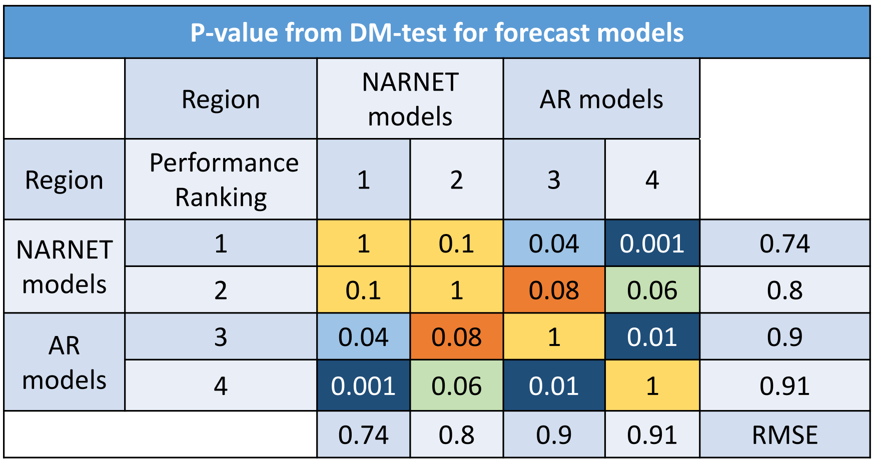
Figure 4.15: The persistence model and AR model are omitted as they do not have activation functions. Tansig refers to the Hyperbolic Tangent Sigmoid function. Logsig refers to the Logistic Sigmoid function.

Table 4.5: DM-test p-values between the best realisation of each activation function of the NARNET by colour.

RMSE	DM	Logsig	ReLU	Tansig
0.7448	Logsig	1	0.0875	0.0012
0.7519	ReLU	0.0875	1	0.006
0.7694	Tansig	0.0012	0.006	1

has the highest variation in performance, being present in all bins.

Table 4.5 is built in the same way as Table 4.4 for the best realisations of each NARNET activation functions. In this case it can be seen that the p-values show that the models are statistically significantly different for every test except between the ReLU and the Logsig comparison, with a weak statistical difference. In all cases the null hypothesis is rejected.



P-value from DM-test for forecast models

	Region	NARNET models		AR models		
Region	Performance Ranking	1	2	3	4	
NARNET models	1	1	0.1	0.04	0.001	0.74
	2	0.1	1	0.08	0.06	0.8
AR models	3	0.04	0.08	1	0.01	0.9
	4	0.001	0.06	0.01	1	0.91
		0.74	0.8	0.9	0.91	RMSE

Figure 4.16: Illustrative table for the presentation of DM-test results.

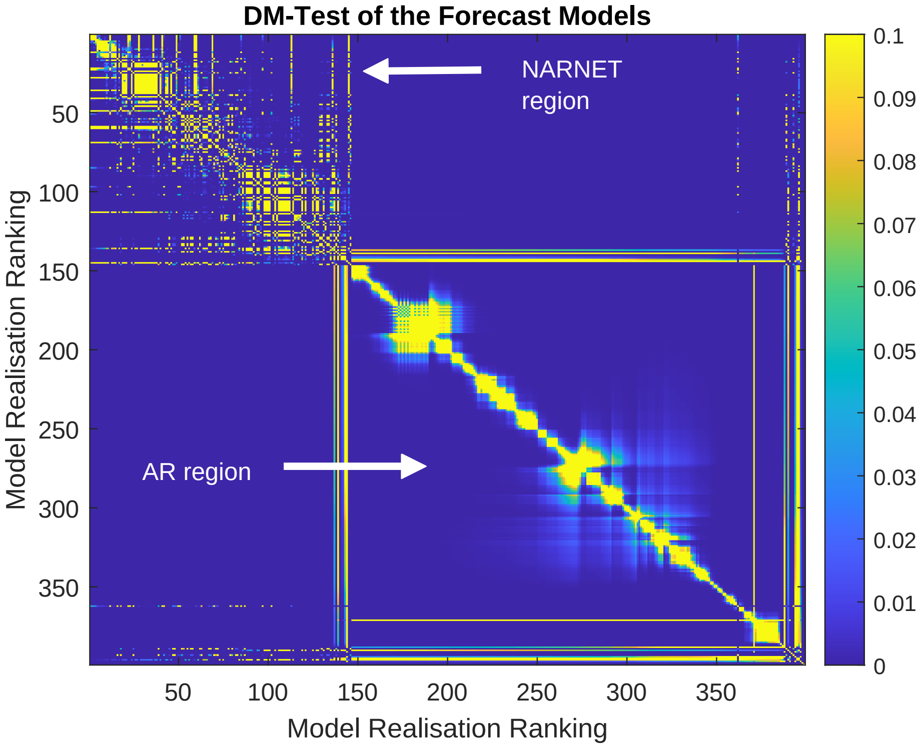


Figure 4.17: Entire DM-test p-value results for every combination of the forecast models by colour. p indicate the probability that the null hypothesis is true, $p < 0.05$ indicate statistical significant difference, $0.05 \leq p < 0.1$ indicate weak statistical difference and $p \geq 0.1$ indicate no statistical difference, shown in yellow. The realisations are ranked from lowest RMSE to highest. The NARNET region is at a lower RMSE and in general shows an statistical significance difference with the AR models indicated by the blue colour.

To present all the p-values for all the possible DM-tests, Figures 4.17 and 4.18 are presented. However, Figure 4.16 is presented first to illustrate the information that is included in those figures. The following observations apply for the DM-test figures: **a)** The performance ranking goes from lowest to highest RMSE, following natural numbers. **b)**Each square in the figure encodes the p-value from the DM-test between the row and column pair of models. **c)** The diagonal is always 1 because a forecast model is compared with itself (100% probability of no difference). **d)** The colour of each square goes from 0 to 0.1, above 0.1 the colour remains yellow, which means no statistical difference. **e)** Any square that is not yellow indicates a statistical difference between the pair of models. **f)** Any square in blue of any shade, has p-value below 0.05, and therefore a statistical significant difference between the models. **g)** In general, the NARNET models outperform the AR models, separating them in two regions in the DM-test figures.

In Figure 4.17 the p-value results of every combination of two forecast realisation for the DM-test is shown. The model realisations are ranked from lowest RMSE to highest in both axes while the colour indicates the p-values, where yellow represents a p-value equal or higher than 0.1, or in other words, no statistically significant difference, the blue colour indicates that the difference in performance in RMSE is statistically significant, therefore, the difference in performance is due to difference in accuracy of the models and not randomness in the data. It can be verified that the NARNET outperforms the AR in terms of RMSE as indicated by the arrows.

Secondly this figure illustrates that the NARNET models in general are either statistically significantly different or not at all from the high contrast in colour among them compared with the smother transition in p-values in the AR region. Figure 4.17 also illustrates that in most cases the NARNET models are statistically significantly different from the AR models except for those with close performance in RMSE. A yellow diagonal is shown across the figure as it shows the comparison of a realisation with itself, and therefore there is no difference calculated.

Figure 4.18 shows in more detail the top 16 forecast model realisation p-values from

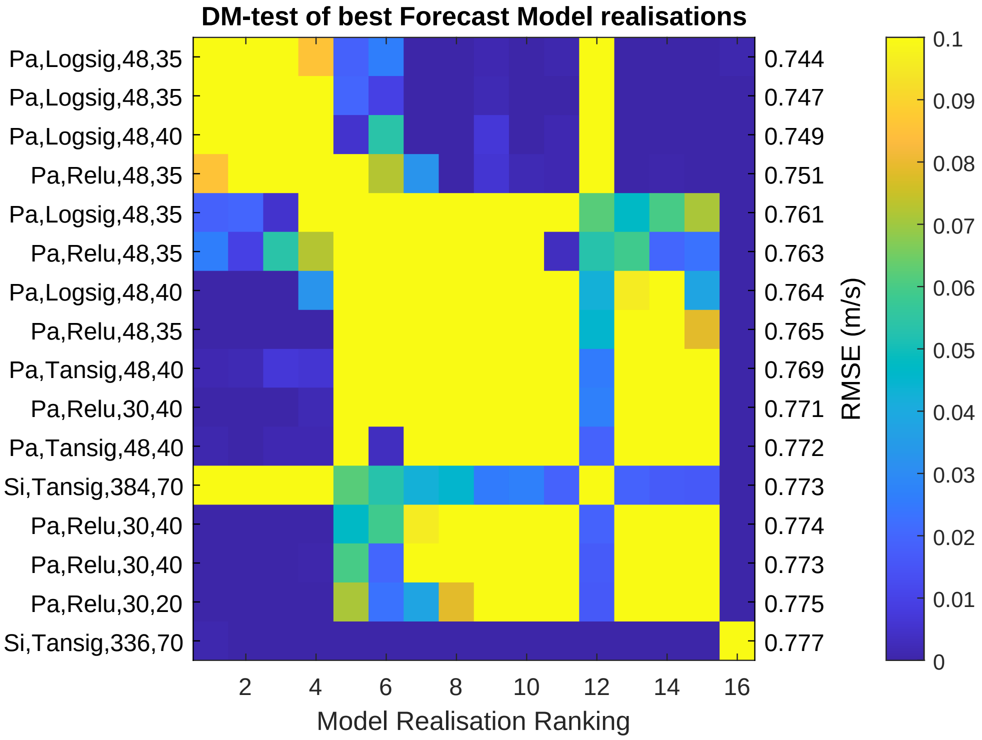


Figure 4.18: Entire DM-test results by colour for the best 16 forecast models realisations ranked from lowest RMSE to highest. p indicate the probability that the null hypothesis is true, $p < 0.05$ indicate statistical significant difference, $0.05 \leq p < 0.1$ indicate weak statistical difference and $p \geq 0.1$ indicate no statistical difference, shown in yellow. In the left axis, the configuration indicates architecture, activation function, size of D and number of neurons per hidden layer.

the DM-test by colour, all of which are NARNET models, describing the specific configuration of the model in the left axis and the RMSE performance in the right axis. The same ranking from top to bottom of the left axis is in the horizontal axis from left to right. Given that the top 3 models are parallel NARNET models with the Logsig activation function with identical RMSE performance for the first two decimal places it is expected that those have no statistically significant difference among them.

The configuration for wind speed forecast with lowest RMSE at 0.74485 has a delay size of 48, or one day of memory, with each hidden layer having 35 neurons, which implies 3360 weights in total for both of the hidden layers. It is also notable that the Single layer with Tansig activation function model, that was found to be best model

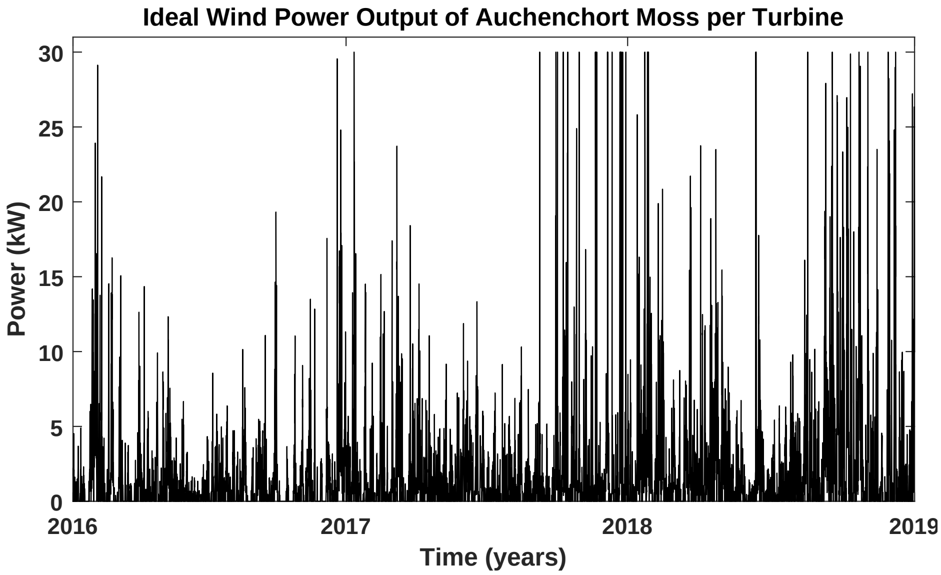


Figure 4.19: Ideal power generation per turbine from the historical data.

in [7], that in this case has a delay of 384, equivalent to 8 days of memory, and 70 neurons in the hidden layer which implies an internal matrix of 26880 weights has 0.773 RMSE.

This study illustrates the importance of not only determine the best forecast model in terms of minimal errors but also the importance of verifying that the difference in performance between two forecast models is actually due to an actual superior model. To finalise, the wind power calculation expected from the case study is presented next, to contextualise the wind speed forecast model for microgrid distributed control.

4.6.2.1 Wind Power Calculation

To help putting in the context of microgrids, this subsection will briefly cover how the forecast results shown previously compare in terms of the expected wind power generation that could be extracted from the site given ideal conditions. First, in Figure 4.19 the total wind power that could be generated using the wind speed data set available is shown.

For the case of wind turbines, the electrical power output P_{wt} is defined in general as:

$$P_{wt}(V_{ws}) = \begin{cases} 0, & V_{ws} < V_{min} \vee V_{ws} > V_{cut} \\ C_p(V_{ws})^3, & V_{rated} < V_{ws} < V_{cut} \\ P_{max}, & else \end{cases} \tag{4.30}$$

where V_{ws} is the wind speed,V_{min} is the minimum wind speed for generation or cut-in speed, V_{rated} is the rated wind speed of the wind turbine, V_{cut} is the maximum wind speed for generation or cut-out speed, P_{max} is maximum wind turbine power output, and C_p is the coefficient of performance of the wind turbine.

The expected power generation $\mathbb{E}(P_{out})$ over a period $\tau = [1 \ldots \tau_{max}]$ with duration τ_{max}, given that the wind speed is random, is defined in this case by [135]:

$$\mathbb{E}(P_{out}) = \sum\nolimits_i P_{wt}(V_{ws} = v_{w_i}) p_{ws,i}, \qquad \forall i \in \tau \tag{4.31}$$

where $p_{ws,i}$ is the probability that V_{ws} equals the i-th realisation v_{w_i}. The expected power generation is then an estimation of electrical power output taking into account the randomness in the wind speed over a finite time period.

For the case of the expected power output $\mathbb{E}$, following equations (4.30) and (4.31), and considering $V_{min} = 3.5\, m/s$, $V_{rated} = 15\, m/s$, $V_{cut} = 35\, m/s$, $P_{max} = 30\, kW$ and $C_p = 0.0089\, kW(m/s)^{-3}$, $\mathbb{E}_1 = 21.9053\, kW$ per turbine using the historical data of the 23/10/2018 as input, $\mathbb{E}_2 = 20.8627\, kW$ per turbine using the model with the lowest RMSE, and $\mathbb{E}_3 = 21.0047\, kW$ for the case of the model ranked 12 in Figure 4.18, which has not statistical significant difference with the rank 1 model, but at a higher computational cost, given that it requires more weights to achieve this. The calculation of $\mathbb{E}$ can be simplified to the mean power output calculated with the wind speed from each case.

4.7 Summary

This chapter presented suitable forecast methods to be applied in microgrid real-time distributed control. In particular the forecast models for electricity grid price in the UK and the wind speed in Auchencorth, UK.

A price forecast method was tested for the UK market using MAS. While there is opportunity to improve the accuracy of the estimation, the system works in a more distributed environment, which makes it less dependant on a central controller. In the same way, in combination with the restoration service as presented in the previous chapter, prevents the loss of the entire agent platform from the disconnection of the main container. Finally, with a good convergence algorithm, its possible to distribute the control with independent price estimations, which leads to minimal cost, which will be shown in the next chapter.

For the case of price forecasting, two main methodologies were tested for price forecast of the UK's electricity market. It was found that the NARNET achieves the highest accuracy if the outliers are removed from the data, while the WA method is better otherwise.

For the case of the Auchencorth Moss wind speed forecast it was found that the optimal forecast model for distributed control is the parallel NARNET with the Logsig function, one day of memory and 35 neurons in each hidden layer. The results show that increasing the complexity of the NARNET does not necessarily imply a better performance with statistically significant difference.

The wind speed forecast study also presents the proposition of a forecast model evaluation method that allows the selection of the best short-term wind speed forecast model, suitable for distributed microgrid control, in terms of accuracy and complexity by comparing the accuracy and number of weights required by the best model found and the second best model, taking into account the randomness in the data.

This study illustrates the importance of not only determining the best forecast model in terms of minimal errors but also the importance of verifying that the difference in performance between two forecast models is actually due to an actual superior model either by accuracy or by efficient use of computational resources.

The NARNET is considered a suitable forecast model for distributed control in microgrids given at it relies in historical data that can be stored locally, which means that the rest of the control system is not subjected to dependence of other external signals, which could represent single points of failure in terms of control reliability. It is also shown that the parallel architecture and Logsig activation function can outperform the other models used for comparison in this study.

This model can also be used to avoid the additional computational cost of running a more complex model without an statistically significant difference in performance. As shown in the wind speed case study, the parallel hidden layers with the Logsig function NARNET model can statistically perform as good as the single hidden layer with the Tansig function NARNET model with 87.50% reduction in required number of weights for the first configuration compared to the second.

This evaluation method can be applied to optimise computational resources in the microgrid control environment in any of the forecast requirements emerging from similar applications, such as price, irradiance or load forecasting, which can minimise the computational cost of the control system and can improve its response speed, while at the same time helps to determine the best model in terms of forecast errors, which in turn contributes to the fulfilment of microgrid control objectives, such as cost minimisation.

Chapter 5

Integrated Full Control for Power Management of Multi-Microgrids

5.1 introduction

Promoting the development of the future energy systems is necessary to contribute to the electricity decarbonisation. Microgrids offer an excellent opportunity to organise and integrate DERs to the electrical grid and contribute to this end. However, given the autonomy of microgrids to pursue their own objectives, increasing the number of microgrids in the distribution network requires an additional control layer for the coordination of multiple microgrids and their interaction with the main grid. Microgrid cluster are circuits that combine multiple microgrids, used to increase the penetration of DERs in the distribution network [22].

In the context of hierarchical control schemes for microgrids, the tertiary control layer is added to regulate the microgrid clusters and coordinate the objectives of each microgrid in the cluster. In the hierarchical control scheme of this thesis, this layer is formed by a MAS platform, with the objective of maximising the use of distributed generation, while maintaining resilient operation properties at the communication level. However, this layer is designed such that some level of independence is maintained by the secondary control layer to allow the individual microgrid to achieve their power management objectives.

This chapter presents the characteristics and interactions of the agents that realise the proposed secondary control layer for the power management of microgrids. In other words, how the MAS platform interact with the physical layer, through the primary control layer, as a distributed control scheme. The MAS platform is used to generate the appropriate power references to achieve individual microgrid's power management objectives, namely, supply cost minimisation and renewable generation maximisation.

Moreover, the formulation of the agents for the tertiary control layer, its general objective and testing are also presented in this chapter. At the tertiary control layer, the control system of the cluster microgrid is used to maximise the power generation contribution from internal generation, which in turn, minimises the use of main grid energy by the cluster, which leads to minimisation of transmission losses and power line congestion. The individual microgrids can be induced to switch from grid supply to internal supply by increasing the grid's electricity price. However, the cluster management system must balance these incentives between the wholesale electricity market and the retail electricity market.

By integrating all of the methods presented in the previous chapters in a hierarchical control scheme, this chapter is intended to demonstrate stability, resilient and optimal operation properties under distributed real-time control of the proposed control scheme for the automatic coordination of interactions within a cluster of microgrids [139]. All of these properties are intended to facilitate the integration of distributed generation in the electricity grid.

As each part of the distributed control scheme is following different objectives at each control layer, and even different objectives for the case of each microgrid, namely supply cost minimisation and renewable generation maximisation, the optimal performance of the entire control scheme will be evaluated using a combination of test cases.

To verify the effectiveness of the cost minimisation agents, regardless of the price forecast accuracy, a cost minimisation case with known prices in advance is presented first, comparing the proposed MAS approach with other optimisation solvers. Next, a test case with a single forecast agent to demonstrate the cost minimisation capabilities with unknown prices is presented, followed by a test case for cost minimisation with multiple forecast agents within a single microgrid, in line with distributed control.

Following this, the performance of the proposed control scheme under real-time operation with uncertainties is evaluated. This is done by implementing the control system presented in chapter 3 with the forecast methods explored in chapter 4 in real-time simulation test cases. For the case of grid price forecast, the evaluation is presented in terms of stability and cost minimisation. For the case of the wind speed forecast, the performance will be evaluated in terms of the difference between power schedule and actual power delivered by a microgrid with renewable generation.

Finally, the results from the subsequent application of each of the control layers are presented to showcase the effect of each layer in the simulated system. The simulation results validate the proposed control scheme as a suitable distributed real-time power management control for cost minimisation while maintaining stability for a cluster of microgrids.

5.2 MAS Design for Secondary Control Layer

The main advantage of the MAS approach comes from the possibility of separating the different control tasks into as many agents as necessary. Some of the agents have specific behaviours that contribute to the general objective of the MAS platform. This section first overviews the properties of standard agents for a MAS application and then the properties of the agents utilised for the microgrid's power management.

As presented in chapter 2, in every MAS application there are three standard agents for the operation of a distributed control: The Agent Management System (AMS), the Directory Facilitator (DF) agent and the Remote Monitoring Agent (RMA). The AMS is in charge of creating and terminating agents. It follows that the container with the AMS is known as the main-container. The DF serves as the yellow pages, receiving requests and offers for agent services, informing the relevant agents when there is a match for service and demand. The operation of the DF agent allows the formation of the communication networks among the rest of the agents. The RMA serves as a communication channel between containers in different computers as the human interface. This means that requests to the AMS can be done in separate locations, which aids with the distribution of the control system.

In addition to these standard agents, five types of agents were programmed in this thesis in Java using the Java Agent Development framework (JADE) environment for single microgrid distributed control optimisation namely, the generator agents that control dispatchable DERs, the ESS agent to regulate batteries, the grid price forecast agent, the Wind speed Forecast Agent (WSFA), and the Renewable Energy Source Regulator Agent (RESRA).

The first three types of agents are used for the supply cost minimisation objective. The global supply cost minimisation is realised by the agents designed for the dispatchable DER operation by solving the total cost minimisation locally of each DER of the microgrid. Then, by cooperating with each other, the agents generate the power schedule to be sent to each of the primary controllers as shown in Figure 3.1 in chapter 3. In a similar manner, the renewable generation optimisation is realised by the corresponding RES agents, however, in this case, the objective is minimising the error between the power schedule and actual generation.

The optimisation problem solved by the entire MAS platform can be formulated as:

$$\min_{P_j} \quad \sum\nolimits_i \sum\nolimits_j c_j((P_j(i)), \forall j \in \mathcal{J} \wedge \forall i \in \mathcal{I} \tag{5.1a}$$

$$\text{s.t.} \quad P_{j_{min}} \leq P_j \leq P_{j_{max}}, \tag{5.1b}$$

$$SOC_{min} \leq SOC \leq SOC_{max}, \tag{5.1c}$$

$$SOC(i+1) = SOC(i) - \eta P(i), \tag{5.1d}$$

$$P_L(i) - \sum\nolimits_j P_j(i) = 0, \forall j \in \mathcal{J}, \forall i \in \mathcal{I}, \tag{5.1e}$$

$$[P\ Q] \in [P(X)\ Q(X)],\ \forall X \in D_X \tag{5.1f}$$

where c_j is the cost function of each source j of DERs and ESS, $\mathcal{J}$ is the set containing the generation sources. P_j and Q_j are the active and reactive power sent by source j at each hour i and $\mathcal{I}$ is the set containing the time steps. Each power P_j is bounded by a minimum $P_{j_{min}}$ and maximum $P_{j_{max}}$. The State of Charge (SOC) is also bounded by a minimum SOC_{min} and a maximum SOC_{max}, the SOC depends on its previous value, the power $P(i)$ and η, which models the capacity and round trip efficiency of the ESS. The Load P_L and generation in the microgrid must be equal. The last constraint is used to guarantee Lyapunov stability and limits the references of the primary control, expressed in terms of the state variable X in the domain D_x, which delimits the stable operation of the bus. This optimisation problem is distributed in simpler problems between the corresponding agent, such that the collective operation of the agents solve the global supply cost minimisation problem.

For the case of multi-microgrid control, a secondary DF agent is required in the secondary control layer to maintain fault-tolerance communication between the secondary and tertiary control layers in the separate MAS platforms. The interactions between the secondary and tertiary control layer are illustrated in figure 5.1. The behaviour of each of these five agents is explained in the following subsections.

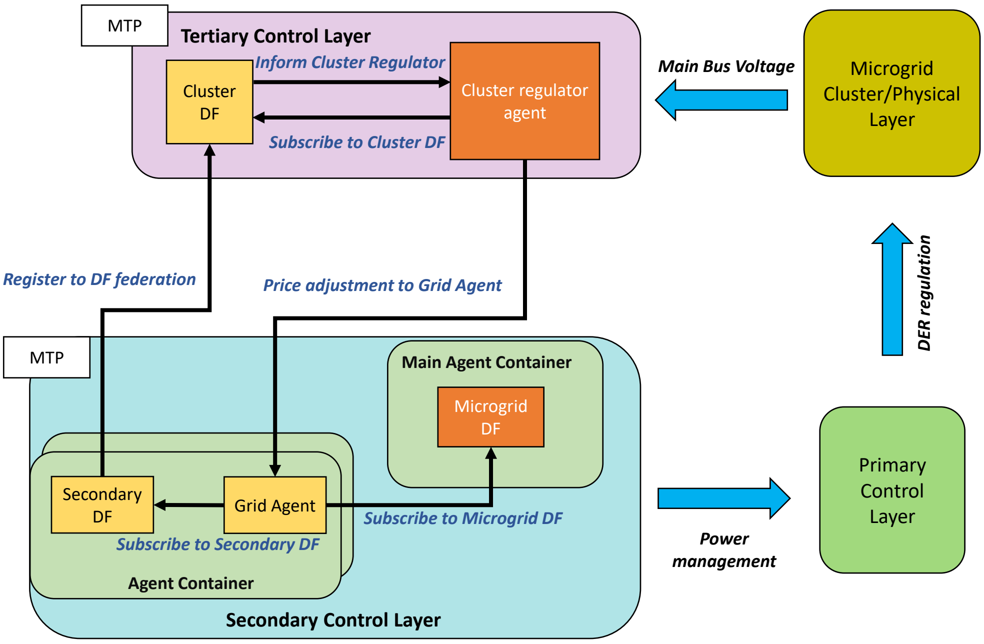


Figure 5.1: Communication between secondary and tertiary control layers. The black arrows and blue labels show the ACL messages sent between the agents. The MTP allows communication across platforms. The blue arrows indicate the general interactions with the rest of the layers.

5.2.1 Distributed Energy Resource Agent

The DER agent's behaviour consists in an internal loop of 10 milliseconds that manages the communication with the DF agent, AMS and a corresponding Grid Agent (GrA), that provides the price forecast π to be used as the actual price p by the DER agent. Once the price forecast messages are received, the agent uses the price forecast π and the DER cost function c_j to generate the corresponding power references for its corresponding primary control using TCP/IP communication.

The DER agent generates a power schedule for a single DER that minimises the total supply cost or maximises energy profits from trading with the grid by solving the following optimisation problem:

$$
\begin{aligned}
\min \quad & \sum\nolimits_i \pi(i)(P_{j_{max}} - P_j(i)) + \sum\nolimits_i c_j((P_j(i)), \qquad \forall i \in \mathcal{I} && (5.2a)\\
\text{s.t.} \quad & P_{j_{min}} \leq P_j \leq P_{j_{max}}, && (5.2b)\\
& \{P_j^{-1}(P_j(X)), Q_j^{-1}(Q_j(X))\} \in D_x, \qquad \forall j \in \mathcal{J} && (5.2c)
\end{aligned}
$$

The local problem solved by the DER agent is limited to minimising the cost of a specific source j with respect to the price forecast, from the set of sources $\mathcal{J}$. P_j is the power output of source j, $\pi(i)$ is the electricity price for the microgrid at time i, in the set $\mathcal{I}$, for the total optimisation time. This means that the DER agent minimises the accumulated total supply cost over time, instead of selecting the cheaper power option at each time step. Each source is constraint to a minimum power output $P_{j_{min}}$ and a maximum power output $P_{j_{max}}$. Additionally, the second constraint refers to the power output boundary of each source must be within the domain of the state variables X (related to the bus voltage and phase angle) to guarantee Lyapunov stability.

The following model is considered for the local cost function solved by the DER agent.

$$
c_j(P_j(i)) = \begin{cases} c_B P_j(i) + c_C + c_s & P_j(i) \neq 0 \\ 0 & P_j(i) = 0 \end{cases} \qquad (5.3)
$$

$$
\forall i \in \mathcal{I} \wedge \forall j \in \mathcal{J}
$$

where c_B and c_C are generator specific cost parameters and s_j is the start-up cost of a DER source j [4]. The start-up cost function c_s is defined as:

$$
c_s = \begin{cases} s_j & (P_j(i-1) = 0 \wedge P_j(i) > 0) \vee P_j(1) > 0 \\ 0 & \textit{otherwise} \end{cases}, \qquad (5.4)
$$

$$
\forall i \in \mathcal{I} \wedge \forall j \in \mathcal{J}
$$

The DER programmed behaviour considers if the start-up cost is worth to be paid based on the future prices and the current state of the DER, especially when the price forecast varies closely to $c_B P_{j_{max}} + c_C$.

Essentially, the jth DER starts supplying power at any time where $\pi \times (P_{j_{max}})$ is higher than $c_j \times (P_{j_{max}})$, and continues generation as long as the following is true:

$$\sum_i (\pi(i) P_{j_{max}} - c_j(P_{j_{max}})) > 0 \tag{5.5}$$

which represents that the accumulated supply cost using internal generation is smaller than the total cost using external generation. The DER agent evaluates the points in time when this condition is not true and compares the accumulated cost of using either external or internal generation to select an optimal power schedule for total supply cost minimisation.

5.2.2 Energy Storage System Agent

Depending on the size of the microgrid, the microgrid on its own can operate as an intermediary between the wholesale electricity market and the retail market. This means that the microgrid control system can minimise the cost of supply by considering the price changes in the main grid by performing hourly energy arbitrage. The ESS agent's behaviour is modelled with these considerations [51].

As mentioned previously, this agent regulates the charge and discharge of the battery by sending the power reference signals to its primary controller. This is done according to the following local cost minimisation problem:

$$\begin{aligned}
\min \quad & \textstyle\sum_i \pi(i)(P_{j_{max}}(i) - P_j(i)), \qquad \forall i \in \mathcal{I} && \text{(5.6a)}\\
\text{s.t.} \quad & P_{j_{min}} \leq P_j \leq P_{j_{max}}, && \text{(5.6b)}\\
& SOC_{min} \leq SOC \leq SOC_{max}, && \text{(5.6c)}\\
& SOC(i+1) = SOC(i) - \eta P(i), && \text{(5.6d)}\\
& \{P_j^{-1}(P_j(X)), Q_j^{-1}(Q_j(X))\} \in D_x, \qquad \forall j \in \mathcal{J} && \text{(5.6e)}
\end{aligned}$$

where the first constraints indicates that output power is bounded by a minimum and maximum, $P_{j_{min}}$ and $P_{j_{max}}$. The second constraint states that the SOC of the battery must be within the minimum charge SOC_{min} and the maximum charge SOC_{max}. The second constraint states that the SoC at time i depends on its immediate preceding timestep and the power $P(i)$ sent or received by the ESS at each time i, multiplied by a constant η, which models the round trip efficiency of the ESS and its capacity.

To solve the local cost minimisation problem of the ESS, considering the dynamic behaviour of the SOC, the ESS agent selects its actions based on a ranking of the forecast prices with respect to the current price and SOC.

The power references are generated by modelling the state of battery in advance to steer the SOC such that is has an optimal value in the future applying offline MPC, presented in chapter 2. MPC is used such that the agent aims to steer the SOC to have some specific value in the future by selecting the appropriate power references based on the internal model of the battery that is stored in the ESS agent. For example, steer the charge to be at maximum at the moment of peak price or to be at minimum at the moment of lowest grid price, to maximise profits from energy arbitrage. For this local objective, this agent requests a price forecast sufficiently long from its local GrA, such that the ESS agent always has enough time to completely charge or discharge the battery to maintain optimal operation.

To maintain a feasible solution for the dynamic behaviour of the SOC, an initial

SOC, which is measured from the physical layer at the start of the optimisation, is defined as SOC_0. Equation (5.7) is evaluated element-wise for each hour and offers an equivalent constraint to the limits of the SOC.

$$\left|SOC - \frac{SOC_{max} + SOCmin}{2}\right| - \frac{SOC_{max} - SOC_{min}}{2} \leq 0 \tag{5.7}$$

To realise its objective, the ESS agent looks at the trends of the current grid price and prices in the future, rating the prices to identify its trends and adjust the power reference accordingly. This means that the ESS drives the battery based on the changes in the main grid's price from one hour to the other, rather than the price itself.

5.2.3 Grid Agent

As part of the distributed control nature of the control system designed in this thesis, it is necessary to provide the control system with the ability to generate the main grid's electricity price, such that the control system does not rely on a centralised control signal for its optimal real-time operation. Because of this, the GrA at each MAS container, generates a price forecast signal based on the historical price data stored locally to be sent to the corresponding DER and ESS agents.

This agent is created in each container, as requested by other DER or ESS agent, such that each container is independent of external control signals in terms of microgrid control, in line with the distributed nature of the control scheme. In Fig 3.1 of chapter 3, the containers in yellow show the agent organisation while the arrows represent the type of ACL messages between them, as mentioned in the communication protocols in chapter 2.

Two price forecast method families were studied in this thesis and implemented into the GrA, as presented in chapter 4 and in [7], the NARNET and the WA model, based on MCMC, which are covered in the previous chapter.

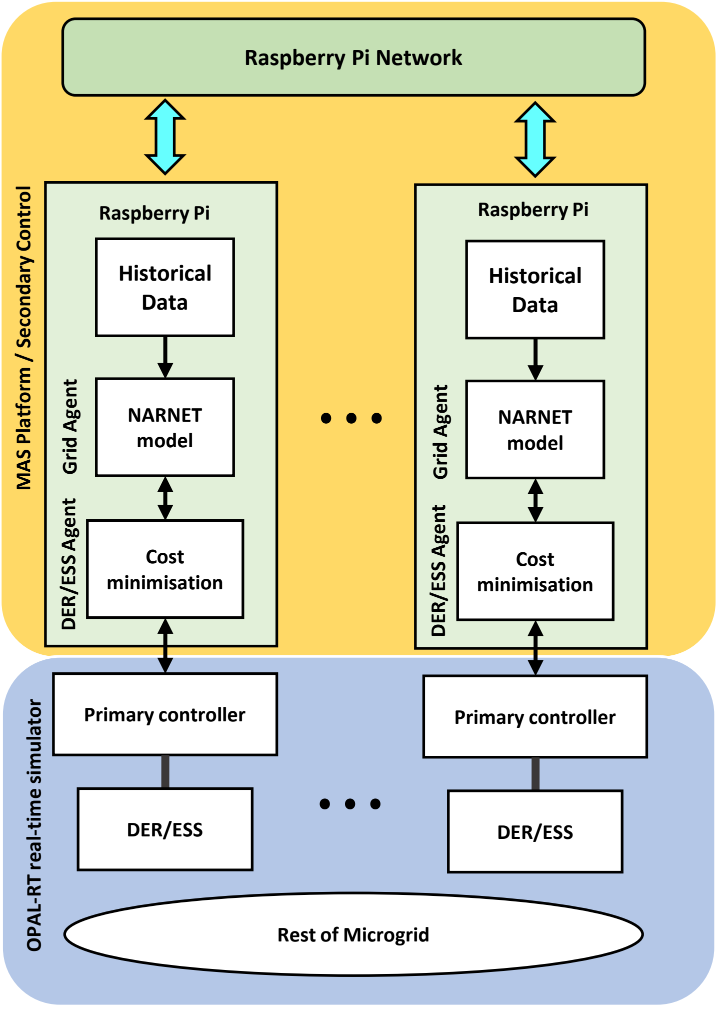


Figure 5.2: Block diagram of the distributed microgrid control system.

Regardless of the price forecast method, its transfer function, that is, the forecast price model based on the previous prices, is obtained and applied to be run continuously by the GrA's behaviours within 10 milliseconds as shown in Figure 5.2.

5.2.4 Wind Speed Forecast Agent

The WSFA is designed essentially as the GrA, with the difference that this agent is used for wind speed forecasting using AR models instead of electricity prices. The wind speed forecast models are covered in the previous chapter. Similarly to its price forecast counterpart, this agent is also created by a request from the RESRA. The properties of the RESRA will be presented in the next section. The objective of this pair of agents is controlling the optimal regulation of renewable generation of a single distributed source. The RES agent generates the appropriate wind speed forecast, while the RESRA maintains the difference between the power schedule and actual power delivered as minimum as possible, which is described next.

5.2.5 RES Regulator Agent

The RESRA uses the wind speed forecast signals from the WSFA to calculate the expected wind power generation using the wind generation model in equation (4.30), presented in chapter 4.

For the case of microgrid control, the accuracy of the power estimation is proportional to the accuracy of the wind speed forecasting method applied.

With the wind power generation values, the RESRA generates the optimal power references for its associated battery to minimise the error between actual renewable generation and power schedule.

The optimisation problem solved by this agent is formulated as:

$$\begin{aligned}
\min \quad & \sum (P_{sch}(i) - P_{wt}(i))^2, \qquad \forall i \in \mathcal{I} && \text{(5.8a)}\\
\text{s.t.} \quad & SOC(i+1) = SOC(i) - \eta P(i) \qquad \forall i \in \mathcal{I}, && \text{(5.8b)}\\
& SOC_{min} \leq SOC \leq SOC_{max} && \text{(5.8c)}
\end{aligned}$$

where P_{sch} is the power schedule and P_{wt} is the wind generation power from (4.30). While P_{sch} is unbounded, P_{wt} is not known in advance, and can only be estimated by the wind speed forecast model. However, the minimisation in real time can be achieved by compensation with a battery. The consequence of this is that if the battery does not have enough capacity to either send or receive power the difference between P_{sch} and P_{wt} will increase. It is therefore necessary to continuously add corrections to the P_{sch} to account for regulation of the SOC of the battery. This has the added benefit that the power from the wind resource is smoothed out over time.

The P_{sch} is then defined as:

$$P_{sch} = P_{fm} + P_{adj} \tag{5.9}$$

where P_{fm} is the calculated power from the forecast model and P_{adj} is the power adjustment to account for SOC net charge or discharge from power regulation.

Given that the power compensation could randomly accumulate a net charge or discharge of the battery over time, this agent also adjusts the power schedule to accommodate for this condition and maintain the state of charge as close to the middle point between SOC_{max} and SOC_{min} as possible. This behaviour of the RESRA maximises the ability of the battery to equally likely absorb or deliver power to compensate for the errors in the wind speed forecast.

The combined behaviours of the agents described previously allow the independent operation of the distributed control realised by the secondary control layer. The next section describes the MAS application to realise the tertiary control layer for the integrated microgrid cluster operation.

5.3 MAS Design for Tertiary Control Layer

This layer has the objective of maintaining stability of the cluster and generating incentives to the individual microgrids inside to use distributed generation. The design presented in this thesis is such that it doesn't override the behaviour of the secondary control layer in the pursue of cost minimisation, avoiding centralisation of the control system. The tertiary control layer implements a price correction mechanism for the internal microgrids.

This allows the entire control hierarchy to cooperate towards the general objective of maintaining the cluster's voltage regulation and provide the incentives to the secondary control to increase distributed generation, even when from the point of view of each of the MAS platforms in the secondary control layer, they are operating to maximise the economic benefit within their microgrids, independently of the rest of the system.

Analogous to the achievement of general objectives of the microgrid by the combination of the behaviours of the individual agents, this interaction is intended to represent the policies that could be implemented in a microgrid cluster to achieve objectives beyond the general objectives of the individual microgrids.

To guarantee that the tertiary control layer does not interfere with the autonomy of the distributed secondary control and the stabilisation mechanisms of the primary control layer, this control layer has a slower response than the first two layers.

The Tertiary control layer in this thesis is mainly directed by the cluster agent at the cluster's main bus, and the DF federation agent, with its corresponding secondary DF agents in the secondary control layers which are presented in the next subsections. The full control hierarchy is shown in Figure 5.3.

5.3.1 Cluster Agent

This agent has the objective of promoting the use of internal generation in a cluster of microgrids connected at a PCC to the main grid by sending correction values to the grid price to each of the DER and ESS agents.

The objective of the cluster agent can be expressed as:

$$\min \quad |V_{ref}(t) - V_{cluster}(t)|, \qquad \forall t \tag{5.10a}$$

where $V_{ref}(t)$ is the reference voltage of the cluster and $V_{cluster}(t)$ is the voltage of the main cluster bus at time t.

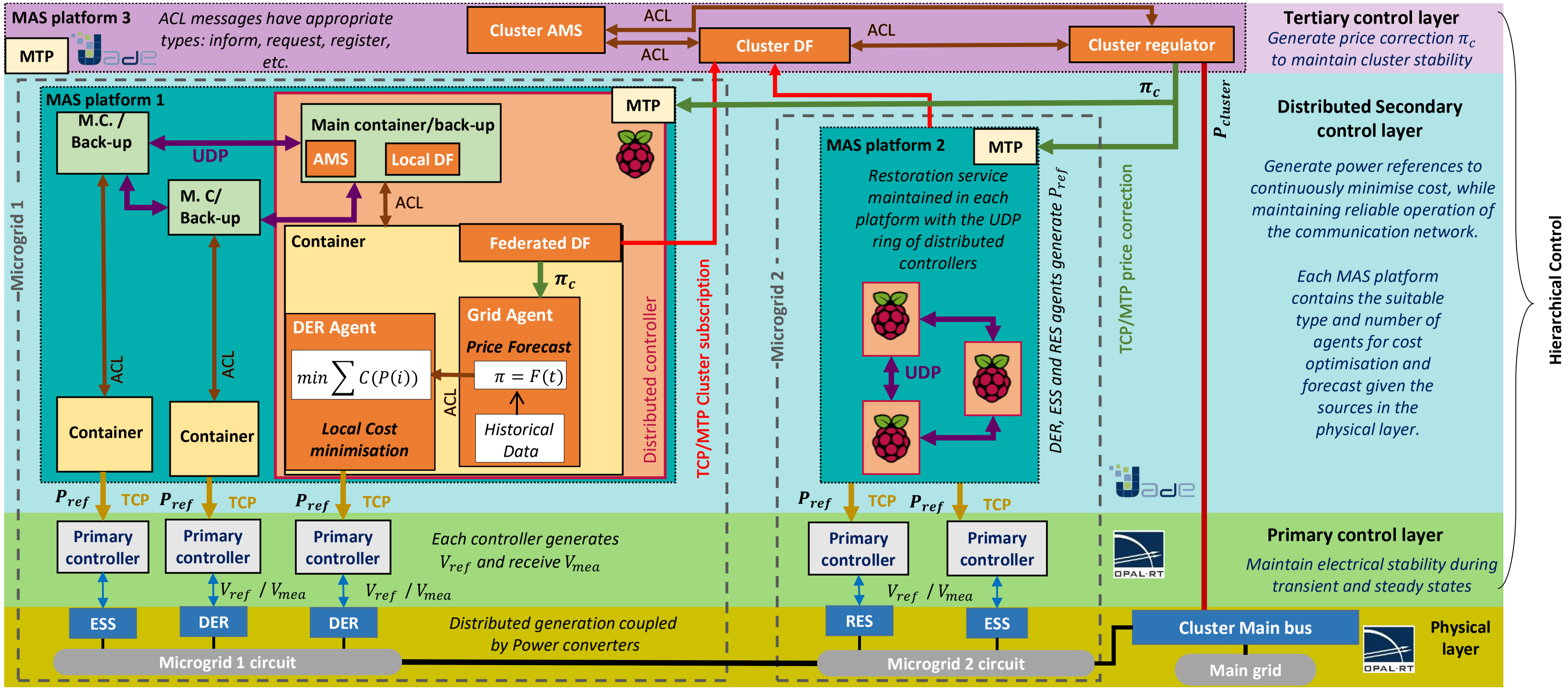


Figure 5.3: Full control hierarchy for the microgrid cluster.

To solve this problem, the cluster agent generates a price correction factor π_c to the registered secondary DFs in the cluster DF, which in turn, is communicated to the DER and ESS agents. The price correction factor can be any non-increasing function with respect to active power consumed by the cluster. In this thesis this function is defined as:

$$\pi_c = -k_1 P_{cluster} - k_2 \tag{5.11}$$

where k_1 and k_2 are positive constants, and $P_{cluster}$ is the power flow of the cluster. For this agent, a positive $P_{cluster}$ indicates that net power is sent to the grid and negative indicates that net power is being consumed by the entire cluster. Because of this, π_c indicate a grid price increase with more power consumed by the cluster and vice versa. The price correction π_c is a function of power to increase the ability of the system to monitor the voltage conditions of the cluster due to power flow, while the primary controllers verify that the cluster remains stable using the voltage directly.

The price forecast sent to the DER and ESS agents is then modified as:

$$\pi = F + \pi_c \tag{5.12}$$

where F is the price forecast generated by the individual GrAs. In the case that the communication between the tertiary and secondary control layers is lost due to any unforeseen failures, the secondary control layer operates with the last π_c received. The communication restoration between the secondary and tertiary control layer is managed by the DF federation agent described next.

5.3.2 DF Federation Agent

This agent allows the restoration of the communication network for the price correction mechanism. The design of this agent follows the same principles as the restoration service presented in chapter 3, except that this agent is capable to form a DF federation across multiple MAS platforms for the control cluster to allow fault-tolerance of the communication network between the tertiary and secondary control layers.

For the formation of the DF federation, the secondary DF agents of the secondary control layer are registered in the DF federation agent in the tertiary control layer. In a similar way, the DER and ESS agents subscribe to the standard DF agent and the secondary agent in their local container. This allows the communication between the cluster agent and the cost optimisation agents to be fault-tolerant against failure of any single agent container or agent main container in any of the MAS platforms in the secondary control layer.

All of the agents used for the operation of the secondary and tertiary control layers are summarised in table 5.1.

5.4 Test cases and Simulation

As different stakeholders in microgrid operation have different objectives, multiple test cases and scenarios are included in this section to analyse the optimal operation of different components of the control scheme presented previously in this chapter. In general, each test case is a real-time simulation model composed of an appropriate physical layer and primary control layer, controlled by hardware in the loop that contains the MAS application of the upper control layers. The test case models presented next validate each of the features implemented for the control scheme presented in this chapter. The test cases are presented such that they subsequently increase the complexity of the problem analysed.

Agent	Inputs	Outputs	Main Actions
Agent Management System (AMS)	Inform and request messages from every agent in the platform	Inform messages to agents	Create and kill agents in the platform. Inform status of platform and messages to relevant agents.
Local Directory Facilitator (DF)	Register and subscription requests from all agents	Inform messages to agents within the same platform.	Form the local platform yellow pages service. Agents subscribe to the DF to request and offer services; this allows the formation of the communication network between agents.
Remote Monitor Agent (RMA)	Inform messages from AMS	Inform and request messages to AMS	Create the GUI to access the platform from different computers and perform requests to AMS. This allows decentralisation of agent management at the hardware level.
Secondary DF	*creation parameters* Cluster DF name and location. Register Messages from GrAs	Register messages to cluster DF	Form the multi-platform yellow pages for cluster coordination in the local containers. This allows resilience against main-container faults.
Cluster DF	Subscription requests from cluster corrector and secondary DFs	Inform messages to notify service and demand match	Form the multi-platform yellow pages service.
Cluster regulator	Main grid bus voltage, Inform messages from cluster DF	Price correction values to remote platforms	Regulate the cluster voltage for cluster regulation. Incentivise the microgrids to use distributed generation.

Distributed Energy resource (DER)	*Creation parameters:* DER Cost function, Pmin and pmax, Local TCP socket. Simulation time Price forecast	Request grid agent to AMS. Power references send to local DER	Calculate the optimal power references to minimise cost in a day ahead scenario, considering current DER status and price forecast.
Grid Agent (GrA)	*Creation parameters:* Name of requester agent (DER/ESS). Simulation time, Inform messages from local DF	Price forecast Subscription to local DF and secondary DF	A DER or ESS requests the creation of a GrA, this allow independence of forecast in each agent container. It access cvs files in the local controller to access historical data. It registers to the local DF and cluster DF Uses a forecast model to generate the price forecast.
Energy Storage System (ESS)	*Creation parameters:* SOC min and SOC max, Round trip efficiency, Local TCP socket. Price forecast, Current SOC	Power references to local ESS for energy arbitrage	This agent realises energy arbitrage. The power references consider both the price forecast and current SOC to steer the availability of the ESS, following offline predictive control.

Renewable Energy Source Regulator Agent (RESRA)	*Creation parameters:* SOC min and SOC max, Round trip efficiency, Local TCP socket. Wind speed forecast, Current SOC	Power references to local ESS for RES compensation	Generate the power references to maximise the RES generation. This agent also considers current SOC to guarantee ESS availability.
Wind speed Forecast Agent (WSFA)	Simulation time, Inform messages from local DF	Wind speed forecast	A RES agent requests the creation of a wind forecast agent. It access cvs files in the local controller to access historical data. Uses a forecast model to generate the wind speed forecast.

Table 5.1: Summary of all of the agents used in this thesis.

5.4.1 Microgrid Cost Optimisation

To establish the feasibility of the MAS approach to secondary control layer, its ability to generate the power schedule is first tested against centralised heuristic optimisation for cost minimisation, independently from forecast models. Secondly, uncertainty in grid prices is introduced to demonstrate that the cost minimisation is feasible in this conditions as well.

5.4.1.1 Optimisation Solver Comparison

The following test cases are intended to demonstrate the ability of the MAS to minimise total supply cost, compared to other solvers. Additionally, the calculation time required is also evaluated to demonstrate that the MAS application is suitable for real-time power management when combined with the primary control layer. The evaluation focuses on the cost minimisation, and therefore these cases do not consider price uncertainty.

The test microgrid model and its scenarios are based on the optimisation problem from [55], which is composed of a MT, a FC, a battery, and a load connected to a single bus and to the main grid.

As in the benchmark optimisation problem, three scenarios for cost minimisation are presented: The low load, where the demand is lower than the distributed generation, the high load case, with load higher than internal DER capacity, and a stand alone case where there is no exchange of power between the grid and the microgrid. In the grid connected scenarios it is assumed that energy trading between the main grid and microgrid is possible at the same price.

The components are modelled with the repository from [140]. The line and control parameters used are found in Table 5.2.

The physical layer of the microgrid and the primary control layer of this test case are modelled in RT-LAB and using the OPAL-RT OP5700 real-time simulator. The

Table 5.2: Microgrid simulation parameters.

DER Lines		**LCL filter**		**Inner loop gains**		**Outer loop gains**	
R	0.1Ω	L grid	$1.5mH$				
X_L	0.002Ω	C	$2.6mF$	P_{inner}	1.5	I_P	1
V_L	$400V$	L converter	$10mH$	I_{inner}	20	I_Q	15
f	$50Hz$						

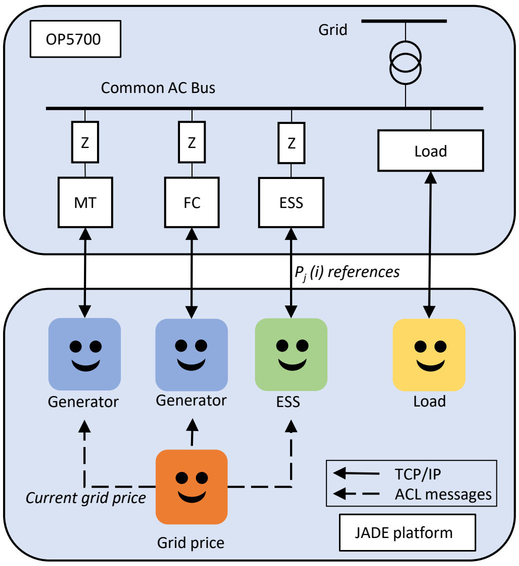


Figure 5.4: Microgrid simulation model and MAS platform used for the first test case in this chapter. In this model, the grid price agent provides the actual prices and the load agent depicted is used for the load variation scenarios.

secondary control layer is a JADE based MAS platform for a distributed secondary control layer [73], interfaced as shown in Figure 5.4. The circuit model operates in 3-phase at 400 Volts RMS from line to line at 50 Hz. The DERs are modelled as a 1500 V DC source connected to a 2-level inverter with an LCL filter.

The circuit model is tested with a simulation time step of 5×10^{-5} seconds, for one day with every hour being represented by 4 seconds of simulation, that is, the price and load that would change after one hour in the test case in the real scenario, change every 4 seconds instead. This is done to focus on the transient state stabilisation, given that after 4 seconds the electrical model is already in steady state. The rest of the variables of the simulated model, such as the electrical variables, vary in real time (no compression is used).

Table 5.3: Test case grid prices.

Hour	Euro ¢/kWh	Hour	Euro ¢/kWh	Hour	Euro ¢/kWh
1	2.5	9	15	17	6.2
2	2	10	40	18	4.4
3	1.5	11	40	19	3.7
4	1.3	12	40	20	5
5	1.2	13	15	21	11.9
6	2.1	14	40	22	5.3
7	2.3	15	21	23	3
8	3.9	16	19.7	24	2.7

Table 5.4: DER cost function parameters.

DER paramter	Micro turbine	Fuel Cell	ESS
P_{min} (kW)	6	6	-30
P_{max}(kW)	30	50	30
c_B (Euro ¢/kWh)	4.37	2.84	0
c_C (Euro ¢)	85.06	255.18	0

As a point of comparison for the MAS approach, the power management of the secondary control layer is also developed for offline optimisation in MATLAB applying heuristics to approach the global minimum cost for three solvers, Genetic Algorithm (GA), Particle Swarm Optimisation (PSO) and Pattern Search (PS), along with the results from [55].[*] The optimisation problem is solved with an i7-6700 CPU at 3.40 GHz with 16 GB of RAM.

In Tables 5.3 and 5.4 the parameters for this test case are provided [55].

5.4.1.2 Cost Minimisation with Price Uncertainty Test case

Next, are included the test cases where the ability to maintain minimum cost is analysed when the electricity price is unknown to the microgrid's secondary control, in line with distributed control. To do this, the GrA implements a grid price forecast model, as opposed to simply inform the actual prices in advance as in the previous case in this section.

[*] Refer to the MathWorks Global Optimisation Tool box for more information about these solvers.

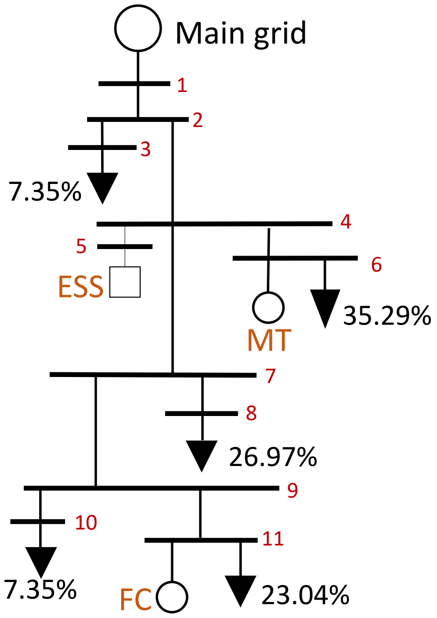


Figure 5.5: Microgrid physical layer and load distribution for the cost minimisation with price forecast test case.

Table 5.5: DER cost function parameters.

DER parameter	**Micro turbine**	**Fuel Cell**	**ESS**
P_{min} (kW)	6	6	-30
P_{max}(kW)	30	50	30
C_B (pence/kWh)	4.37	2.84	0
c_C (pence)	85.06	255.18	0
s	9	16	0

Centralised and distributed forecast comparison

The physical model of this test case implements further non-linear parameters in the cost minimisation problem, in the form of the start-up cost, which makes necessary the use of price forecast methods for the DER agents. In the previous case only the ESS state is affected by the changes in price as it cannot change its SOC within one hour. The microgrid model for this case is shown in Figure 5.5.

The generators parameters are described in Table 5.5 and the total electricity demand is found in Table 5.6. Table 5.7 describes the admittance of the power lines.

For the first scenario applying price forecast, the GrA applies the MCMC method

Table 5.6: Microgrid total demand.

Hour	kW	Hour	kW	Hour	kW	Hour	kW
1	16.32	7	18.36	13	33.66	19	52.02
2	15.30	8	24.48	14	36.72	20	61.20
3	13.26	9	26.52	15	36.72	21	55.08
4	11.22	10	27.54	16	30.60	22	46.92
5	12.24	11	30.60	17	30.60	23	33.66
6	14.28	12	33.66	18	40.80	24	18.36

Table 5.7: Line admittance of the test microgrid.

Line	Siemens	Line	Siemens
$y_{1,2}$	46.3435 - 13.5440i	$y_{4,7}$	46.3435 - 13.5440i
$y_{2,3}$	9.0275 - 0.2299i	$y_{7,8}$	205.7613 - 205.7613i
$y_{2,4}$	92.6870 - 27.0881i	$y_{7,9}$	3.3505 - 0.0979i
$y_{4,5}$	2.4153 - 0.0143i	$y_{9,10}$	205.7613 - 208.7613i
$y_{4,6}$	17.5886 - 3.6600i	$y_{9,11}$	19.2675 - 2.0134i

based on the data of UK hour prices from 16/05/19 to 26/09/19 in GBP/MWh, obtained from Nord Pool, as described in chapter 4, where the accuracy of this method is validated.

To test the cost minimisation performance with grid price forecast, four cases for different price scenarios are used: case 1a is double the average Friday's price for the prices between 16/05/19 to 26/09/19, case 1b is double the price from 27/09/2019, case 2a is triple the Average Friday price and in case 2b the price is triple the price from 27/09/2019.

This test cases compare the cost minimisation between three forecast conditions: Sharing a single price forecast used by all the agents in a centralised manner; using multiple GrAs, labelled as independent forecast for increased autonomy of each DER within the microgrid, and sending the actual prices in advance to the agents, such that no forecast error exist, labelled as ideal case.

NARNET and WA Method Cost Comparison

This case is used to analyse the differences between the NARNET and WA price forecast models in terms of cost minimisation as part of the MAS application, complementing the analysis presented in chapter 4 in terms of error.

This test case uses the same microgrid model as the previous case, with an ESS and 2 dispatchable generators, a MT and a FC and a varying total demand between 11.22kW and 61.2 kW during the day which is distributed in the circuit as shown in Figure 5.5. The physical layer and the primary control layer are modelled within an OPAL-RT real-time simulator. The MAS platform is deployed using two Raspberry Pi3 model B+ and a PC over a local area network for the secondary control layer.

In the following cases, the total supply cost of the microgrid is calculated for the best iteration of each of the price forecast models implemented in the GrA, as illustrated in chapter 4, with the real price considered the ideal case as reference. This is done for each of the following price data scenarios: no weekends and no outliers, no weekends, no outliers and all data.

Following the cost analysis done in [7], all the prices are adjusted to double and triple the values of the original historical UK price data set, as the original data set is intended for wholesale market, which does not accurately reflect the electricity price to the consumers at the retail market, this also contributes to examine the effect of the accuracy of the forecast model in terms of supply cost minimisation with respect to local cost functions of distributed sources. Similarly to previous test cases, it is assumed that the microgrid can trade with the grid at the UK price to buy and sell energy, in all cases the reactive power reference of the primary control layer is set to zero.

5.4.2 Microgrid Renewable Optimisation

Similarly to the case of uncertain prices, randomness in the power management is also introduced with the use of renewable distributed resources. In this thesis, an AR model approach is used to generate a short term wind speed forecast. As the wind speed forecast cannot directly be analysed in terms of cost as in the price case, the validation of these forecast methods is presented in chapter 4 of this thesis.

In this case study, the ability of the RES agent to maximise the use of wind power

Line	Siemens	Line	Siemens
$y_{16,17}$	$100.6035 - 0.0935i$	$y_{16,1}$	$100.6035 - 0.0935i$
$y_{16,8}$	$50.3017 - 0.0467i$	$y_{1,18}$	$100.6035 - 0.0935i$
$y_{18,3}$	$100.6035 - 0.0935i$	$y_{18,2}$	$9.0334 - 0.0007i$
$y_{3,19}$	$50.3017 - 0.0467i$	$y_{3,4}$	$18.3503 - 0.01215i$
$y_{19,20}$	$33.5345 - 0.0311i$	$y_{19,5}$	$411.5184 - 1.3090i$
$y_{20,6}$	$411.5184 - 1.3090i$	$y_{20,7}$	$19.4779 - 0.0064i$
$y_{8,9}$	$83.9626 - 0.1875i$	$y_{9,10}$	$58.0718 - 0.0947i$
$y_{10,22}$	$41.9813 - 0.0939i$	$y_{10,11}$	$58.0718 - 0.0947i$
$y_{21,12}$	$27.3672 - 0.0227i$	$y_{21,14}$	$27.9875 - 0.0626i$
$y_{12,13}$	$2.7367 - 0.0002$	$y_{14,15}$	$83.9626 - 0.1875i$

Table 5.8: Line admittance for the microgrid cluster.

generation based on the forecast with the highest error is tested. This will be used to validate the wind speed forecast and power compensation methods in the worst case scenario available in the dataset. The scenario uses the data from the 3rd of October of 2017. On this date, the largest sudden increase of wind speed during the period covered by the data set is found. This data is chosen given that the NARNET model, as any AR model, may have an accuracy decrease with sudden changes in the time-series, which forces the use of the compensation mechanism.

The simulation for this test case and the cluster test case is described in Figure 5.6. The values of admittance of each power line is found in Table 5.8. For the renewable optimisation test case, the microgrid 2 is modelled in the real-time simulator.

As in the previous test cases, the physical layer and the primary control layer are modelled using an OPAL-RT real-time simulator and the secondary control is built externally using Raspberry-Pis.

5.4.3 Microgrid Cluster Optimisation

The final case study is the integration of all the control and forecast methods interacting simultaneously in a single circuit. This includes the three control layers in a single physical layer, where microgrid 1 has the objective of cost minimisation and microgrid 2 has the objective of renewable generation maximisation at the secondary control layer.

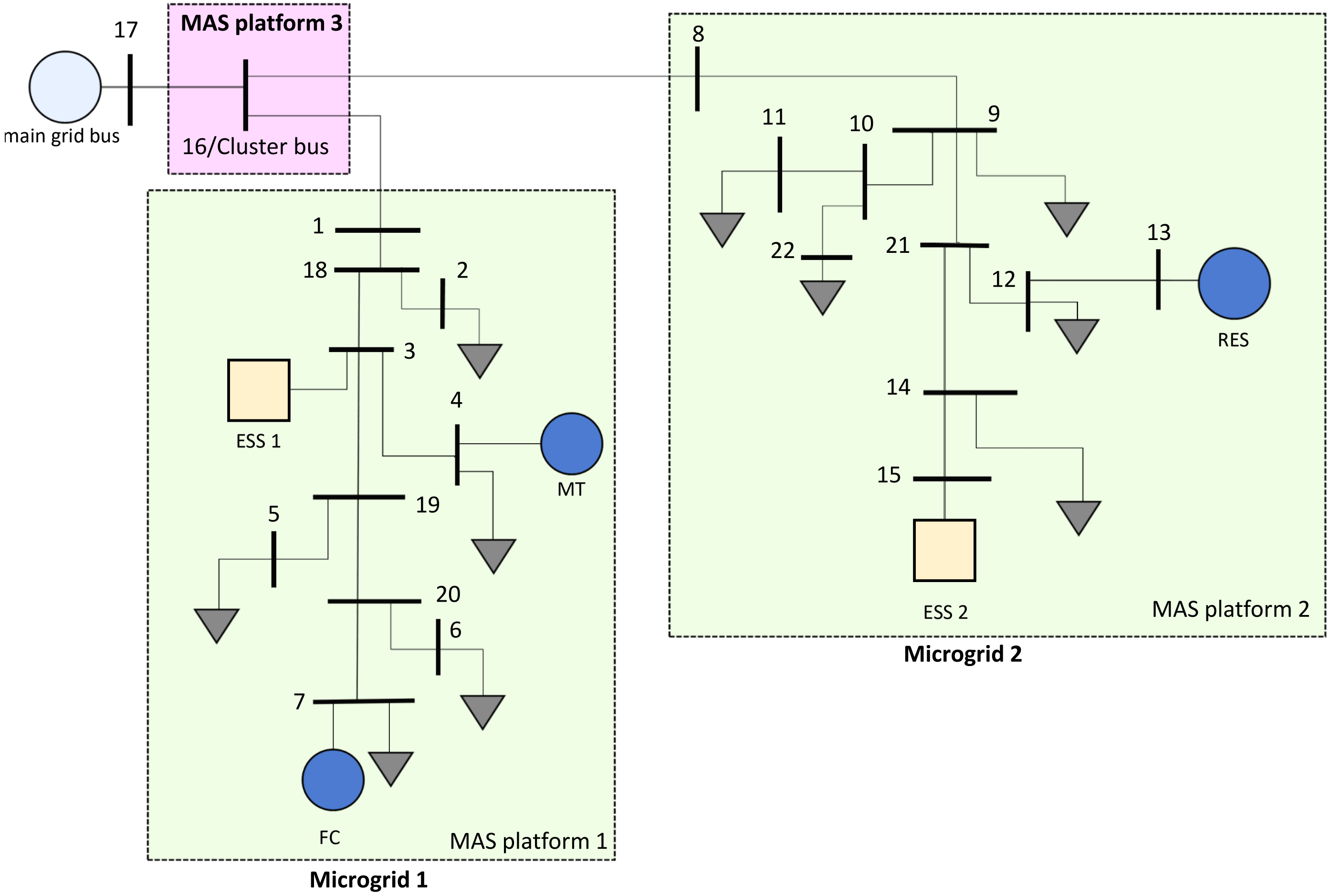


Figure 5.6: Physical layer of the microgrid cluster test case. The boxes represent the area on which the different MAS have direct communication. The buses are enumerated for reference in the simulation results section of this chapter.

To verify the functionality of the proposed model, the price correction parameters of the cluster agent at the tertiary control layer are $k_1 = 0.177$ and $k_2 = 22.222$. This values, in combination with the cost functions used in previous chapters and the net power flow will influence the secondary control layer of microgrid enough to see the effect of increased internal generation.

However, some room has been left for microgrid 1 to take advantage of grid prices such that it can be verified that the secondary control maintains its capacity of solving its own cost minimisation problem, omitting a trivial solution, such as setting the internal generation to be activated at all times.

As mentioned in the previous test case, the physical layer of this case is illustrated in Figure 5.6, the power lines are described in Table 5.8. In this case, the entire circuit is modelled in the OPAL-RT real-time simulator, along with the corresponding primary controllers in each of the DERs.

5.5 Simulation Results and Discussion

This sections presents the simulation results of the test cases described previously in terms of performance of the entire simulation model with the control system. The accuracy analysis for the price and wind speed forecast methods are covered in the previous chapter.

5.5.1 Microgrid Cost Optimisation Results

5.5.1.1 Solver Comparison Results

The supply cost results are plotted against computing time for 10 runs for each solver and each scenario as shown in Figure 5.7. The time axis is presented in all cases in logarithmic scale for this test case.

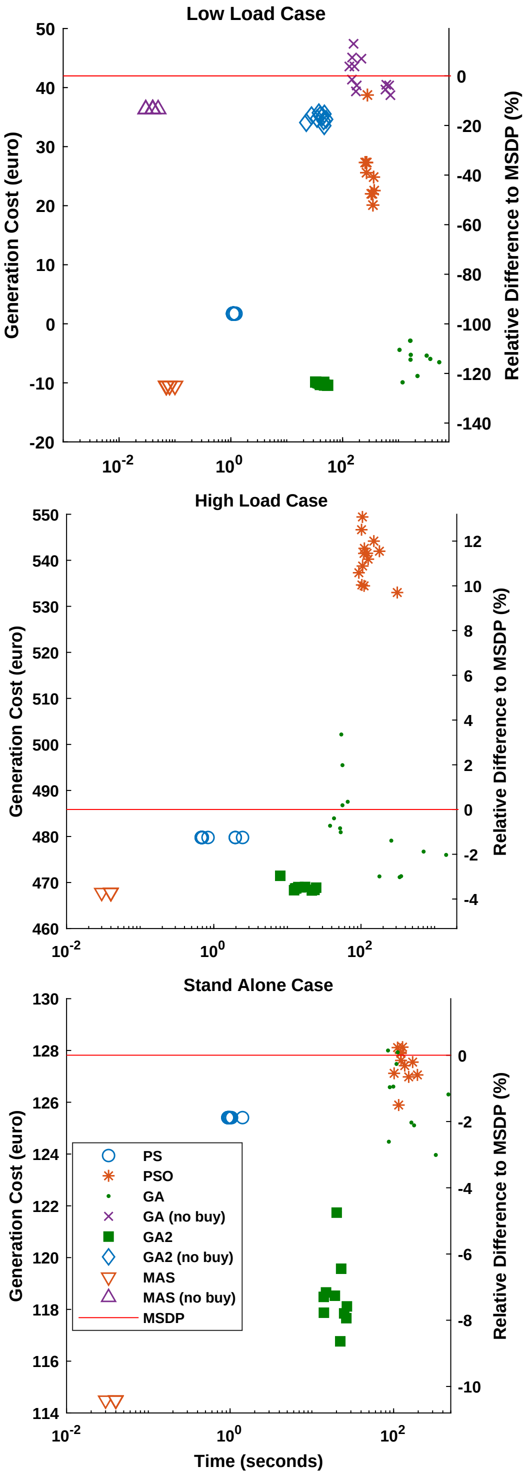


Figure 5.7: Optimisation results for cost minimisation. Top, low load case. Middle, high load case. Bottom, stand-alone case.

Table 5.9: Supply cost minimisation per solver Summary.

Solver	Average Cost (euro)	Average CPU calculation time (seconds)	Type of Solver	Relative Difference to benchmark(%)
Low Load Case				
PS	1.71	1.11	Heuristic	-95.91
PSO	25.78	308	Heuristic	-38.57
GA	-5.81	2323	Heuristic	-113.84
GA (no buy)	42.06	325	Heuristic	0.21
GA2	-10.14	42.3	Combined	-124.16
GA2 (no buy)	34.82	39.6	Combined	-17.03
MAS	-10.49	0.08	Numerical	-124.99
MAS (no buy)	36.33	0.04	Numerical	-13.43
High Load Case				
PS	479.78	1.03	Heuristic	-1.25
PSO	540.47	132.6	Heuristic	11.23
GA	481.9	263	Heuristic	-0.82
GA2	468.97	16.2	Combined	-3.47
MAS	467.82	0.39	Numerical	-3.71
Stand Alone Case				
PS	125.4	1.01	Heuristic	-1.88
PSO	127.42	134.7	Heuristic	-0.30
GA	129.16	172	Heuristic	1.05
GA2	118.52	20.5	Combined	-7.27
MAS	114.49	0.39	Numerical	-10.42

The vertical left axis measures the cost found in euros, while the vertical right axis measures the relative difference to the benchmark total cost in [55], where Multi-Stage Decision Programming (MSDP) was used for the cost minimisation. The relative difference is calculated as the difference of the cost found and the MSDP cost divided by the MSDP cost, shown with a red line. GA refers to the fully heuristic approach and GA2 to the combination of the heuristic and numerical approach.

For the Low load case, an additional no buy policy is tested for GA, GA2 and MAS, as the DERs capacity is higher than the load, no energy is supplied from the grid. For the High load and Stand-alone case, all solvers reach a relatively similar solution. A summary of the average costs and calculation time for each solver is found in Table 5.9, the optimisation solvers stop when the total cost variation with respect to the previous iteration is below 1×10^{-6} euro cents.

In all of these cases it can be seen that the MAS approach is significantly faster, taking milliseconds to obtain the power schedule, while the other methods take minutes and the result varies even with the same initial conditions. The MAS approach consistently reaches the global minimum in the 3 scenarios except in the no buy scenario of the low load case where the GA2 is able to find a slightly better solution, however the average calculation time required is 39.6 seconds compared to 40 milliseconds for the MAS approach.

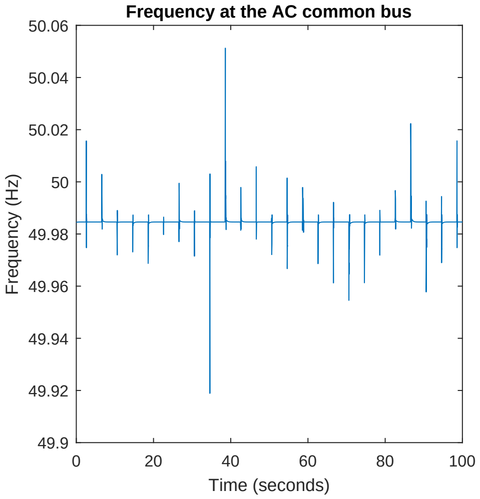
The energy management capacity of the MAS control can produce consistent, fast and optimal results compared to the heuristic methods, the small calculation time allows real-time optimisation as the power schedule is generated while the simulation is running.

In Figure 5.9 the response of the DER to the set points sent by the ESS agent are presented. Smooth operation can be seen for reference changes from -30kW to 30kW. Figure 5.8 illustrates compliance with the GB National Electricity Transmission System Grid Code. This validates the primary controller as well as a suitable control system for the microgrid in terms of stability and response time. A more detailed response for the physical layer will be presented in the last test case of this thesis.

5.5.1.2 Cost Minimisation with Price Uncertainty Results

Centralised and distributed forecast comparison

Three scenarios are evaluated for supply cost with price uncertainty to verify the ability of the distributed control system to achieve optimal power management. For these simulation results, the first case consist in providing each of the DER and ESS agents the real price values in advance, such that the forecast has error equal to zero. This case is labelled as ideal. In the second case, a single price forecast is realised and transmitted to the rest of the agents, which is labelled as shared



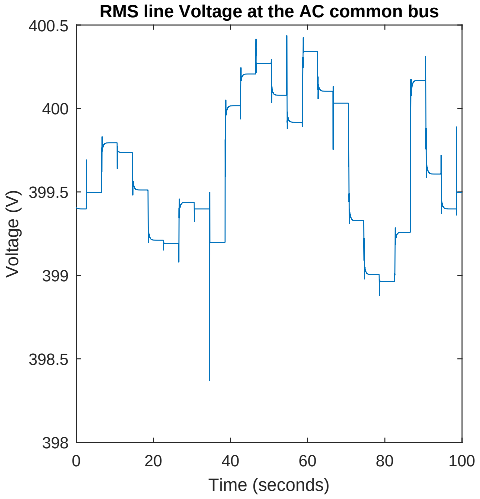


Figure 5.8: The frequency and voltage variation are within the 50 Hz ±1% range and the 400V ±5% range.

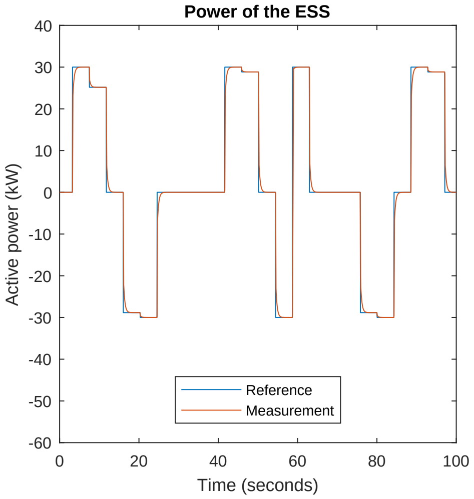


Figure 5.9: Control response for the active power of the ESS.

Table 5.10: Average extra cost per day in GBP for the MCMC price forecast method.

Price	Date	Shared	Independent
Double	Average Friday	0.373	0.377
	27/09/2019	1.347	1.426
Triple	Average Friday	0.222	0.329
	27/09/2019	3.223	3.287

forecast. In the third case, each of the DER and ESS agents request an independent GrA to realise a local price forecast, which is labelled as independent.

The price π_0 estimated by the MCMC was doubled and tripled as the price data is intended for large scale suppliers and does not reflect the actual end consumer price, which for the UK was in average 13 pence per kWh in 2019. This is done to better study the effect of the estimation compared to the actual price at the distribution level.

The Table 5.10 shows the extra cost from the MCMC price forecast compared to the ideal case, it can also be observed that sharing a single price estimation or using the same data to generate price signals independently does not have a significant effect on the total cost for the test microgrid, but allows further distribution of the control.

The results from these scenarios are summarised in Figure 5.10, for each price case, the bar represents the average cost after 10 runs. The results are compared with the Ideal case of having the actual prices of each case.

NARNET and WA method cost comparison

From the scenarios discussed in the test case section, the total supply cost is obtained. Each of the scenarios is run with the WA method and the NARNET method implemented in the GrA and compared with the ideal case, where the forecast has no errors, in other words, the real price is known in advance. From the Quasi-Newton optimisation of the WA, the weights of the best performance are: $W_{am} = [0.48\ 0.34\ 0.96\ 0.03\ -0.82]$. The cost results are shown in table 5.11.

Using double and triple prices illustrates the effect that the price has in the cost

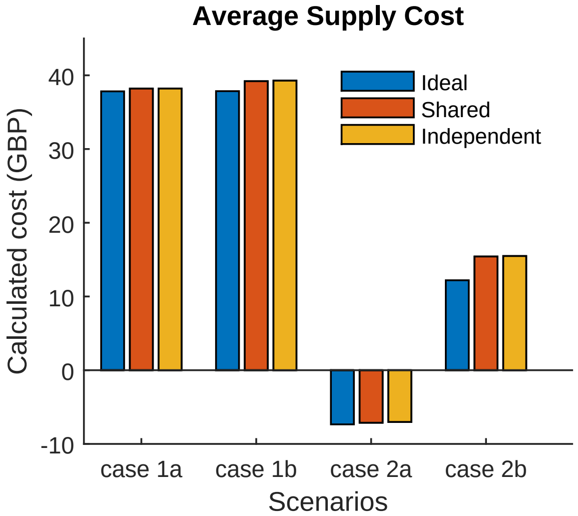


Figure 5.10: Comparison of centralised and distributed price forecasting with the MCMC method. Case 1a is double the average Friday prices. Case 1b is double the price from 27/09/2019. Case 2a is triple the average Friday prices. Case 2b is triple the price from 27/09/2019.

Table 5.11: Total Cost comparison in GBP for the NARNET and WA forecast method comparison. The Perfect information scenario refers to using the actual prices in advance instead of forecasting them.

Price	Scenario	Perfect Information	NARNET	MCMC
Double	no weekends, no outliers	17285.19	18358.56	19317.81
	no weekends	17250.52	19103.90	19384.47
	no outliers	24243.42	26046.88	27929.09
	all data	24259.32	27248.01	27983.01
Triple	no weekends, no outliers	-12470.42	-11733.26	-11631.11
	no weekends	-13908.33	-12348.67	-12913.53
	no outliers	-17185.72	-15063.10	-15868.16
	all data	-18513.67	-16140.25	-16712.16

minimisation combined with the accuracy of the forecast method and the particular load of the microgrid. For the high price case, the negative values in Table 5.11 represent net profits of the microgrid from energy trading with the grid. This shows that for the load in this case is sufficiently low for the microgrid to benefit from higher grid prices.

It can be see that the NARNET method performs worse than the WA when the outliers are included in the data, which is consistent with the previous publication presented in [7]. However, for the scenario of triple price without outliers and weekends, the NARNET is below 1% difference compared to the perfect information case over two years, which showcases the forecasting capacity of this method in terms of total cost.

5.5.2 Renewable Optimisation results

The power response simulated for the renewable wind generation simulation is presented in Figure 5.11. The top subplot shows the wind generation during the simulation. The middle subplot shows the power and SOC of the compensation battery. It is notable that full charge is reached between 20:00 and 21:00. At this point, as shown in the bottom subplot, the compensation cannot occur between the wind generation and the power schedule. However, after 21:00, the compensation is restored and the error between actual generation and forecast is minimised.

As the test case presented represents the worst case scenario, it can be concluded that for any other conditions, the secondary control system is capable of maintaining its power schedule with minimal deviations with the use of the compensation mechanisms.

To better understand the effects of the proposed control schemes for distributed optimisation with uncertainties, the simultaneous operation of the control schemes in a single microgrid cluster is presented next. This is done to demonstrate that the performance of the combined control schemes is maintained as described by the simulations results presented up to this point.

5.5.3 Microgrid Cluster Optimisation Results

To summarise the effects of the different control layers, the cluster model is run for one day in three different cases. For first case of this subsection, only the primary control layer is active. In the second case, the primary and secondary control layer are activated as shown in the case studies of previous chapters. In the third case, all control layers are activated, to validate the effect of the tertiary control layer

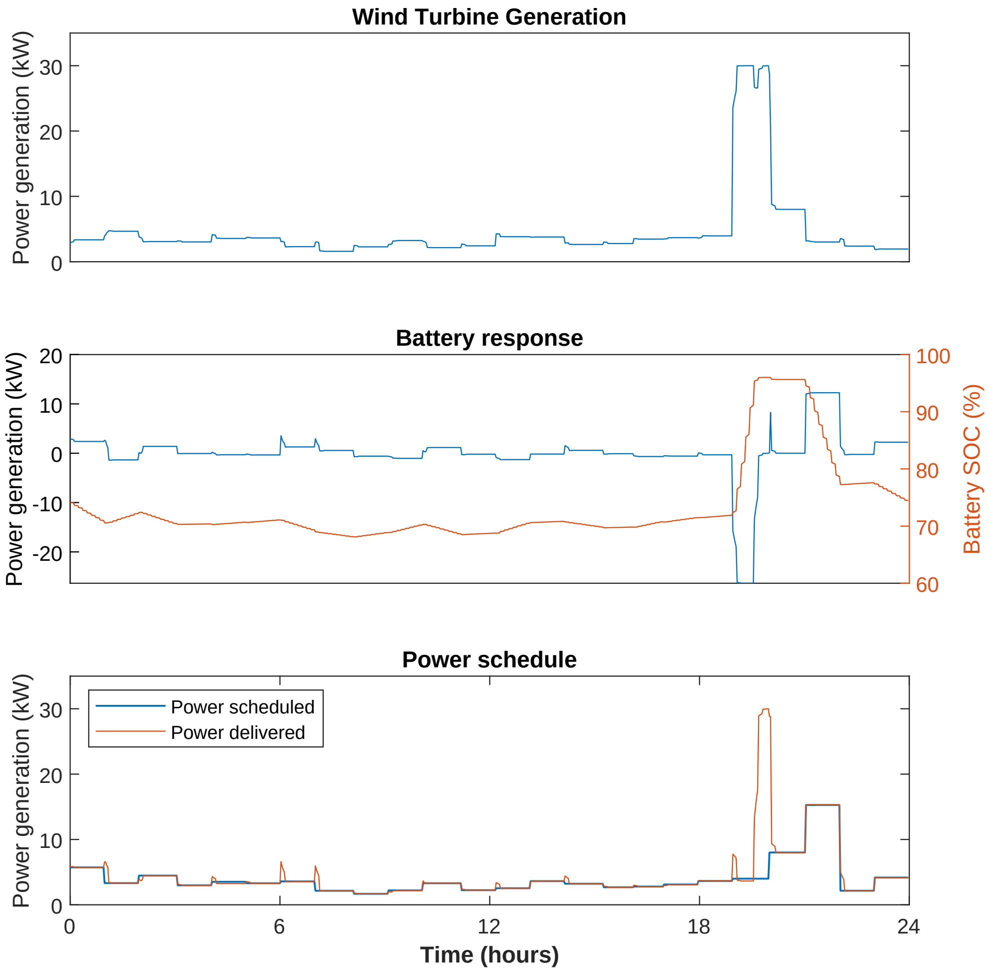


Figure 5.11: Microgrid 2 response for the worst case scenario. The power schedule refers to the combined power of the turbine and the battery of this microgrid.

as a cluster regulator, in conjunction with the other control layers. The voltage response of the cluster for each case is explored next.

Simulation results of the first case are shown in Figure 5.12. Results from this case demonstrate that the individual primary controllers can also maintain stability in the cluster circuit. As mentioned previously, this case establishes a point of reference for the effect of the secondary and third control layers.

Simulation results of the second case are shown in Figure 5.13. With the activation of the secondary control layer in both microgrids, optimal power management is achieved as described in the previous subsection. This case illustrates that independent secondary control layers can operate in the same circuit simultaneously.

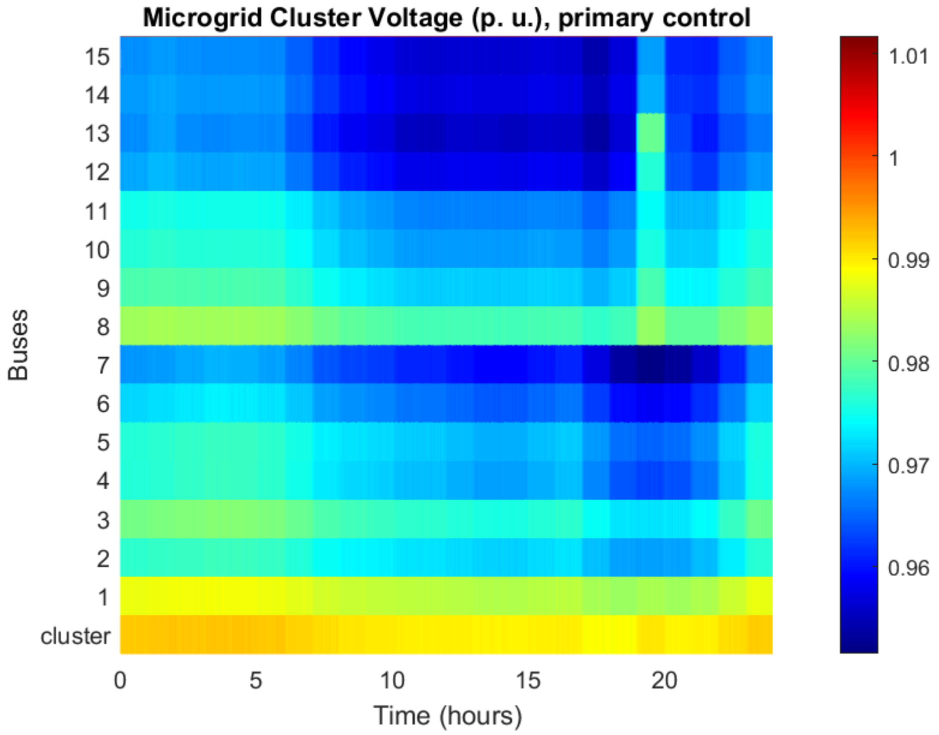


Figure 5.12: Voltage response of the microgrid cluster with only the primary control layer active. The bus are enumerated according to Figure 5.6.

Results also reassure that stability is maintained within range.

Simulation results of the third case are shown in Figure 5.14. Here it can be seen that the internal generation is increased from the contributions of the fuel cell. Given that the internal generation remains the same for the micro turbine, it is shown that the autonomy of microgrid 1 is preserved in line with distributed control. Another important result from this simulation is in microgrid 2. While the secondary control of microgrid 2 is not affected by the change of the main grid's electricity price, the third layer control contributed to its voltage regulation. By increasing the internal generation, the voltage of the entire cluster is increased. This demonstrates that the cluster congestion is decreased, and therefore, the capacity for renewable generation is increased.

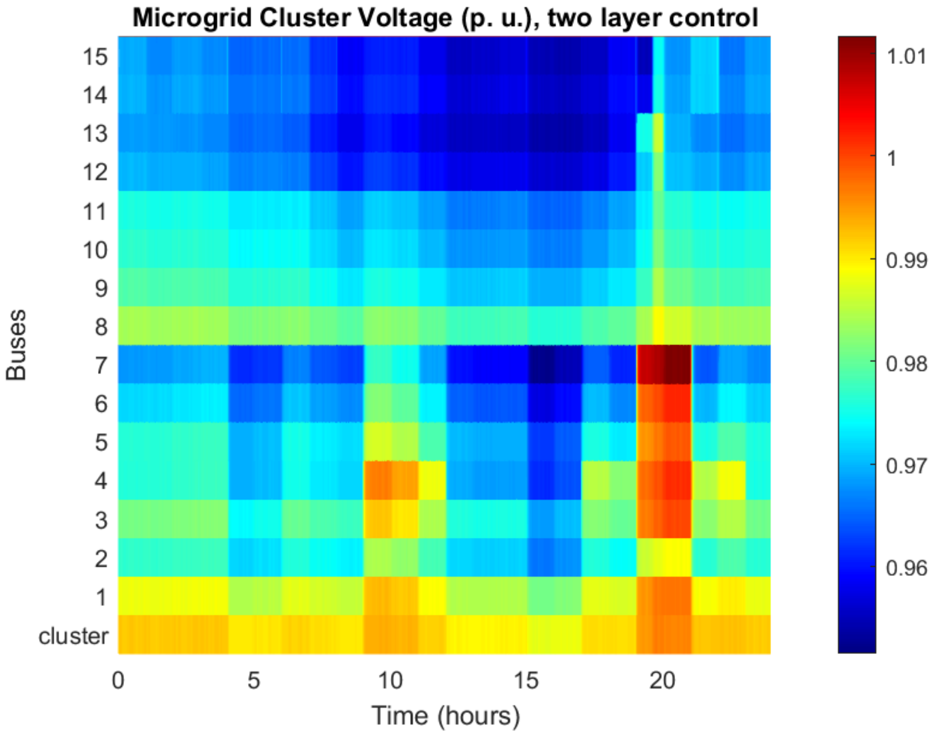


Figure 5.13: Voltage response of the microgrid cluster with the primary and secondary control active. The bus are enumerated according to Figure 5.6.

5.6 Summary

In this chapter the real-time distributed control is presented. This is an application that demonstrates the ability of the designed control schemes to achieve optimal operation. This chapter also validates the stability and reliability of the primary controllers evaluated in chapter 3. In a similar manner, the generation of the power schedule used in this chapter is a result of the implementation of the distributed forecast methods explored in chapter 4.

This chapter serves as an integration of the previous concepts presented in this thesis. The ability of the control system to achieve multi objective optimisation in a reliable, real-time distributed scheme is demonstrated. It also shows improvement on the benchmark case and therefore shows the potential for distributed generation to facilitate an electricity service model that could lead to a cheaper and more

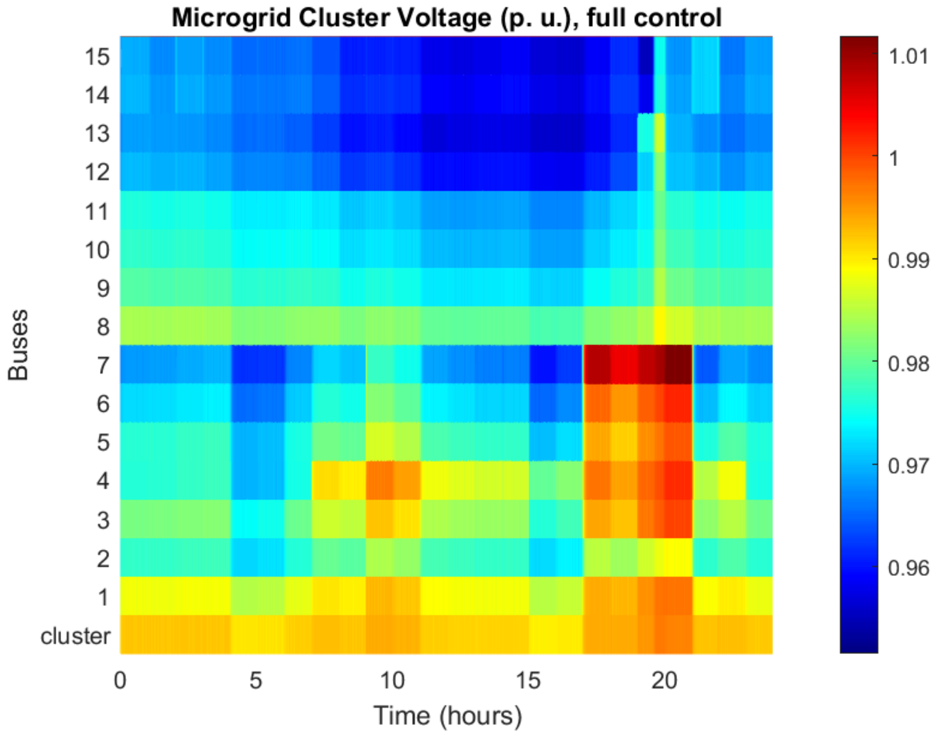


Figure 5.14: Voltage response of the microgrid cluster with the full hierarchical control active. The bus are enumerated according to Figure 5.6.

reliable smart grid.

It is also an important part on the design of this system to consider that the system is continuously optimising its cost function based on forecast values, this means that, as opposed to similar works in microgrid cost minimisation with a defined point for the end of operation, the design implemented here does not contemplate such condition, which means that even at the end of the scenario studied, the control system behaves as it would continue operation.

While real systems cannot operate Ad infinitum, it is reasonable to expect that they should be ready to continue operation as long as possible at least for several days to minimise the management required by an operator or end user.

Finally, The cost analysis shows that the effectiveness of the secondary control layer in terms of supply cost minimisation and renewable generation maximisation of the microgrid cluster depends on the accuracy of the forecast methods. Another factor in supply cost minimisation is the relation between electricity price and electric load, as the higher price case at the load tested actually achieved smaller cost than the lower price.

CHAPTER 6

Conclusion and Future Work

This thesis has contributed with the formulation, simulation and analysis of a suitable control scheme to further increase the economic and technical viability of distributed generation, which aims to promote their use, and in turn, contribute to the realisation of the smart grid. This chapter summarises the main findings from the test cases of the previous chapters. The thesis finalises with the layout of the future work in which the control system designed can be further improved and validated.

6.1 Conclusions and Main Contributions

In this thesis, a control system for microgrids is presented and analysed. As presented in chapter 2, the control system studied is based on distributed control to mimic the distributed generation nature of microgrids as part of the bidirectional power flow paradigm. The control system is organised in a hierarchy of three main layers: The primary control layer used for stabilisation and regulation of the electrical parameters of each generation source in real-time; the secondary control layer for the coordination of generation sources in individual microgrids; and the tertiary control layer for the power management of microgrids in a microgrid cluster, aiming at maximising the use of distributed generation assets.

The design of the primary control presented in chapter 2 of this thesis allows the operation following a voltage reference or the power references provided by the secondary control layer. This allows the primary control to maintain regulation of the system, independently from the state of the power management system.

As shown in chapter 3, the design of the control scheme implemented in this thesis for the primary control layer is stable and reliable, even in the event of faults in the underlying communication network in the secondary and tertiary control layers. From the perspective of stability of the primary control, aimed with regulating the electrical variables of the system, the design of the primary control layer, with the line model used in this thesis (short line RL model), is guaranteed to be Lyapunov stable, within the analysed constraints. Due to the nature of the power flow between two buses, it is also verified in chapter 3 that any control system that regulates power flow between two AC buses based phase angle and voltage, as in the outer loop of the primary controller implemented, cannot be globally asymptotically stable as infinitely many equilibrium points exist.

The evaluation of the forecast models designed for the distributed control paradigm of the secondary control layer, explained in Chapter 4, shows that the models used are suitable for this type of control either from cost reduction, when combined with the appropriate cost minimisation agents, or with statistical hypothesis testing for the case of wind speed forecast. The evaluation framework for the wind speed forecast models also provides insight on the efficient use of computational resources required to perform the wind speed forecast when compared to other models. In this regard, the evaluation by the Diebold-Mariano test verifies the fact that a forecast model has a statistical significant difference with another method with similar performance in terms of errors. This is of great importance for distributed control as one of the trade-offs with fully decentralised control methods is the amount of computational resources dedicated to maintaining the communication network of the control system.

It is verified in chapter 5 that the combined artificial intelligence methods, namely, multi-agent systems and non-linear auto-regression artificial neural networks allow for an optimal operation of the microgrid, and microgrid clusters in economic terms in a distributed control scheme. The artificial neural network allows the system to forecast the grid price and wind speed, which is communicated with the corresponding optimisation agents without the need of an external signal that would otherwise centralise the system. This removes single points of failure, which improves the reliability of the entire system.

Design and testing of the proposed control schemes for microgrids with the use of a real-time simulator, given that the secondary and tertiary control layer has realised by a network of Raspberry Pis, allows a better consideration of the time response requirements of such system under more realistic conditions than offline simulation. Time response requirements, combined with the distributed control requirements directed the design of the control system, from the control loops to the artificial intelligence models employed to realise the optimal power management under electricity price and renewable generation uncertainties. In chapter 5 is shown that real-time simulators allowed the verification of the capabilities of the distributed control system to simultaneously achieve reliable and optimal operation under real-time conditions.

6.2 Future Work

Future work will be focused in further validation of the proposed control scheme, and implementing conditions not covered in this thesis. Further verification of the proposed control scheme can be done by increasing the complexity, type of generation and consumption of the tested cluster model. Additionally, the hardware, software, datasets and parameters involved in the test cases can be modified to reflect conditions that are more realistic.

It is become evident during the development of this thesis that many of the objectives in a distributed control environment could be greatly improved with the addition of load forecasting methods applying the same methodologies used in this thesis for equivalent problems in electricity price and wind speed forecast. It is therefore part of the future work to establish a load forecasting model for the smart power management system to realise other objectives such as peer-to-peer electricity trading, such that the control system can automatically purchase energy for its assigned microgrid or microgrid cluster, with the least amount of manual input from the end users.

Secondly, to distribute the process of adding new stakeholders into the system, it will become necessary to implement the ability of the system to discover the circuit topology in a distributed environment. Once the distributed control system is capable of doing this, it will also become more resilient to faults in the physical network, as it could self-adjust its control layers depending on the available power flow channels.

Finally, further control methods are required to remove assumptions made on the simulated microgrid models, in particular the nature of loads, in other words, further development of the control system can be channelled to address the system stability in the presence of non-linear loads or unbalanced loads. Additionally to this, further work can be done to develop the necessary primary controllers to accommodate other types of electrical sources, for example, DC microgrids to the cluster, with the focus on generalising the control schemes to any type of electrical circuits.

Bibliography

[1] Fabio Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[2] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart grid- the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2012.

[3] "Chengshan Wang, Jinyue Yan, Chris Marnay, Ned Djilali, Erik Dahlquist, Jianzhong Wu, and Hongjie Jia". Distributed energy and microgrids (dem). *Applied Energy*, (210):685–689, 2018.

[4] M. E. Cruz Victorio, B. Kazemtabrizi, and M. Shahbazi. Real-time cost optimisation for power management in microgrids using multi-agent control. In *2019 9th International Conference on Power and Energy Systems (ICPES)*, pages 1–6, 2019.

[5] M. E. Cruz Victorio, B. Kazemtabrizi, and M. Shahbazi. Distributed real-time power management in microgrids using multi-agent control with provisions for fault-tolerance. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 108–113, 2020.

[6] M. E. Cruz Victorio, B. Kazemtabrizi, and M. Shahbazi. Real-time resilient

microgrid power management based on multi-agent systems with price forecast (accepted). *IET Smart Grid*, 2022.

[7] M. E. Cruz Victorio, B. Kazemtabrizi, and M. Shahbazi. Price forecast methodologies comparison for microgrid control with multi-agent systems. In *2021 IEEE Madrid PowerTech*, 2021.

[8] M. E. Cruz Victorio, B. Kazemtabrizi, and M. Shahbazi. Statistical evaluation of wind speed forecast models for microgrid distributed control (accepted). *IET Smart Grid*, 2022.

[9] Asrul Bin-Ibrahim, Ahmad. *Operational Planning and Optimisation in Active Distribution Systems for Flexible and Resilient Power*. PhD thesis, Durham University, 2018.

[10] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou. Trends in microgrid control. *IEEE transactions on smart grid*, (4):1905–1919, 2014.

[11] F. Martin-Martínez, A. Sánchez-Miralles, and M. Rivier. A literature review of microgrids: A functional layer based classification. *Renewable and Sustainable Energy Reviews*, 62:1133–1153, 2016.

[12] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna, and M. Castilla. Hierarchical control of droop-controlled dc and ac microgrids– a general approach towards standardization. *IEEE Transactions on Industrial Electronics*, 58(1):158–172, 2011.

[13] Magdi S Mahmoud, S Azher Hussain, and Mohammad Ali Abido. Modelling and control of microgrid: An overview. *Journal of the Franklin Institute*, (351):2822–2859, 2014.

[14] Wassim M. Haddad and VijaySekhar Chellaboina. *Chapter Three. Stability Theory for Nonlinear Dynamical Systems*, pages 135 – 206. Princeton University Press, Princeton, 2008.

[15] Jan Machowski, Janusz W. Bialek, and Jim Bumby. *Power System Dynamics : Stability and Control.* John Wiley & Sons, Incorporated, 2008.

[16] Dao H. Vu, Kashem M. Muttaqi, Ashish P. Agalgaonkar, and Abdesselam Bouzerdoum. Short-term forecasting of electricity spot prices containing random spikes using a time-varying autoregressive model combined with kernel regression. *IEEE Transactions on Industrial Informatics*, 15(9):5378–5388, 2019.

[17] H. Golmohamadi and R. Keypour. Stochastic optimization for retailers with distributed wind generation considering demand response. *Journal of Modern Power Systems and Clean Energy*, 6(4):733–748, 2018.

[18] F. D. Kanellos. Real-time control based on multi-agent systems for the operation of large ports as prosumer microgrids. *IEEE Access*, 5:9439–9452, 2017.

[19] Christopher M Colson and M Hashem Nehrir. Comprehensive real-time microgrid power management and control with distributed agents. *IEEE transactions on smart grid*, 4(1):617–627, 2013.

[20] "Lexuan Meng, Eleonora Riva Sanseverino, Adriana Luna, Tomislav Dragicevic, Juan C. Vasquez, and Josep M. Guerrero". Microgrid supervisory controllers and energy management systems: A literature review. *Renewable and Sustainable Energy Reviews*, (60):1263–1273, 2016.

[21] "Junjie Hu, Arshad Saleem, Shi You, Lars Nordström, Morten Lind, and Jacob Østergaard". A multi-agent system for distribution grid congestion management with electric vehicles. *Engineering applications of artificial intelligence*, (38):45–58, 2015.

[22] Y. Han, K. Zhang, H. Li, E. A. A. Coelho, and J. M. Guerrero. Mas-based distributed coordinated control and optimization in microgrid and microgrid clusters: A comprehensive overview. *IEEE Transactions on Power Electronics*, 33(8):6488–6508, 2018.

[23] José Luis Rueda-Torres and Francisco González-Longatt. *Probabilistic Indicators for the Assessment of Reliability and Security of Future Power Systems*, pages 41–61. 2018.

[24] Christos G. Kaloudas, Luis F. Ochoa, Ben Marshall, Shanti Majithia, and Ian Fletcher. Assessing the future trends of reactive power demand of distribution networks. *IEEE Transactions on Power Systems*, 32(6):4278–4288, 2017.

[25] Hongbin Sun, Qinglai Guo, Junjian Qi, Venkataramana Ajjarapu, Richard Bravo, Joe Chow, Zhengshuo Li, Rohit Moghe, Ehsan Nasr-Azadani, Ujjwol Tamrakar, Glauco N. Taranto, Reinaldo Tonkoski, Gustavo Valverde, Qiuwei Wu, and Guangya Yang. Review of challenges and research opportunities for voltage control in smart grids. *IEEE Transactions on Power Systems*, 34(4):2790–2801, 2019.

[26] Stephen F. Bush. *Introduction to Power Systems Before Smart Grid*, pages 1–53. "Wiley-IEEE press", 2013.

[27] Vehbi C. Gungor, Bin Lu, and Gerhard P. Hancke. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, 57(10):3557–3564, 2010.

[28] Mehrdad Yazdanian and Ali Mehrizi-Sani. Distributed control techniques in microgrids. *IEEE Transactions on Smart Grid*, 5(6):2901–2909, 2014.

[29] Enrique Espina, Jacqueline Llanos, Claudio Burgos-Mellado, Roberto Cárdenas-Dobson, Manuel Martínez-Gómez, and Doris Sáez. Distributed control strategies for microgrids: An overview. *IEEE Access*, 8:193412–193448, 2020.

[30] Amir-Hamed Mohsenian-Rad, Vincent WS Wong, Juri Jatskevich, Robert Schober, and Alberto Leon-Garcia. Autonomous demand side management based on game theoretic energy consumption scheduling for the future smart grid. *IEEE transactions on smart grid*, 1(3):320–331, 2010.

[31] Vitor N. Coelho, Miri Weiss Cohen, Igor M. Coelho, Nian Liu, and Frederico Gadelha Guimaraes. Multi-agent systems applied for energy systems integration: State-of-the-art applications and trends in microgrids. *Applied Energy*, 187:820 – 832, 2017.

[32] "Amjad Anvari-Moghaddam, Ashkan Rahimi-Kian, Maryam S. Mirian, and Josep M. Guerrero". A multi-agent based energy management solution for integrated buildings and microgrid system. *Applied Energy*, 203:41 – 56, 2017.

[33] C. Dou and B. Liu. Multi-agent based hierarchical hybrid control for smart microgrid. *IEEE Transactions on Smart Grid*, 4(2):771–778, 2013.

[34] Su Ma, Hong Zhang, and Xiaomin Xing. Scalability for smart infrastructure system in smart grid: A survey. *Wireless Personal Communications*, 99:161–184, 2018.

[35] Chunxia Dou, Zhanqiang Zhang, Dong Yue, and Yuhang Zheng. Mas-based hierarchical distributed coordinate control strategy of virtual power source voltage in low-voltage microgrid. *IEEE Access*, 5:11381–11390, 2017.

[36] Chunxia Dou, Dong Yue, Zhanqiang Zhang, and Kai Ma. Mas-based distributed cooperative control for dc microgrid through switching topology communication network with time-varying delays. *IEEE Systems Journal*, 13(1):615–624, 2019.

[37] Mehmet H. Cintuglu, Tarek Youssef, and Osama 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*, 9(3):1759–1768, 2018.

[38] D. I. Makrygiorgou and A. T. Alexandridis. Distributed stabilizing modular control for stand-alone microgrids. *Applied Energy*, (210):925–935, 2018.

[39] Zhaohong Bie, Yanling Lin, Gengfeng Li, and Furong Li. Battling the extreme: A study on the power system resilience. *Proceedings of the IEEE*, 105(7):1253–1266, 2017.

[40] S. K. Sahoo, A. K. Sinha, and N. K. Kishore. Control techniques in ac, dc, and hybrid ac–dc microgrid: A review. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 6(2):738–759, 2018.

[41] Zhao xia Xiao, Josep M. Guerrero, Jia Shuang, Dezso Sera, Erik Schaltz, and Juan C. Vásquez. Flat tie-line power scheduling control of grid-connected hybrid microgrids. *Applied Energy*, 210:786–799, 2018.

[42] T. Dragičević, J. M. Guerrero, J. C. Vasquez, and D. Škrlec. Supervisory control of an adaptive-droop regulated dc microgrid. *IEEE transactions on power electronics*, 29(2):695–706, 2014.

[43] Stefano Riverso, Michele Tucci, Juan C Vasquez, Josep M Guerrero, and Giancarlo Ferrari-Trecate. Stabilizing plug-and-play regulators and secondary coordinated control for ac islanded microgrids with bus-connected topology. *Applied Energy*, (210):914–924, 2018.

[44] Mingyu Sun, Gaoyuan Liu, Marjan Popov, Vladimir Terzija, and Sadegh Azizi. Underfrequency load shedding using locally estimated rocof of the center of inertia. *IEEE Transactions on Power Systems*, 36(5):4212–4222, 2021.

[45] M. Chidambaram and Vivek Sathe. *Cascade Controllers Tuning by Relay Autotune Method*, page 30–42. Cambridge University Press, 2014.

[46] Dmitry Baimel, Juri Belikov, Josep M. Guerrero, and Yoash Levron. Dynamic modeling of networks, microgrids, and renewable sources in the dq0 reference frame: A survey. *IEEE Access*, 5:21323–21335, 2017.

[47] Hirofumi Akagi, Edson Hirokazu Watanabe, and Mauricio Aredes. *The Instantaneous Power Theory*, pages 41–107. 2007.

[48] "Mehdi Tavakoli, Fatemeh Shokridehaki, Mudathir Funsho Akorede, Mousa Marzband, Ionel Vechiu, and Edris Pouresmaeil". Cvar-based energy management scheme for optimal resilience and operational cost in commercial building microgrids. *Electrical Power and energy systems*, 100:1–9, 2018.

[49] Sundari Ramabhotla. *Energy management in microgrids*. PhD thesis, Texas Tech University, 2015.

[50] G. Liu, Y. Xu, and K. Tomsovic. Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization. *IEEE Transactions on Smart Grid*, 7(1):227–237, 2016.

[51] B. Gundogdu, D. Gladwin, and D. Stone. Battery energy management strategies for uk firm frequency response services and energy arbitrage. *The Journal of Engineering*, 2019(17):4152–4157, 2019.

[52] Frank Allgower and Alex Zheng. *Nonlinear model predictive control*, volume 26. Birkhauser, 2012.

[53] Liuping Wang, Shan Chai, Dae Yoo, Lu Gan, and Ki Ng. *PID and predictive control of electrical drives and power converters using MATLAB/Simulink*. John Wiley & Sons, 2015.

[54] Jose Rodriguez and Patricio Cortes. *Predictive control of power converters and electrical drives*, volume 40. John Wiley & Sons, 2012.

[55] Zheng Zhao. *Optimal Energy Management for Microgrids*. PhD thesis, Clemson University, 2012.

[56] Chen Yuan. *Resilient distribution systems with community microgrids*. PhD thesis, Ohio State University, 2016.

[57] "Xiaolong Jin, Jianzhong Wu, Yunfei Mu, Mingshen Wang, Xiandong Xu, and Hongjie Jia". Hierarchical microgrid energy management in an office building. *Applied Energy*, (208):480–494, 2017.

[58] Pan Wu, Wentao Huang, Nengling Tai, and Shuo Liang. A novel design of architecture and control for multiple microgrids with hybrid ac/dc connection. *Applied Energy*, 210:1002–1016, 2018.

[59] 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*, 31(2):1600–1617, 2016.

[60] O. Samuel, N. Javaid, A. Khalid, W. Z. Khan, M. Y. Aalsalem, M. K. Afzal, and B. S. Kim. Towards real-time energy management of multi-microgrid using a deep convolution neural network and cooperative game approach. *IEEE Access*, 8:161377–161395, 2020.

[61] L. Ding, Q. Han, and X. Zhang. Distributed secondary control for active power sharing and frequency regulation in islanded microgrids using an event-triggered communication mechanism. *IEEE Transactions on Industrial Informatics*, 15(7):3910–3922, 2019.

[62] John Kabouris and F. D. Kanellos. Impacts of large-scale wind penetration on designing and operation of electric power systems. *IEEE Transactions on Sustainable Energy*, 1(2):107–114, 2010.

[63] R. Billinton and R.N. Allan. Basic power system reliability concepts. *Reliability Engineering & System Safety*, 27(3):365–384, 1990.

[64] M. Mohiuddin, W. Saab, S. Bliudze, and J. Le Boudec. Axo: Detection and recovery for delay and crash faults in real-time control systems. *IEEE Transactions on Industrial Informatics*, 14(7):3065–3075, 2018.

[65] Jianzhou Wang, Shiqi Wang, and Wendong Yang. A novel non-linear combination system for short-term wind speed forecast. *Renewable Energy*, 143:1172–1192, 2019.

[66] Y. Liu, Z. Qu, H. Xin, and D. Gan. Distributed real-time optimal power flow control in smart grid. *IEEE Transactions on Power Systems*, 32(5):3403–3414, 2017.

[67] Giulia De Zotti, S Ali Pourmousavi, Henrik Madsen, and Niels Kjølstad Poulsen. Ancillary services 4.0: A top-to-bottom control-based approach. *IEEE Access*, 6:11694–11706, 2018.

[68] Qobad Shafiee, Vahidreza Nasirian, Juan C Vasquez, Josep M Guerrero, and Ali Davoudi. A multi-functional fully distributed control framework for ac microgrids. *IEEE transactions on smart grid*, 9(4):3247–3258, 2018.

[69] M. Kabalan, P. Singh, and D. Niebur. A design and optimization tool for inverter-based microgrids using large-signal nonlinear analysis. *IEEE Transactions on Smart Grid*, 10(4):4566–4576, 2019.

[70] M. Mehrasa, E. Pouresmaeil, S. Zabihi, and J. P. S. Catalão. Dynamic model, control and stability analysis of mmc in hvdc transmission systems. *IEEE Transactions on Power Delivery*, 32(3):1471–1482, 2017.

[71] Dragoš Cvetković, Peter Rowlinson, and Slobodan Simić. *Graph operations and modifications*, pages 24–51. London Mathematical Society Student Texts. Cambridge University Press, 2009.

[72] Dragoš Cvetković, Peter Rowlinson, and Slobodan Simić. *Laplacians*, page 184–227. London Mathematical Society Student Texts. Cambridge University Press, 2009.

[73] "Allal M. Bouzid, Josep M. Guerrero, Ahmed Cheriti, Mohamed Bouhamida, Pierre Sicard, and Mustapha Benghanem". A survey on control of electric

power distributed generation systems for microgrid applications. *Renewable and Sustainable Energy Reviews*, (44):751–766, 2015.

[74] V. Bui, A. Hussain, and H. Kim. A multiagent-based hierarchical energy management strategy for multi-microgrids considering adjustable power and demand response. *IEEE Transactions on Smart Grid*, 9(2):1323–1333, 2018.

[75] L. Wu and M. Shahidehpour. A hybrid model for day-ahead price forecasting. *IEEE Transactions on Power Systems*, 25(3):1519–1530, 2010.

[76] Yixian Liu, Matthew C. Roberts, and Ramteen Sioshansi. A vector autoregression weather model for electricity supply and demand modeling. *Journal of Modern Power Systems and Clean Energy*, 6(4):763–776, 2018.

[77] Adil Ahmed and Muhammad Khalid. An intelligent framework for short-term multi-step wind speed forecasting based on functional networks. *Applied Energy*, 225:902–911, 2018.

[78] Fulin Fan, Keith Bell, and David Infield. Probabilistic real-time thermal rating forecasting for overhead lines by conditionally heteroscedastic auto-regressive models. *IEEE Transactions on Power Delivery*, 32(4):1881–1890, 2017.

[79] John Harlim. *Data-Driven Computational Methods: Parameter and Operator Estimations*. Cambridge University Press, 2018.

[80] I. Kim. Markov chain monte carlo and acceptance–rejection algorithms for synthesising short-term variations in the generation output of the photovoltaic system. *IET Renewable Power Generation*, 11(6):878–888, 2017.

[81] X. Yang, Y. Yang, Y. Liu, and Z. Deng. A reliability assessment approach for electric power systems considering wind power uncertainty. *IEEE Access*, 8:12467–12478, 2020.

[82] C. Wan, Z. Xu, Y. Wang, Z. Y. Dong, and K. P. Wong. A hybrid approach for probabilistic forecasting of electricity price. *IEEE Transactions on Smart Grid*, 5(1):463–470, 2014.

[83] A. Mohsenian-Rad and A. Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *IEEE Transactions on Smart Grid*, 1(2):120–133, 2010.

[84] Y. Du and F. Li. Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning. *IEEE Transactions on Smart Grid*, 11(2):1066–1076, 2020.

[85] Musaed Alhussein, Syed Irtaza Haider, and Khursheed Aurangzeb. Microgrid-level energy management approach based on short-term forecasting of wind speed and solar irradiance. *Energies*, 12(8), 2019.

[86] Zhongju Wang, Jing Zhang, Yu Zhang, Chao Huang, and Long Wang. Short-term wind speed forecasting based on information of neighboring wind farms. *IEEE Access*, 8:16760–16770, 2020.

[87] H. Mosbah and M. El-hawary. Hourly electricity price forecasting for the next month using multilayer neural network. *Canadian Journal of Electrical and Computer Engineering*, 39(4):283–291, 2016.

[88] Jean-François Toubeau, Thomas Morstyn, Jérémie Bottieau, Kedi Zheng, Dimitra Apostolopoulou, Zacharie De Grève, Yi Wang, and François Vallée. Capturing spatio-temporal dependencies in the probabilistic forecasting of distribution locational marginal prices. *IEEE Transactions on Smart Grid*, 12(3):2663–2674, 2021.

[89] Bartosz Uniejewski, Rafał Weron, and Florian Ziel. Variance stabilizing transformations for electricity spot price forecasting. *IEEE Transactions on Power Systems*, 33(2):2219–2229, 2018.

[90] G. Zhang, H. Zhou, C. Wang, H. Xue, J. Wang, and H. Wan. Forecasting time series albedo using narnet based on eemd decomposition. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3544–3557, 2020.

[91] A. Botterud, Z. Zhou, J. Wang, R. J. Bessa, H. Keko, J. Sumaili, and V. Miranda. Wind power trading under uncertainty in lmp markets. *IEEE Transactions on Power Systems*, 27(2):894–903, 2012.

[92] A. Y. Alanis. Electricity prices forecasting using artificial neural networks. *IEEE Latin America Transactions*, 16(1):105–111, 2018.

[93] F. Llorente, L. Martino, and D. Delgado. Parallel metropolis–hastings coupler. *IEEE Signal Processing Letters*, 26(6):953–957, 2019.

[94] M. Jawad, S. M. Ali, B. Khan, C. A. Mehmood, U. Farid, Z. Ullah, S. Usman, A. Fayyaz, J. Jadoon, N. Tareen, A. Basit, M. A. Rustam, and I. Sami. Genetic algorithm-based non-linear auto-regressive with exogenous inputs neural network short-term and medium-term uncertainty modelling and prediction for electrical load and wind speed. *The Journal of Engineering*, 2018(8):721–729, 2018.

[95] Kevin L. Priddy and Paul E. Keller. chapter 1, pages 1–12. SPIE, 2005.

[96] A. C. Luna, L. Meng, N. L. Diaz, M. Graells, J. C. Vasquez, and J. M. Guerrero. Online energy management systems for microgrids: Experimental validation and assessment framework. *IEEE transactions on power electronics*, 33(3):2201–2215, 2018.

[97] Yurong Wang, Dongchuan Wang, and Yi Tang. Clustered hybrid wind power prediction model based on arma, pso-svm, and clustering methods. *IEEE Access*, 8:17071–17079, 2020.

[98] Siddhartha Bhattacharyya, Vaclav Snasel, Aboul Ella Hassanien, Satadal Saha, and B. K. Tripathy. *Convolutional Neural Networks: A Bottom-Up Approach*, pages 21 – 50. De Gruyter, Berlin, 2020.

[99] Ling Xiang, Zeqi Deng, and Aijun Hu. Forecasting short-term wind speed based on iewt-lssvm model optimized by bird swarm algorithm. *IEEE Access*, 7:59333–59345, 2019.

[100] Tianqiao Zhao and Zhengtao Ding. Distributed agent consensus-based optimal resource management for microgrids. *IEEE Transactions on Sustainable Energy*, 9(1):443–452, 2018.

[101] Yongjing He, Wei Wang, and Xuezhi Wu. Multi-agent based fully distributed economic dispatch in microgrid using exact diffusion strategy. *IEEE Access*, 8:7020–7031, 2020.

[102] Mircea Eremia, Chen-Ching Liu, and Abdel-Aty Edris. *Multiagent Systems*, pages 903–930. 2016.

[103] Stephen Knox, Philipp Meier, Jim Yoon, and Julien J Harou. A python framework for multi-agent simulation of networked resource systems. *Environmental Modelling & Software*, (103):16–28, 2018.

[104] X. Yi, L. Yao, T. Yang, J. George, and K Johansson. Distributed optimization for second-order multi-agent systems with dynamic event-triggered communication. In *2018 IEEE Conference on Decision and Control*, 2018.

[105] Y. Zhang and Y Hong. Distributed optimization design for second-order multi-agent systems. In *Proceedings of the 33rd Chinese Control Conference*, 2014.

[106] H. Kim, H. Shim, and J. Seo. Output consensus of heterogeneous uncertain linear multi-agent systems. *IEEE Transactions on automatic control*, 56(1):200–206, 2011.

[107] Y. Su and J. Huang. Cooperative output regulation of linear multi-agent systems. *IEEE Transactions on automatic control*, 57(4):1062–1066, 2012.

[108] W. Ni and D. Cheng. Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, pages 209–217, 2010.

[109] Shihua Li, Haibo Du, and Xiangze Lin. Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics. *Automatica*, pages 1706–1712, 2011.

[110] Q. Li, C. Peng, M. Chen, F. Chen, W. Kang, J. M. Guerrero, and D. Abbott. Networked and distributed control method with optimal power dispatch for islanded microgrids. *IEEE Transactions on Industrial Electronics*, 64(1):493–504, 2017.

[111] A. Bidram, A. Davoudi, F. L. Lewis, and Z. Qu. Secondary control of microgrids based on distributed cooperative control of multi-agent systems. *IET Generation, Transmission Distribution*, 7(8):822–831, 2013.

[112] Mònica Aragüés-Peñalba, Tung Lam Nguyen, Raphael Caire, Andreas Sumper, Samuel Galceran-Arellano, and Quoc-Tuan Tran. General form of consensus optimization for distributed opf in hvac-vsc-hvdc systems. *International Journal of Electrical Power & Energy Systems*, page 106049, 2020.

[113] P. Trichakis. *Multi Agent Systems for the Active Management of Electrical Distribution Networks*. PhD thesis, Durham University, 2009. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/57/.

[114] D. Dimarogonas, E. Frazzoli, and K. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on automatic control*, 57(5):1291–1296, 2012.

[115] Yuan Fan, Gang Feng, Yong Wang, and Cheng Song. Distributed event-triggered control of multi-agent systems with combinational measurements. *Automatica*, pages 671–675, 2013.

[116] William J. Terrell. *Chapter Nine. Cascade Systems*, pages 189 – 211. Princeton University Press, Princeton, 2009.

[117] William J. Terrell. *Chapter Eight. Stability Theory*, pages 161 – 188. Princeton University Press, Princeton, 2009.

[118] Wassim M. Haddad and VijaySekhar Chellaboina. *Advanced Stability Theory*, pages 135 – 206. Princeton University Press, Princeton, 2008.

[119] E. K. Çetinkaya, M. J. F. Alenazi, J. P. Rohrer, and J. P. G. Sterbenz. Topology connectivity analysis of internet infrastructure using graph spectra. In *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pages 752–758, 2012.

[120] M. Lydia, S. Suresh Kumar, A. Immanuel Selvakumar, and G. Edwin Prem Kumar. Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Conversion and Management*, 112:115–124, 2016.

[121] Nathaniel S. Pearre and Lukas G. Swan. Statistical approach for improved wind speed forecasting for wind power production. *Sustainable Energy Technologies and Assessments*, 27:180–191, 2018.

[122] Gyu Gwang Kim, Jin Ho Choi, So Young Park, Byeong Gwan Bhang, Woo Jun Nam, Hae Lim Cha, NeungSoo Park, and Hyung-Keun Ahn. Prediction model for pv performance with correlation analysis of environmental variables. *IEEE Journal of Photovoltaics*, 9(3):832–841, 2019.

[123] A. Aranizadeh, A. Zaboli, O. Asgari Gashteroodkhani, and B. Vahidi. Wind turbine and ultra-capacitor harvested energy increasing in microgrid using wind speed forecasting. *Engineering Science and Technology, an International Journal*, 22(5):1161–1167, 2019.

[124] Lifang Zhang, Yao Dong, and Jianzhou Wang. Wind speed forecasting using a two-stage forecasting system with an error correcting and nonlinear ensemble strategy. *IEEE Access*, 7:176000–176023, 2019.

[125] Iram Akhtar, Sheeraz Kirmani, Mohmmad Ahmad, and Sultan Ahmad. Average monthly wind power forecasting using fuzzy approach. *IEEE Access*, 9:30426–30440, 2021.

[126] Yagang Zhang, Shuang Gao, Jingyi Han, and Minghui Ban. Wind speed prediction research considering wind speed ramp and residual distribution. *IEEE Access*, 7:131873–131887, 2019.

[127] Wenjuan Li, Yungang Liu, Huijun Liang, and Yanjun Shen. A new distributed energy management strategy for smart grid with stochastic wind power. *IEEE Transactions on Industrial Electronics*, 68(2):1311–1321, 2021.

[128] Chao Zhang, Ming Ding, Weisheng Wang, Rui Bi, Leying Miao, Haibiao Yu, and Lian Liu. An improved elm model based on ceemd-lzc and manifold learning for short-term wind power prediction. *IEEE Access*, 7:121472–121481, 2019.

[129] David Harvey, Stephen Leybourne, and Paul Newbold. Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291, 1997.

[130] Mirza Naveed Shahzad, Saiqa Kanwal, and Abid Hussanan. A new hybrid arar and neural network model for multi-step ahead wind speed forecasting in three regions of pakistan. *IEEE Access*, 8:199382–199392, 2020.

[131] Katarzyna Hubicka, Grzegorz Marcjasz, and Rafał Weron. A note on averaging day-ahead electricity price forecasts across calibration windows. *IEEE Transactions on Sustainable Energy*, 10(1):321–323, 2019.

[132] Edmore Ranganai and Caston Sigauke. Capturing long-range dependence and harmonic phenomena in 24-hour solar irradiance forecasting: A quantile regression robustification via forecasts combination approach. *IEEE Access*, 8:172204–172218, 2020.

[133] Jianzhou Wang, Qiwei Li, and Bo Zeng. Multi-layer cooperative combined forecasting system for short-term wind speed forecasting. *Sustainable Energy Technologies and Assessments*, 43:100946, 2021.

[134] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[135] Peter Zörnig. *Probability Theory and Statistical Applications : A Profound Treatise for Self-Study*. De Gruyter Textbook. Princeton University Press, 2016.

[136] Auchencorth moss atmospheric observatory (au): Annual half-hourly meteorology since 1995, near edinburgh, uk. `https://data.ceda.ac.uk/badc/deposited2017/auchencorth_moss_met/data`. Accessed: 28-10-2021.

[137] G. Li, C. Liu, C. Mattson, and J. Lawarree. Day-ahead electricity price forecasting in a grid environment. *IEEE Transactions on Power Systems*, 22(1):266–274, 2007.

[138] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan. Electricity price forecasting with extreme learning machine and bootstrapping. *IEEE Transactions on Power Systems*, 27(4):2055–2062, 2012.

[139] Xiangyu Wu, Yin Xu, Jinghan He, Xiaojun Wang, Juan C. Vasquez, and Josep M. Guerrero. Pinning-based hierarchical and distributed cooperative control for ac microgrid clusters. *IEEE Transactions on Power Electronics*, 35(9):9865–9885, 2020.

[140] R. Salcedo, E. Corbett, C. Smith, E. Limpaecher, R. Rekha, J. Nowocin, G. Lauss, E. Fonkwe, M. Almeida, P. Gartner, S. Manson, B. Nayak, I. Celanovic, C. Dufour, M. O. Faruque, K. Schoder, R. Brandl, P. Kotsampopoulos, T. Ham Ha, A. Davoudi, A. Dehkordi, and K. Strunz. Banshee distribution network benchmark and prototyping platform for hardware-in-the-loop in-

tegration of microgrid and device controllers. *The Journal of Engineering*, 2019.

Appendix **A**

Quasi-Newton Algorithm

This algorithm is based on gradient descent methods to minimise a cost function f_{QN} with input x_{QN}. At the start of the QN algorithm, a starting point x_{QN_0} and a H_0 as a symmetric positive definite matrix are chosen to represent the estimated Hessian Matrix, which will be updated along the estimated solution. With these, an optimal direction for gradient descent g_k is calculated:

$$g_k = -H_k^-1 \cdot \nabla f_{QN}(x_{QN_k}) \tag{A.1}$$

then, an α_k is selected such that:

$$f_{QN}(x_{QN_k} + a_k g_k) < f_{QN}(x_{QN_k}) \tag{A.2}$$

with the resulting a_k and g_k, the next iteration is calculated:

$$x_{QN_{k+1}} = x_{QN_k} + a_k g_k \tag{A.3}$$

The difference from the Newton method is that the Hessian matrix update in its algorithm is estimated from its previous values instead of calculated analytically:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k} \tag{A.4}$$

where:

$$s_k = x_{QN_{k+1}} - x_{QN_k} \tag{A.5}$$

$$q_k = \nabla f_{QN}(x_{QN_{k+1}}) - \nabla f_{QN}(x_{QN_k}) \tag{A.6}$$

The method finalises when the following is true for a given tolerance ϵ, otherwise, the algorithm repeats with the next iteration $k+1$:

$$\left|\nabla f_{QN}(x_{QN_k})\right| < \epsilon \tag{A.7}$$

Appendix **B**

Inner Loop Function Stability

This function models the dynamics of the LCL filter for voltage control. The requirements are close loop stability with respect to the DER side voltage source and open loop stability for the Grid side voltage source. The inner loop dynamics are described by the following system of equations:

$$
\begin{gathered}
v_U(t) = L_1 \frac{di_1}{dt} + v_C \\
v_C(t) = \frac{1}{C}\int (i_1 - i_2)dt \\
v_C(t) = L_2 \frac{di_2}{dt} + Ri_2 + v_S
\end{gathered}
$$

Applying the Laplace transform, assuming initial conditions at zero we have:

$$
\begin{gathered}
\mathscr{L}[v_U(t)] = \mathscr{L}\left[L_1 \frac{di_1}{dt} + V_C\right] \\
\mathscr{L}[v_C(t)] = \mathscr{L}\left[\frac{1}{C}\int (i_1 - i_2)dt\right] \\
\mathscr{L}[v_C(t)] = \mathscr{L}\left[L_2 \frac{dI_2}{dt} + Ri_2 + v_S\right]
\end{gathered}
$$

$$
\begin{gathered}
V_U(S) = L_1 S I_1(S) + V_C(S) \\
V_C(S) = \frac{1}{CS}(I_1 - I_2) \\
V_C(S) = L_2 S I_2(S) + RI_2(S) + V_S(S)
\end{gathered}
$$

Using the superposition principle, we analyse the response of I_2 with respect to V_U and V_S independently. For $V_S = 0$ we have:

$$V_U = L_1SI_1 + \frac{1}{CS}(I_1 - I_2)$$
$$\frac{1}{CS}(I_1 - I_2) = L_2SI_2 + RI_2$$
$$I_1 = L_2CS^2I_2 + RCSI_2 + I_2$$
$$V_U = L_1S(L_2CS^2I_2 + RCSI_2 + I_2) + L_2SI_2 + RI_2$$
$$V_U = L_1L_2CS^3I_2 + L_1RCS^2I_2 + L_1SI_2 + L_2SI_2 + RI_2$$
$$V_U = (L_1L_2CS^3 + L_1RCS^2 + L_1S + L_2S + R)I_2$$
$$T_1 = \frac{I_2}{V_U} = \frac{1}{L_1L_2CS^3 + L_1RCS^2 + (L_1 + L_2)S + R}$$

Similarly, for $V_U = 0$ we have:

$$0 = L_1SI_1 + \frac{1}{CS}(I_1 - I_2)$$
$$\frac{1}{CS}(I_1 - I_2) = L_2SI_2 + RI_2 + V_S$$
$$I_1 = L_2CS^2I_2 + RCSI_2 + CSV_S + I_2$$
$$0 = L_1S(L_2CS^2I_2 + RCSI_2 + CSV_S + I_2) + L_2SI_2 + RI_2 + V_S$$
$$0 = L_1L_2CS^3I_2 + L_1RCS^2I_2 + L_1CS^2V_S + L_1SI_2 + L_2SI_2 + RI_2 + V_S$$
$$-V_S(L_1CS^2 + 1) = (L_1L_2CS^3 + L_1RCS^2 + (L_1 + L_2)S + R)I_2$$
$$T_2 = \frac{I_2}{V_s} = -\frac{L_1CS^2 + 1}{L_1L_2CS^3 + L_1RCS^2 + (L_1 + L_2)S + R}$$

It can be concluded that there are no poles in zero for T_1 and T_2 and all poles have negative real part. This is because there is no change in sign in the denominator and all the parameters of the circuit are positive, therefore they are open loop stable. For the PI control, the parameters of the gains and the filter can be selected to be close loop stable for T_1.

Appendix C

Lyapunov Stability of Outer Loop

The following formulation describes step by step the process to derive the Lyapunov's candidate function, used in the Lyapunov's direct method of stability for the power flow between two buses of the microgrid.

Starting from the power equation between two buses, V_S and V_R,in a short line with impedance $Z = R + i\omega L$ we have:

$$P + iQ = V_S(I)^*$$
$$P + iQ = V_S\frac{(V_S - V_R\angle\delta)^*}{Z}$$
$$P + iQ = V_S\frac{(V_S - V_R\angle\delta)^*}{Z}$$
$$P + iQ = V_S\frac{((V_S - V_R\angle\delta)(R - i\omega L))^*}{R^2 + \omega^2L^2}$$
$$P + iQ = V_S\frac{((V_S - V_Rcos\delta - iV_Rsin\delta)(R - i\omega L))^*}{R^2 + \omega^2L^2}$$
$$P + iQ = V_S\frac{(RV_S - RV_Rcos\delta - iRV_Rsin\delta - i\omega LV_S + i\omega LV_Rcos\delta - \omega LV_Rsin\delta)^*}{R^2 + \omega^2L^2}$$
$$P + iQ = V_S\frac{(RV_S - RV_Rcos\delta - iRV_Rsin\delta - i\omega LV_S + i\omega LV_Rcos\delta - \omega LV_Rsin\delta)^*}{R^2 + \omega^2L^2}$$
$$P + iQ = \frac{V_S}{R^2 + \omega^2L^2}(RV_S - RV_Rcos\delta - iRV_Rsin\delta - i\omega LV_S +$$
$$i\omega LV_Rcos\delta - \omega LV_Rsin\delta)^*$$

$$P + iQ = \frac{V_S}{R^2 + \omega^2 L^2}(RV_S - RV_R cos\delta + iRV_R sin\delta + i\omega LV_S -$$

$$i\omega LV_R cos\delta - \omega LV_R sin\delta)$$

$$P + iQ = \frac{V_S}{R^2 + \omega^2 L^2}[(RV_S - RV_R cos\delta - \omega LV_R sin\delta) +$$

$$i(RV_R sin\delta + \omega LV_S - \omega LV_R cos\delta)]$$

Separating the active power P and the reactive power Q we have:

$$P = \frac{V_S}{R^2 + \omega^2 L^2}(RV_S - RV_R cos\delta - \omega LV_R sin\delta)$$
$$Q = \frac{V_S}{R^2 + w^2 L^2}(RV_R sin\delta + \omega LV_S - \omega LV_R cos\delta)$$

Selecting V_S as the grid bus and the V_R as the DER or ESS bus, and reversing the sign of the power equations, such that power sent by the microgrid is positive and power received is negative we have the power equations:

$$P = \frac{V_S}{R^2 + \omega^2 L^2}(RV_R cos\delta + \omega LV_R sin\delta - RV_S)$$
$$Q = \frac{V_S}{R^2 + \omega^2 L^2}(\omega LV_R cos\delta - RV_R sin\delta - \omega LV_S)$$

Selecting as variables V_R and δ, and leaving the rest as constant we have the functions and constant:

$$P(V_R, \delta) = C_1(RV_R cos\delta + \omega LV_R sin\delta - RV_S)$$

$$Q(V_R, \delta) = C_1(\omega LV_R cos\delta - RV_R sin\delta - \omega LV_S)$$
$$C_1 = \frac{V_S}{R^2 + \omega^2 L^2}$$

With the definition of the state space variables X_1 and X_2 as follows:

$$V_R = X_1 + V_s$$

$$-\delta = X_2$$

Which are substituted in the power equations, to have:

$$P(X_1, X_2) = C_1(X_1(RcosX_2 - \omega LsinX_2) + V_s(RcosX_2 - \omega LsinX_2) - RV_s)$$

$$Q(X_1, X_2) = C_1(X_1(\omega LcosX_2 + RsinX_2) + V_s(\omega LcosX_2 + RsinX_2) - \omega LV_s)$$

Which is a time invariant system with solution $[P(0)\ Q(0)] = [0\ 0]$ with infinitely many solutions of the form:

$$X_1 = 0 \qquad X_2 = \pm 2\pi k \qquad \forall k \in \mathbb{N}_0$$

Applying integral control with the appropriate starting conditions in a negative feedback loop, we can define:

$$\dot{X}_1 = -C_1(X_1(RcosX_2 - \omega LsinX_2) + V_s(RcosX_2 - \omega LsinX_2) - RV_s)$$
$$\dot{X}_2 = -C_1(X_1(\omega LcosX_2 + RsinX_2) + V_s(\omega LcosX_2 + RsinX_2) - \omega LV_s)$$

To demonstrate Lyapunov stability through the direct approach, a candidate function V is selected such that it has the following properties:

$$\mathcal{V}(0) = 0$$
$$\mathcal{V}(X) \geq 0, X \neq 0, X \in D_X$$
$$\mathcal{V}(X) \rightarrow \infty, |X| \rightarrow \infty$$
$$\dot{\mathcal{V}}(0) = 0$$
$$\dot{\mathcal{V}}(X) \leq 0, X \neq 0, X \in D$$

where D_X is some domain for X where $X = 0 \in D_X$. The candidate function for this system is:

$$\mathcal{V}(X) = \frac{1}{2}X_1^2 + \frac{1}{2}X_2^2, X \in D_X$$
$$\dot{\mathcal{V}}(X) = X_1\dot{X}_1 + X_2\dot{X}_2, X \in D_X$$

D_X is any convex set where $\mathcal{V} \leq \alpha$ that contains $X = 0$, for any constant α.

Finally, substituting the values of $\dot{X}$ in $\dot{\mathcal{V}}$:

$$\dot{\mathcal{V}}(X) = -C_1(X_1(X_1(RcosX_2 - \omega LsinX_2) + V_s(RcosX_2 - \omega LsinX_2) - RV_s)$$
$$-X_2(X_1(\omega LcosX_2 + RsinX_2) + V_s(\omega LcosX_2 + RsinX_2) - \omega LV_s))$$

Which can always be found to have the desired properties for positive parameters in the domain D_X, provided that the $R/\omega L$ ratio is sufficiently large for the proposed state variables and candidate function. These conditions are true in a distribution circuit. The domain D_X is selected such that it delimits the origin around a value of $\mathcal{V}$, such that the domain is continuous for X. As the outer loop function is asymptotic stable and the cascade or inner loop function is stable, the entire primary control can always be tuned to be stable.

Appendix D

Current Stability between buses

The previous outer loop is designed to operate under the assumption that the voltages behave as phasors, which require matching frequency between buses and the reactive components behave as close as possible to the steady state. The duration of transient state in a short line model can be calculated as follows:

$$\begin{gathered}
\sqrt{2}V_s sin(\omega t) = Ri(t) + L\frac{di(t)}{dt} + \sqrt{2}V_R sin(\omega t + \delta) \\
\mathscr{L}[\sqrt{2}V_s sin(\omega t) - \sqrt{2}V_R sin(\omega t + \delta)] = \mathscr{L}\left[Ri(t) + L\frac{di(t)}{dt}\right] \\
\frac{\sqrt{2}V_s w}{S^2+\omega^2} - \frac{\sqrt{2}V_R(Ssin\delta + \omega cos\delta)}{S^2+\omega^2} = RI(S) + SLI(S) \\
\frac{\sqrt{2}V_s w}{S^2+\omega^2} - \frac{\sqrt{2}V_R(Ssin\delta + \omega cos\delta)}{S^2+\omega^2} = I(S)(R+SL) \\
\frac{\sqrt{2}V_s\omega - \sqrt{2}V_R(Ssin\delta + \omega cos\delta)}{(S^2+\omega^2)(R+SL)} = I(S) \\
I(S) = \frac{\mathcal{A}s}{S^2+\omega^2} + \frac{\mathcal{B}}{S^2+\omega^2} + \frac{\mathcal{C}}{\frac{R}{L}+S} \\
\frac{\sqrt{2}V_s\omega - \sqrt{2}V_R(Ssin\delta + \omega cos\delta)}{(S^2+\omega^2)(r+LS)} = \frac{(\mathcal{A}S+\mathcal{B})(R+LS) + \mathcal{C}(S^2+\omega^2)}{(S^2+\omega^2)(R+LS)} \\
\sqrt{2}V_s\omega - \sqrt{2}V_R(Ssin\delta + \omega cos\delta) = \mathcal{A}RS + RB + L\mathcal{A}S^2 + \mathcal{B}LS + \mathcal{C}S^2 + \mathcal{C}\omega^2
\end{gathered}$$

Re arranging the terms by order, the following can be solved in matrix form:

$$-\sqrt{2}V_R sin\delta S + \sqrt{2}V_S\omega - \sqrt{2}R\omega cos\delta = (L\mathcal{A}+\mathcal{C})S^2 + (\mathcal{A}R+\mathcal{B}L)S + \mathcal{C}\omega^2 + R\mathcal{B}$$

$$\begin{bmatrix} L & 0 & 1 \\ R & L & 0 \\ 0 & R & \omega^2 \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{B} \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} 0 \\ -\sqrt{2}V_R sin\delta \\ \sqrt{2}V_S\omega - \sqrt{2}V_R cos\delta \end{bmatrix} = \begin{bmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \end{bmatrix}$$

$$\begin{bmatrix} L & 0 & 1 \\ R & L & 0 \\ 0 & R & \omega^2 \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{B} \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \end{bmatrix}$$

Omitting the constants and solving the matrix we have:

$$\begin{bmatrix} 1 & 0 & \frac{1}{L} \\ 0 & L & -\frac{R}{L} \\ 0 & R & \omega^2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{L} \\ 0 & 1 & -\frac{R}{L^2} \\ 0 & R & \omega^2 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\mathcal{X}_2}{L} \\ \mathcal{X}_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{L} \\ 0 & 1 & -\frac{R}{L^2} \\ 0 & 0 & \omega^2 + \frac{R^2}{L^2} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\mathcal{X}_2}{L} \\ \mathcal{X}_3 - \frac{R\mathcal{X}_2}{L} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{L} \\ 0 & 1 & -\frac{R}{L^2} \\ 0 & 0 & \frac{L^2\omega^2 + R^2}{L^2} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\mathcal{X}_2}{L} \\ \frac{L\mathcal{X}_3 - R\mathcal{X}_2}{L} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{1}{L} \\ 0 & 1 & -\dfrac{R}{L^2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{\mathcal{X}_2}{L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{1}{L} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{\mathcal{X}_2}{L} + \dfrac{(L\mathcal{X}_3 - R\mathcal{X}_2)R}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{1}{L} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{\mathcal{X}_2}{L} + \dfrac{(L\mathcal{X}_3 - R\mathcal{X}_2)R}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{1}{L} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{L^2\omega^2\mathcal{X}_2 + R^2\mathcal{X}_2 + LR\mathcal{X}_3 - R^2\mathcal{X}_2}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dfrac{1}{L} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{L^2\omega^2\mathcal{X}_2 + LR\mathcal{X}_3}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \\ 0 & 1 & 0 \\ \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \dfrac{R\mathcal{X}_2 - L\mathcal{X}_3}{L^2\omega^2 + R^2} \\ \dfrac{L^2\omega^2\mathcal{X}_2 + LR\mathcal{X}_3}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

From which the time domain solution can be written as:

$$\begin{aligned} I(S) &= \frac{\mathcal{A}S}{S^2 + \omega^2} + \frac{\mathcal{B}}{S^2 + \omega^2} + \frac{\frac{\mathcal{C}}{L}}{\frac{R}{L} + S} \\ \mathscr{L}^{-1}[I(S)] &= \mathscr{L}^{-1}\left[\frac{\mathcal{A}S}{S^2 + \omega^2} + \frac{\mathcal{B}}{S^2 + \omega^2} + \frac{\frac{\mathcal{C}}{L}}{\frac{R}{L} + S}\right] \\ i(t) &= \mathcal{A}cos(\omega t) + \frac{\mathcal{B}}{w}sin(\omega t) + \frac{\mathcal{C}}{L}e^{-\frac{R}{L}t} \end{aligned}$$

Putting this together with:

$$\begin{bmatrix} \mathcal{A} \\ \\ \mathcal{B} \\ \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} \dfrac{R\mathcal{X}_2 - L\mathcal{X}_3}{L^2\omega^2 + R^2} \\ \dfrac{L^2\omega^2\mathcal{X}_2 + LR\mathcal{X}_3}{(L^2\omega^2 + R^2)L} \\ \dfrac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{L^2\omega^2 + R^2} \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ \\ -\sqrt{2}V_R sin\delta \\ \\ \sqrt{2}V_S\omega - \sqrt{2}V_R cos\delta \end{bmatrix} = \begin{bmatrix} \mathcal{X}_1 \\ \\ \mathcal{X}_2 \\ \\ \mathcal{X}_3 \end{bmatrix}$$

$$\begin{gathered} i(t) = \frac{R\mathcal{X}_2 - L\mathcal{X}_3}{L^2\omega^2 + R^2}cos(\omega t) + \frac{L^2\omega^2\mathcal{X}_2 + LR\mathcal{X}_3}{(L^2\omega^2 + R^2)L}sin(\omega t) + \\ \frac{L^2\mathcal{X}_3 - RL\mathcal{X}_2}{(L^2\omega^2 + R^2)L}e^{-\frac{R}{L}t} \end{gathered}$$

$$i(t) = \frac{R\mathcal{X}_2 - L\mathcal{X}_3}{L^2\omega^2 + R^2}cos(\omega t) + \frac{L\omega^2\mathcal{X}_2 + R\mathcal{X}_3}{L^2\omega^2 + R^2}sin(\omega t) + \\ \frac{L\mathcal{X}_3 - R\mathcal{X}_2}{L^2\omega^2 + R^2}e^{-\frac{R}{L}t}$$

$$i(t) = \frac{-\sqrt{2}RV_Rsin\delta - L(\sqrt{2}V_S\omega - \sqrt{2}V_Rcos\delta)}{L^2\omega^2 + R^2}cos(\omega t) + \\ \frac{-\sqrt{2}L\omega^2V_Rsin\delta + R(\sqrt{2}V_S\omega - \sqrt{2}V_Rcos\delta)}{L^2\omega^2 + R^2}sin(\omega t) + \\ \frac{L(\sqrt{2}V_S\omega - \sqrt{2}V_Rcos\delta) + \sqrt{2}RV_Rsin\delta}{L^2\omega^2 + R^2}e^{-\frac{R}{L}t}$$

$$i(t) = \frac{\sqrt{2}}{L^2\omega^2 + R^2}((-RV_Rsin\delta - L(V_S\omega - V_Rcos\delta))cos(\omega t) + \\ (L\omega^2V_Rsin\delta + R(wV_S - V_Rcos\delta))sin(\omega t)) + \\ (L(wV_S - V_Rcos\delta) + RV_Rsin\delta)e^{-\frac{R}{L}t}$$

Where it can be observed that the transient state depends only on the last term, lasting $5R/L$ seconds, after which the current stabilises. If the current and voltage are stable, then the power will also be stable. This sets an upper limit on how fast the outer loop can operate and a lower limit on how slow the inner loop must be to not cause interference between the two.

As part of the inner loop, the control must also minimise the distortion caused by the power converter, to make the system behave like a PQ bus from the point of view of the outer loop and for the rest of the system. For this effect an LCL filter is used along with DQ0 control the three-phase system using two constant values assuming linear loads. The range of operation of the inner controller is also limited to be within 0.85:1 of the reference signal such that the comparison with the carrier results in the least amount of distortion after the filter. This range can be adjusted from the gains of the inner loop and the DC voltage supplying the converter.

Appendix **E**

Laplacian and L-polynomial calculation

E.1 Main Container fault

This section illustrates the calculation of the L-polynomial before and after the fault of the main-container to show that the network remains fully connected.

Before the fault, the Laplacian $\mathfrak{L}_o$ is:

$$\mathfrak{L}_o = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathfrak{L}_o = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

From these Laplacian, the characteristic polynomial is:

$$|\mathfrak{L}_o - \lambda I| = \left\| \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{bmatrix} \right\|$$

$$|\mathcal{L}_0 - \lambda I| = \begin{vmatrix} 3-\lambda & -1 & -1 & -1 & 0 & 0 \\ -1 & 3-\lambda & -1 & 0 & -1 & 0 \\ -1 & -1 & 3-\lambda & 0 & 0 & -1 \\ -1 & 0 & 0 & 1-\lambda & 0 & 0 \\ 0 & -1 & 0 & 0 & 1-\lambda & 0 \\ 0 & 0 & -1 & 0 & 0 & 1-\lambda \end{vmatrix}$$

This will be refered as $\mathfrak{C}_0$. To obtain the polynomial form, it is proceeded as a determinant calculation:

$$|\mathcal{L}_0 - \lambda I| = \mathfrak{C}_0$$

$$\mathfrak{C}_0 = \begin{vmatrix} 3-\lambda & -1 & -1 & -1 & 0 & 0 \\ -1 & 3-\lambda & -1 & 0 & -1 & 0 \\ -1 & -1 & 3-\lambda & 0 & 0 & -1 \\ -1 & 0 & 0 & 1-\lambda & 0 & 0 \\ 0 & -1 & 0 & 0 & 1-\lambda & 0 \\ 0 & 0 & -1 & 0 & 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_0 = 1 \begin{vmatrix} 3-\lambda & -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 \\ -1 & 3-\lambda & 0 & 0 & -1 \\ -1 & 0 & 1-\lambda & 0 & 0 \\ 0 & -1 & 0 & 0 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda) \begin{vmatrix} 3-\lambda & -1 & -1 & -1 & 0 \\ -1 & 3-\lambda & -1 & 0 & 0 \\ -1 & -1 & 3-\lambda & 0 & -1 \\ -1 & 0 & 0 & 1-\lambda & 0 \\ 0 & 0 & -1 & 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_0 = (3-\lambda)\begin{vmatrix} -1 & 0 & -1 & 0 \\ 3-\lambda & 0 & 0 & -1 \\ 0 & 1-\lambda & 0 & 0 \\ -1 & 0 & 0 & 1-\lambda \end{vmatrix} + 1\begin{vmatrix} -1 & 0 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ -1 & 1-\lambda & 0 & 0 \\ 0 & 0 & 0 & 1-\lambda \end{vmatrix} -$$

$$1\begin{vmatrix} -1 & -1 & -1 & 0 \\ -1 & 3-\lambda & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1-\lambda \end{vmatrix} + (1-\lambda)\begin{vmatrix} -1 & -1 & -1 & 0 \\ 3-\lambda & -1 & 0 & 0 \\ -1 & 3-\lambda & 0 & -1 \\ 0 & -1 & 0 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2\begin{vmatrix} 3-\lambda & -1 & -1 & 0 \\ -1 & 3-\lambda & -1 & 0 \\ -1 & -1 & 3-\lambda & -1 \\ 0 & 0 & -1 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_0 = -(3-\lambda)\begin{vmatrix} 0 & 0 & -1 \\ 1-\lambda & 0 & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} - (3-\lambda)\begin{vmatrix} 3-\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} -$$

$$1\begin{vmatrix} 0 & 0 & -1 \\ 1-\lambda & 0 & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} - 1\begin{vmatrix} -1 & 0 & -1 \\ -1 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} +$$

$$1\begin{vmatrix} -1 & -1 & 0 \\ 3-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} -1 & 0 & 0 \\ 3-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)\begin{vmatrix} 3-\lambda & 0 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 1-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 3-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2\begin{vmatrix} 3-\lambda & -1 & -1 & 0 \\ -1 & 3-\lambda & -1 & 0 \\ -1 & -1 & 3-\lambda & -1 \\ 0 & 0 & -1 & 1-\lambda \end{vmatrix}$$

Eliminating the components that have columns with only zeros we have:

$$\mathfrak{C}_0 = -(3-\lambda)\begin{vmatrix} 3-\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} - 1\begin{vmatrix} -1 & 0 & -1 \\ -1 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} +$$

$$1\begin{vmatrix} -1 & -1 & 0 \\ 3-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 3-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2\begin{vmatrix} 3-\lambda & -1 & -1 & 0 \\ -1 & 3-\lambda & -1 & 0 \\ -1 & -1 & 3-\lambda & -1 \\ 0 & 0 & -1 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_0 = -(3-\lambda)\begin{vmatrix} 3-\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} - 1\begin{vmatrix} -1 & 0 & -1 \\ -1 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} +$$

$$1\begin{vmatrix} -1 & -1 & 0 \\ 3-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 3-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 3-\lambda & 0 \\ -1 & -1 & -1 \end{vmatrix} + (1-\lambda)^3\begin{vmatrix} 3-\lambda & -1 & -1 \\ -1 & 3-\lambda & -1 \\ -1 & -1 & 3-\lambda \end{vmatrix}$$

$$\mathfrak{C}_0 = -(3-\lambda)(1-\lambda)((3-\lambda)(1-\lambda)-1) + (1-\lambda)^2 -$$

$$1\begin{vmatrix} 0 & -1 \\ 0 & 1-\lambda \end{vmatrix} + 1\begin{vmatrix} 3-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} - (1-\lambda)(3-\lambda)\begin{vmatrix} 3-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} -$$

$$(1-\lambda)\begin{vmatrix} -1 & -1 \\ 0 & 1-\lambda \end{vmatrix} + (1-\lambda)^2(3-\lambda)\begin{vmatrix} 3-\lambda & 0 \\ -1 & -1 \end{vmatrix} +$$

$$(1-\lambda)^2\begin{vmatrix} -1 & 0 \\ -1 & -1 \end{vmatrix} + (1-\lambda)^3(3-\lambda)\begin{vmatrix} 3-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} +$$

$$(1-\lambda)^3\begin{vmatrix} -1 & -1 \\ -1 & 3-\lambda \end{vmatrix} - (1-\lambda)^3\begin{vmatrix} -1 & 3-\lambda \\ -1 & -1 \end{vmatrix}$$

$$\mathfrak{C}_0 = -(3-\lambda)(1-\lambda)((3-\lambda)(1-\lambda)-1) + (1-\lambda)^2 +$$

$$(3-\lambda)(1-\lambda) - 1 - (1-\lambda)(3-\lambda)((3-\lambda)(1-\lambda)-1) +$$

$$(1-\lambda)^2 - (1-\lambda)^2(3-\lambda)^2 + (1-\lambda)^2 +$$

$$(1-\lambda)^3(3-\lambda)((3-\lambda)^2-1) +$$

$$(1-\lambda)^3(-(3-\lambda)-1) - (1-\lambda)^3(1+(3-\lambda))$$

$$\begin{gathered}
\mathfrak{C}_0 = (3-\lambda)(1-\lambda)(1-(3-\lambda)(1-\lambda)) + (1-\lambda)^2 + \\
(3-\lambda)(1-\lambda) - 1 - (1-\lambda)^2(3-\lambda)^2 + (3-\lambda)(1-\lambda) + \\
(1-\lambda)^2 - (1-\lambda)^2(3-\lambda)^2 + (1-\lambda)^2 + \\
(1-\lambda)^3(3-\lambda)^3 - (1-\lambda)^3(3-\lambda) - \\
(1-\lambda)^3 - (3-\lambda)(1-\lambda)^3 - (1-\lambda)^3 - (3-\lambda)(1-\lambda)^3
\end{gathered}$$

$$\begin{gathered}
\mathfrak{C}_0 = (3-\lambda)(1-\lambda) - (3-\lambda)^2(1-\lambda)^2 + (1-\lambda)^2 + \\
(3-\lambda)(1-\lambda) - 1 - (1-\lambda)^2(3-\lambda)^2 + (3-\lambda)(1-\lambda) + \\
(1-\lambda)^2 - (1-\lambda)^2(3-\lambda)^2 + (1-\lambda)^2 + \\
(1-\lambda)^3(3-\lambda)^3 - (1-\lambda)^3(3-\lambda) - \\
(1-\lambda)^3 - (3-\lambda)(1-\lambda)^3 - (1-\lambda)^3 - (3-\lambda)(1-\lambda)^3
\end{gathered}$$

$$\begin{gathered}
\mathfrak{C}_0 = 3(3-\lambda)(1-\lambda) - 3(3-\lambda)^2(1-\lambda)^2 + 3(1-\lambda)^2 - \\
1 + (1-\lambda)^3(3-\lambda)^3 - 3(3-\lambda)(1-\lambda)^3 - 2(1-\lambda)^3
\end{gathered}$$

$$\begin{gathered}
\mathfrak{C}_0 = 3(3-4\lambda+\lambda^2) - 3(9-6\lambda+\lambda^2)(1-2\lambda+\lambda^2) + 3(1-2\lambda+\lambda^2) - 1 + \\
(1-3\lambda+3\lambda^2+\lambda^3)(3-\lambda)^3 - 3(3-\lambda)(1-3\lambda+3\lambda^2-\lambda^3) - 2(1-3\lambda+3\lambda^2-\lambda^3)
\end{gathered}$$

$$\begin{gathered}
\mathfrak{C}_0 = 9 - 12\lambda + 3\lambda^2 - 3(9-6\lambda+\lambda^2)(1-2\lambda+\lambda^2) + 3 - 6\lambda + 3\lambda^2 - 1 + \\
(1-3\lambda+3\lambda^2+\lambda^3)(27-27\lambda+9\lambda^2-\lambda^3) - 3(3-\lambda)(1-3\lambda+3\lambda^2-\lambda^3) - \\
2 + 6\lambda - 6\lambda^2 + 2\lambda^3
\end{gathered}$$

$$\begin{gathered}
\mathfrak{C}_0 = 9 - 12\lambda - 27 + 72\lambda - 66\lambda^2 + 24\lambda^3 - 3\lambda^4 + \\
27 - 108\lambda + 171\lambda^2 - 136\lambda^3 + 57\lambda^4 - 12\lambda^5 + \lambda^6 - \\
9 + 30\lambda - 36\lambda^2 + 18\lambda^3 - 3\lambda^4 + 2\lambda^3
\end{gathered}$$

$$\mathfrak{C}_0 = -18\lambda - 69\lambda^2 - 92\lambda^3 + 51\lambda^4 - 12\lambda^5 + \lambda^6$$

It can be observed that $\mathfrak{C}_0$ has a minimum exponent of one, therefore there is only one eigenvalue $\lambda_0 = 0$.

For the case after the fault, the L-polynomial $\mathfrak{C}_1$ is calculated as follows:

$$\mathcal{L}_1 = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_1 = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

$$|\mathcal{L}_1 - \lambda I| = \left| \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix} \right|$$

$$|\mathcal{L}_1 - \lambda I| = \begin{vmatrix} 3-\lambda & -1 & -1 & -1 & 0 \\ -1 & 2-\lambda & 0 & 0 & -1 \\ -1 & 0 & 1-\lambda & 0 & 0 \\ -1 & 0 & 0 & 1-\lambda & 0 \\ 0 & -1 & 0 & 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_1 = -1 \begin{vmatrix} -1 & -1 & -1 & 0 \\ 2-\lambda & 0 & 0 & -1 \\ 0 & 0 & 1-\lambda & 0 \\ -1 & 0 & 0 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda) \begin{vmatrix} 3-\lambda & -1 & -1 & 0 \\ -1 & 2-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda & 0 \\ 0 & -1 & 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_1 = 1 \begin{vmatrix} 0 & 0 & -1 \\ 0 & 1-\lambda & 0 \\ 0 & 0 & 1-\lambda \end{vmatrix} - 1 \begin{vmatrix} 2-\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} +$$

$$1 \begin{vmatrix} 2-\lambda & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} - (1-\lambda) \begin{vmatrix} -1 & -1 & 0 \\ 2-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2 \begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix}$$

Similarly as before, the terms with columns full of zeros are eliminated:

$$\mathfrak{C}_1 = -1 \begin{vmatrix} 2-\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{vmatrix} -$$

$$(1-\lambda) \begin{vmatrix} -1 & -1 & 0 \\ 2-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} + (1-\lambda)^2 \begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_1 = -(1-\lambda)((2-\lambda)(1-\lambda)-1)+$$
$$(1-\lambda)\begin{vmatrix} 0 & -1 \\ 0 & 1-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} +$$
$$(1-\lambda)^2(3-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} + (1-\lambda)^2\begin{vmatrix} -1 & -1 \\ 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_1 = -(1-\lambda)((2-\lambda)(1-\lambda)-1)-$$
$$(1-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} + (1-\lambda)^2(3-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 1-\lambda \end{vmatrix} +$$
$$(1-\lambda)^2\begin{vmatrix} -1 & -1 \\ 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_1 = -(1-\lambda)((2-\lambda)(1-\lambda)-1)-$$
$$(1-\lambda)^2(2-\lambda)+(1-\lambda)+(1-\lambda)^2(3-\lambda)((2-\lambda)(1-\lambda)-1)-$$
$$(1-\lambda)^3$$

$$\mathfrak{C}_1 = -(1-\lambda)^2(2-\lambda)+1-\lambda-(1-\lambda)^2(2-\lambda)+$$
$$1-\lambda+(1-\lambda)^3(3-\lambda)(2-\lambda)-(1-\lambda)^2(3-\lambda)-(1-\lambda)^3$$

$$\mathfrak{C}_1 = -2(1-\lambda)^2(2-\lambda)+2-2\lambda+$$
$$(1-\lambda)^3(3-\lambda)(2-\lambda)-(1-\lambda)^2(3-\lambda)-(1-\lambda)^3$$

$$\mathfrak{C}_1 = -2(1-2\lambda+\lambda^2)(2-\lambda)+2-2\lambda+$$
$$(1-\lambda)^3(3-\lambda)(2-\lambda)-(1-2\lambda+\lambda^2)(3-\lambda)-(1-\lambda)^3$$

$$\mathfrak{C}_1 = -2(1-2\lambda+\lambda^2)(2-\lambda)+2-2\lambda+$$
$$(1-3\lambda+3\lambda^2-\lambda^3)(6-5\lambda+\lambda^2)-(1-2\lambda+\lambda^2)(3-\lambda)-1+3\lambda-3\lambda^2+\lambda^3$$

$$\mathfrak{C}_1 = -2(2 - 5\lambda + 4\lambda^2 - \lambda^3) + 2 - 2\lambda +$$

$$6 - 23\lambda + 34\lambda^2 - 24\lambda^3 + 8\lambda^4 - \lambda^5 - 3 + 7\lambda - 5\lambda^2 + \lambda^3 - 1 + 3\lambda - 3\lambda^2 + \lambda^3$$

$$\mathfrak{C}_1 = -5\lambda + 18\lambda^2 - 20\lambda^3 + 8\lambda^4 - \lambda^5$$

Again, it can be seen that the smallest exponent is one, which indicates that the eigenvalue $\lambda_0 = 0$ is not repeated in $\mathfrak{C}_1$, which in turn shows that the network is fully connected.

E.2 Container fault

This section illustrates the calculation of the L-polynomial before and after the fault of a container to show that the network remains fully connected.

Before the fault, the Laplacian Matrix and L-polynomial are equal to $\mathcal{L}_0$ and $\mathfrak{C}_0$. After the fault the Laplacian matrix becomes:

$$\mathcal{L}_2 = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_2 = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$|\mathcal{L}_2 - \lambda I| = \left| \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix} \right|$$

$$|\mathcal{L}_2 - \lambda I| = \begin{vmatrix} 3-\lambda & -1 & -1 & -1 & 0 \\ -1 & 2-\lambda & -1 & 0 & 0 \\ -1 & -1 & 3-\lambda & 0 & -1 \\ -1 & 0 & 0 & 1-\lambda & 0 \\ 0 & 0 & -1 & 0 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_2 = 1 \begin{vmatrix} -1 & -1 & -1 & 0 \\ 2-\lambda & -1 & 0 & 0 \\ -1 & 3-\lambda & 0 & -1 \\ 0 & -1 & 0 & 1-\lambda \end{vmatrix} + (1-\lambda) \begin{vmatrix} 3-\lambda & -1 & -1 & 0 \\ -1 & 2-\lambda & -1 & 0 \\ -1 & -1 & 3-\lambda & -1 \\ 0 & 0 & -1 & 1-\lambda \end{vmatrix}$$

$$\mathfrak{C}_2 = -1 \begin{vmatrix} -1 & 0 & 0 \\ 3-\lambda & 0 & -1 \\ -1 & 0 & 1-\lambda \end{vmatrix} + 1 \begin{vmatrix} 2-\lambda & 0 & 0 \\ -1 & 0 & -1 \\ 0 & 0 & 1-\lambda \end{vmatrix} - 1 \begin{vmatrix} 2-\lambda & -1 & 0 \\ -1 & 3-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix} +$$

$$(1-\lambda) \begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & 0 \\ -1 & -1 & -1 \end{vmatrix} + (1-\lambda)^2 \begin{vmatrix} 3-\lambda & -1 & -1 \\ -1 & 2-\lambda & -1 \\ -1 & -1 & 3-\lambda \end{vmatrix}$$

$$\mathfrak{C}_2 = -1 \begin{vmatrix} 2-\lambda & -1 & 0 \\ -1 & 3-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{vmatrix} + (1-\lambda) \begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & 0 \\ -1 & -1 & -1 \end{vmatrix} +$$

$$(1-\lambda)^2 \begin{vmatrix} 3-\lambda & -1 & -1 \\ -1 & 2-\lambda & -1 \\ -1 & -1 & 3-\lambda \end{vmatrix}$$

$$\mathfrak{C}_2 = -1\begin{vmatrix} 2-\lambda & 0 \\ -1 & -1 \end{vmatrix} - (1-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} -$$

$$(1-\lambda)\begin{vmatrix} -1 & 0 \\ 2-\lambda & 0 \end{vmatrix} + (1-\lambda)\begin{vmatrix} 3-\lambda & 0 \\ -1 & 0 \end{vmatrix} - (1-\lambda)\begin{vmatrix} 3-\lambda & -1 \\ -1 & 2-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2(3-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} + (1-\lambda)^2\begin{vmatrix} -1 & -1 \\ -1 & 3-\lambda \end{vmatrix} - (1-\lambda)^2\begin{vmatrix} -1 & 2-\lambda \\ -1 & -1 \end{vmatrix}$$

$$\mathfrak{C}_2 = -1\begin{vmatrix} 2-\lambda & 0 \\ -1 & -1 \end{vmatrix} - (1-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} - (1-\lambda)\begin{vmatrix} 3-\lambda & -1 \\ -1 & 2-\lambda \end{vmatrix} +$$

$$(1-\lambda)^2(3-\lambda)\begin{vmatrix} 2-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} + (1-\lambda)^2\begin{vmatrix} -1 & -1 \\ -1 & 3-\lambda \end{vmatrix} - (1-\lambda)^2\begin{vmatrix} -1 & 2-\lambda \\ -1 & -1 \end{vmatrix}$$

$$\mathfrak{C}_2 = 2-\lambda-(1-\lambda)((2-\lambda)(3-\lambda)-1)-(1-\lambda)((3-\lambda)(2-\lambda)-1)+$$

$$(1-\lambda)^2(3-\lambda)((2-\lambda)(3-\lambda)-1)+(1-\lambda)^2(-(3-\lambda)-1)-$$

$$(1-\lambda)^2(1+(2-\lambda))$$

$$\mathfrak{C}_2 = 2-\lambda-(1-\lambda)(2-\lambda)(3-\lambda)+1-\lambda-(1-\lambda)(3-\lambda)(2-\lambda)+1-\lambda+$$

$$(1-\lambda)^2(3-\lambda)^2(2-\lambda)-(1-\lambda)^2(3-\lambda)-(1-\lambda)^2(3-\lambda)-(1-\lambda)^2-$$

$$(1-\lambda)^2-(2-\lambda)(1-\lambda)^2$$

$$\mathfrak{C}_2 = 4-3\lambda-2(1-\lambda)(2-\lambda)(3-\lambda)+(1-\lambda)^2(3-\lambda)^2(2-\lambda)-$$

$$2(1-\lambda)^2(3-\lambda)-2(1-\lambda)^2-(2-\lambda)(1-\lambda)^2$$

$$\mathfrak{C}_2 = 4-3\lambda-2(2-3\lambda+\lambda^2)(3-\lambda)+(1-2\lambda+\lambda^2)(3-\lambda)^2(2-\lambda)-$$

$$2(1-2\lambda+\lambda^2)(3-\lambda)-2(1-2\lambda+\lambda^2)-$$

$$(2-\lambda)(1-2\lambda+\lambda^2)$$

$$\mathfrak{C}_2 = 4-3\lambda-2(2-3\lambda+\lambda^2)(3-\lambda)+(1-2\lambda+\lambda^2)(9-6\lambda+\lambda^2)(2-\lambda)-$$

$$2(1-2\lambda+\lambda^2)(3-\lambda)-2+4\lambda-2\lambda^2-$$

$$2+5\lambda-4\lambda^2+\lambda^3$$

$$\mathfrak{C}_2 = 6\lambda - 6\lambda^2 + \lambda^3 - 2(6 - 11\lambda + 6\lambda^2 - \lambda^3) +$$

$$(1 - 2\lambda + \lambda^2)(9 - 6\lambda + \lambda^2)(2 - \lambda) - 6 + 14\lambda - 10\lambda^2 + 2\lambda^3$$

$$\mathfrak{C}_2 = 6\lambda - 6\lambda^2 + \lambda^3 - 12 + 22\lambda - 12\lambda^2 + 2\lambda^3 +$$

$$18 - 57\lambda + 68\lambda^2 - 38\lambda^3 + 10\lambda^4 - \lambda^5 - 6 + 14\lambda - 10\lambda^2 + 2\lambda^3$$

$$\mathfrak{C}_2 = -15\lambda - 40\lambda^2 - 33\lambda^3 + 10\lambda^4 - \lambda^5$$

As mentioned before, the exponents of the characteristic polynomial $\mathfrak{C}_2$ indicate that the network after the fault remains fully connected.